

Target selection for a hypervelocity asteroid intercept vehicle flight validation mission

Sam Wagner^a, Bong Wie^a, Brent W. Barbee^{a,b,*}

^a Asteroid Deflection Research Center, Department of Aerospace Engineering, Iowa State University, 2271 Howe Hall, Room 2348, Ames, IA 50011-2271, USA

^b NASA/GSFC, Code 595, 8800 Greenbelt Road, Greenbelt, MD 20771, USA



ARTICLE INFO

Article history:

Received 15 August 2014

Received in revised form

16 November 2014

Accepted 26 November 2014

Available online 3 December 2014

Keywords:

Asteroids

Near-Earth objects

Planetary defense

Trajectory optimization

ABSTRACT

Asteroids and comets have collided with the Earth in the past and will do so again in the future. Throughout Earth's history these collisions have played a significant role in shaping Earth's biological and geological histories. The planetary defense community has been examining a variety of options for mitigating the impact threat of asteroids and comets that approach or cross Earth's orbit, known as near-Earth objects (NEOs). This paper discusses the preliminary study results of selecting small (100-m class) NEO targets and mission analysis and design trade-offs for validating the effectiveness of a Hypervelocity Asteroid Intercept Vehicle (HAIV) concept, currently being investigated for a NIAC (NASA Advanced Innovative Concepts) Phase 2 study. In particular this paper will focus on the mission analysis and design for single spacecraft direct impact trajectories, as well as several mission types that enable a secondary rendezvous spacecraft to observe the HAIV impact and evaluate its effectiveness.

© 2014 Published by Elsevier Ltd. on behalf of IAA.

1. Introduction

Geological evidence shows that asteroids and comets have collided with the Earth in the past and will do so in the future. Such collisions have played an important role in shaping the Earth's biological and geological histories. Many researchers in the planetary defense community have examined a variety of options for mitigating the impact threat of Earth approaching or crossing asteroids and comets, known as near-Earth objects (NEOs).

As early as 1992, the idea of discovering and tracking near-Earth objects (NEOs) was proposed to the U.S. Congress [1]. That search effort, called the Spaceguard Survey, was later implemented in 1998 with the ultimate goal of finding 90% of

the estimated asteroid population 1 km in diameter or larger by 2008. By focusing on only 1 km size or larger NEOs, that survey only intended to find NEOs large enough to cause global catastrophes. While not large enough to affect the entire globe, impacts by objects smaller than 1 km occur more frequently and are capable of causing significant damage. In 2005, the George E. Brown, Jr. Near-Earth Object Survey Act expanded the original Spaceguard search to include the detection and characterization of 90% of NEOs as small as 140 m by the year 2020. To date, none of the discovered objects are predicted to be on a collision course with the Earth, but the survey still has several more years before the mission is complete. Should a new NEO be discovered on a collision course with the Earth, a mitigation effort would be necessary in order to prevent a collision with the Earth.

Given a lead time (from initial detection of the incoming NEO) of at least 10–20 years, depending on circumstances, various proposed technologies such as kinetic impactors [2] or slow-pull gravity tractors [3] could be employed to successfully

* Corresponding author at: NASA/GSFC, Code 595, 8800 Greenbelt Road, Greenbelt, MD 20771, USA. Tel.: +1 301 286 1837.
E-mail addresses: thewags@iastate.edu (S. Wagner), bongwie@iastate.edu (B. Wie), brent.w.barbee@nasa.gov (B.W. Barbee).

mitigate an impact threat by deflecting the NEO's heliocentric orbit just enough to avoid a collision with Earth. When the warning time is short, nuclear technologies for a standoff, contact, or subsurface explosion may be the only viable options [4,5]. However, as of the time of this writing none of the aforementioned mitigation options have been validated with a flight demonstration mission. The Asteroid Deflection Research Center (ADRC) has conducted a preliminary design for Hypervelocity Asteroid Intercept Vehicle (HAIV), a spacecraft capable of performing hypervelocity (> 5 km/s) intercept of asteroids as small as a 50–100 m in diameter [6–8]. The HAIV concept, depicted in Fig. 1, involves a two-body spacecraft that creates a small crater in an asteroid's surface and then detonates a nuclear explosive within the crater, thus effecting a subsurface detonation. A subsurface detonation is more effective at coupling energy into an asteroid than a standoff or surface detonation, permitting disruption of an asteroid with a smaller nuclear device yield than would otherwise be required. The HAIV design is necessitated by the fact that penetrator technology can only protect the explosive payload from impact velocities of several hundred m/s, while impact velocities of > 5 km/s are required for asteroid intercept missions, especially in the case of short warning times.

In this paper a variety of mission analysis results will be presented to illustrate candidate target asteroids for a flight validation of the HAIV concept. The mission concepts considered include direct intercept missions, in which the impactor is inserted directly on an intercept course by the

launch vehicle and is only allowed to perform small ΔV maneuvers prior to impact, as well as missions which allow an observer spacecraft to rendezvous with the asteroid prior to the HAIV impact.

To measure the performance and success of the HAIV it would be useful to have an observer spacecraft at the asteroid prior to the time at which the HAIV impacts the asteroid, as suggested in other studies [2]. However, due to mission and launch vehicle cost constraints it is highly desirable to perform the entire mission using one launch vehicle. Towards that end, we have designed the flight validation mission such that an observer spacecraft is not strictly required; instead, the HAIV transmits adequate telemetry to Earth for reconstruction of the asteroid impact event. The flight validation mission design is made even more cost-effective by incorporating advanced interplanetary mission design techniques including optimally placed deep space maneuvers (DSMs) and both powered and unpowered gravity assists via planetary flybys.

1.1. Previous and future NEO missions

To help determine the mission requirements and constraints it is useful to examine past and proposed future robotic missions to NEOs. Space agencies such as ESA, JAXA, and NASA have had several successful missions that demonstrate technologies and mission capabilities that are relevant to the proposed HAIV demonstration mission, including terminal guidance targeting and/or landing capabilities.

Hypervelocity Asteroid Intercept Vehicle (HAIV) Mission Architecture

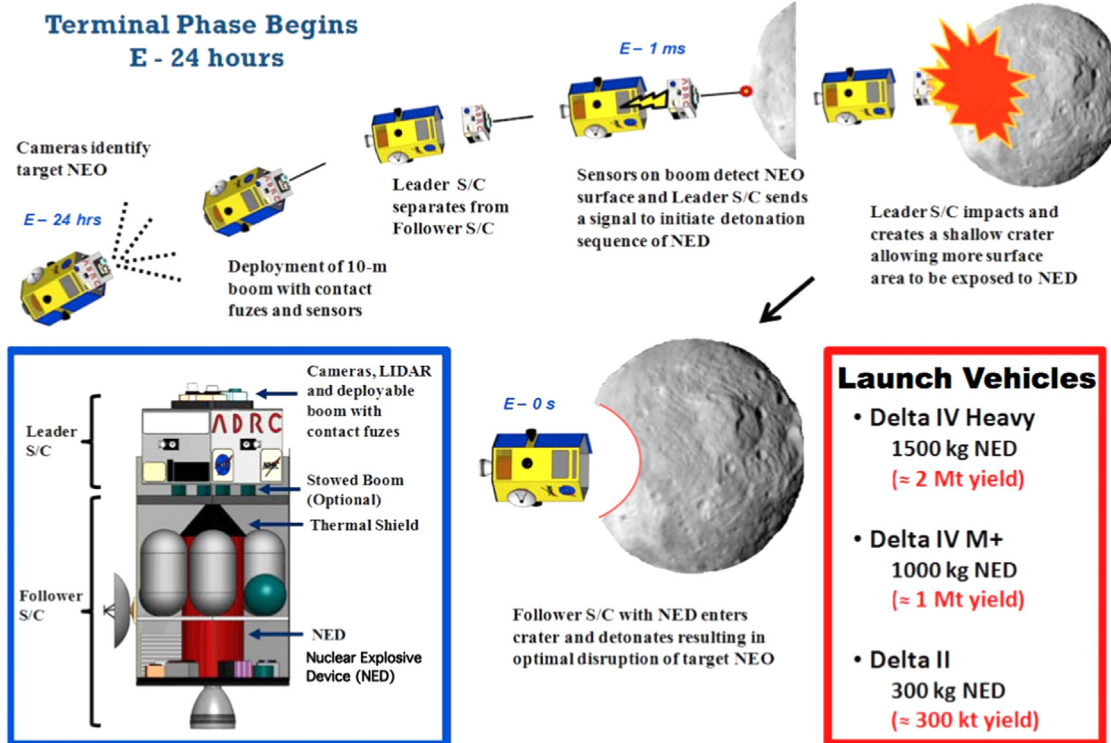


Fig. 1. A baseline HAIV and its terminal-phase operational concept [9].

