AIAA 2012-4874

Analysis and Selection of Optimal Targets for a Planetary Defense Technology Demonstration Mission

Sam Wagner, * Tim Winkler, † and Bong Wie ‡

Asteroid Deflection Research Center, Iowa State University, Ames, IA, USA

In the past; various options have been proposed for mitigating the impact threat of near-Earth objects (NEOs. This paper presents mission analysis and design to potential NEO candidates for a demonstration of such planetary defense technologies. In particular, a flight demonstration mission is necessary to validate the effectiveness of blending a hypervelocity kinetic impactor with a penetrated nuclear subsurface explosion. Possible asteroid candidates are found by determining optimal missions for several possible mission architectures for nearly 3400 potential asteroids. The optimization is performed using a genetic algorithm developed to solve complex mission design problems.

I. Introduction

As early as 1992, the idea of discovering and tracking near-Earth objects (NEOs) was proposed to the U.S. Congress.[1] The search, called the Spaceguard Survey, was later implemented in 1998 with the ultimate goal of finding 90 percent of the estimated asteroid population of 1 km in diameter or larger by 2008. This survey only intended to find asteroids that were large enough to cause catastrophe on a global scale. While not large enough to affect the entire globe, impacts with objects smaller than 1 km occur more frequently, and cause significant damage upon impact. In 2005, the George E. Brown, Jr. Near-Earth Object Survey Act expanded the search to include the detection and characterization of NEOs as small as 140 m. The new objectives retained the goal of detecting 90 percent of the NEO population, but extended the completion date to the end of 2020. To date, none of the discovered objects are predicted to be on a collision course with the Earth, but the survey still has 8 more years until its completion. Should a new NEO be discovered to impact the Earth, mitigation measures would have to be taken to prevent a potential disaster.

Given a lead time of at least 10 years, various proposed technologies such as kinetic impactors, slow-pull gravity tractors, or solar sails could be employed to successfully mitigate an impact threat. When the warning time is short, however, nuclear technologies for a standoff, contact, or subsurface explosion may be the only viable options. Currently none of the aforementioned mitigation options have been validated on a flight demonstration mission. To that end, research projects at the Asteroid Deflection Research Center (ADRC) have proposed a hypervelocity nuclear interceptor system (HNIS) concept to be used in a planetary defense technology (PDT) demonstration mission.[2–4] Preliminary target recommendations for a PDT demo mission have been made, but are restricted to a time frame which may not provide sufficient time to develop and launch the HNIS. This study will expand target list presented in Ref. 4, but with later launch windows to allow the development of the the HNIS. Consideration will also be given for an observation spacecraft to determine the success of the PDT mission spacecraft as well as a possible, separate sample return mission.

A. Previous NEO Missions and Proposals

Space agencies such as JAXA, ESA, and NASA have already conducted several successful exploration missions to asteroids and comets. Some notable missions that demonstrated NEO targeting and/or landing abilities were Hayabusa by JAXA, NEAR Shoemaker, and Deep Impact by NASA. Hayabusa (formerly MUSES-C) was sent to the asteroid 25143 Itokawa, a 330-m diameter asteroid. While there, the spacecraft

*Graduate Research Assistant, Department of Aerospace Engineering, 2271 Howe Hall, AIAA Student Member.
†Graduate Research Assistant, Department of Aerospace Engineering, 2271 Howe Hall, AIAA Student Member.
‡Vance Coffman Endowed Chair Professor, Dept. of Aerospace Engineering, 2271 Howe Hall, AIAA Associate Fellow.

performed two landings to collect surface samples, which were returned to Earth in June 2010.[5] The spacecraft also had a small lander called MINERVA that was to be guided to the surface of the asteroid, but unfortunately drifted into space due to the low gravity. NEAR Shoemaker was designed to study 433 Eros, one of the largest NEOs. The spacecraft became the first to orbit an asteroid as well as the first to land on one. Whereas Hayabusa and NEAR softly touched down on their respective asteroids, Deep Impact aimed to do just the opposite. Approximately 24 hours before impact with the comet Tempel 1, the impactor was separated from the flyby craft and autonomously navigated to ensure a hypervelocity impact close to 10 km/s with the 5-km target.

More recently, the ESA proposed a demonstration mission for a kinetic-impactor called the Don Quijote mission.[6,7] The mission concept called for two separate spacecraft to be launched at the same time, but follow different interplanetary trajectories. Sancho, the orbiter spacecraft, would be the first to depart Earth's orbit, and rendezvous with a target asteroid approximately 500 m in diameter. Sancho would measure the position, shape, and other relevant characteristics before and after a hypervelocity impact by Hidalgo, the impactor spacecraft. After Sancho has studied the target for some months, Hidalgo approaches the target at a closing speed around 10 km/s. Sancho then observes any changes in the asteroid after the kinetic impact to assess the effectiveness of this deflection strategy. Don Quijote was planned to launch in early 2011 and conclude in mid to late 2017. However, the mission concept was never realized due to higher than expected mission costs.

Table 1. Target Selection Criteria for the Don Quijote Mission.

Orbit Characteristics	Preferred Range
Rendezvous ΔV	< 7 km/s
Orbit type	Amor
MOID	large and increasing
Orbit accuracy	well determined orbits
Physical Characteristics	Preferred Range
Size	< 800 m
Density	$\sim$1.3 g/cm^3
Absolute magnitude	20.4 - 19.6
Shape	not irregular
Taxonomic type	C-type
Rotation period	< 20 hours
Binarity	not binary

The selection process for the Don Quijote mission was based on a set of NEO characteristics defined by ESA's NEOMAP in Table 1.[8] Their analysis resulted in the selection of the asteroids 2002 AT4 and 1989 ML. As can be noticed in Table 2, 2002 AT4 is roughly half the size of 1989 ML, but requires a higher ΔV in order to intercept. A realistic deflector spacecraft would require a versatile design capable of intercepting and deflecting or disrupting both kinds of targets on short notice.

Currently at the ADRC, a hypervelocity nuclear interceptor system (HNIS) concept is being investigated for a high-energy disruption/fragmentation mission that may be inevitable for the most probable impact threat with a short warning time.[9–11] While the Don Quijote mission concept considered deflecting an asteroid with a kinetic impactor, the ADRC is focusing on a high-energy deflection/disruption demo mission by means of a standoff or contact explosion or disrupting/fragmenting the asteroid into smaller, less threatening pieces using a penetrating subsurface nuclear explosion. The latter option is accomplished by an innovative, two body penetrator design, which allows a nuclear explosive device (NED) to be detonated inside the asteroid itself to facilitate a more efficient energy transfer from the explosion.[9–11]

B. Near-Earth Asteroid (NEA) Groups

For the purposes of this study, only asteroids in the near-Earth asteroid (NEA) groups Atira, Apollo, Aten, and Amor were considered. Comparisons of typical asteroids in each of these groups are shown in figure 1. Asteroids in these groups all have perihelion distances of 1.3 AU or less, and many of them also cross the

Table 2. Properties of Candidate Targets Considered for the Don Quijote Mission.

