2017

Optimal Trajectory Determination and Mission Design for Asteroid/Deep-Space Exploration via Multibody Gravity Assist Maneuvers

Sean Fritz
San Jose State University

Kamran Turkoglu
San Jose State University, kamran.turkoglu@sjsu.edu

Recommended Citation

Sean Fritz and Kamran Turkoglu. "Optimal Trajectory Determination and Mission Design for Asteroid/Deep-Space Exploration via Multibody Gravity Assist Maneuvers" *International Journal of Aerospace Engineering* (2017). https://doi.org/10.1155/2017/6801023

Hindawi
International Journal of Aerospace Engineering
Volume 2017, Article ID 6801023, 12 pages
https://doi.org/10.1155/2017/6801023

Research Article

Optimal Trajectory Determination and Mission Design for Asteroid/Deep-Space Exploration via Multibody Gravity Assist Maneuvers

Sean Fritz and Kamran Turkoglu

Aerospace Engineering, San José State University, San José, CA 95192-0179, USA

Correspondence should be addressed to Kamran Turkoglu; kamran.turkoglu@sjsu.edu

Received 19 April 2017; Accepted 28 June 2017; Published 3 August 2017

Academic Editor: Angel Velazquez

This paper discusses the creation of a genetic algorithm to locate and optimize interplanetary trajectories using gravity assist maneuvers to improve fuel efficiency of the mission. The algorithm is implemented on two cases: (i) a Centaur-class target close to the ecliptic plane and (ii) a Centaur-class target with a high inclination to the ecliptic plane. Cases for multiple numbers of flybys (up to three) are discussed and compared. It is shown that, for the targets considered here, a single flyby of Jupiter is the most efficient trajectory to either target with the conditions and limitations discussed in this paper. In this paper, we also iterate on possible reasons for certain results seen in the analysis and show how these previously observed behaviors could be present in any trajectory found. The parameters and methods used in the algorithm are explained and justified over multiple real-life interplanetary missions to provide deeper insights into the development choices.

1. Introduction

On satellites, or generally speaking any type of deep-space exploration spacecraft, every kilogram of mass becomes a crucial component of overall mission design. Controlling the mass of the spacecraft throughout the design process is vital to the success of the assigned task. Getting to a target in space requires fuel to be expended, and every kilogram of fuel that must be launched is a kilogram of scientific equipment that must be sacrificed in order for the mission to remain feasible. The ability to optimize the deep-space exploration trajectories in a way that minimizes the fuel requirement (i.e., ΔV) is key to greater scientific return (such as maximized life time of the spacecraft and maximum distance traveled). In some cases, this becomes the only way for the desired mission to succeed. In that sense, by limiting the number of maneuvers (or ΔV) that are needed to reach the target of interest, a mission's effectiveness can be maximized through more equipment and/or an extended life at the target, if the mission is propellant-limited.

From this perspective, this paper aims to explore the use of a class of nondeterministic optimization algorithms in finding indirect trajectories to interplanetary targets and minimize the associated fuel consumption throughout the mission. This class of algorithms, namely, genetic algorithms, have been studied before as means of improving interplanetary trajectories. In Xu et al. [1], a genetic algorithm was used to optimize launch and deep-space maneuver parameters to enable closer approaches to two opportunistic asteroids en route to their main target. In another study, a genetic algorithm tool (GALOMUSIT) for finding trajectories to predetermined targets has been developed (in FORTRAN 77) and improved upon by many parties, which was then used by Molenaar [2] to determine an optimized trajectory for an Uranus orbiter. This tool is rather sophisticated in a sense that it allows for multimodal optimization and was improved later on allowing for deep-space maneuver optimization as well. The trajectories found make use of planetary flybys in an attempt to reduce the fuel consumption over a direct target trajectory. Solórzano et al. [3] discuss the planning of trajectories to Neptune using a variety of gravity assist maneuver schemes, highlighting the difficulty in finding an optimized solution with so many combinations of the planets considered. In another valuable study, Izzo et al. [4] discuss a deterministic search space pruning algorithm for multiple gravity assist problem. Vasile and Pascale [5] investigates a preliminary design of multiple gravity assist maneuvers as a global optimization method based on a hybrid genetic algorithm methodology. Gad and Abdelkhalik [6] concentrated their efforts on implementation of genetic algorithms on free number of deep-space maneuvers. Petropoulos and Longuski [7] provided results for the gravity assists which are modeled as discontinuities in velocity arising from an instantaneous turning of the spacecraft. Lee et al. [8] used a genetic algorithm to adjust optimization parameters used to search for Pareto fronts of low-thrust orbit transfers.

The genetic algorithm based trajectory mission design presented herein, completing the existing studies, is used to optimize increasingly difficult optimization problems starting with simple Hohmann transfer timing problems. This methodology then progresses to target rendezvous with intermediate flyby of a chosen planet, finally progressing to multiple planet flybys in which the specific planets, and their order, are determined as part of the optimization process. This constitutes one of the major novelties introduced by this methodology. In this study, we also show genetic algorithm based on methodology's ability to find optimized trajectories to both targets whose trajectories lend themselves to easy rendezvous, such as Main Belt Asteroids or planets, as well as targets that would not be accessible to our current spacecraft technology without the aid of at least one planetary flyby, such as Trans-Neptunian Asteroids or comets, as discussed in Wen et al. [9] and refined further in Wen et al. [10]. This constitutes the second major novelty introduced by this paper. The ultimate goal of such an endeavor is to be capable of finding nonintuitive trajectories that could take advantage of gravitational anomalies (if possible multiple times) that are sensitive to position, velocity and timing.

This paper looks at the following trajectory problems with origins at Earth: (i) an optimal (min. fuel) transfer to a Centaur-class object with multiple flybys and (ii) an optimal (min. fuel) transfer to a Centaur-class object with higher inclination and eccentricity via multiple flybys.

By having the algorithm take on increasingly difficult problems in trajectory optimization, this paper aims to demonstrate the value of such methods in determining interplanetary trajectories that might be hard to find but easy to verify once the trajectory is obtained. It is also the aim of this paper to show the capabilities of such algorithm when deployed on hardware accessible to the common consumer and thus to any institution that seeks to do research at this juncture between computing science and astrodynamics.