Some of the most notable missions are the Hayabusa Mission by JAXA, and the NEAR-Shoemaker and Deep Impact missions by NASA. The Hayabusa spacecraft, formerly known as MUSES-C, was sent to the asteroid 25143 Itokawa, a near-Earth asteroid (NEA) $535 \times 294 \times 209$ m in size. While at the asteroid, the spacecraft performed two landings for the purpose of collecting surface samples, which were subsequently returned to Earth in June 2010. However, problems with the sample collection mechanism resulted in only tiny grains of asteroid material being returned. The spacecraft also had a small lander onboard, called MINERVA, which was to be guided to the surface of the asteroid. Unfortunately, the lander drifted into space and was unable to complete its mission. The NEAR-Shoemaker mission was designed to study the asteroid 433 Eros, which is one of the largest NEOs at $34.4 \times 11.2 \times 11.2$ km in size. This spacecraft was the first to orbit an asteroid as well as the first to land on one. While the Hayabusa and NEAR-Shoemaker missions were designed to softly touch down on the surface of their respective asteroids, the Deep Impact mission was designed to do just the opposite. Approximately 24 h prior to impact with the comet Tempel 1, which is 7.6×4.9 km in size, the impactor was separated from the flyby spacecraft and autonomously navigated to ensure a hypervelocity impact at a relative speed of 10.3 km/s.

More recently, ESA proposed a demonstration mission for a kinetic-impactor called the Don Quijote mission [10,11]. 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 studied the target for some months, Hidalgo would approach the target at a relative speed of approximately 10 km/s. Sancho then observes any changes in the asteroid and its heliocentric orbit after the kinetic impact to assess the effectiveness of this deflection strategy. Don Quijote was planned to launch in early 2011 and complete its mission in mid to late 2017. However, the mission concept was never realized due to higher than expected mission costs.

The selection process for the Don Quijote mission was based on a set of NEO characteristics defined by ESA's NEOMAP. Their analysis resulted in the selection of the asteroids 2002 AT₄ and 1989 ML. The asteroid 2002 AT₄ is roughly half the size of 1989 ML, but intercepting it requires a higher ΔV . A realistic deflector spacecraft would require a versatile design capable of intercepting and deflecting or disrupting either target on short notice.

One last notable future mission planned by NASA is the OSIRIS-REx asteroid sample return mission, which will return a sample from NEA 101955 Bennu (formerly known as 1999 RQ₃₆). This mission will launch in September of 2016 and will return the sample to Earth in September of 2023. This mission will utilize large DSMs, Earth gravity assist (GA), rendezvous and proximity maneuvers, and an asteroid departure maneuver. The proposed HAIV demonstration mission will incorporate a combination of the knowledge gained from the development and execution of these NEO science missions.

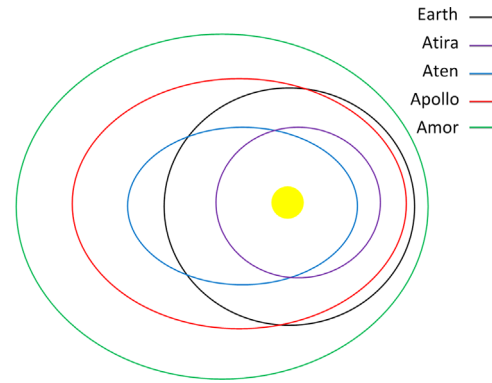


Fig. 2. Comparison of Atira, Apollo, Aten, and Amor class asteroid orbits in relation to the Earth's orbit.

1.2. Near-earth asteroid (NEA) groups

For the purposes of this study, NEAs in the Atira and Amor orbit groups were considered. A comparison of NEA orbit families is shown in Fig. 2. NEAs in these groups all have perihelion distances < 1.3 AU, and many of them also cross the Earth's orbit at some point. The proximity of NEA orbits to Earth's orbit means that the ΔV required for intercept is usually small. As such, we expect that a number of NEAs will prove to be viable candidates for an NEA deflection/disruption flight validation mission. Apollo and Aten NEA orbits cross Earth's orbit, and in some cases this leads to lower mission ΔV requirements as compared to Atiras or Amors. On the other hand, this same fact means that any significant perturbation to the NEA's orbit could cause it to later impact the Earth. While unlikely, we do not want our demonstration of deflection technologies to cause such a thing 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 [10].

To preclude the possibility of inadvertently perturbing a previously harmless NEA onto an Earth collision course, we consider only Atira and Amor NEAs in our study. 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 the time of this study, July 21st, 2012, there were 3398 Amor and Atira asteroids listed in NASA's NEO Program database.¹ While Amor asteroids are entirely outside of the Earth's orbit, the Atira asteroid group orbits are contained entirely within the orbit of the Earth. Because the orbits of Atiras and Amors are entirely interior or exterior to Earth's orbit, respectively, disturbances to the orbits of those asteroids are not likely to cause an impact with the Earth at any time after the mission.

1.3. Mission design software

In addition to the HAIV concept design, the ADRC has also developed mission design software tools capable of performing advanced mission analysis for thousands of

¹ <http://neo.jpl.nasa.gov/stats/>.

potential target asteroids. 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 a hybrid algorithm that utilizes both an evolutionary algorithm and traditional non-linear programming solvers. The types of missions considered in this paper can easily be formulated as constrained optimization problems, making them ideal for hybrid genetic-nonlinear programming (GNLP) algorithm. For this approach, each mission type must first be formulated as a single-valued cost function with constraints.

Utilizing a combination of evolutionary and nonlinear programming algorithms, and the modular mission design software, it is possible for multiple mission architectures for thousands of possible target asteroids to be quickly and efficiently analyzed. Given the capabilities of the mission design software several types of missions have been considered and are detailed in the following section. Details of the hybrid GNLP algorithm are presented in [Appendix A](#).

This hybrid optimization algorithm has previously been utilized to determine optimal trajectories for planetary defense demonstration missions utilizing single and multiple launch vehicle and spacecraft scenarios [12], as well as to determine optimal trajectories for multiple gravity assist outer-planetary missions similar to NASA's Galileo and Cassini missions [13]. If addition, this optimization algorithm has also been utilized to determine optimal trajectories for possible asteroid redirect missions (ARM) [14].

1.4. Mission types analyzed

In this paper a total of 5 separate mission types will be analyzed. The first mission type is the two impulse direct intercept in which the HAIV impactor is sent on a direct intercept trajectory for the target asteroid and is referred to as direct intercept mission.

A total of 4 mission types are evaluated in which a second spacecraft it utilized to observe the impact at the target asteroid. With these 4 mission types only missions with a single launch are considered, with the main driver being that the rendezvous spacecraft must arrive prior to the HAIV impactor. In this paper Type 1 missions are mission defined in which HAIV interceptor is launched on a direct intercept mission. The observer spacecraft then utilized a deep-space maneuver (DSM) to target and asteroid arrival prior to impact. Type 2 missions are similar to Type 1 missions, except that the observer spacecraft is launch on a direct intercept trajectory and the HAIV impactor utilizes a DSM to target the asteroid after the initial observer rendezvous.

The last two mission types utilized a gravity assist at a planet chosen by the optimization algorithm. With Type 3 missions the observer spacecraft is launch on a direct intercept trajectory, while the HAIV spacecraft utilized a DSM to target a single gravity assist planet. Type 4 missions are identical to Type 3 missions, except the HAIV impactor utilized 2 gravity assists prior to impacting the target asteroid.

2. Problem formulation and mission constraints

The purpose of this paper is to determine realistic mission designs for a HAIV demonstration that can be flown

in the near future. Several mission constraints are imposed during the mission design survey to ensure that the resulting missions are realistic at the level of the current analysis. These constraints are enforced by the GNLP algorithm and are used to shape the solutions and ensure the best optimal solutions are found while satisfying constraints.

The first constraint imposed is designed to ensure that the HAIV spacecraft will be able to communicate with Earth ground stations during the final impact phase of the mission. This is done by adding a constraint that ensure the impact will not occur on the opposite side of the sun from the Earth. The exact angle limitations for the impact Earth–Asteroid angle are that the angle must be $< 175^\circ$ from the Earth and $> 185^\circ$ from the Earth. [Fig. 3](#) illustrates the line of sight angular constraints. The second main constraint is a guidance and navigation constraint which requires that the impactor approaches from the sun facing side of the asteroid. This constraint ensures proper lighting conditions for the terminal guidance phase of the mission.

The purpose of this mission is to demonstrate the feasibility of the HAIV for the purpose of planetary defense. The HAIV spacecraft is designed for hypervelocity impacts, which are likely to be required by planetary defense missions executed with short warning time. Therefore, missions designed in this paper must have a minimum impact velocity of 5 km/s. Due to anticipated technological limitations, impact velocities over 30 km/s are penalized as well. However, it should be noted that none of the mission analyzed have had an impact velocity above 30 km/s, meaning the HAIV impactor never has to decrease its approach velocity. All other major mission constraints can be found in [Table 1](#). The exact penalty functions are also discussed at the end of this section. It is worth noting that limitations on NEA orbit group and absolute magnitude (H) reduce the number of asteroids that must be searched from approximately 3500 to 1500.