	2002 AT4	1989 ML
Orbital period (yr)	2.549	1.463
e	0.447	0.137
i (deg)	1.5	4.4
ΔV (km/s)	6.58	4.46
Orbit type	Amor	Amor
MOID	large	large
Absolute magnitude	20.96	19.35
Taxonomic type	D-type	E-type
Diameter (m)	380	800
Rotational period (hr)	6	19

Earth's orbit at some point. Asteroids in these groups are relatively close to the Earth, and have low ΔV requirements to achieve intercept. As such, objects in these groups are the most likely candidates for an asteroid deflection/disruption demonstration mission. Apollo and Aten class asteroids are characterized by asteroids with orbits that intersect that of the Earth, which could potentially lead to lower ΔV requirements for a mission. On the other hand, this same fact means that any significant perturbation in the object's trajectory could cause it to later impact the Earth. While unlikely, a demonstration of deflection technologies could cause this to happen. The ESA also had this in mind when they selected the asteroids 2002 AT4 and 1989 ML from the Amor group for the Don Quijote mission concept.[6] With that in mind, the Amor and Atira groups shall be the focus for determining suitable candidates in this paper.

The Amor asteroid group is characterized by asteroids that approach the Earth, but do not actually cross its orbit. By definition the perihelion distances of these asteroids lie between 1.017 and 1.3 AU. As of July 21st, 2012, there are 3398 Amor and Atira class asteroids listed in NASA's Near Earth Object Program database. While Amor asteroids are entirely outside of the Earth's orbit, the Atira asteroid group is characterized by asteroids whose orbits are contained entirely within the orbit of the Earth. As the orbits of both of the groups is entirely inside or outside that of the Earth, any disturbance in the trajectories of these asteroids is not likely to cause an impact the Earth at any time after the mission.

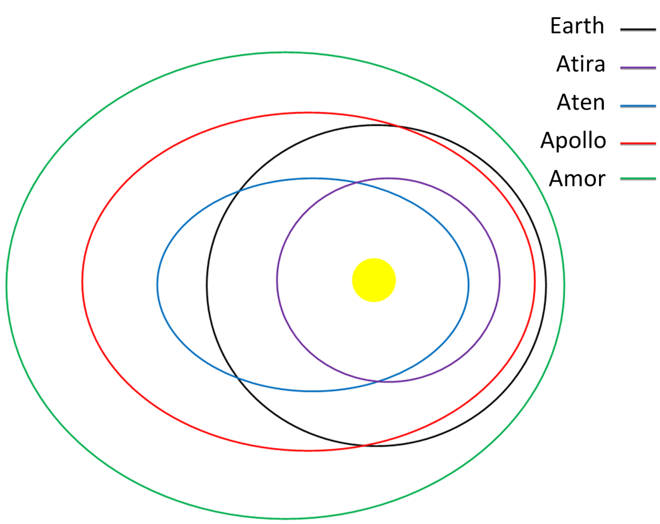

Figure 1. Comparison of Atira, Apollo, Aten, and Amor class asteroid orbits in relation to the Earth's orbit.

C. Mission Design Software

Due to the large of variables in these types of missions, an exhaustive search of all 3398 asteroids would be impractical. All mission design computation will be performed using an evolutionary algorithm known as a genetic algorithm. The types of missions considered in this paper can easily be formulated as constrained optimization problems, making them ideal for evolutionary algorithms. Each mission type must first formulated as a single-valued cost function. Several types of missions have been considered and are detailed in the following section. Detail of the development of the genetic algorithm is also provided in this paper.

II. Problem Formulation

All missions evaluated in this paper have two main requirements. The main spacecraft must return a sample of the asteroid to the Earth and the hypervelocity impactor must impact the asteroid at no less than 5 km/s. A list other various mission requirement is shown in Table 3.

Table 3. List of mission requirement.

Asteroid Types	Amor, Atira
Earliest Launch Date	1-Jan-15
Mission Completed by	1-Jan-40
Max. Earth Entry Velocity (km/s)	12.000
Minimum Impact Vel (km/s)	5.000
Impactor Limitations	Must occur while the main s/c is at the asteroid

Three types of missions have been considered. The first mission type, referred to as Type 1 missions, consists of two separate launches. One launch for the main spacecraft and one for the impactor as well. This is the simplest type of mission to formulate. The main spacecraft has two mission legs, Earth departure through asteroid arrival and asteroid departure up to Earth atmospheric entry. The impactor has only one mission phase, Earth departure to asteroid impact. The two other missions require a single launch. The main spacecraft has the same two phases as before, while the hypervelocity impactor performs a deep-space maneuver prior to impact. In the case of the third mission type (referred to as Type 3) the impactor has also have a gravity assist at Venus prior to impact. In this type of mission the impactor typically gains velocity during the Venus flyby, resulting in higher impact velocities and significantly lower mission costs. Table 4 illustrates the different elements that impact the cost function for each mission type.

Table 4. Break down of mission critical elements for the 3 mission types.

	Type 1 Mission		Type 2 Mission		Type 3 Mission	
	Main S/C	Impactor	Main S/C	Impactor	Main S/C	Impactor
Launch	X	X	X		X	
DSM (km/s)				X		X
Venus Powered GA						X
Asteroid Arrival ΔV (km/s)	X	X	X	X	X	X
Asteroid Departure ΔV (km/s)	X		X		X	
Earth Arrival Δ V (km/s)	X		X		X	

The purpose of this section is to develop all the necessary pieces of the cost functions for each of the three methods. The cost functions typically consists of all of the required mission ΔVs, the Earth departure V_∞, and any other required mission constraints. There are several possible mission ΔVs. The main spacecraft's atmospheric entry velocity is limited to a maximum of 12 km/s, often times requiring an additional ΔV to slow down the spacecraft prior to reentry. The hypervelocity impactor leg(s) also contribute to the cost function through deep-space targeting maneuvers and powered gravity assists. If the impactor has a v_∞ of less than 5 km/s a penalty is added to ensure the impact velocity mission requirement is met.

Each leg of the missions are calculated using solutions to Kepler's equation and Lambert's problem. In missions types 2 and 3 the main spacecraft and hypervelocity impactor are launched together. The impactor performs a deep-space maneuver prior to the main spacecraft asteroid arrival. The ΔV required by this deep-space maneuver is found in the following manner. First, a burn index ϵ, which ranges from 0 to 1, is used to define when the impactor will separate and perform the maneuver. To determine the magnitude of the deep-space maneuver, the orbit is propagate out by a time ϵT_0, where T_0 is the time of flight for the Earth departure to asteroid arrival leg of the main spacecraft. A solution to Lambert's problem is then used to target the next phase of the impactor mission. In type 2 missions, this is the asteroid impact. While in type 3 missions then next target is a gravity assist at Venus. The magnitude of the ΔV is simply the magnitude of the velocity differences from the Kepler and Lambert solutions. In the following section the model for the powered gravity assist, sometimes referred to as the MGA model, is discussed.

A. Multiple Gravity-Assist Model

For the multiple gravity-assist model, two Lambert solutions are essentially "patched" together using the standard patched conic method. This results in a powered hyperbolic orbit for each gravity assist in which a ΔV is allowed only at the perigee passage for each gravity assist. The ΔV at each gravity assist is part of the final cost function, but is usually driven to a near zero value.

For each gravity assist, the incoming and outgoing velocity vectors (in the heliocentric frame) are given from Lambert solutions. The velocities relative to the planets are then found as

$$\vec{v}_{\infty-in} = \vec{V}_{s/c-in} - \vec{V}_{\oplus} \tag{1}$$

$$\vec{v}_{\infty-out} = \vec{V}_{s/c-out} - \vec{V}_{\oplus} \tag{2}$$

From this point the goal is to determine a method to find the perigee radius that is required to patch the two solutions together. The first step is to determine the semi-major axis of the incoming and outgoing hyperbolic trajectories.

$$a_{in} = -\frac{\mu_{\oplus}}{v_{\infty-in}^2} \tag{3}$$

$$a_{out} = -\frac{\mu_{\oplus}}{v_{\infty-out}^2} \tag{4}$$

where $\mu_{\oplus}$ is the target planet's gravitational parameter.