The paper is organized as follows: in Section 2.1 through Section 2.3, a high level discussion of genetic algorithms, Lambert's solution, and gravity assist maneuvers is presented. In Section 4, the genetic algorithm is used to find various trajectories to the target asteroid Hylonome. Section 5 is devoted to the interception of an asteroid of high inclination to the ecliptic plane. Finally, there is a revisit of work done to examine nonintuitive results and the analysis if presented in Section 6. The study is finalized with conclusions and future work in Section 7.

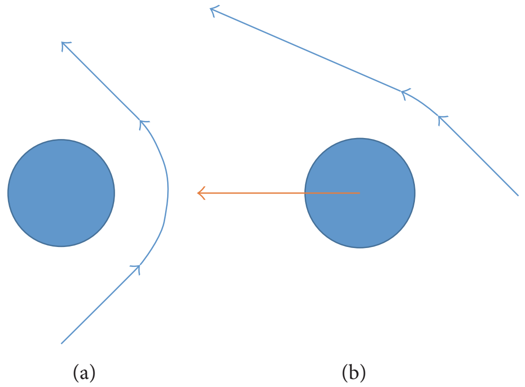

Figure 1: An example of a gravity assist maneuver as seen in a reference frame fixed to the planet (a) and fixed inertially (b).

2. Preliminaries

In this section, for the sake of completeness, we review some of the fundamental concepts that will play a role in the development and/or use of the genetic algorithm (as further discussed in Section 2.3).

2.1. Gravity Assist Maneuvers. Generally speaking, gravity assist maneuver is a technique by which a probe (and/or spacecraft) can change its orbital energy (and/or momentum) with respect to a primary body by flying past a large secondary body with the utilization of the gravitational force of that specific body (as depicted in Figure 1).

The close flyby of the secondary body will allow its gravity to affect the probe and affect a momentum exchange. Because the secondary body will typically be much larger than the probe, this will result in a large effect on the probe's trajectory with respect to the primary body. In addition to the natural momentum exchange, an opportunity exists to perform a trajectory correction maneuver at the periapse of the flyby. The periapse is preferred to take advantage of the Oberth Effect, as further defined in Oberth [11]. The Oberth Effect arises from the fact that a change in energy is equal to a force applied over a distance (as shown in (1); assuming that all vector quantities are parallel):

$$\Delta E = F\Delta d. \tag{1}$$

Here, since the distance over which the force is applied is equal to the velocity of the spacecraft multiplied by the duration of the burn,

$$\Delta d = \frac{\delta d}{\delta t}\Delta t = v\Delta t, \tag{2}$$

and since the fuel usage is proportional to the duration of the burn (assuming a fixed force),

$$m_{\text{fuel}} \propto \Delta t, \tag{3}$$

it is easy to find that, for a desired change in orbital energy, the fuel used can be minimized by using the engines when the spacecraft is at a maximum velocity of the trajectory.

$$m_{\text{fuel}} \propto v^{-1}. \tag{4}$$

By using a number of flybys, the possibility exists of being able to send instrument-laden probes to targets that might otherwise be outside the reach of our current launch vehicle capability. The key is to know what trajectory to be on in order to intercept the next planet or the target itself. This is where the solution to Lambert's problem comes into play which is revisited next.

2.2. Lambert's Equation. According to this fundamental principle, it is well known that the transfer time between two points on an orbit is independent of the eccentricity and depends on a small number of factors: the magnitudes of the two position vectors, the semimajor axis, and the length of the chord connecting the two points, where further details could be investigated in many classical references including Prussing and Conway [12] and Curtis [13]. The solution to Lambert's problem in which the two positions and the time of flight are known is found with the following technique as derived in Curtis [13]:

(1) Calculate the magnitudes of the two position vectors $\vec{r}_1$ and $\vec{r}_2$.

(2) Calculate the angle $\Delta\theta$ between the two position vectors taking the direction of travel in to account.

(3) Calculate A, where

$$A = \sin\Delta\theta\sqrt{\frac{r_1 r_2}{1-\cos\Delta\theta}}. \tag{5}$$

(4) Solve the following equation by iterating the value of z, where $S(z)$ and $C(z)$ are the Stumpff equations:

$$\sqrt{\mu}\Delta t = \left[\frac{y(z)}{C(z)}\right]^{3/2} S(z) + A\sqrt{y(z)}, \tag{6}$$

where

$$y(z) = r_1 + r_2 + A\frac{zS(z)-1}{\sqrt{C(z)}}. \tag{7}$$

(5) Use the resulting value of z to calculate the Lagrange coefficients:

$$\begin{aligned} f &= 1 - \frac{y(z)}{r_1} \\ g &= A\sqrt{\frac{y(z)}{\mu}} \\ \dot{f} &= \frac{\sqrt{\mu}}{r_1 r_2}\sqrt{\frac{y(z)}{C(z)}}\left[zS(z)-1\right] \\ \dot{g} &= 1 - \frac{y(z)}{r_2}. \end{aligned} \tag{8}$$

(6) Finally, use the Lagrange coefficients and $\vec{r}_1$ and $\vec{r}_2$ to obtain $\vec{v}_1$ and $\vec{v}_2$:

$$\begin{aligned} \vec{v}_1 &= \frac{1}{g}\left(\vec{r}_2 - f\vec{r}_1\right) \\ \vec{v}_2 &= \dot{f}\vec{r}_1 + \dot{g}\vec{v}_1. \end{aligned} \tag{9}$$

This leads to the fact that, in order to calculate a trajectory between two planets in the solar system, we only need to specify the departure time and the travel time. A series of flight stages can be strung together to allow for any number of gravity assists along the way to the target. This suggests that a possible solution vector would take the following form:

$$\left[J_0, P_1, \Delta t_1, P_2, \Delta t_2, \ldots, P_n, \Delta t_n, \Delta t_t\right], \tag{10}$$

where J_0 is *launch date from Earth*, P_x is *target planet*, and Δt_x is time of flight to next planet/target.

This vector provides the complete set of inputs to specify a particular solution. Given J_0, we have the position of Earth at launch from a vector table covering the entire allowable time within which the mission can occur. With Δt_1, we have the time of arrival at P_1 and its position from its vector table. This chain of time spans continues until we have the final time span, Δt_t, that determines the arrival time of the space craft and the position of the target. By altering these parameters, different trajectories can be created and these trajectories can be evaluated for their fuel usage.