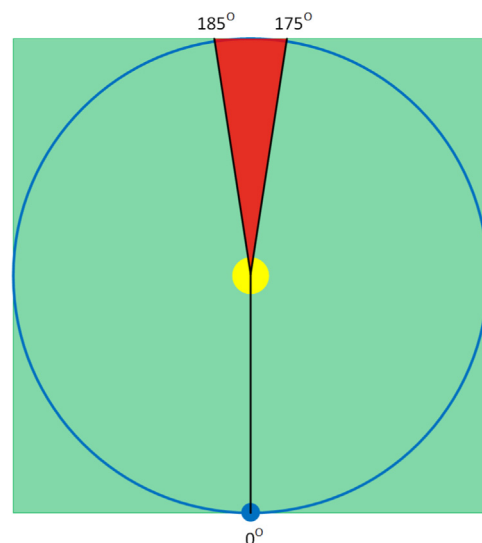


Fig. 3. Illustration of the Earth–Sun–Asteroid line-of-site communication angle for the HAIV mission. Green indicates communicable from Earth ground stations, while red indicates area where communications are not possible. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Table 1
List of mission constraints.

Asteroid types	Amor, Atira
Earlier launch date	1-Jan-2018
Latest launch date	31-Dec-2022
Minimum impact velocity	5 km/s
Maximum impact velocity	30 km/s
<i>H</i> magnitude range	20.75–23.62
Communication LOS constraints	> 185 and < 175 °
Impact angle constraint	Penalized dark-side approaches
Impactor limitations	For dual missions the impact must occur after the rendezvous S/C arrival
Maximum departure C_3	12.5 km ² /s ² for single S/C mission 30 km ² /s ² for dual S/C missions

Depending on mission complexity, there are different ways to formulate multiple gravity-assist (MGA) problems. Simple direct intercept missions are a subset of the MGA model. Missions such as NASA's Voyager, Pioneer, and Galileo missions, which don't require large deep space maneuvers (DSMs), can be formulated using the MGA model. This model requires fewer variables than the three impulse MGA-DSM model, meaning they have a much lower computational cost. On many occasions, small DSMs, when compared to the departure and arrival maneuvers, can significantly lower the total overall mission cost in terms the total ΔV or maximizing arrival mass. The formulation for both types of missions are included in the following subsections. Most of the solution algorithms needed for both problem types will be included. However, expositions of solutions to Lambert's problem, solutions to Kepler's equation, date conversion algorithms, and so on are omitted for brevity.

The intent of both methods is to determine a final cost function that the hybrid GNLP algorithm can optimize. That is, minimize or maximize, depending on how the individual problem is formulated. Cost functions typically consist of all of the required mission ΔV 's, the Earth departure v_∞ , the final arrival v_∞ , and any other mission constraints. Any permutation of these elements can be used for each specific mission type analyzed in this paper. For instance in this study the departure C_3 only affects the final cost function if it is above 12.5 km²/s² for the single spacecraft missions and 30 km²/s² for the dual spacecraft missions. In space mission design this is often done because the launch vehicle upper stage can be leveraged for the Earth departure maneuver (typically, limited by C_3). It is worth noting that for the GNLP algorithm used in this paper, all mission constraints can be modeled as nonlinear constraints or applied directly to the cost function.

2.1. Multiple gravity-assist model

The goal of this section is to develop a description of how each of the mission types and accompanying cost functions are developed. For all of these missions the problem variables are the launch dates and times-of-flight for each mission leg. The planetary and asteroid positions are known as a function of time. From solutions to Lambert's problem and the multiple gravity-assist model (MGA) described in this section, all of the missions analyzed in this study can be constructed.

With the multiple gravity-assist (MGA) two-impulse model planetary flybys are modeled using the 3-dimensional patched

conic method developed by Minovitch [15]. The flyby orbit geometry is defined by the incoming and outgoing spacecraft velocity vectors, which are given by two solutions to Lambert's problem [16]. This *patching* results in a powered hyperbolic orbit for each gravity-assist that requires a ΔV at the perigee of each flyby. It should be noted that the ΔV required for each gravity-assist is typically driven to a near 0 value, resulting in a free flyby.

For each gravity assist, the incoming and outgoing velocity vectors (in the heliocentric frame) are given directly from solutions to Lambert's problem [17–21]. The incoming and outgoing velocity vectors, relative to each planet, are then found by subtracting the planet's velocity from the incoming and outgoing spacecraft velocities, given from the solutions to Lambert's problems as follows:

$$\vec{v}_{\infty-in} = \vec{v}_{s/c-in} - \vec{v}_\oplus \quad (1)$$

$$\vec{v}_{\infty-out} = \vec{v}_{s/c-out} - \vec{v}_\oplus \quad (2)$$

where \vec{v}_\oplus is the planetary velocity of the gravity assist planet. The variables $\vec{v}_{s/c-in}$ and $\vec{v}_{s/c-out}$ are the spacecraft incoming and outgoing velocity vectors.

The perigee radius, which is required to patch the two solutions together, is a function of the incoming and outgoing orbits. The first step is to determine the semi-major axis of the incoming and outgoing hyperbolic trajectories as

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

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

where μ_\oplus is the target planet's gravitational parameter, while $v_{\infty-in}$ and $v_{\infty-out}$ are the magnitude of the incoming and outgoing velocities relative to the planet.

The required turning angle for the gravity-assist is a function of the incoming and outgoing v_∞ vectors, defined as

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

With this model the flyby perigee radius must be equal for both legs of the hyperbolic orbit, as described by

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

where e_{in} and e_{out} are the incoming and outgoing orbit eccentricities. It should be noted that two eccentricities should be greater than 1 because hyperbolic incoming and

outgoing orbits are required to avoid being captured by the planet. The turning angle δ can also be represented as the sum of the transfer angles for the incoming and outgoing orbits, represented as follows:

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

Eqs. (6) and (7) can be written into a single equation that can be iteratively solved in order to determine the incoming and outgoing eccentricities, as follows:

$$f = \left(\frac{a_{out}}{a_{in}}(e_{out} - 1)\right) \sin\left(\delta - \sin^{-1}\left(\frac{1}{e_{out}}\right)\right) - 1 = 0 \quad (8)$$

The above equation, which is a function of the unknown parameter e_{out} , must be solved with an iterative root-finding method. For this problem, a simple Newton root-finding method works well. To start the Newton iteration an initial value for e_{out} of 1.5 works well. In a Newton iteration scheme a while loop is used until e_{out} stops changing within a specified tolerance. The number of iterations should also be monitored, so the process can be terminated if a solution does not converge. The required first derivative of f with respect to e_{out} is

$$\begin{aligned} \frac{df}{de_{out}} = & \left(\frac{a_{out}}{a_{in}}e_{out} - \frac{a_{out}}{a_{in}} + 1\right) \frac{\cos\left(\delta - \sin^{-1}\frac{1}{e_{out}}\right)}{e_{out}^2 \sqrt{1 - \frac{1}{e_{out}^2}}} \\ & + \frac{a_{out}}{a_{in}} \sin\left(\delta - \sin^{-1}\frac{1}{e_{out}}\right) \end{aligned} \quad (9)$$

When a converged e_{out} is found the perigee radius is calculated from Eq. (6). Finally, the ΔV required to patch the two orbits at the perigee can be determined as follows:

$$\Delta V_{GA} = \left| \sqrt{v_{\infty-in}^2 + \frac{2\mu_{\oplus}}{r_p}} - \sqrt{v_{\infty-out}^2 + \frac{2\mu_{\oplus}}{r_p}} \right| \quad (10)$$

The flyby perigee radius and ΔV are functions of 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, so the only decision variables with this model are the date of Earth departure and time-of-flight for each additional leg of the trajectory. The final cost function, C , for the MGA optimization problem is represented as a function of the decision variables, \mathbf{X} , as follows:

$$C = f(\mathbf{X}) + g(\mathbf{X}) \quad (11)$$

$$\mathbf{X} = [T_0, T_1 \dots T_f]^T \quad (12)$$

where T_0 is the launch date and T_i are the time-of-flights for each leg of the mission. Typical penalty function, $g(\mathbf{X})$, will be discussed at the end of this section.

The final cost functions are then constructed by summing all of the required mission ΔV 's, as well as any other terms the user wishes to minimize, such as the initial departure or arrival v_{∞} 's. Globally optimal solutions of the MGA objective function(s) can then be found with the proposed hybrid optimization algorithm.

2.2. Problem constraints

In optimization problems, constraints are often used to help shape the final solution and ensure only feasible solutions are found. When using optimization algorithms, constraints are used instead of hard variable limits. This ensures all possible solutions are continuous and can be optimized with the gradient based hybrid optimization algorithm.

Common constraints for mission design problems include flyby altitude limits, mission time constraints, and penalties designed to protect against low velocity flybys. During a low-velocity flyby the spacecraft would be captured by the flyby planet, rather than gaining velocity in the heliocentric frame. These are rare and can typically only occur at the first planet in the flyby sequence. Any additional constraint which help shape the solution can be added, as long as the constraint values are approximately the same order of magnitude at the intended objective function values. It should be noted that when using the MGA-DSM model in conjunction with the hybrid algorithm all of the variables are explicitly constrained. However, low-velocity flybys may still occur.

The MGA model often results in a perigee radius that passes through the planet or close to the planet's atmosphere. In this situation a constraint function to move the solution toward more feasible trajectories can be expressed as [22]

$$g_i(\mathbf{X}) = -2 \log \frac{R_{pi}}{kR_{\oplus}} \quad (13)$$

where k is a multiplier used to define how close the spacecraft is allowed to flyby a target planet. For the Cassini mission a value of 1.05 is used. However, the Galileo mission had an Earth close approach altitude of 300 km corresponding to a k value of approximately 1.047.