The required turning angle is

$$\delta = \cos^{-1}\left(\frac{\vec{v}_{\infty-in} \cdot \vec{v}_{\infty-out}}{v_{\infty-in} \cdot v_{\infty-out}}\right) \tag{5}$$

The flyby perigee radius is equal for both legs of the hyperbolic orbit. That is, we have

$$r_p = a_{in}(1 - e_{in}) = a_{out}(1 - e_{out}) \tag{6}$$

where e_{in} and e_{out} are the incoming and outgoing orbit eccentricities. It should be noted that the orbits will be hyperbolic, so both eccentricities will be greater than 1. The turning angle δ can also be represented as the sum of the transfer angles for the incoming and outgoing orbits.

$$\delta = \sin^{-1}\left(\frac{1}{e_{in}}\right) + \sin^{-1}\left(\frac{1}{e_{out}}\right) \tag{7}$$

Equation 6 can be then rewritten for e_{in}, as

$$e_{in} = \frac{a_{out}}{a_{in}}\left(e_{out} - 1\right) + 1 \tag{8}$$

Substituting Eq. (8) into Eq. (7) gives

$$\delta = \sin^{-1}\left(\frac{1}{\frac{a_{out}}{a_{in}}\left(e_{out} - 1\right) + 1}\right) + \sin^{-1}\left(\frac{1}{e_{out}}\right) \tag{9}$$

which can be rewritten as

$$f = \left(\frac{a_{out}}{a_{in}}(e_{out} - 1)\right) \sin\left(\delta - \sin^{-1}\left(\frac{1}{e_{out}}\right)\right) - 1 = 0 \tag{10}$$

The above equation, which is now only a function of e_{out}, can be iterated upon to solve for e_{out}. A simple Newton iteration scheme works well. For this the derivative of f with respect to e_{out} must first be found.

$$\frac{df}{de_{out}} = \left(\frac{a_{out}}{a_{in}}e_{out} - \frac{a_{out}}{a_{in}} + 1\right) \frac{\cos\left(\delta - \sin^{-1}\frac{1}{e_{out}}\right)}{e_{out}^2\sqrt{1 - \frac{1}{e_{out}^2}}} + \frac{a_{out}}{a_{in}} \sin\left(\delta - \sin^{-1}\frac{1}{e_{out}}\right) \tag{11}$$

To start the Newton iteration an initial value for e_{out} of 1.5 works well. In a typical Newton iteration scheme a do while loop is used until e_{out} stops changing within a certain tolerance. The iteration number inside the loop should also be monitored, so the loop can be exited if convergence doesn't occur. Each new e_{out} is calculated as

$$e_{new} = e_{old} - \frac{f}{\frac{df}{de_{out}}} \tag{12}$$

When a converged e_{out} is found, the perigee radius is calculated from Eq. (6). Finally the ΔV that must be applied to the perigee is obtained as.

$$\Delta V_{GA} = \left| \sqrt{v_{\infty-in}^2 + \frac{2\mu_{\oplus}}{r_p}} - \sqrt{v_{\infty-out}^2 + \frac{2\mu_{\oplus}}{r_p}} \right| \tag{13}$$

The flyby perigee radius and ΔV are found directly from the spacecraft's incoming and outgoing velocities. These are found from solutions to Lambert's problem, which are a function of planetary positions and time of flight. The planetary positions are also a function of time, meaning that the decision variables for the MGA model are the time of Earth departure, time of each gravity assist, and final arrival time. Thus the final cost function C for the MGA portion of the problem is:

$$C = f(\mathbf{X}) + g(\mathbf{X}) \tag{14}$$

$$\mathbf{X} = [T_1, T_2]^T \tag{15}$$

In this case T_i is the time of flight for each leg of the mission, and T_2 is time of flight for the final mission leg. The penalty function, $g(\mathbf{X})$ will be covered later. As shown above each element of the cost function, ΔVs, etc, are only a function of time. This MGA cost function can then be optimized by the genetic algorithm to find optimal or near optimal solutions.

B. Problem Constraints

In optimization problems, constraints are often used to help shape the final solution. When implementing genetic algorithms, constraints are used, rather than hard limits, in order to keep to solution space open. If the solution space isn't sufficiently large the genetic algorithm will be unable to start, due to the random initialization process.

Common constraints for mission design problems include, perigee radius during a gravity assist, mission time/leg length constraints, and guarding against low velocity flybys. During a low-velocity flyby the spacecraft is captured by the planet, rather than gaining velocity in the heliocentric frame. Any other constraint the user wishes impose to help shape the solution can be added, as long as the constraint values are approximately the same order of magnitude at the actual cost function values. It should be noted that when using the MGA-DSM model in conjunction with a genetic algorithm all of the variables can be explicitly constrained. However, low-velocity flybys can still occur.

Often time, the MGA model results in a perigee radius that passes through the planet or too close to the planets atmosphere. In this situation a constraint-handling method is necessary to move the solution toward more feasible solutions. The method below is from Englander et al.[12]

$$g_i(\mathbf{X}) = -2\log\frac{R_{pi}}{kR_{\oplus}} \tag{16}$$

Here k is a multiplier used to define how close the spacecraft is allowed to flyby a target planet. For all of the examples in this paper a value of 1.1 was used. The only exception was when reproducing the Galileo mission, which had an Earth close approach altitude of 300 km for the second gravity assist (k value of approximately 1.047).

The second constraint penalty method penalizes low velocity flybys. The method has also been adapted from Englander et al.[12] Low velocity flybys are very rare, but solutions should still be protected from converging on them. The orbital energy, about the flyby planet, can be calculated as.

$$E = \frac{\left|\vec{v}_{\infty-in}\right|^2}{2} - \frac{\mu_{\oplus}}{R_{soi}} \tag{17}$$

For the flyby orbit to be hyperbolic about the planet, E must be greater than zero. However, the sphere of influence model is an approximation, so an additional 10% margin on the incoming velocity needs to be added. A simple penalty that is scaled inversely to the flyby arrival velocity scales well with. For relatively large v_∞ the penalty is zero or very small compared to the overall cost function value. Alternatively for very low v_∞ values the penalty is large enough to severely influence the final shape of the solution. The flyby orbital energy, adjusted for the 10% margin, and final constraint are calculated using the following two equations:

$$E = \frac{\left|0.9\vec{v}_{\infty-in}\right|^2}{2} - \frac{\mu_{\oplus}}{R_{soi}} \tag{18}$$

$$g_i(\mathbf{X}) = \begin{cases} 0 & E \geq 0 \\ \frac{1}{\left|\vec{v}_{\infty-in}\right|} & E < 0 \end{cases} \tag{19}$$

Time constraints are also very important for this problem because the hypervelocity impactor must impact the asteroid after the main spacecraft arrives and prior to asteroid departure. The time penalty constraints are calculated as follows.

$$g_i(\mathbf{X}) = \begin{cases} 0.1(T_{S/C-arr} - T_{impact}) & T_{impact} < T_{S/C-arr} \\ 0 & T_{S/C-arr} \leq < T_{impact} \leq T_{S/C-dep} \\ 0.1(T_{impact} - T_{S/C-dep}) & T_{impact} > T_{S/C-dep} \end{cases} \tag{20}$$

With this constraint the spacecraft is penalized if the impactor arrives at the asteroid prior to the main spacecraft or a is the impactor arrives after the main spacecraft has left the asteroid. The ensure that only solutions are found where the main spacecraft will be able to observe the impact and collect a sample of the impacted area/debris.

$$g(\mathbf{X}) = \sum g_i(\mathbf{X}) \tag{21}$$

With the cost function for each method finalized then next step is to develop the genetic algorithm, which will be capable of optimizing these types of advance mission design problems.