2.3. Genetic Algorithms. Genetic algorithms (GAs) attempt to harness the strength of natural evolution in an attempt to solve problems that do not lend themselves to finding solutions in an analytical manner, as presented in Eiben and Smith [14]. Some of the terminology common to discussions of GAs is given in Table 1.

One concept used extensively in GA is that of *crossover*. It is well known from literature that this process takes two solutions and mixes them together to generate two new solutions. This is the primary method of exploring the available search space in the outset of the simulation.

The solution to this problem with crossover leads to another concept named *mutation*. Mutation changes the value of a given value within a solution by a chosen method. This generates a new value that was not present among any of the solutions prior to its alteration. This plays a critical role in optimizing the solutions in the final generations. Towards the end of the simulation most of the solutions will be very similar and there will be very little variety in the values on each of the search axes. Mutation allows for each value to be varied by a small amount that will hopefully yield an improvement in the best fitness.

Embedded within the mutation concept is the idea of *cooling*. Cooling is a process in which the possible range by which the mutation process can alter a value is slowly diminished as the simulation proceeds. This allows the simulation to "dial in" on the minimum value that it has found by minimizing the overshoot caused by the mutation of any single gene. This also allows for a large mutation range

Table 1: Genetic algorithm terminology.

Terminology	Definition
Gene	A single value representing a parameter of a potential solution
Chromosome	A vector representation of the parameters of a potential solution
Population	A collection of chromosomes
Crossover	A process by which two or more existing chromosomes are recombined to yield two different chromosomes
Mutation	A process by which gene values are changed yielding a new chromosome
Selection	A process that determines which chromosomes from the population to use for the crossover process
Fitness function	Algorithm that assigns a value or "fitness" to each chromosome This value being typically minimized or maximized

at the beginning of the simulation to create as many unique points as possible in the next generation without suffering the large range for the entire simulation.

Selection is the process by which the two parent chromosomes on which to apply the above concepts are chosen. The GA created for this work relies on an unbalanced mapping of the chromosomes to a uniformly distributed random number to favor the selection of solutions with better fitness. Here, unbalanced mapping defines the situation where more chromosomes are represented by a smaller range of the random number domain and thus have a smaller chance of being selected. This aims to maintain the capability of selecting any solution from the current generation, even if it was the worst solution tested. The mapping chosen in this study is shown in Figure 2 for a population of 300 chromosomes. As is also seen in Figure 2, solutions that rank above the median fitness (i.e., have ΔV lower than that of half the population and thus are ranked at position 150 or higher in this example) value have a cumulative 69% chance of being selected to seed the next generation, while solutions below the median fitness value have a cumulative 31% chance of being selected as seen on the x-axis which corresponds to the value of a uniformly distributed random number for selection. With the disassociation of the test value from the actual calculation result, a situation in which "lucky" few chromosome are dominating a selection round is avoided. This preserves diversity in solutions while a thorough exploration of the search space is made and prevents rapid convergence to a solution that starts strongly but is too distant from a better minimum location in the search space.

If Figure 2 is compared to Figure 3, it can be seen that the mapping used by the GA is one that favors solutions with higher fitness which in this problem are solutions that will have lower ΔV.

Finally if Figure 2 is compared to Figure 4, it can be seen that the selection pressure applied to the GA is not so extreme to run into the issue discussed earlier of squeezing out weaker solutions before their genes can be used to explore the search space over a few generations. Again, this preserves diversity in the solution parameters while the search space is explored via crossover.

Another critical concept utilized was that of *survivorship*. The two solutions with the best fitness score were carried into the next generation without any changes made to their

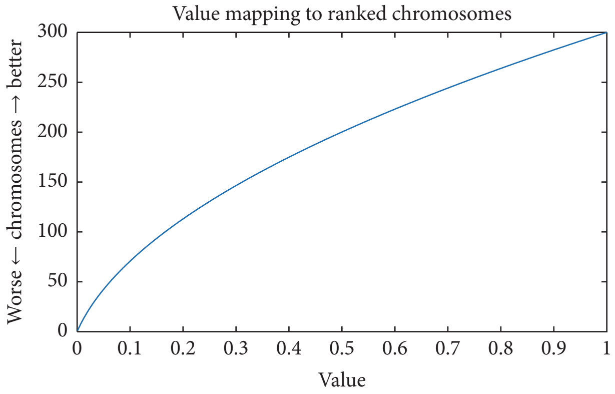

Figure 2: The mapping of the selection value to the ranked chromosomes. When testing from bottom to top, the first chromosome to have a higher value than a randomly generated number in the range (0, 1) is selected.

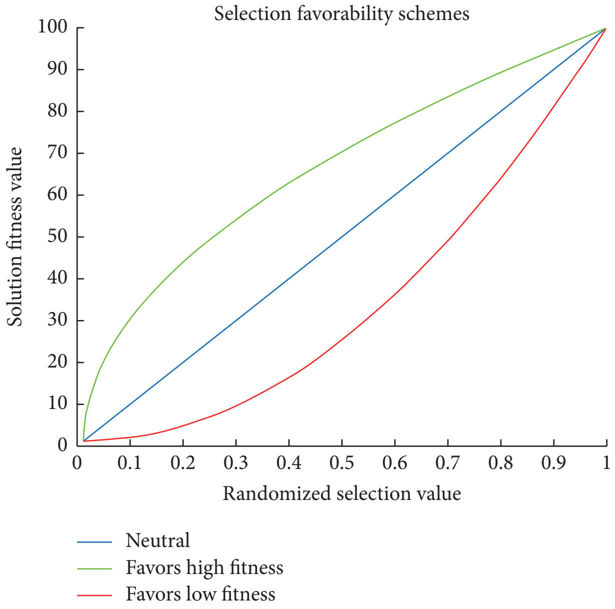

Figure 3: Three selection schemes for choosing parent chromosomes. A randomized value in the range (0, 1) is chosen on the x-axis and solution associated with the corresponding fitness value is used as a parent.

Table 2: Gene specific parameters.