The next constraint penalizes low velocity flybys. This method has also been adapted from [22]. Low velocity flybys are very rare, but solutions should still be protected from these unfeasible trajectories. The orbital energy, about the flyby planet, can be calculated as

$$E = \frac{|\vec{v}_{\infty-in}|^2}{2} - \frac{\mu_{\oplus}}{R_{soi}} \quad (14)$$

The radius of the planetary sphere of influence, denoted as R_{soi} , is crucial for all patched conic methods and is defined as

$$R_{soi} = r_{\odot-\oplus} \left(\frac{m_{\oplus}}{m_{\odot}}\right)^{2/5} \quad (15)$$

where $r_{\odot-\oplus}$ is the radius from the sun to the targeted planet.

For the flyby orbit to be hyperbolic about the planet, E must be greater than 0. However, the sphere of influence model is an approximation, so an additional 10% margin on the incoming velocity is added. A simple penalty that is scaled inversely to the flyby velocity is used. For relatively large v_{∞} values, the penalty is near 0 and is small compared to the overall cost function value. Alternatively, for very low v_{∞} values, the penalty is large enough to 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:

$$E = \frac{|0.9\vec{v}_{\infty-in}|^2}{2} - \frac{\mu_{\oplus}}{R_{soi}} \quad (16)$$

$$g_i(\mathbf{X}) = \begin{cases} 0 & E \geq 0 \\ \frac{1}{|\vec{v}_{\infty-in}|} & E < 0 \end{cases} \quad (17)$$

For the MGA model, trajectories can be found with Earth departure C_3 values that exceed the specified limit. The penalty function is formulated to represent that additional Earth departure ΔV beyond that provided by the launch vehicle's upper stage would be required from the HAIV spacecraft in order to attain the required C_3 . The exact penalty function is

$$g_i(\mathbf{X}) = \begin{cases} v_{\infty-l} - \sqrt{C_{3max}} & v_{\infty-l} > \sqrt{C_{3max}} \\ 0 & v_{\infty-l} \leq \sqrt{C_{3max}} \end{cases} \quad (18)$$

A time constraint is used to ensure that the rendezvous spacecraft arrives at the asteroid prior to the arrival of the HAIV spacecraft and is represented as follows:

$$g_i(\mathbf{X}) = \begin{cases} 0.1(T_{rend.-arrival} - T_{impactor.-arrival}) & T_{rend.-arrival} > T_{impactor.-arrival} \\ 0 & T_{rend.-arrival} \leq T_{impactor.-arrival} \end{cases} \quad (19)$$

The next constraint is added to ensure the communication line-of-sight angle is feasible during the final terminal impact phase. This is especially important in the absence of an observer spacecraft, as communications with the Earth just prior to impact will be the only way to determine mission success. The line-of-sight angle is then found as

$$LOS = \cos^{-1} \left(\frac{\vec{R}_{\oplus} \cdot \vec{R}_{ast}}{R_{\oplus} R_{ast}} \right) \quad (20)$$

where \vec{R}_{\oplus} is the Earth radius vector at impact, while \vec{R}_{ast} is the asteroid radius vector at impact. To ensure the LOS angle is in the right quadrant, the z component of the cross product of the two radius vectors is utilized as follows:

$$\vec{c} = \vec{R}_{\oplus} \times \vec{R}_{ast} \quad (21)$$

$$LOS = 2\pi - \arccos \left(\frac{\vec{c} \cdot \vec{c}}{|\vec{c}|^2} \right) \quad (22)$$

The final penalty function for the LOS angle constraints is given in the following constraint equation:

$$g_i(\mathbf{X}) = \begin{cases} \exp \left(\frac{-1}{1 - (LOS - 180)^2} \right) & 175^\circ \leq LOS \leq 185^\circ \\ 0 & \text{for all other angles} \end{cases} \quad (23)$$

The final penalty function used to ensure mission feasibility penalizes the relative asteroid velocity with respect to the asteroid's position relative to the sun. This ensures that the spacecraft arrives on the sunward side of the asteroid, which results in lighting conditions that are favorable for the terminal guidance system. The asteroid radius and impactor relative velocity unit vectors are utilized to preclude numerical scaling issues. Unlike the time and LOS angle penalties, all angles greater than 0 (which corresponds to approaching directly along the asteroid-sun line from the sunward side) are penalized with a linear function. This is done because the

approach angle is one of the most critical parameters to ensure mission success, as follows:

$$SA = \cos^{-1}(\vec{e}_r \cdot \vec{e}_v) \quad (24)$$

$$g_i(\mathbf{X}) = \frac{1}{\pi} SA \quad (25)$$

Additional penalty constraints can be added to shape the solution as desired. For MGA and MGA-DSM missions, the final constraint function is represented as the sum of each individual constraint by

$$g(\mathbf{X}) = \sum g_i(\mathbf{X}) \quad (26)$$

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

3. Mission analysis results

In this section the mission design results for the various mission architectures considered in the study are presented. Each of these mission types can be constructed using the methods described in the previous section. The hybrid GNLP algorithm described in [Appendix A](#) was then used to optimize each constructed cost function along with all mission constraints. More details on common limits used for each variable can be found in [\[22,12,13\]](#). For each mission type approximately 1500 potential target asteroids were scanned for feasible missions. The results presented in the following three subsections are subsets of both the standard MGA and MGA-DSM models.

3.1. Direct intercept missions

For the direct intercept mission the launch vehicle is used to inject the HAIV into an interplanetary orbit that directly impacts with the target asteroid. This is the simplest mission type analyzed, with only two variables to optimize (launch date and time-of-flight to the asteroid), which yields the largest number of feasible target asteroids. For direct intercept missions a relatively low C_3 limit of $12.5 \text{ km}^2/\text{s}^2$ is used. By limiting missions to low C_3 it may be possible to use a less capable (and therefore less expensive) launch vehicle than for missions that include both an impactor and rendezvous (observer) spacecraft.

With the low C_3 limit, hundreds of potential target asteroids were found during the search. These asteroids require no ΔV from the HAIV impactor, other than small statistical course corrections during the terminal impact phase of the mission. The results presented in [Table 2](#) represent the top 3 asteroids determined by a joint study with the Mission Design Lab (MDL) at the Goddard Space Flight Center and the ADRC. These asteroids were chosen because their orbits are fairly well known (or future observations of them will be possible prior to the mission launch dates), the estimated diameters (assuming and albedo of 0.25) are close to the desired diameter of 100 m, they have low sun approach angles, and the HAIV impactor arrival velocities are high enough to be technically challenging, yet feasible. How well the asteroid orbits are known is expressed by the Orbit Condition Code

(OCC) for the orbit, which is an integer scale ranging from 0 (a very well-known orbit) to 9 (a very poorly known orbit). The OCC value is the same as the Minor Planet Center's (MPC) "U" parameter. In Table 2 the label TOF stands for time-of-flight and will be used throughout the results section.

3.2. Combined rendezvous and direct intercept missions

Direct intercept missions are the baseline reference mission for the HAIV studies at the ADRC; however it may be

Table 2

Top 3 asteroids for the single direct intercept mission.

Asteroid	2006 CL ₉	2009 QO ₅	2004 BW ₁₈
<i>a</i> (AU)	1.35	1.59	1.37
<i>e</i>	0.24	0.24	0.25
Diameter (m)	105	106	97
Departure C_3 (km ² /s ²)	12.0	12.5	12.5
Require S/C ΔV (km/s)	0.00	0.00	0.00
LOS angle	349.01°	349.33°	333.17°
Sun approach angle	3.04°	28.05°	34.21°
Departure date	2-Aug-19	27-Mar-19	7-Apr-19
TOF (days)	121	124	268
OCC	5	1	5
Arrival velocity (km/s)	11.5	9.2	6.6

useful to have a spacecraft at the asteroid prior to impact. This rendezvous spacecraft would likely be a small spacecraft attached to the HAIV spacecraft that would separate at some point during the mission. The goal of the following sections is to determine a trajectory solution such that the HAIV spacecraft has a relative intercept velocity greater than 5 km/s and the observer spacecraft can rendezvous with the asteroid prior to impact. Several mission types are presented that attempt to minimize the total ΔV required by both the HAIV impactor and the rendezvous spacecraft such that the mission design can be flown using existing launch vehicles.

During initial mission analysis we determined that no feasible trajectories could be found with the relatively low C_3 used for the direct intercept mission. We therefore revised the maximum allowable C_3 to 30 km²/s² and considered Delta IV and Atlas V launch vehicles.

The first mission type that we analyzed employs a single DSM after the separation of the two spacecraft. For this type of mission, referred to herein as a Type 1 mission, the HAIV interceptor continues on a direct intercept course, while the rendezvous spacecraft uses a DSM to target asteroid intercept at a later date. The results for Type 1 missions are shown in Table 3. The lowest total ΔV required is 3.299 km/s for asteroid 2010 KU₇. However, this ΔV is likely too high to be feasible. By examining the results it appears that the impact velocity was often driven

Table 3

Top 5 asteroids by total required ΔV for Type 1 missions.