III. Overview of Genetic Algorithms

The heart of a genetic algorithm is the simulation of natural selection, reproduction, and mutations found in nature. Genetic operators are used to 'evolve' an initial population, through genetic operators, in order to determine a best fitness design.[13] The purpose of this section is to develop and implement a genetic algorithm for the MGA, MGA-DSM, and other complex mission design problems.

A genetic algorithm(GA) is a stochastic optimization method based on the principles of evolution. Genetic algorithms perform a probabilistic search by evolving a randomly chosen initial population. The population is just a series of sets of variables that are evaluated by a cost function, in this case the cost functions previously covered. The advantage of using evolutionary methods over traditional optimization methods is that no initial solution is necessary. This helps ensure, but does not guarantee, that solutions are not confined to locally optimal solutions. Genetic algorithms also perform well in very complex nonlinear design spaces. Despite all the advantages, evolutionary algorithms do have their downside. They almost always require a greater number of cost function evaluations, increasing the computational requirements. Additionally evolutionary algorithms do not make use of gradients, so there is no proof of convergence. It should also be noted that genetic algorithms were developed for constrained minimization problems and may not perform well for unconstrained minimization.

The basic genetic operators are, selection, reproduction/crossover, mutation, and elitism. The genetic algorithm developed in this section can utilize a number of different selection, reproduction, crossover, and mutation methods. Each of the methods will be discussed in detail in later sections.

In a simple genetic algorithm (SGA), each variable is represented by a binary string where individual bits are represented by 1's and 0's. This binary string is referred as a gene. The separate genes for each variable are then concatenated to form a complete chromosome. One chromosome correspond to one member of the entire population, which can number in the hundreds of thousands for some problems. This binary representation of variables means that each variable is discretized to a certain resolution. In our case the desired resolution for each variable is one of the inputs to the genetic algorithm, along with the individual variable upper and lower bounds, size of the population, and number of generations the population is to be run out. For an individual variable the resolution is defined as

$$r = \frac{x^U - x^L}{2^b - 1} \tag{22}$$

where x^U and x^L are the user specified variable upper and lower bounds, while b is the minimum number of bits required to obtain the desired variable accuracy. Equation (22) can then be solved for b for each individual variable. With the variable b known for each variable the size of each chromosome can be determined.

Then next step in the process is to randomly, via a "digital coin flip", assign a value (1 or 0) to each position in the chromosome. The coin flip is performed using a uniform random number generator, that generates random numbers between 0 and 1. If the random number is greater than 0.5, the a value of 1 is assigned, otherwise a value of 0 is assigned. This process is run out to generate an entire initial population, which is randomly distributed throughout the design space.

From this point each member of the population is assigned a fitness value, via a user supplied cost function. The genetic operators are then used to generate a new population from the initial parent population. This is the crux of how genetic algorithms are able to evolve to more fit populations. This process continues until the genetic algorithm is told to exit and output the final population. Common stopping conditions include stopping after a certain number of iterations, monitoring when the best fit solution stop changing, or monitoring when the average cost function value approaches the population minimum. For this study the stopping condition is always running the genetic algorithm out by a user specified number of generations. The sequence of operations of the core operations of the genetic algorithm is illustrated in Fig. 2. One secondary operator, elitism, was used in this genetic algorithm as well. The elitism operator ensure that the best fit solution(s) are not lost from generation to generations. A simple elitism operator is used, in which the top two solutions from the parent population are directly inserted into the next generation.

A. Selection Types

In genetic algorithms, selection operator, also the first genetic operator, is used to ensure that the best fit solutions are chosen to pass on their genes through the reproduction and crossover operators. The selection operator is essentially the operator that represents the survival of the fittest principle. A total of two selection types have been implemented in this genetic algorithm one based solely on a roulette selection method, and another that combines roulette and tournament selection. These two operators are by no means the only possible selection methods, but that are some of the simplest to use and prove to work well for the mission design problems.

Roulette selection is a fitness proportionate selection method in which better fit individuals are more likely to be chosen for reproduction and crossover. There are four fitness types that are used in the roulette selection method described in this section. The ultimate goal is to get a normalized fitness that ranges from 0 to 1. More fit members of the population will have a higher normalized fitness value, thus they will be more likely to be chosen for further genetic operations.

The first fitness type is the raw fitness, $r(i)$, which is the fitness values provided directly from the cost function for each population member. In this case i represents the chromosome/population number. The

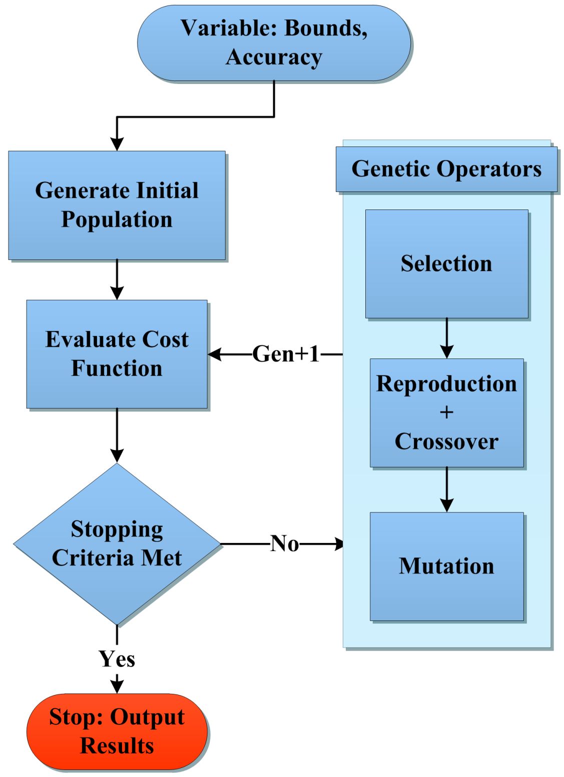

Figure 2. Simple genetic algorithm flow chart.

raw fitness is then standardized, depending on whether the problem is being minimized and maximized. If the problem is a minimization problem, the standardized fitness is the same as the raw fitness as

$$s(i) = r(i) \tag{23}$$

However, if the problem is a maximization problem, the standardized fitness becomes

$$s(i) = r_{max} - r(i) \tag{24}$$

Now the fitness is adjusted so that individual fitnesses lie between 0 and 1. The adjusted fitness is larger for better individuals. The advantage of this optional step is that small differences in the most fit individuals, those that approach an adjusted of 0, are exaggerated. This has a larger benefit when the population improves by emphasizing small differences in individuals. This effect is most exaggerated in problems where the best solution has a cost function near 0,[14,15] as follows

$$a(i) = \frac{1}{1 + s(i)} \tag{25}$$

Now that the adjusted fitness values lie between 0 and 1 the next step is to normalize the whole population, so the sum of each normalized fitness value is 1. Ultimately, this is necessary because individuals are chosen through the use of random numbers, that also in 0 and 1 range, as

$$n(i) = \frac{a(i)}{\sum\limits_{j}^{n} a(j)} \tag{26}$$

The actual selection of individuals, based on normalized fitness, is done in the following manner. Firstly the normalized fitnesses are sorted from largest to smallest. Next a random number is generated. This random number will decide which individual is then selected. The actual selection of an individual is done by a summation of the normalized fitness values. The individual that causes the summation value to be greater than the random number is chosen to move along through the rest of the genetic operators. This process is repeated and both of the individuals then go through the reproduction or crossover operators. Thus, more fit individuals are more likely to be passed on to the next generation.