Parameter	Maximum	Minimum	Standard deviation
J_0 (Julian Date)	2462489.5	2451545	100 (days)
P_x (planet number)	8	1	1
Δt_x (seconds)	∞	0	28512000

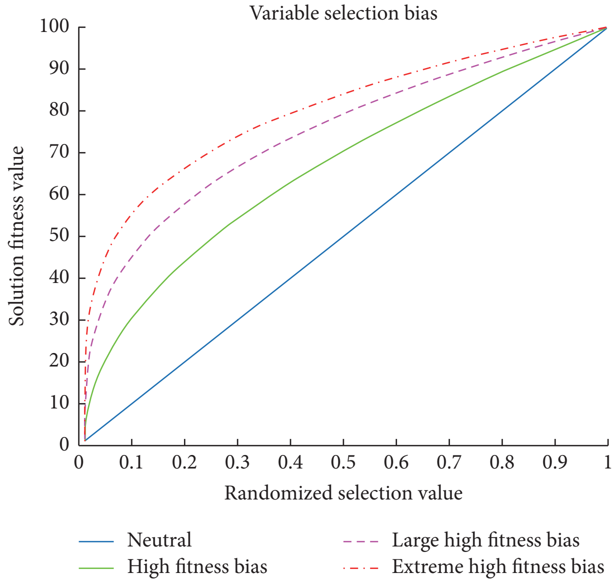

Figure 4: Four selection schemes for choosing parent chromosomes. A randomized value in the range $(0, 1)$ is chosen on the x-axis and solution associated with the corresponding fitness value is used as a parent. The most extreme bias gives solutions in the bottom half of the ranking a combined chance of less than 10% of being selected.

chromosomes. Without this technique, the best solution found thus far would be lost with no improvement in the next generation.

3. Parameter Limits and Design

In the following, we discuss the limits that were applied to each of the conducted studies. Section 3.1 through Section 3.2 discusses input limits, while Section 3.3 through Section 3.5 discusses limits used directly by the GA.

3.1. Date Range. The GA was given as inputs of the position and velocity data for the planets and the targets for dates ranging from 01 Jan 2000 to 28 Dec 2029 in five-day increments. This date range was chosen because it seemed a reasonable range over which to look for mission planning purposes. A mission that would have to occur more than 30 years from the conception time would likely not be considered in the real world. Thus the ability of the GA to find solutions even if the perfect launch time was not available would be important. The date range used was to enable the possibility of converging to known trajectories during testing of the fitness function and GA while also providing enough time range to explore future trajectories to distant targets.

A linear interpolation method was used to calculate positions and velocities in between the data points provided. Five-day increments were chosen to minimize the error in linear interpolation of planetary position and velocity data should, for an inner planet like Mercury, be chosen for evaluation of a gravity assist.

3.2. Gene Specific Parameters and Limits. The genes had several parameters in common that were used by the GA, although the values of these parameters were altered depending on the gene that they were being applied to. All genes have a maximum value, a minimum value, and a standard deviation value used to scale the randomization of any mutation that the gene undergoes. The values for the used ones for these parameters are shown in Table 2. J_0 is the Julian Date of launch and has units of days for the standard deviation. Δt_x is transfer time in seconds with its standard deviation in the same units. P_x is the next planet around which a flyby will occur. When this gene is mutated, it still utilizes a normally distributed random value, but it is then truncated to an integer value subject to the minimum and maximum limits. It should be noted that the lack or constraint on the Δt_x parameters had no adverse effect on the converged solutions, though they may have contributed to the long convergence times that will be seen later. If the total flight time duration was a constraint, it would be possible to make an educated guess on the allowable transfer times.

3.3. Population Size. The GA utilized in this paper uses a fixed population of 300 for finding trajectories to Hylonome and Hidalgo. This specific number was the result of trying to optimize between the time needed to converge on a solution and providing sufficient density of initial test points in the search space. Increasing the population to a higher value was found to quickly increase the run time for a single simulation especially for the triple flyby algorithm.

Being run on a personal Windows computer, the operating system was known to occasionally suffer instabilities and reboot. Long runtimes risked losing work prior to convergence under these conditions. On the other hand, small population sizes were found to be at risk of converging too quickly to solutions that were not optimal based on subsequent simulations. This is because they did not have sufficient diversity to explore the search space and the variance in the final ΔV solutions left doubt as to whether the algorithm was working properly.

3.4. Crossover and Mutation Rates. The crossover rate was set to a 30% chance of crossover per gene in the chromosome to ensure a high probability that crossover would occur somewhere within the chromosome. This means that the

Table 3: Convergence times for various simulation types for trajectories to Hylonome.

Hylonome	Direct	Single	Double	Triple
Minimum	0:42:00	2:26:00	9:08:00	3:14:00
Maximum	1:35:00	6:06:00	50:18:00	28:12:00
Average	1:07:23	3:43:44	26:41:20	8:21:13
Median	1:08:00	3:41:00	20:38:00	6:33:00
Standard deviation	0:13:45	0:58:46	21:14:27	7:35:12
Total runs	19	12	4	10

Table 4: Convergence times for various simulation types for trajectories to Hidalgo.

Hidalgo	Direct	Single	Double	Triple
Minimum	0:41:00	1:59:00	1:46:00	1:05:00
Maximum	1:34:00	15:48:00	8:33:00	9:08:00
Average	0:59:13	4:32:00	5:47:22	4:37:32
Median	0:50:00	2:42:30	6:13:30	5:30:00
Standard deviation	0:17:31	4:38:25	1:58:57	2:51:15
Total runs	19	11	10	15

chance of failing to perform a crossover at any one particular gene is 70%. Since the next gene in the chromosome is tested only after the previous one fails, the chance of n-gene chromosomes failing to perform a crossover is given in (12). The likelihood of the crossover point being after the nth gene is shown in (13):

$$\begin{aligned} F(1) &= 0.7 \\ F(2) &= 0.7 \times 0.70 \\ F(3) &= 0.7 \times 0.7 \times 0.7 \\ &\vdots \end{aligned} \tag{11}$$

$$F(x) = (0.7)^{x}\Big|_{1}^{n-1} \tag{12}$$

$$C(x) = 1 - (0.7)^{x}\Big|_{1}^{n-1}, \tag{13}$$

where n is the number of genes in the chromosomes, x is the current gene, and $C(x)$ is the cumulative likelihood of a crossover occurring after gene x as x increases from 1 to $n-1$. the upper limit is $n-1$ since a crossover after the last gene results in no genes swapping between the two chosen chromosomes and thus no crossover.