Asteroid	2010 KU ₇	2012 JX ₁₁	2001 CK ₄₂	2009 CR ₄	2000 RD ₅₃
<i>a</i> (AU)	1.67	1.97	1.42	1.75	1.79
<i>e</i>	0.38	0.48	0.28	0.42	0.43
Diameter (m)	95	58	266	123	280
Launch date	14-Jul-21	7-Jun-18	18-May-18	28-Apr-21	19-Oct-22
TOF _{HAIV} (days)	668	801	543	677	652
TOF _{Rend} (days)	629	758	495	657	615
C_3 (km ² /s ²)	25.6	33.9	16.8	33.8	24.1
$\Delta V_{HAIV-DSM}$ (km/s)	2.387	1.990	3.258	2.153	2.030
$\Delta V_{Rend-Arr}$ (km/s)	0.912	1.186	0.453	1.247	1.725
LOS	26.2°	257.8°	128.8°	19.9°	39.9°
Sun angle	97.8°	102.3°	93.0°	94.4°	106.0°
Impactor velocity (km/s)	5.0	5.0	5.0	5.0	5.0
Total ΔV (km/s)	3.299	3.518	3.711	3.737	3.755

Table 4

Top 5 asteroids by total required ΔV for type 2 missions.

Asteroid	2000 WO ₁₄₈	2009 TV ₄	1998 UM ₁	1996 FO ₃	2008 XB
<i>a</i> (AU)	1.64	1.69	1.7	1.44	1.51
<i>e</i>	0.38	0.37	0.40	0.29	0.31
Diameter (m)	199	59	60	212	81
Launch date	19-Jan-20	27-Sep-20	20-Sep-18	11-Feb-22	2-Dec-21
TOF _{Rend} (days)	515	336	405	354	396
TOF _{HAIV} (days)	770	786	791	626	661
C_3 (km ² /s ²)	26.6	29.2	33.1	16.8	24.2
$\Delta V_{DSM-Rend}$ (km/s)	1.265	0.562	1.448	0.951	1.632
$\Delta V_{Rend-Arr}$ (km/s)	1.338	2.095	0.976	1.975	1.310
LOS	64.0°	198.2°	149.6°	202.8°	163.2°
Sun angle	91.7°	101.0°	60.9°	90.1°	100.7°
Impact velocity (km/s)	5.0	5.0	5.0	5.0	5.0
Total ΔV (km/s)	2.603	2.657	2.709	2.926	2.942

to the lower limit of 5 km/s, which indicates that lowering the minimum required impact velocity may lower the total mission ΔV . For mission types 1 to 4 all the times-of-flight (TOF) values represent the total time of flight from launch for both the rendezvous and HAIV spacecraft, including flight times prior to separation.

The second type of mission analyzed, referred to herein as Type 2, and is similar to the Type 1 mission except that the HAIV interceptor performs the DSM to target the asteroid at a different date instead of the rendezvous spacecraft. The top 5 results for this search are shown in Table 4. Examination of the results reveals that, on average, the total required ΔV is reduced by approximately 1 km/s, with the lowest required ΔV being approximately 2.6 km/s. Like the Type 1 mission results the impact arrival velocity is often driven to 5 km/s. As mentioned previously, reducing the minimum allowable impact velocity may reduce the total mission ΔV required.

3.3. Gravity-assist missions using the MGA model

Planetary gravity assists are often used to reduce the required ΔV for outer planet missions, and this is also possible for the missions considered herein. Gravity assist (s) can lower the average required total mission ΔV to the 1–1.5 km/s range, with some solutions found that require as little as approximately 600 m/s. For the following Type 3 and Type 4 missions the optimizer was used to decide which planet(s) should be used for the gravity assist(s). In all of the best cases presented, the Earth was found to be the best planet for the gravity assist(s). Given that most of the target asteroids have a semi-major axis between 1 and 1.3 AU, this is an intuitively satisfying result.

Type 3 missions use a single gravity assist achieved by having the HAIV impactor perform a DSM to target the planetary flyby. The gravity assist is then used to increase the velocity of the impactor, thus lowering the required ΔV

Table 5

Top 5 asteroids by total required ΔV for Type 3 missions.

Asteroid	2012 OO	2008 XB	2000 WO ₁₄₈	2009 CO ₅	1996 FO ₃
a (AU)	1.7	1.51	1.64	1.66	1.44
e	0.38	0.31	0.38	0.35	0.29
Diameter (m)	214	81	199	120	212
Launch date	3-Sep-21	13-Dec-21	15-Jan-20	14-Mar-22	21-Feb-22
TOF _{Rend} (days)	628	232	570	554	245
TOF _{HAIV – Leg1} (days)	729	730	718	736	730
TOF _{HAIV – Leg2} (days)	71	227	157	717	276
C_3 (km ² /s ²)	34.5	29.7	27.6	30.4	30.5
ΔV_{HAIV} (km/s)	0.546	0.764	0.264	0.496	1.094
ΔV_{Rend} (km/s)	0.599	0.558	1.146	0.934	0.445
Gravity assist planet	Earth	Earth	Earth	Earth	Earth
R_p (km)	7,398	17,249	8,033	30,908	13,098
LOS	348.5°	289.5°	316.5°	332.9°	281.7°
Sun angle	134.7°	6.2°	157.0°	164.7°	4.9°
Impact velocity (km/s)	5.8	5.1	6.1	5.0	5.6
Total ΔV (km/s)	1.145	1.322	1.410	1.430	1.539

Table 6

Top 5 asteroids by total required ΔV for type 4 missions.

Asteroid	2012 CR	2011 FR ₁₇	2010 XB ₇₃	2010 GZ ₃₃	2011 AL ₂₄
a (AU)	1.77	1.7	1.71	1.91	1.72
e	0.38	0.30	0.31	0.42	0.35
Diameter (m)	120	56	102	91	91
Launch date	25-Feb-19	10-Mar-22	30-Nov-19	3-Apr-18	10-Jan-20
TOF _{Rend} (days)	367	406	351	325	391
TOF _{HAIV – Leg1} (days)	731	730	733	730	731
TOF _{HAIV – Leg2} (days)	45	569	352	1224	1126
TOF _{HAIV – Leg3} (days)	605	533	540	768	565
C_3 (km ² /s ²)	26.8	29.5	30.6	30.0	32.1
Impactor S/C ΔV (km/s)	0.461	0.303	0.230	1.150	0.373
Rendezvous arrival ΔV (km/s)	0.147	0.702	0.988	0.104	0.933
First gravity assist planet	Earth	Earth	Earth	Earth	Earth
R_p (km)	34,570	32,342	40,987	7,151	19,7861
Second gravity assist planet	Earth	Earth	Earth	Earth	Earth
R_p (km)	28,357	87,772	21,174	20,646	91,587
LOS angle	341.4°	54.1°	30.2°	201.4°	26.9°
Sun angle	175.9°	168.5°	167.6°	174.9°	167.5°
Impact velocity (km/s)	12.3	9.5	7.2	19.4	10.2
Total ΔV (km/s)	0.608	1.005	1.218	1.254	1.306

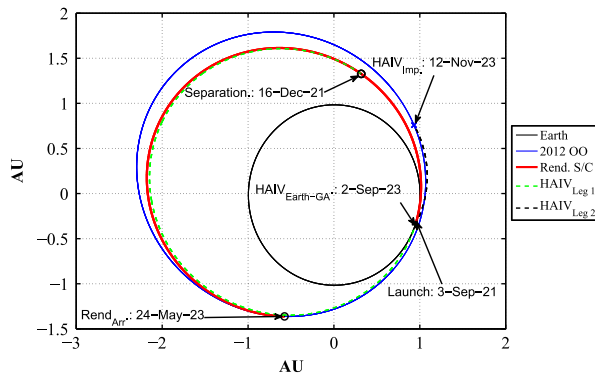


Fig. 4. Trajectory plot for Type 3 mission to asteroid 2012 OO.

when compared to Type 1 and Type 2 missions. Missions where the rendezvous spacecraft performs a gravity assist to lower the arrival ΔV were also considered, however this did not improve the total required ΔV compared to Type 1 and Type 2 missions.

The list mission type, Type 4, considered for the MGA model employs two gravity assists. The HAIV impactor targets the first gravity assist planet with a DSM. After the first gravity assist the spacecraft flies on a ballistic trajectory until the second gravity assist. This resulted in total mission ΔV being reduced from the Type 3 solutions. As was the case with Type 3 missions, the optimal gravity assist planet is the Earth for both assists. Type 4 mission results are shown in Table 5. For these missions the lowest required total ΔV is reduced to approximately 600 m/s, ranging up to 1.3 km/s for the top 5 asteroids (Table 6).

A summary of the best Type 3 mission results is shown in Table 5. The best mission requires a total ΔV of approximately 1.15 km/s, which represents a nearly 1.5 km/s reduction from the best solution found for the previous mission types. To illustrate the schematic of these gravity assist missions, the trajectory of the Type 3 mission to asteroid 2012 OO is shown in Fig. 4.