Tournament is a greedy over selection method that also utilized the roulette selection method. Two individual are chosen using roulette selection and then compete to see which individual is allowed to pass on its genetic material. This is done through the direct comparison of their normalized fitness function. The individual with the best fitness is chosen as the one that gets to pass through to the reproduction and crossover operations. The process is repeated twice, resulting in two individuals who's genetic material can be passed on.

This selection method has a distinct advantage in that it drives down the raw fitness and average of the fitness function in fewer generations than pure roulette selection. With all greedy over selection methods there is a risk that genetic diversity, which could help out the population in later generations, will be lost. Therefore tournament selection should be used with care by the user. In the orbital problems discussed in this paper, tournament selection does appear to work well.

B. Reproduction and Crossover Operator Types

While the selection operator determines which population members get the privilege of reproducing and passing on their genetic material to future generations, the crossover and reproduction operators decided what to do with that genetic material. In the algorithm utilized in this paper the user must supply the probability in which an individual will under go either reproduction or crossover. These probabilities must total up to 1.0 (100%), with values typically being close to 0.9 and 0.1 for crossover and reproduction respectively. It should also noted that both operations require two parent and produce two offspring.

The simplest of these two operators is reproduction. In traditional genetic algorithm terms this means that the operation is a fitness-proportionate process in which individuals are allowed to directly pass to the next generation with a probability proportionate to their fitness values. This essentially means that when reproduction occurs the two parents are passed directly from the parent generation into the next generation. As with every other critical procedure in the genetic algorithm, whether the parents undergoes reproduction or crossover is chosen through the use of random number. In the algorithm a random number is generated. If the random number is greater than the user given crossover probability the parents undergoes reproduction. If not, the a crossover operation is applied. This, of course, is the reason that the two probabilities must total up to 100%.

The crossover process is a process in which biological reproduction is used to allow new individuals, with new and unique genetics, to be created. Unlike the reproduction process, crossovers allow new points in the design space to be searched. Without reproduction (and also mutations) genetic algorithms would be no more useful and a completely random search. The crossover operation produces to offspring from two parents that contain genetic material from both parents. A total of 4 distinct crossover types have been implemented in for use with this genetic algorithm. The user supplies which crossover type should be used, making the final genetic algorithm suitable for many different types of optimization problems. For the mission design problems discussed in this paper the double point crossover proved to be the best crossover type. However for other types of problems this may not necessarily be the cause.

The purpose of each crossover type is to promote genetic diversity and expand, in a controlled way, the search of the design space. The first type considered is uniform crossover. It has been shown to be a very effective method for promoting genetic diversity, and in turn the discovery of new useful chromosomes.[13,16] In uniform crossover each bit in the chromosome is a crossover point. The two offspring are generated by the same virtual coin flip that was used to initially generate the population for the first generation. If two bit are the same for both parents no coin flip is necessary. However, if two bits are not identical in the parents the coin flip is used to decide which offspring gets the genetic material from the separate parent individuals. This process is graphically shown in the first entry in Fig. 3. In this case if the coin flip is heads, i.e. the random number is greater than or equal to 0.5, the bit from the second parent is inserted into the chromosome for the first child and the bit from the first parent is inserted into the chromosome of the second child. The exact opposite is true if the coin flip is tails, i.e. the random number is less than 0.5.

A second crossover operator has also been developed that is similar to uniform crossover but only operates on an individual gene. This process only changes the value of one gene (i.e. variable). In some situations changes to an individual variable can greatly improve (or alternatively worsen) the individuals fitness. For

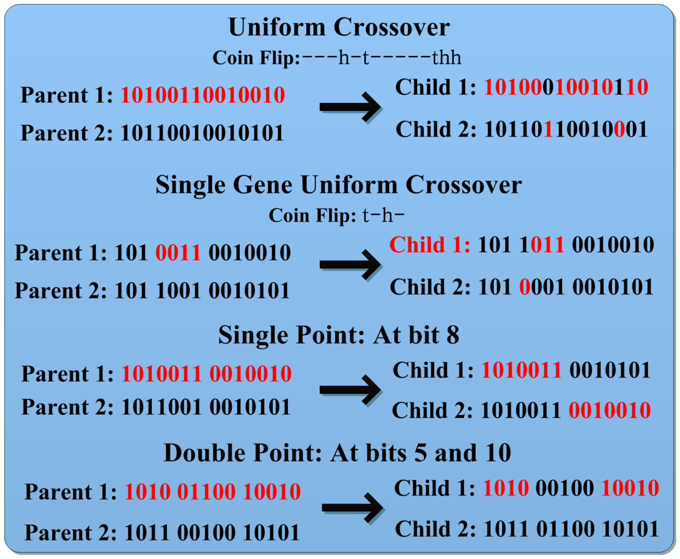

Figure 3. Simple illustration of the four implemented crossover types.

this process a gene is randomly selected. The same gene then undergoes uniform crossover. An illustration of this method can be seen in Fig. 3. In this example the second gene (bits 4-7) undergoes uniform crossover through the same virtual coin flip method described above.

In single point crossover one bit is randomly chosen to be the crossover point. All bits after the random crossover point are swapped between the two points. Double point crossover is the same thing with two randomly chosen crossover points. While these two methods are the simplest of the tested crossover methods they have proven extremely useful in practical applications.

C. Mutation Types

The last genetic operator, which the newly generated population must pass through, is the mutation operator. The probability that a mutation will occur and the desired type of mutation are the last two user inputs. The probability that a mutation should occur should be small, typically less than 0.05 (5%). When the mutation probability is set too high the genetic algorithm will start to resemble a simple random search. If the user doesn't wish to make use of the mutation operator, a probability of 0 can be entered as well.

Allowing the genetic algorithm to utilize a mutation operator has several advantages. Mutations are used to help maintain genetic diversity in a population as it ages. Often times after many generations the population can lose genetic diversity and stagnate at a local minimum. Mutations help to prevent this from happening. By introducing (or in some cases reintroducing) changes in a chromosome or individual gene it is possible for better more fit individuals to appear.

As with the crossover operator four separate types of mutations have been implemented. Unlike reproduction and crossover mutations operate on a single individual in the population. Different mutation types may work well for individual problems. The four types introduced here are representative of some of the most common types of mutations used in genetic programming. Each mutation type is illustrated in Fig. 4.

The simplest type of mutation is the flip bit mutation. In this type of mutation the program randomly chooses a single bit to be flipped. When a bit is flipped that value is either changed from 1 to 0 or from 0 to 1. The next three mutation types are only allowed to operate on a single randomly chosen gene.