The mutation rate is specified per gene in the chromosomes to allow for certain genes to mutate more often than others if desired. Mutation is a very powerful modifier, Eiben and Smith [14], and so the chance of it occurring should be kept low so as not to obscure the effects of crossover. In the GA discussed in this paper, every gene was given a 5% chance of modification via mutation. Whereas the crossover check is performed until it succeeds once on the chromosome, the mutation check is performed on each gene, regardless of the outcome of previous checks on prior genes. While the chance of any individual gene to mutate is 5% (i.e., a 95% chance of failure), the likelihood of the entire chromosome remaining unchanged is given

$$F(x) = (0.95)^{x}\Big|_{1}^{n} \tag{14}$$

where $F(x)$ is the likelihood of the chromosome not mutating on any gene, and the rest of the symbology is as in (13). For a single flyby solution with four parameters ($[J_0, P_1, \Delta t_1, \Delta t_t]$), the possibility of no changes due to mutation is approximately 80% while a flyby solution with 8 parameters ($[J_0, P_1, \Delta t_1, P_2, \Delta t_2, P_3, \Delta t_3, \Delta t_t]$) has a probability of no change due to mutation closer to 66%. Increasing the mutation rate to 10% would make the 8-parameter solution be affected by mutation more than half of the time, while a 1% rate would make the 4-parameter solution nearly immune to mutation effects.

3.5. Simulation Termination Criteria. In this study, the simulations conducted had two conditions under which they could be terminated. The first condition was a simple maximum generation count. If a solution was not found after 2000 generations, the simulation would terminate and complete as though it had converged. No flag was implemented to indicate that these termination criteria were the one used, but the number of generations needed to converge is in the output of the simulation and was therefore easy to verify whether or not this was the criteria used to terminate the simulation. In no simulation was this maximum generation count encountered.

The second termination criteria were reached if the simulation converged on a solution. Convergence was defined as seeing no improvement of more than 1 m/s of ΔV for a certain number of generations. Thirty generations were the range of a priori information used in this GA.

With the parameters discussed above, the time (hh:mm:ss) to converge for each simulation type is shown in Tables 3 and 4, where the computational platform is a laptop PC with i3 Intel core processor and 8 Gb RAM.

3.6. Fitness Function. The fitness function is the workhorse of the designed GA. This algorithm is the part of the GA that evaluates the potential solutions and assigns a fitness value to each chromosome based on ΔV calculated for the trajectory specified by the chromosome. Any quantitative measurement that is primarily affected by the genes present in the chromosome can be used by the GA as an operator to create the next generation. The fitness function in this paper focused on calculating ΔV of the proposed trajectory in an attempt to find a trajectory that used the minimum amount of fuel and had the specified number of planetary flybys.

The fitness function starts by determining the dates of each planetary encounter based on the launch date and the time of flight of each leg of the trajectory. It then acquires the position and velocity of Earth at the time of launch and the specified planet(s) as well as the target at the dates calculated. This provides all the information needed to use Lambert's problem solution to determine the velocities at the endpoints of the trajectory that connects the two positions in the travel time given. In some cases, the algorithm to determine the velocities is unable to converge. Should this happen, an infinite amount of ΔV is applied to the mission total to penalize the solution.

At this point, the arrival and departure velocities at each flyby are known as well as the departure velocity from Earth and the arrival velocity at the target. The algorithm assumes that the spacecraft starts in a 300 km altitude circular orbit around Earth and includes the propellant used to depart Earth in the total mission cost. Given the departure velocity dictated by Lambert's problem, ΔV needed to leave Earth is calculated. Since our target is so small compared to any planet, the arrival ΔV is simply the ΔV needed to match velocity with the target.

All that is left at this point is to calculate ΔV consumed at each flyby. An ideal flyby would not require any ΔV as the approach and departure speeds would be the same and the planet's gravity would suffice to perform any required velocity turn without having to approach too close. In reality, the speeds will likely not match, requiring a maneuver to change orbital energy and the periapse will be dictated by the velocity turn required. If the speeds are too high and the required turn too great, the only mathematical solutions may have the periapse within the planet itself which in reality is a null solution.

To calculate the flyby maneuver, the algorithm treats the problem as the patching of two hyperbolic orbits that meet at the periapse and with V_∞ speeds determined by the solution to Lambert's problem. This is illustrated in Figure 5. The algorithm iterates on the periapse to find an altitude at which the combined half-turns from the two hyperbolas equal the required turn. From the difference in orbital energy of the two planet-centric trajectories at the final altitude, ΔV is calculated and added to the mission total. If the required periapse falls below the planet's surface, an infinite amount of ΔV is added to the mission total to penalize the solution.

4. Trajectory Planning to Hylonome

4.1. Hylonome's Orbit. Hylonome is classed as a Centaur object, having a semimajor axis that is between that of Jupiter and Neptune, Jet Propulsion Laboratory [15]. Its orbit is shown in Figure 6. Its orbital characteristics are shown in Table 5, Jet Propulsion Laboratory [15].

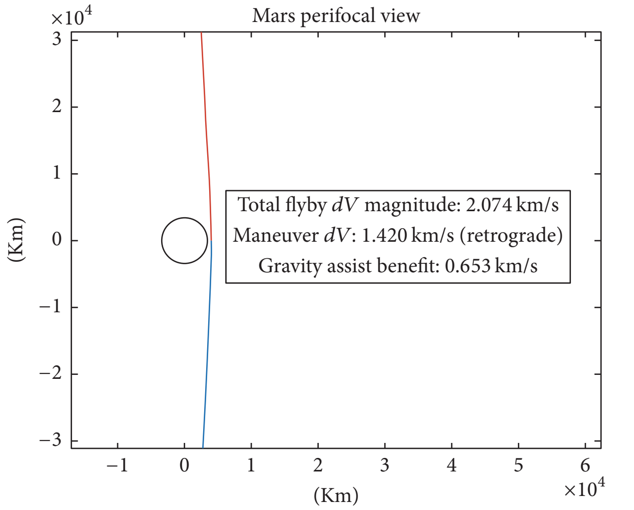

Figure 5: Illustration of the periapse patching of the planet-centric inbound and outbound hyperbolic orbits. Inbound is the blue (bottom) and outbound is the red (top).