4. Conclusions

Throughout this paper several possible mission types for a HAIV demonstration mission have been analyzed. A simple direct intercept mission was found to be the best option for the HAIV flight demo mission. The direct intercept mission has the largest number of feasible candidate target asteroids, requires a minimal total post-launch ΔV (close to 0 km/s for the optimal cases presented herein), and is a representative of the sort of worst-case asteroid mitigation mission scenario that we should be prepared for. More advanced missions, which enable an observer spacecraft to arrive at the asteroid prior to the main spacecraft impact, have also been analyzed. The results generally show that allowing the impactor spacecraft to perform multiple gravity assists lowers the total required ΔV for each mission considered as well as increases the impactor arrival velocity. There are multiple possible target asteroids which require a total ΔV of 1.5 km/s or less, with the lowest combined rendezvous/impact mission ΔV of approximately 600 m/s.

Acknowledgments

This research work has been supported by a NASA Innovative Advanced Concepts (NIAC) Phase 2 Project Grant, NASA Grant and Cooperative Agreement Number NNX12AQ60G. The authors would like to thank Dr. John (Jay) Falker, the NIAC Program Executive, for his support.

Appendix A. Hybrid genetic-nonlinear programming optimization algorithm

Given the basics for each objective function formulation the next step is to develop an optimization algorithm capable of determining global minimum solutions for low thrust problems formulated in the previous section. The motivation for the optimization algorithm discussed in this section is to develop an algorithm that is capable of solving complex problems with no prior knowledge of the solution structure or approximate solutions, as is required for typical gradient based non-linear programming optimization. To accomplish this, a hybrid algorithm has been developed that is able to combine the global optimization capabilities of evolutionary algorithms with the local optimization capabilities of traditional NLP solvers. It will be shown that the final algorithm is capable of determining optimal (or at least very near optimal) solutions for preliminary low-thrust trajectories. The low-thrust trajectory optimization problem is known to be one of most difficult optimization problems in astrodynamics and is an area of active search [23–28]. The hybrid GNLP algorithm is able to find complete solutions, including the number of gravity-assists and flyby order, for the two and three-impulse class of problems.

A.1. Genetic algorithm

The heart of a genetic algorithm is the stochastic simulation of natural selection, reproduction, and mutations found in natural evolution. In these simulations, genetic operators are used to 'evolve' an initial population, through genetic operators, to determine the best fitness design [27]. The purpose of this section is to discuss the basics of the genetic algorithm developed for the final hybrid optimization algorithm. The genetic algorithm is responsible for the global optimization capabilities of the final algorithm.

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 the objective function, which was developed in the previous section. 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 a single locally optimal solution. Genetic algorithms also perform well in complex nonlinear and discontinuous design spaces. Despite all the advantages, evolutionary algorithms also have a downside. They almost always require a greater number of cost function evaluations than traditional gradient based methods, 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 bound-constrained minimization problems and may not always perform well for unconstrained minimization when very large bounds are used. Many shortcomings of evolutionary algorithms are alleviated in the final hybrid algorithm. By utilizing a local NLP gradient based solver the hybrid algorithm requires significantly smaller populations and will have provide at least locally optimal solutions.

The basic genetic operators include selection, reproduction, crossover, mutation, and elitism. The algorithm developed in this section can utilize a number of different user selected selection, reproduction, crossover, and mutation methods. In a real valued genetic algorithm (as implemented for this study), each variable is represented by either its integer or real number value. Each real and integer variable is referred to as a gene. A complete set of variables, which defines a solution to the problem, forms a complete chromosome. A single chromosome then corresponds to one member of the entire population, which can contain hundreds to thousands of chromosomes. This real valued approach has an advantage over the traditional binary representation of variables because the real valued variables do not have a discrete resolution. This is especially important for problems that are sensitive to small changes in the variables. For the pure GA individual input variables include upper and lower bounds, size of the population, and various other parameters to control the flow and output of the final optimization routine.

The first step in the evolutionary process is to use uniform random numbers to generate the initial population. This is done in a way that ensures a completely random, evenly distributed initial population is used to start the problem. Each individual chromosome is generated using the integer and real valued user supplied upper and lower bounds. The process is then repeated until the entire population has been filled.

From this point each member of the population is assigned a fitness value via a user supplied objective function. The genetic operators are then used to generate a new population from the initial parent population. These genetic operators are crucial to the performance of the genetic algorithms because they enable a more fit population to be evolved from the initial population. This process continues until the genetic algorithm exit condition occurs, at which point the final population is returned. Common stopping conditions include limiting the number of generations, monitoring when the best fit solution stop changing, or monitoring when the average cost function value approaches the population minimum. For this study the optimization algorithm is run out until the best solutions has not changed for at least 25 generations. The sequence of operations of the core operations of the genetic algorithm is illustrated in Fig. A1. One secondary operator, elitism, is also required for the genetic algorithm. This operator simply ensures that the best fit solution(s) are not lost from generation to generations by directly inserting the best members of the parent population into the next generation.

A.1.1. Selection types

In genetic algorithms a selection operator is used to determine which individuals in the parent population pass on their genes to the next population, through the reproduction

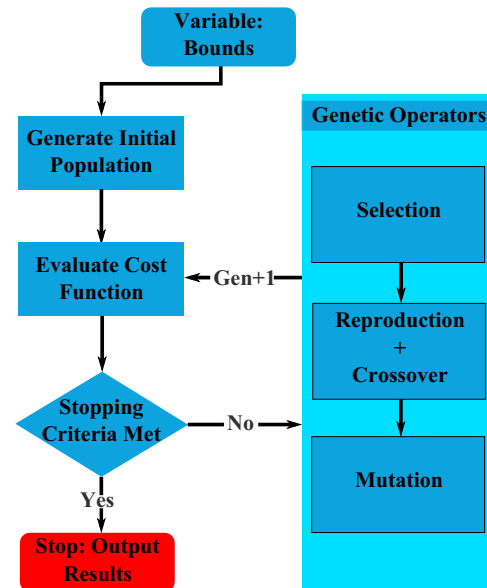


Fig. A1. Flow chart for the simple real and integer valued genetic algorithm.

and crossover operators. This operator is representative of the survival of the fittest evolutionary principle. In this algorithm, a total of two selection types have been implemented, one based solely on a roulette selection method, and another that combines roulette and tournament selection.

Roulette selection is a fitness proportionate selection method in which better fit individuals are more likely to be chosen. In this selection method, a normalized fitness that ranges from 0 to 1 is used. The fitness is reformulated so that more fit members of the population have a higher normalized fitness value and will be more likely to be chosen to become parents.

The first step in this process is to evaluate the objective function for each member of the population. The value from the objective function evaluation is known as raw fitness, $r(i)$. In this case i represents the chromosome/population number. The 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, represented by

$$s(i) = r(i) \quad (\text{A.1})$$

However, if the problem is a maximization problem, the standardized fitness is the individual raw fitness value subtracted from the largest raw fitness value. This maximization problem standardized fitness is represented by

$$s(i) = r_{\max} - r(i) \quad (\text{A.2})$$

Next the standardized fitness is adjusted so that each individual has a value between 0 and 1, with larger adjusted fitness values representing better fit individuals. The advantage of this step is that as the population matures, fitness values are exaggerated in population members that are close to the most fit solution. However, this adjustment is most effective for problems where the optimal solution is

near 0 [29,30]. To ensure the adjusted fitness is between 0 and 1 the adjusted fitness is calculated as follows:

$$a(i) = \frac{1}{1 + s(i)} \quad (\text{A.3})$$

Now that the adjusted fitness values lie between 0 and 1 the next step is to normalize the whole population, so the sum of the normalized fitness values is 1. This is done so that selected individual can be chosen from pseudo random number, which also have a range from 0 to 1. Individual normalized fitness values are determined by dividing the individual adjusted fitness value by the sum of adjusted fitness values, which is expressed as

$$n(i) = \frac{a(i)}{\sum_j^n a(j)} \quad (\text{A.4})$$

The actual selection of individuals is performed as follows. First the normalized fitness values are sorted from largest to smallest. A random number is then used to decide which individual is selected. The selection of an individual is done by a summing the normalized fitness values. The individual that causes the summation value to be greater than the random number is chosen to be a parent and move on to the next set genetic operators. This process is repeated until enough parents have been selected to maintain the same population size as the parent generation. Roulette selection ensures that the best fit individuals have the highest probability of surviving to the next generation.

The second selection method implemented is tournament selection. This method is a greedy over selection method which uses roulette selection in conjunction with standard tournament selection. In this case, greedy over selection methods means the algorithm performs additional measures to try to evolve to an optimal solution more quickly than the standard selection method. Two individuals are chosen, using roulette selection, to compete to determine which individual is allowed to pass on its genetic material. The winner of the competition is done by directly comparing the normalized fitness values. The most fit individual is then selected to become a parent. This process is continued until enough parents are chosen to fill the next population.

This selection method has an advantage because it drives down the raw fitness and average of the fitness function in fewer generations than pure roulette selection. As with all greedy over selection methods, there is a risk that genetic diversity, which could help the population in later generations, will be lost. Although tournament selection will likely result in faster convergence, it should be used with care. For both the MGA and MGA-DSM problems both selection methods work well.