The boundary mutation type is as simple as it sounds. An individual gene is set either to corresponding variables user supplied minimum or maximum. The digital coin flip is used to decided whether the minimum or maximum values will be used. To set the gene to the maximum every bit is set to 1. Alternative is the gene is set to the minimum every bit is changed to 0.

Uniform mutation is similar to uniform crossover and the method used to initialize the population. A digital coin flip is used to completely redefine an individual gene. The last mutation type studied, inversion, is a simple mutation operator that has proven to work extremely well for mission design problems. As the name implies, when inversion is applied the bits that make up the gene are simply inverted.

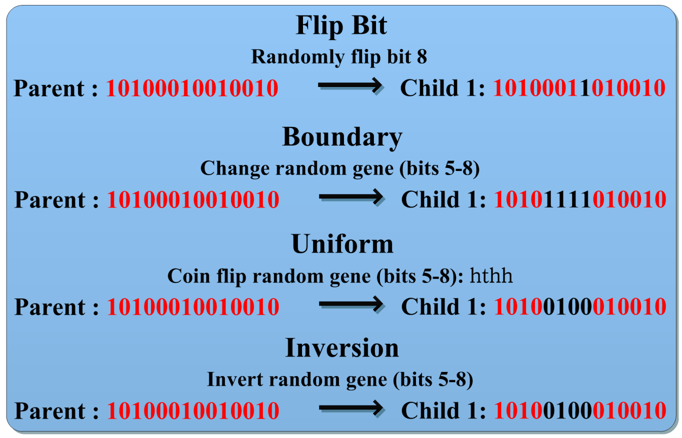

Figure 4. Simple illustration of the four mutation types implemented.

D. Comparison of Genetic Operator Performances

When using genetic algorithms, it is beneficial to perform a study to determine which combination or selection, crossover, and mutation operators work best with the type of problem being solved. A much more detailed study than that illustrated in Table 5 was performed when designing and implementing the genetic algorithm. None the less, Table 5 is representative of the results. In general, the best crossover type for the MGA and MGA-DSM problems is double point. It is less clear which mutation type is the best. Both the uniform and inversion operators perform well. For the following example problems, the selection, crossover, and mutation types used are tournament, double point, and inversion respectively.

Table 5. Comparison of mutation and crossover types for a complex multiple gravity assist mission with a population size of 20,000 run for 200 generations. The numbers represent the final cost function values in km/s.

Crossover Type	Mutation Type			
	Flip Bit	Uniform	Boundary	Inversion
Uniform	6.510	6.938	6.805	6.536
Uniform Gene	4.903	5.050	5.626	5.368
Single Point	5.182	4.148	4.636	4.687
Double Point	4.389	3.731	5.137	3.363

IV. Mission Search Results

The genetic algorithm was used to evaluate each mission type for the 3398 Amor and Atira asteroids. Each mission type took approximately 3 days to run. For the genetic algorithm the tournament selection, double point crossover, and inversion mutation operators were used. The reproduction, crossover mutation probabilities for each mission type are 0.1, 0.9, and 0.05 respectively. The results only reflect the mission design for each asteroid. Specific asteroid parameters such as asteroid diameter, rotational period, and brightness should taken into account at a later phase in the study.

A. Type 1 Mission Results

The variable bounds for each mission variable is shown in Table 6. A total of 6 variables are required to completely define the mission. The main spacecraft variables are launch date, time-of-flight or the Earth departure and Earth return legs and asteroid stay time. Only 2 variables are required to define the mission for the hypervelocity impactor, the launch date and the time-of-flight for the Earth departure to asteroid impact leg.

Table 6. Type 1 mission variable bounds.

	Lower	Upper
Main Spacecraft		
T_0	1-Jan-15	1-Jan-35
T_1 (days)	1.000	500
T_2 (days)	1.000	500
T_3 (days)	1.000	500.000
Hypervelocity Impactor		
T_0	1-Jan-15	1-Jan-35
T_1 (days)	1.000	1000

The results from the genetic algorithm for type 1 missions are shown in Table 7. This table shows the 10 asteroids with the lowest cost functions. The cost functions range from 6.1 to 7.3 km/s. When compared to mission types 2 and 3 these mission have very high cost. Not just from a total ΔV perspective, but from actual missions cost, do to the dual launch vehicle requirement. Type 1 missions serve as a baseline to compare the single launch missions (type 2 and 3). They have relatively high cost functions when compare to other mission types.

B. Type 2 Mission Results

A total of 6 variables are required to fully define type 2 missions. The main spacecraft requires 4 variables, departure date, Earth departure leg length, asteroid stay time, and Earth departure leg length (the same definitions as type 1 mission). Two variables are required for the hypervelocity impactor portion of the mission. The deep-space maneuver burn index and the time-of-flight from the maneuver up to the asteroid impact. The variable bounds are shown in Table 9. The launch date ranges and mission length ranges are formulated in such a way that no portion of the mission will be at the asteroid after Jan 1st, 2040, which is the last entry for the asteroid ephemeris data.

Examination of the results in Table 8. While the actual required cost function values are lower than top type 1 missions, the ΔV required for the main spacecraft and hypervelocity impactor are much higher. The solutions appear to be minimized by minimizing the departure v_∞ more than the ΔV required after Earth departure. For type 2 missions the minimum ΔV required by the two spacecraft is approximately 5 km/s, making these trajectories far from ideal. The hypervelocity impact speeds tend to be driven towards the minimum required 5 km/s. Relaxing the impact velocity requirements and emphasizing the Earth departure portion of the cost function may result in more feasible solutions. However solutions requiring total ΔV's in the 1-2 km/s range are found for the type 3 missions. It is unlikely that modifying the type 2 mission requirements will lower the required ΔV by the 3-4 km/s required to match the type 3 mission solutions.

C. Type 3 Mission Results

The last mission type included takes advantage of a gravity assist at Venus. The gravity-assist is used to increase the velocity of the hypervelocity impactor, resulting in higher impact speeds and lower total cost functions. By increasing the impact velocities the rest of the mission constraint a relaxed, allowing the genetic algorithm to find solutions requiring total spacecraft ΔV's much lower than bothe type 1 and 2 mission. Impact velocities range from 10.9 to 24.7 km/s for top 10 missions. This mission type requires 7 decision

Table 7. Top 10 Asteroids for Type 1 missions.