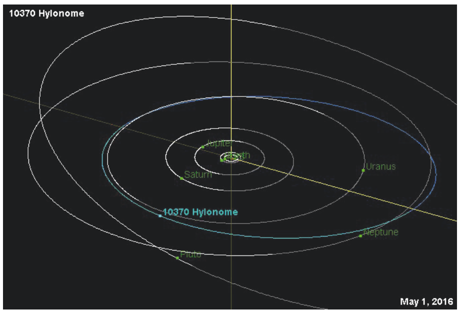

Figure 6: Hylonome's orbit in relation to the orbits of the planets as referenced on 1 May 2016.

4.2. Simulation Results. Hylonome was chosen as a target due to its large orbit relative to all other asteroids and small bodies that have been investigated to date. A direct launch to this target would be very difficult if not impossible with today's rocket technology due to tremendous ΔV that would be required to travel out to the target and then to match speed with it. For this reason, a flyby of an intervening planet would be necessary to reach the target with a reasonable fuel allocation. The genetic algorithm was run as before for a direct trajectory, a single flyby, two flybys, and three flybys. Table 6 shows the best ΔV from all simulation types.

The best result occurred with a single flyby of Jupiter and is discussed here in more detail. Table 7 shows the chromosome from the single flyby case that yielded the best results of all simulations run. It was obtained from performing 12 runs of the algorithm. The results in Table 8 were obtained for the trajectory to Hylonome with a flyby

Table 5: Orbital characteristics of Hylonome.

Element	Value	Uncertainty (1 – sigma)	Units
e	.2482	$3.8913e-05$	
a	25.1017	0.0016667	au
q	18.8690	0.00028158	au
i	4.1442	$8.9499e-05$	deg
Node	178.0727	0.00055742	deg
Peri	6.5554	0.0045953	deg
M	60.5576	0.0086221	deg
tp	2449873.32 (1995-Jun-04.82286853)	0.33896	JED
Period	45936.0887346103 125.77	4.575 0.01253	d yr
n	.007836	$7.8052e-07$	deg/d
Q	31.3343	0.0020805	au

Table 6: Mission ΔV to Hylonome for multiple flyby scenarios.

	Direct	Single	Double	Triple
Total ΔV [km/sec]	13.452	9.420	10.714	93.611
Orbiter ΔV [km/sec]	3.823	2.543	3.799	89.866

Table 7: Optimized chromosome for single planet flyby to Hylonome.

Parameter	Value
J_0	November 16, 2003 00:46:14.6 UTC
p_1	Jupiter
dt_1	679 days, 8 hours
dt_2	8720 days, 7 hours

Table 8: Single planet flyby results from Earth to Hylonome.

Result	Value
V_∞ from Earth	[−6.706 6.666 1.961] km/s
V_∞ to p_1	[−8.356 1.634 −0.093] km/s
V_∞ from p_1	[−6.492 −5.393 1.129] km/s
V_∞ to Target	[−1.913 −1.497 0.152] km/s
Total Flyby ΔV at p_1	[1.864 −7.027 1.222] km/s
Total Flyby ΔV	
Magnitude at p_1	7.372 km/s
Maneuver ΔV at p_1	0.000 km/s
Gravity assist benefit	7.372 km/s
Maneuver ΔV at Earth	6.856 km/s
Maneuver ΔV at target	2.434 km/s
Total ΔV	9.290 km/s
Total ΔV of orbiter	2.434 km/s

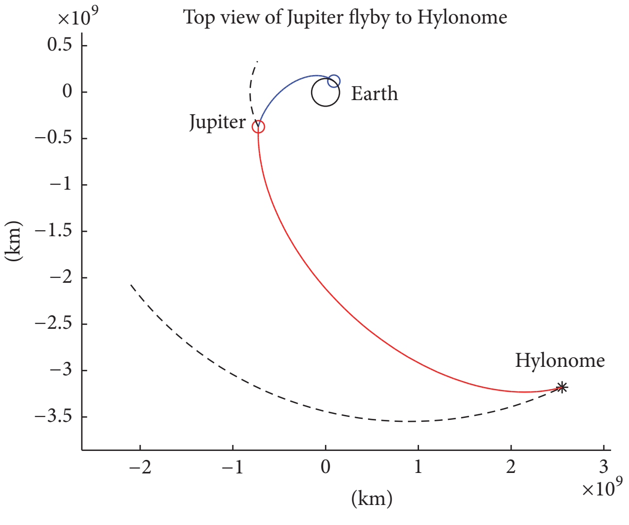

Figure 7: Single flyby trajectory simulation results.

gravity assist from Jupiter. Figure 7 shows the trajectory computed.

5. Trajectory Planning to Hidalgo

5.1. Hidalgo's Orbit. Like Hylonome, Hidalgo is classed as a Centaur object, having a semimajor axis that is between that of Jupiter and Neptune. Its orbit is shown in Figure 8. Its

Table 9: Orbital characteristics of Hidalgo.

Element	Value	Uncertainty (1 − sigma)	Units
e	.6611	$8.6408e-08$	
a	5.7388	$6.3347e-08$	au
q	1.9451	$4.9074e-07$	au
i	42.5231	$2.2714e-05$	deg
Node	21.4255	$1.4532e-05$	deg
Peri	56.7263	$2.1274e-05$	deg
M	301.3931	$8.0923e-06$	deg
tp	2458417.9712 (2018-Oct-26)	0.0001236	JED
Period	5021.4195 13.75	$8.3142e-05$ $2.276e-07$	d yr
n	.07169	$1.1871e-09$	deg/d
Q	9.5324	$1.0522e-07$	au

Table 10: Mission ΔV to Hidalgo for multiple flyby scenarios.

	Direct	Single	Double	Triple
Total ΔV	13.889	8.043	11.187	29.255
Orbiter ΔV	7.055	0.574	7.485	25.580

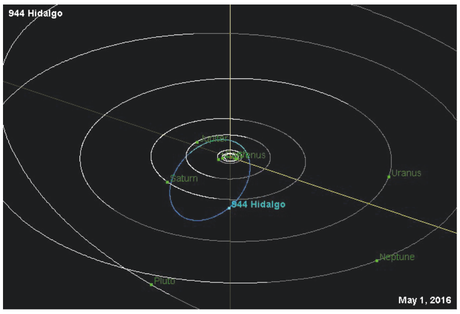

Figure 8: Hidalgo's orbit in relation to the orbits of the planets as referenced on 1 May 2016.