A.1.2. 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 is done with the genetic material. The user must supply the probability that an individual will under go either reproduction or crossover. To maintain a constant population size, these two probabilities must total up to 1.0 (100%). Crossover and reproduction values should typically be chosen close to 0.9 and 0.1,

respectively. Both of these operators require two parent and produce two offspring.

The simplest of these two operators is reproduction. Reproduction occurs when 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 undergo reproduction or crossover is chosen by a random process using the user supplied probabilities. In the implemented genetic algorithm a random number is generated, if the random number is greater than the user given crossover probability the parents undergo reproduction. If not, a crossover operation is performed. Without the reproduction and elitism operators genetic algorithms would be no more useful than random searches.

The crossover operator, which represents biological reproduction, is used to allow individuals to be created with new and unique genetics. The purpose of each crossover type is to promote genetic diversity and expand, in a controlled way, the search of the design space. Unlike the reproduction process, crossovers allow new points in the design space to be searched. The crossover operators produce two offspring that contain genetic material from both parents. A total of five distinct crossover types have been implemented for use with the hybrid GNLP algorithm. The crossover type is selected by the user, making the final hybrid algorithm suitable for many different optimization problems. For typical mission design problems either the double point or uniform crossover operators produce the best results.

Single point crossover: The first crossover operator is single point crossover. In the single point crossover operator one variable is randomly chosen to be the crossover point. The first child is created by copying everything from the first parent prior to the crossover point and everything from the second parent after the crossover point. The second child is the inverse of this process. This process is shown graphically in Fig. A2.

Double point crossover: Double point crossover is similar to the single point crossover operator. In this case two crossover points are randomly chosen. After the two points are chosen two new individuals are formed, as shown in Fig. A3, by swapping the variables between the two crossover points.

Uniform crossover: Uniform crossover has been shown to be a very effective method for promoting genetic diversity,



Fig. A2. Illustration of the single point crossover with the 3rd variable chosen as the crossover point.



Fig. A3. Illustration of the double point crossover with the 3rd and 5th variables chosen as the crossover points.

and, in turn, discovering new useful chromosomes [27,31]. With the uniform crossover operator each variable in the chromosome is a crossover point. The two offspring are then generated by a series of virtual coin flips. The virtual coin flip is used to decide which offspring gets the genetic material from the separate individual parents. While this process promotes genetic diversity, it is best used to determine optimal solutions near an approximate solution that has been previously determined. This can be done by using this operator late in the search, when genetic diversity have begun to stagnate or by running a smaller search localized around a sub-optimal point. Graphically this process is similar to both the single and double point crossover methods.

Arithmetic: The arithmetic crossover operator, sometimes referred to as the whole arithmetic operator, linearly combines the chromosomes from the two parents. The two offspring are generated from is by linearly combining the two parents with a randomly chosen multiplier, α , which has the range of 0 and 1. It is possible to use other bounds for α , as long as the upper and lower bounds are checked for each resulting gene. The two offspring are determined from the random multiplier as follows:

$$C_1 = \alpha P_1 + (1 - \alpha) P_2 \quad (\text{A.5})$$

$$C_2 = (1 - \alpha) P_1 + \alpha P_2 \quad (\text{A.6})$$

A simple illustration of the arithmetic operator can be seen in Fig. A4. This illustration shows a chromosome consisting of only real valued variables, but the method can be easily extended to include integer value variables by rounding to the nearest integer.

Heuristic: The heuristic crossover operator determines a search direction from the two parent members. This is similar to the arithmetic operator, but modifies the search direction from by moving the worst solution towards the better parent. The two offspring are created using a randomly chosen variable, r , which has a range from 0 to 1, as outlined below:

$$C_1 = P_{\text{best}} + r(P_{\text{best}} - P_{\text{worst}}) \quad (\text{A.7})$$

$$C_2 = P_{\text{best}} \quad (\text{A.8})$$

A.1.3. Mutation types

The last genetic operator, which the newly generated population must pass through, is the mutation operator. The two user inputs are the probability that a mutation will occur and the desired type of mutation. The probability that a mutation will 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 does not wish to make

use of the mutation operator, a probability of 0 can be entered.

Allowing the genetic algorithm to utilize a mutation operator has several advantages. Mutations help maintain genetic diversity in a population as it ages. Often times, after many generations the population tends to lose genetic diversity and stagnate near local minimums. The mutation operator helps to prevent this from happening by randomly mutating genes in the chromosomes. By introducing (or in some cases reintroducing) changes in a chromosome or individual gene it is possible for better more fit individuals to appear. In this algorithm the mutation operator changes the value of one randomly selected gene (i.e. variable) in the chromosome, which in some situations can greatly improve the individuals fitness.

As with the crossover operator several separate types of mutations have been implemented. In practice, certain mutation types may work best for each type of problem, so the user should experiment with different crossover and mutations types to see which combinations work best. A total of 3 mutation types have been implemented in the hybrid algorithm, which can operate on both integer and real valued genes.

Uniform mutation: The uniform mutation operator replaces the value of a randomly chosen gene with a uniform random value between the variable's upper and lower bounds. The advantage of this operator is that it allows the population's genetic diversity to be maintained by inserting a random gene that other mutation methods may not be able to obtain. Similar non-uniform mutation, based on other distributions, can be used as well.

Boundary mutation: With boundary mutations, a single, randomly selected, variable is changed to either the variables upper or lower bounds. The choice to use either the upper or lower bound is done with a virtual coin flip. This method best promotes genetic diversity when used in combination with either the arithmetic or heuristic crossover operators.

Sliding: The sliding mutation operator is similar in performance to uniform mutations. A randomly selected gene is multiplied by a randomly selected value between 0.8 and 1.2. In essence, this operator allows the gene to be slightly mutated by $\pm 20\%$. The operator then checks to ensure that the variable bounds have not been violated. If a violation occurs the variable is set to the closest bound.

A.2. Non-linear programming solver

Evolutionary algorithms, particularly genetic algorithms, are well suited for global optimization. However, when genetic algorithms are used to optimize impulse multiple gravity-assist and low thrust problems they often only find solutions near the global optimum. Alternatively, non-linear programming (NLP) solvers typically only converge to locally optimal solutions. By modifying the genetic algorithm to utilize an NLP solver to determines locally optimal solutions a robust global optimization algorithm can be developed. This hybrid algorithm, known as the GNLP optimization algorithm, is able determine near globally optimal solutions by combining the global convergence of the genetic algorithm with accuracy of the nonlinear programming algorithm. The

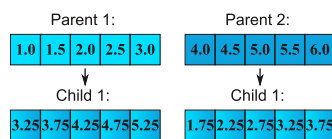


Fig. A4. Illustration of the arithmetic crossover operator with $\alpha = 0.25$.

proposed algorithm is able to efficiently solve complex problems by significantly reducing the population size and number of generations required to converge on near-globally optimal solutions.

The GNL algorithm should only be used to optimize functions that are continuous and at least twice differentiable, at least in the neighborhood of the proposed solution. The NLP solver does not iterate on integer variables, because this often introduces large discontinuities. The genetic algorithm is used exclusively to optimize integer variables. These properties make optimizing both high and low thrust mission design problems good candidates for proposed GNL optimization algorithm.

A.2.1. UNCMIN based non-linear programming solver

The NLP solver implemented in the hybrid algorithm is based on the UNCMIN optimization algorithm, originally written in FORTRAN 77, and introduced in [32,33]. This algorithm is a quasi-newton algorithm based on steepest descent methods, which requires only the objective function values. However, to improve convergence and run times this algorithm does allow user supplied gradient and Hessians. The NLP algorithm implemented in the hybrid algorithm is an updated version of the original UNCMIN algorithm written in Fortran 90. Additional information on this algorithm can be found in [32,33].

As implemented, the UNCMIN algorithm is an extremely modular system of algorithms capable of multiple step selection strategies and multiple derivative calculation techniques (for both 1st and 2nd order derivatives). First order derivatives can be calculated analytically, or by forward or central finite difference. The second derivative Hessian matrix can be calculated analytically, with a Broyden–Fletcher–Goldfarb–Shanno (BFGS) approximation [34–37], or by finite difference. The BFGS method approximates the Hessian from gradient information and is often used in quasi-Newton methods. The BFGS method gives a sufficiently accurate approximation, for all the problems in this paper, for the Hessian matrix, which is computationally expensive to evaluate numerically. Utilizing this approximation significantly increases the efficiency of the final algorithm. Having all of these options allow for a very robust algorithm that can be adjusted by the user to work for a large variety of problems. For the MGA and MGA-DSM problems gradients are calculated with the forward finite difference method, while the Hessian matrices are calculated with the BFGS approximation. Calculating the Hessian via finite difference increased the solution accuracy, but also severely increased the computational requirements of the NLP solver. For the preliminary trajectory design problems, using the BFGS approximation does not have a noticeable effect the convergence of the hybrid algorithm when compared to the more accurate finite difference Hessians. Near locally minimum solution, the Hessian approximation gives sufficiently accurate results, while minimizing the computations the NLP solver must perform.

A.2.2. CONMIN based non-linear programming solver

In addition to the UNCMIN unconstrained optimization algorithm the classical CONMIN constrained optimization has also been implemented. This is necessary for problems

such as low-thrust mission optimization problems, which have match-point constraints that must be met in order to determine feasible trajectories. Like the UNCMIN algorithm, this algorithm was originally written in FORTRAN 77, but has since been updated to modern Fortran. This algorithm can determine locally optimal solutions to problems that have linear and non-linear constraints.