	2012 LA	2011 MD	2003 YT70	2006 UN	2006 UB17	2007 TT24	2010 SZ3	2006 HX30	2004 KG17	1993 BD
	Main Spacecraft Earth Departure									
Launch Date	13-Aug-29	11-Jul-23	6-May-17	24-Dec-28	9-Nov-22	16-Nov-23	16-Oct-18	24-May-21	23-Jun-25	31-Mar-17
$v_{\infty-0}$ (km/s)	0.346	0.446	0.586	1.160	1.201	2.239	0.712	1.988	1.949	0.696
	Main Spacecraft Asteroid Parameters									
Time-of-Flight (days)	287.544	349.851	235.394	336.266	288.706	407.405	328.469	412.298	418.133	310.735
Δ V$_{arr}$ (km/s)	1.647	1.083	4.791	3.898	1.404	2.177	2.127	2.755	2.822	4.845
Asteroid Stay Time (days)	262.195	26.340	219.218	212.079	95.536	345.935	52.151	172.531	151.699	209.244
Δ V$_{dep}$ (km/s)	1.056	1.091	0.364	0.392	2.497	0.297	2.589	0.406	0.555	0.941
	Main Spacecraft Earth Arrival Parameters									
Time-of-Flight (days)	270.868	337.199	495.872	484.404	311.785	299.230	326.520	484.404	500.000	500.000
Δ V$_{arr}$ (km/s)	0.000	0.000	0.191	0.118	0.000	0.000	0.000	0.010	0.034	0.217
	Impactor Paramerters									
Launch Date	13-Mar-30	22-Jan-24	24-May-17	9-Feb-29	14-Feb-23	7-Mar-24	26-Dec-18	2-Aug-21	26-Sep-25	3-Jul-17
$v_{\infty-0}$ (km/s)	0.545	2.537	0.543	1.130	0.632	2.321	1.646	1.978	1.823	0.619
Time-of-Flight (days)	254.653	181.544	234.932	326.881	276.952	383.465	309.319	391.479	375.660	247.797
Arrival v_∞ (km/s)	2.483	3.771	5.487	6.261	3.911	10.415	4.952	7.917	9.353	6.680
	Summary									
Main S/C ΔV (km/s)	2.703	2.174	5.346	4.409	3.901	2.474	4.717	3.171	3.411	6.002
Impactor Δ V (km/s)	2.517	1.229	0.000	0.000	1.089	0.000	0.048	0.000	0.000	0.000
Total Cost (km/s)	6.110	6.385	6.475	6.700	6.823	7.033	7.122	7.137	7.182	7.317

Table 8. Top 10 asteroids for type 1 missions.

	2008 NX	2011 MD	2004 KG17	2003 YT70	2005 GN22	2011 TP6	2010 VA140	2012 LA	2005 EZ169	2006 UN
Main Spacecraft Earth Departure										
Launch Date	23-Aug-16	3-Jul-22	12-Jul-27	24-Apr-17	30-May-17	13-Feb-17	21-Dec-26	30-Aug-29	15-May-19	27-Jan-31
$v_{\infty-0}$ (km/s)	0.900	0.839	0.801	0.503	0.974	1.012	1.058	0.379	1.005	0.609
Main Spacecraft Asteroid Parameters										
Time-of-Flight (days)	336.711	328.001	352.525	254.502	352.385	290.622	340.419	267.968	333.859	291.046
Δ V$_{arr}$ (km/s)	2.694	1.667	3.981	4.967	2.672	3.048	4.075	1.662	3.020	4.574
Asteroid Stay Time (days)	637.817	74.249	96.147	240.185	649.887	668.888	230.674	56.900	647.692	258.490
Δ V$_{dep}$ (km/s)	0.720	0.353	0.537	0.358	0.320	0.697	0.339	0.905	0.751	0.728
Main Spacecraft Earth Arrival Parameters										
Time-of-Flight (days)	427.813	313.187	608.725	467.818	406.509	375.624	459.223	297.578	418.189	453.671
Δ V$_{arr}$ (km/s)	0.000	0.000	0.207	0.192	0.000	0.000	0.143	0.000	0.000	0.183
Impactor Paramerters										
Burn Index ϵ	0.131	0.295	0.162	0.763	0.213	0.039	0.157	0.822	0.051	0.154
ΔV$_{DSM}$ (km/s)	1.547	2.481	0.589	0.067	1.397	1.487	0.650	0.000	1.551	0.297
Time-of-Flight (days)	335.005	301.069	315.755	61.037	317.927	317.657	305.529	47.715	346.957	256.262
Arrival v_∞ (km/s)	4.995	4.299	5.004	4.926	4.192	5.007	5.002	1.662	4.997	4.999
Summary										
Main S/C ΔV (km/s)	3.414	2.020	4.724	5.516	2.991	3.745	4.557	2.567	3.771	5.486
Impactor Δ V (km/s)	1.552	3.182	0.589	0.141	2.205	1.487	0.650	3.338	1.555	0.298
Total Cost (km/s)	5.866	6.041	6.115	6.161	6.170	6.244	6.265	6.284	6.331	6.392

variables, which are the same as the type 2 with an additional time-of-flight added from the gravity-assist to impact.

As the results show from Table 11 show, the total cost function and required ΔV for both the main spacecraft and the hypervelocity impactor are significantly improved over type 1 and 2 missions. The lowest cost functions range from 3.1 to 5.8 km/s. Assuming the launch vehicle and provide the necessary Earth departure v_∞ the combined total ΔV required for the main spacecraft and hypervelocity impactor range from 1.2 to 4.23 km/s. Most of the top ten missions, by required ΔV are type 3 mission (8 out of 10).

Table 9. Type 2 mission variable bounds

	Lower	**Upper**
Main Spacecraft		
T_0	1-Jan-15	1-Jan-32
T_1 (days)	1	750
T_2 (days)	1	750
T_3 (days)	1	1000
Hypervelocity Impactor		
ϵ	0.02	0.9
T_1 (days)	1	1500

Table 10. Type 3 mission variable bounds

	Lower	**Upper**
Main Spacecraft		
T_0	1-Jan-15	1-Jan-32
T_1 (days)	50	1000
T_2 (days)	50	1000
T_3 (days)	50	1000
Hypervelocity Impactor		
ϵ	0.05	0.9
T_1 (days)	50	1000
T_2 (days)	50	1000

Table 11. Top 10 asteroids for Type 3 Missions

	2011 MD	2012 LA	2005 ER95	2005 GN22	2005 OH3	2008 NX	2006 KL103	2010 DJ	2011 PN1	2002 TC70
					Main Spacecraft Earth Departure					
Launch Date	27-Jun-22	4-Jul-28	11-Mar-28	17-Apr-21	17-Jul-16	1-Sep-19	24-May-20	26-Jan-18	2-Aug-19	8-Jul-26
$v_{\infty-0}$ (km/s)	1.047	1.331	2.673	3.038	3.557	1.249	4.205	2.221	3.188	3.644
					Main Spacecraft Asteroid Parameters					
Time-of-Flight (days)	333.416	294.655	98.239	175.014	317.159	325.589	372.392	200.294	360.441	145.298
Δ V$_{arr}$ (km/s)	1.518	1.774	0.723	1.105	0.745	2.489	0.494	0.963	1.380	0.906
Asteroid Stay Time (days)	70.421	340.381	999.999	495.313	757.834	658.602	937.940	855.491	797.778	591.623
Δ V$_{dep}$ (km/s)	0.317	0.506	0.422	1.038	0.897	0.992	0.758	1.839	0.608	1.248
					Main Spacecraft Earth Arrival Parameters					
Time-of-Flight (days)	323.447	229.343	369.144	425.060	399.270	405.662	505.696	391.638	319.533	377.684
Δ V$_{arr}$ (km/s)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
					Impactor Paramerters					
Burn Index	0.061	0.810	0.674	0.736	0.763	0.891	0.785	0.807	0.803	0.626
ΔV$_{DSM}$ (km/s)	0.196	0.155	0.058	0.027	0.121	0.723	0.034	0.588	0.461	0.012
GA Time-of-Flight (days)	242.971	74.889	106.157	109.708	120.533	99.289	171.969	143.121	101.406	130.119
ΔV$_{GA}$ (km/s)	0.000	0.013	0.008	0.004	0.010	0.025	0.007	0.043	0.016	0.003
Ast. Time-of-Flight (days)	100.217	321.419	427.247	431.760	382.051	486.365	445.317	385.898	355.148	515.692
Arrival v_∞ (km/s)	14.323	24.700	16.594	16.283	18.107	16.521	11.334	10.934	19.443	18.726
					Summary					
Main S/C ΔV (km/s)	1.835	2.280	1.146	2.144	1.641	3.481	1.252	2.802	1.988	2.154
Impactor Δ V (km/s)	0.196	0.169	0.066	0.031	0.132	0.749	0.041	0.631	0.477	0.015
Total Cost (km/s)	3.079	3.779	3.884	5.213	5.330	5.479	5.498	5.654	5.654	5.814

V. Mission Design Summary

Several possible mission architectures have been examined throughout this paper. Multiple mission candidates have also been determined through the implementation of a genetic algorithm and searching program. The top ten missions are summarized in Table 12, sorted by the total combined ΔV for the main spacecraft and hypervelocity impactor. It is assumed the launch vehicle(s) will provide the necessary C_3. By utilizing a gravity-assist at Venus mission requiring ΔV's as low as 1.2 km/s have been found.