Table 11: Optimized chromosome for single planet flyby to Hidalgo.

Parameter	Value
J_0	November 16, 2003 00:41:08.7 UTC
p_1	Jupiter
dt_1	539 days, 4 hours
dt_2	4045 days, 16 hours

Table 12: Single planet flyby results from Earth to Hidalgo.

Result	Value
V_∞ from Earth	[−8.211 6.459 1.539] km/s
V_∞ to p_1	[−12.131 1.683 0.059] km/s
V_∞ from p_1	[−10.010 2.206 −6.700] km/s
V_∞ to Target	[0.211 0.105 0.522] km/s
Total flyby ΔV at p_1	[2.121 0.523 −6.759] km/s
Total flyby ΔV Magnitude at p_1	7.103 km/s
Maneuver ΔV at p_1	0.001 km/s
Gravity assist benefit	7.101 km/s
Maneuver ΔV at Earth	7.469 km/s
Maneuver ΔV at target	0.573 km/s
Total ΔV	8.043 km/s
Total ΔV of orbiter	0.574 km/s

orbital characteristics are shown in Table 9, Jet Propulsion Laboratory [15].

5.2. Results. Hidalgo was chosen as a target due to the large inclination of its orbit to the ecliptic plane. A direct launch to this target would be very difficult if not impossible with today's rocket technology. For this reason, a flyby of an intervening planet would be necessary to reach the target with a reasonable fuel allocation. The genetic algorithm was run as before for a direct trajectory, a single flyby, two flybys, and three flybys. Table 10 shows the best ΔV from all simulation types.

The best result occurred with a single flyby of Jupiter and is discussed here in more detail. Table 11 shows the chromosome from the single flyby case that yielded the best results of all simulations run. It was obtained from performing 10 runs of the algorithm. The results in Table 12 were obtained for the trajectory to Hidalgo with a flyby

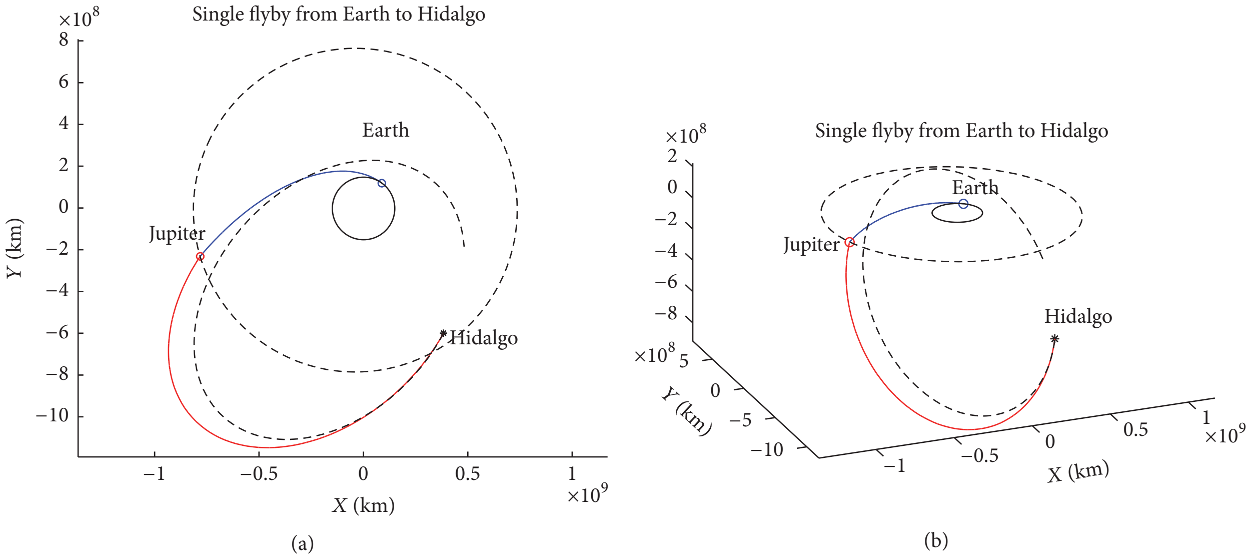

FIGURE 9: Single flyby trajectory simulation results.

gravity assist from Jupiter. Figure 9 shows the trajectory computed.

6. Trajectory Planning to Ceres: A Revisit

Finally, we revisit analysis done previously on planning a trajectory to Ceres and provide an extension to the study of Fritz and Turkoglu [16]. In previous work, it was noted that the mission ΔV was not decreasing for every flyby added to the trajectory. Rather, although there was a decrease in mission ΔV from the direct trajectory to the single flyby trajectory, the mission ΔV then increased for two flybys and again for three flybys. This may seem counterintuitive given that flybys are used to increase mission fuel efficiency and/or perform maneuvers (such as change inclination) that would be impossible to do with propulsion alone. There are two main reasons for seeing this behavior where two or more flybys are less efficient than a single flyby.

First, the fitness function is not presently capable of handling all possible combinations of values in the chromosomes. Examples of situations that the fitness function cannot correctly calculate include (a) performing multiple flybys of a single planet in a row; (b) completing more than a single orbit prior to encountering the next target; and (c) performing deep-space maneuvers that would optimize the approach to flyby encounters. These limitations preclude a number of known trajectory optimization techniques. The first limitation in particular is a technique commonly used to reach the outer solar system by receiving a gravity assist from Earth one or more times after launch with no other encounters in between.

Second, the limited date range used means that system level synodic periods may not complete even a single cycle in the time allowed for trajectories involving more than two or three bodies. Figure 10 shows the angular misalignment from an optimum configuration over time. Systems involving three planets and the target Ceres have a synodic period greater than even 200 years. If the system presented to the GA should look like the time period after the 100-year mark in the third plot of Figure 10, the GA will likely not find a solution better than the single flyby option which has a much shorter cycle if one is willing to approximate.