The basic method of optimization utilized in this algorithm is the method of feasible directions. Like the UNCMIN algorithm, only a user supplied objective function, which calculates the objective function and problem constraints is required. The algorithm can calculate derivatives analytically with user gradient functions or numerically via finite difference methods. This algorithm should only be used with problems that have a relatively low dimensionality. Further information on the CONMIN optimization algorithm can be found in [38,39].

A.3. Hybrid algorithm implementation

The final hybrid GNL algorithm is able to achieve significant performance increases, in terms of algorithm efficiency, over the standard GA implementation. This is possible because each individual population member is optimized with the NLP solver, rather than simply evaluating the objective function and relying on the GA for the entire optimization process. Through this process the GNL algorithm is able to combine the global convergence properties of the genetic algorithm with the local solution accuracy of the non-linear programming solver. A flow chart of the hybrid GNL algorithm is shown in Fig. A5. This algorithm flow is similar to the genetic algorithm, except the objective function evaluation step has been replaced by the NLP optimization step. In this step a driver for the NLP solver is called to set the NLP user inputs for the particular problem. The NLP solver then determines a locally optimal solution and outputs the results to the genetic algorithm. After this the genetic operators create a new population and the process is continued until the algorithm exits and outputs the final results. Which NLP algorithm is used is one of the user input of the final algorithm. If the problem does not have strict constraints that are commonly violated the UNCMIN algorithm should be used.

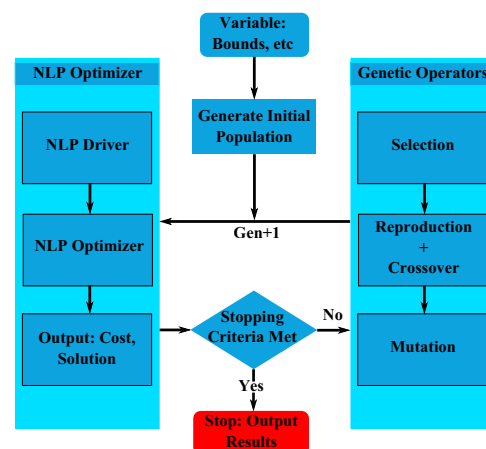


Fig. A5. Flow chart for hybrid GNL algorithm.

References

- [1] D. Morrison, The Spaceguard Survey: Report of the NASA International Near-Earth-Object Detection Workshop, NASA STI/Recon Technical Report No. 92 (1992) 34245.
- [2] S. Hernandez, B.W. Barbee, S. Bhaskaran, K. Getzandanner, Mission opportunities for the flight validation of the kinetic impactor concept for asteroid deflection, *Acta Astronaut.* 103 (2014) 309–321.
- [3] B. Wie, Dynamics and control of gravity tractor spacecraft for asteroid deflection, *J. Guid. Control Dyn.* 31 (2008) 1413–1423.
- [4] B.W. Barbee, W.T. Fowler, Spacecraft mission design for the optimal impulsive deflection of hazardous Near-Earth Objects (NEOs) using nuclear explosive technology, in: Proceedings of the 2007 Planetary Defense Conference, Washington, DC.
- [5] B. Kaplinger, B. Wie, D. Dearborn, Nuclear fragmentation/dispersion modeling and simulation of hazardous near-earth objects, *Acta Astronaut.* 90 (2013) 156–164.
- [6] A. Pitz, B. Kaplinger, B. Wie, Preliminary design of a hypervelocity nuclear interceptor spacecraft for optimal disruption/fragmentation of NEOs, in: 22nd AAS/AIAA Space Flight Mechanics Meeting, 2012.
- [7] G. Vardaxis, A. Pitz, B. Wie, Conceptual design of planetary defense technology demonstration mission, in: 22nd AAS/AIAA Space Flight Mechanics Meeting, 2012.
- [8] T. Winkler, S. Wagner, B. Wie, Optimal target selection for a Planetary Defense Technology (PDT) demonstration mission, in: 22nd AAS/AIAA Space Flight Mechanics Meeting, 2012.
- [9] A. Pitz, B. Kaplinger, G. Vardaxis, T. Winkler, B. Wie, Conceptual design of a hypervelocity asteroid intercept vehicle (HAIV) and its flight validation mission, *Acta Astronaut.* 94 (2014) 42–56.
- [10] I. Carnelli, A. Gálvez, D. Izzo, Don Quijote: A NEO deflection precursor mission, in: 2006 NASA Near-Earth Object Detection and Threat Mitigation Workshop, 2006.
- [11] J. Cano, M. Sánchez, I. Carnelli, Mission analysis for the Don Quijote phase-A study, in: Proceedings of the 20th International Symposium on Space Flight Dynamics, 2007.
- [12] S. Wagner, T. Winkler, B. Wie, Analysis and selection of optimal targets for a planetary defense technology demonstration mission, in: AIAA/AAS Guidance Navigation and Control Conference, 2012.
- [13] S. Wagner, B. Kaplinger, B. Wie, GPU accelerated genetic algorithm for multiple gravity-assist and impulsive ΔV maneuvers, in: AIAA/AAS Guidance Navigation and Control Conference, 2012.
- [14] S. Wagner, B. Wie, Low-thrust trajectory optimization for asteroid exploration, redirect, and deflection mission, in: 24th AAS/AIAA Spaceflight Mechanics Meeting, 2014.
- [15] M. Minovitch, The invention that opened the solar system to exploration, *Planet. Space Sc.* 58 (2010) 885–892.
- [16] K. Qadir, Multi-gravity assist design tool for interplanetary trajectory optimisation, Master's Thesis, Lulea University of Technology, 2009.
- [17] D. Vallado, *Fundamentals of Astrodynamics and Applications*, 2nd edition, Microcosm Press, El Segundo, CA, 2004.
- [18] R. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*, Revised edition, AIAA Educational Series, Reston, VA, 1999.
- [19] E.R. Lancaster, R.C. Blanchard, A Unified Form of Lambert's Theorem, NASA Technical Note, TN D-5368, 1969.
- [20] F.T. Sun, On the minimum time trajectory and multiple solutions of Lambert's problem, in: AAS/AIAA Astrodynamics Specialist Conference, 1979.
- [21] R.H. Gooding, On the Solution of Lambert's Orbital Boundary-Value Problem, Royal Aerospace Establishment, 1988.
- [22] J. Englander, B. Conway, T. Williams, Optimal autonomous mission planning via evolutionary algorithms, in: 21th AAS/AIAA Space Flight Mechanics Meeting, 2011.
- [23] M. Kim, Continuous low-thrust trajectory optimization: techniques and applications (Ph.D. thesis), Virginia Polytechnic Institute and State University, 2005.
- [24] C.H. Yam, D. Di Lorenzo, D. Izzo, Constrained global optimization of low-thrust interplanetary trajectories, in: 2010 IEEE Congress on Evolutionary Computation (CEC), 2010, pp. 1–7.
- [25] T. McConaghy, J. Longuski, Parameterization effects on convergence when optimizing a low-thrust trajectory with gravity assists, in: AIAA/AAS Astrodynamics Specialist Conference and Exhibit, 2004.
- [26] G. Lantoine, A methodology for robust optimization of low-thrust tin multi-body environments (Ph.D. thesis), Georgia Institute of Technology, 2010.
- [27] M. Vavrina, A hybrid genetic algorithm approach to global low-thrust trajectory optimization (Ph.D. thesis), Purdue University, 2010.
- [28] B. Conway, *Spacecraft Trajectory Optimization*, 1st edition, Cambridge University Press, New York, NY, 2010.
- [29] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st edition, Addison-Wesley, 1989.
- [30] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, 1st edition, The MIT Press, Cambridge, MA, 1992.
- [31] G. Syswerda, Uniform crossover in genetic algorithms, in: Proceedings of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989, pp. 2–9.
- [32] R. Schnabel, J. Koontz, B. Weiss, *A Modular System of Algorithms for Unconstrained Minimization*, University of Colorado at Boulder: Department of Computer Science, 1985.
- [33] D. Kahaner, C. Moler, S. Nash, *Numerical Methods and Software (Prentice Hall Series in Computational Mathematics)*, 2nd edition, Prentice Hall, Englewood Cliffs, NJ, 1989.
- [34] C.G. Broyden, The convergence of a class of double rank minimization algorithms: 2. The new algorithm, *J. Inst. Math. Appl.* 6 (1970) 76–231.
- [35] R. Fletcher, A new approach to variable metric algorithms, *Comput. J.* 13 (1970) 317–322.
- [36] D. Goldfarb, A family of variable metric methods derived by variational means, *Math. Comput.* 24 (1970) 23–26.
- [37] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, *Math. Comput.* 24 (1970) 647–650.
- [38] G.N. Vanderplaats, Conmin User's Manual, Technical Report X-62282, NASA Technical Memorandum, 1978.
- [39] G.N. Vanderplaats, F. Moses, Structural Optimization by Methods of Feasible Directions, National Symposium on Computerized Structural Analysis and Design, 1972.