Table 12. Top 10 asteroids sorted by require spacecraft ΔV in km/s

	Mission Type	Departure C_3 km^2/s^2	Main S/C ΔV km/s	Impactor Δ V km/s	Total Required ΔV km/s
2005 ER95	3	7.144	1.146	0.066	**1.211**
2006 KL103	3	17.681	1.252	0.041	**1.293**
2005 OH3	3	12.650	1.641	0.132	**1.773**
2011 MD	3	1.097	1.835	0.196	**2.031**
2002 TC70	3	13.280	2.154	0.015	**2.169**
2005 GN22	3	9.231	2.144	0.031	**2.175**
2012 LA	3	1.771	2.280	0.169	**2.448**
2011 PN1	3	10.165	1.988	0.477	**2.466**
2007 TT24	1	5.013, 5.385	2.474	0.000	**2.474**
2006 HX30	1	3.951, 3.914	3.171	0.000	**3.171**

Appendix

A few of the noteworthy functions required for the genetic algorithm are outlined in this appendix. Although crucial to implementing a genetic algorithm, random number generators and sorting functions, are beyond the scope of this paper. They are not included in this appendix or anywhere else in this paper. Further information can be found in Refs. [17–19].

Number of Bits per Gene

$$b = INT\left(\frac{\ln\left(\frac{X^U - X^L}{r} - 1\right)}{\ln(2)} + 1\right) \tag{27}$$

This equation is derived by solving Eq. (22) for b, the number of bits required for the desired variable resolution. The addition of 1 in Equation (22) ensures that the minimum accuracy is preserved when the integer function rounds down.

Converting Binary Gene to Real Number Value

The conversion from a single gene to the gene's corresponding value is outlined below. A is the gene's binary string, b is the number of bits in the gene, and X^U and X^L are the user supplied upper and lower bounds the the variable represented by the gene.

$$sum = \sum_{i=1}^{b} A(i)2^{i-1} \tag{28}$$

$$scale = \frac{X^U - X^L}{2^{b-1}} \tag{29}$$

The final variable value X is then:

$$X = X^L + scale \cdot sum \tag{30}$$

Acknowledgments

This work was supported by the Iowa Space Grant Consortium (ISGC) through a research grant to the Asteroid Defection Research Center at Iowa State University. The authors would like to thank Dr. Ramanathan Sugumaran (Director, ISGC) for his interest and support of this research work.

References

[1]Morrison, D., "The Spaceguard Survey: Report of the NASA International Near-Earth-Object Detection Workshop," *NASA STI/Recon Technical Report N*, Vol. 92, Jan. 1992, pp. 34245.

[2]Pitz, A., Vardaxis, G., and Wie, B., "A Hypervelocity Nuclear interceptor System (HNIS) for Optimal Disruptions of Near-Earth Objects," *22nd AAS/AIAA Space Flight Mechanics Meeting*, Vol. AAS-12-125, Jan. 2012.

[3]Vardaxis, G., Pitz, A., and Wie, B., "Conceptual Design and Analysis of Planetary Defense Technology (PDT) Demonstration Missions," *22nd AAS/AIAA Space Flight Mechanics Meeting*, Vol. AAS-12-228, Jan. 2012.

[4]Winkler, T., Wagner, S., and Wie, B., "Optimal Target Selection for a Planetary Defense Technology (PDT) Demonstration Mission," *22nd AAS/AIAA Space Flight Mechanics Meeting*, Vol. AAS-12-226, Jan. 2012.

[5]Fujiwara, A., Kawaguchi, J., Yeomans, D. K., Abe, M., Mukai, T., Okada, T., Saito, J., Yano, H., Yoshikawa, M., Scheeres, D. J., Barnouin-Jha, O., Cheng, A. F., Demura, H., Gaskell, R. W., Hirata, N., Ikeda, H., Kominato, T., Miyamoto, H., Nakamura, A. M., Nakamura, R., Sasaki, S., and Uesugi, K., "The Rubble-Pile Asteroid Itokawa as Observed by Hayabusa," *Science*, Vol. 312, No. 5778, 2006, pp. 1330–1334.

[6]Carnelli, I., Gálvez, A., and Izzo, D., "Don Quijote: A NEO Deflection Precursor Mission," *2006 NASA Near-Earth Object Detection and Threat Mitigation Workshop*, June 2006.

[7]Cano, J., Sánchez, M., and Carnelli, I., "Mission Analysis for the Don Quijote Phase-A Study," *Proceedings of the 20th International Symposium on Space Flight Dynamics*, Sept. 2007.

[8]Gálvez, A. and Carnelli, I., "ESA's Don Quijote Mission: an Opportunity for the Investigation of an Artificial Impact Crater on an Asteroid," *25th International Symposium on Space Technology and Science*, 2006.

[9]Wie, B., "Hypervolocity Nuclear Interceptors for Asteroid Deflection or Disruption," *Proceedings of 2nd IAA Planetary Defense Conference*, May 2011.

[10]Kaplinger, B. and Wie, B., "NEO Fragmentation and Disperion Modeling and Simulation," *Proceedings of 2nd IAA Planetary Defense Conference*, May 2011.

[11]Wagner, S. and Wie, B., "Target Asteroid Selection for Human Exploration of Near Earth Objects," *20th AAS/AIAA Space Flight Mechanics Meeting*.

[12]Englander, J., Conway, B., and Williams, T., "Optimal Autonomous Mission Planning Via Evolutionary Algorithms," *21th AAS/AIAA Space Flight Mechanics Meeting*, Vol. AAS-11-159, Feb. 2011.

[13]Vavrina, M., *A Hybrid Genetic Algorithm Approach to Global Low-thrust Trajectory Optimization*, Ph.D. thesis, Purdue University, 2010.

[14]Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1st ed., 1989.

[15]Koza, J., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge, Massachusetts, 1st ed., 1992.

[16]Syswerda, G., "Uniform Crossover in Genetic Algorithms," *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989, pp. 2–9.

[17]Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes in Fortran: The art of Scientific Computing*, Cambridge University Press, 2nd ed., Sept. 1992.

[18]Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in Fortran 90 (2nd ed.): The art of Parallel Scientific Computing*, Cambridge University Press, New York, NY, USA, 1996.

[19]Cormen, T. H., Stein, C., Rivest, R. L., and Leiserson, C. E., *Introduction to Algorithms*, McGraw-Hill Higher Education, 2nd ed., 2001.