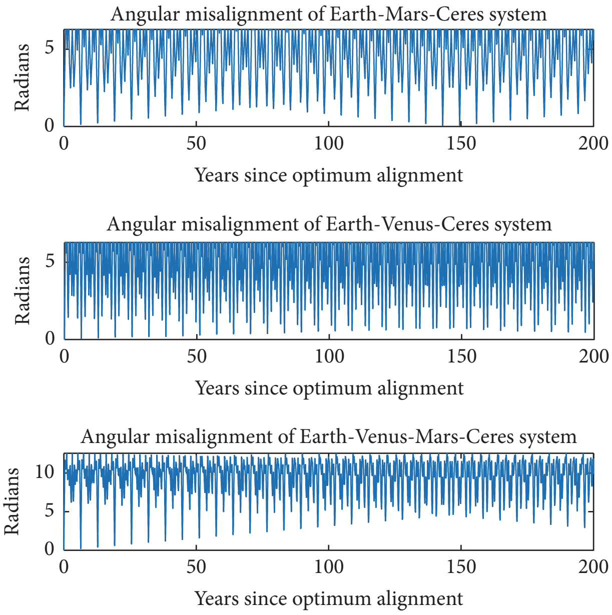

FIGURE 10: Illustration of the system misalignment from a presumed optimum at year 0.

To test the second statement, the GA was set to find trajectories to Ceres with a greatly expanded time range. Launch dates starting as early as 1904 were included with arrival dates ending in 2099. The GA had its parameters and limits modified slightly to account for the reduced travel time

Table 13: Mission total and orbiter ΔV to Ceres for 30-year time span and 200-year time span.

	Single	Double	Triple
30 years			
Total	10.971 Km/s	10.021 Km/s	16.203 Km/s
Orbiter	5.277 Km/s	6.118 Km/s	10.101 Km/s
200 years			
Total	10.340 Km/s	10.365 Km/s	13.229 Km/s
Orbiter	5.457 Km/s	5.934 Km/s	8.796 Km/s

needed to reach Ceres over the Centaur asteroids discussed earlier. All Δt_x genes were given a minimum of one month and a maximum of 16 months and the date range was also widened to accommodate the new data set. The single flyby, double flyby, and triple flyby were reexamined and compared to the trajectories obtained with the thirty-year data. The results are tabulated in Table 13. It is immediately obvious that trajectories involving multiple flybys suffer from a reduced time span in which to find a solution. For this reason, any future work on this project should focus on removing the shortcomings listed earlier from the fitness function. The ability to return to the same planet two or more times in a row alone would alleviate the time span problem by providing many more solutions.

7. Conclusions

This paper has discussed the development and application of a genetic algorithm to the optimization of interplanetary trajectories to distant and unusual targets. The algorithm is capable of converging on a physically viable solution on every simulation, and only a very small number of simulation runs are required to obtain confidence in the strength of the best solution. The algorithm proved capable of reaching targets located in the inner solar system and the outer solar system, as well as targets whose orbits are highly inclined to the ecliptic plane. This level of robustness was obtained with just the most basic of genetic algorithm capabilities and with severe limitations on the capabilities of the fitness function to assess a valuable class of trajectories.

The algorithm, though working, could use some improvements to enable better searching and refinement of results. It has been proposed that occasional repopulating of a portion of certain generations could aid in potentially avoiding local minimums while minimizing the number of low population simulations required. Improvements in the fitness function could yield much better solutions if the capability to assess resonant gravity assists could be added.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] R. Xu, P. Cui, D. Qiao, and E. Luan, “Design and optimization of trajectory to Near-Earth asteroid for sample return mission using gravity assists,” *Advances in Space Research*, vol. 40, no. 2, pp. 220–225, 2007.

[2] S. Molenaar, *Optimization of interplanetary trajectories with deep space maneuvers - model development and application to a uranus orbiter mission* [Master’s thesis], Delft University of Technology, Netherlands, 2009.

[3] C. R. H. Solórzano, A. A. Sukhanov, and A. F. B. A. Prado, “A study of trajectories to the Neptune system using gravity assists,” *Advances in Space Research*, vol. 40, no. 1, pp. 125–133, 2007.

[4] D. Izzo, V. M. Becerra, D. R. Myatt, S. J. Nasuto, and J. M. Bishop, “Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories,” *Journal of Global Optimization*, vol. 38, no. 2, pp. 283–296, 2007.

[5] M. Vasile and P. De Pascale, “Preliminary design of multiple gravity-assist trajectories,” *Journal of Spacecraft and Rockets*, vol. 43, no. 4, pp. 794–805, 2006.

[6] A. Gad and O. Abdelkhalik, “Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization,” *Journal of Spacecraft and Rockets*, vol. 48, no. 4, pp. 629–641, 2011.

[7] A. E. Petropoulos and J. M. Longuski, “Shape-based algorithm for automated design of low-thrust, gravity-assist trajectories,” *Journal of Spacecraft and Rockets*, vol. 41, no. 5, pp. 787–796, 2004.

[8] S. Lee, P. Von Allmen, W. Fink, A. E. Petropoulos, and R. J. Terrile, “Design and optimization of low-thrust orbit transfers,” in *Proceedings of the 2005 IEEE Aerospace Conference*, IEEE, Big Sky, MT, USA, March 2005.

[9] C. Wen, Y. Zhao, and P. Shi, “Orbital accessibility problem for spacecraft with a single impulse,” *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1260–1271, 2014.

[10] C. Wen, Y. Zhao, and P. Shi, “Precise determination of reachable domain for spacecraft with single impulse,” *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, pp. 1767–1779, 2014.

[11] H. Oberth, “Ways to spaceflight. Number 622,” *National Aeronautics and Space Administration*, 1972.

[12] J. E. Prussing and B. A. Conway, *Orbital mechanics*, Oxford University Press, USA, 1993, Orbital mechanics.

[13] H. D. Curtis, *Orbital Mechanics for Engineering Students*, Elsevier, 2007.

[14] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Natural Computing Series, Springer, Berlin, Germany, 1st edition, 2003.

[15] Jet Propulsion Laboratory, “JPL small-body database browser,” http://ssd.jpl.nasa.gov/sbdb.cgii.

[16] S. Fritz and K. Turkoglu, “Optimal trajectory determination and mission design for asteroid/deep space exploration via multi-body gravity assist maneuvers,” in *Proceedings of the 2016 IEEE Aerospace Conference, (AERO ’16)*, Big Sky, MT, USA, March 2016.