Optimal Triangular Lagrange Point Insertion Using Lunar Gravity Assist

Julio César Benavides
Aerospace Engineering
The Pennsylvania State University
306 George Building
(814) 865-7396
jcb339@psu.edu

Elizabeth Davis
Aerospace Engineering
The Pennsylvania State University
306 George Building
(814) 865-7396
ecd134@psu.edu

Brian Wadsley
Aerospace Engineering
The Pennsylvania State University
306 George Building
(814) 865-7396
bjw196@psu.edu

ABSTRACT

In this paper, we design optimal trajectories that transfer satellites from Earth orbit to the Sun-Earth triangular Lagrange points (L_4 and L_5) using a lunar gravity assist during the year 2010. An optimal trajectory is defined as one that minimizes total mission velocity change (Δv), which is directly proportional to fuel consumption and operational cost. These optimal trajectories are determined using two heuristic algorithms: Differential Evolution and Covariance Matrix Adaptation. The three parameters being varied are the mission commencement time, Hohmann transfer target radius above the center of the moon, and the midcourse correction maneuver time of flight.

Optimal trajectory parameters predicted by Differential Evolution and Covariance Matrix Adaptation agree closely. The results show that an optimal trajectory to L_4 using this problem formulation requires a total mission Δv of 6.15 km/s. The mission commencement time was 23.1 days, the Hohmann transfer target radius for this trajectory is 59993 km, and the midcourse correction maneuver time of flight is 294.9 days. An optimal trajectory to L_5 requires a total mission Δv of 5.056 km/s. The mission commencement time is 363.7 days, the Hohmann transfer target radius for this trajectory is 59993 km, and the midcourse correction maneuver time of flight is 410.9 days.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization–*Constrained Optimization*; J.2 [**Physical Sciences and Engineering**]: Aerospace

General Terms

Design, Algorithm, Experimentation, Theory

Keywords

Triangular Lagrange Point, Hohmann Transfer, Gravity Assist, Lambert's Problem

1. INTRODUCTION

This section defines triangular Lagrange point geometry, discusses previous work involving trajectory optimization to these points, and describes the characteristics of Differential Evolution (DE) and Covariance Matrix Adaptation (CMA).

1.1 Lagrange Point Geometry

Given two massive bodies in circular orbits around their common center of mass, there are five positions in space where a third body of comparatively negligible mass will maintain its position relative to the two primary bodies. These five points, known as Lagrange points, are the stationary solutions of the circular, restricted three-body problem [1]. Figure 1.1 shows the location of the five Lagrange points relative to the positions of two primary masses in the rotating coordinate frame.

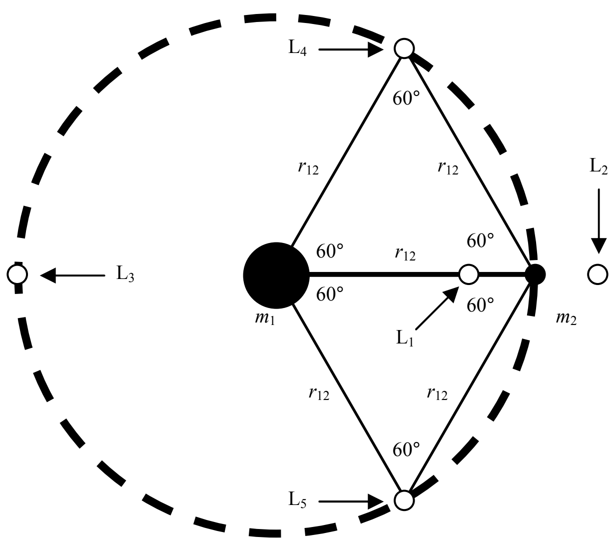

Figure 1.1: Lagrange Point Geometry

As seen in a frame of reference which rotates with the same period as the two co-orbiting bodies, the centrifugal force and gravitational fields of the two principal bodies are in balance at the Lagrange point, allowing a third object to be stationary with respect to the primary bodies. The first three Lagrange points (L_1, L_2, and L_3) are referred to as the collinear Lagrange points and the fourth and fifth points (L_4 and L_5) are referred to as the triangular Lagrange points.

1.2 Previous Work

Sun-Earth triangular Lagrange point insertion using a lunar gravity assist has never been attempted. In fact, no trajectory optimization to these specific Lagrange points had been published in astrodynamics conference proceedings until 2007 [2]. The majority of literature related to this topic involves insertion into the Sun-Earth co-linear Lagrange points [3, 4, and 5] or the Lagrange points of the Earth-Moon system [6, 7].

Benavides and Spencer [2] use co-orbital rendezvous (co-orbital phasing) orbits to transfer satellites initially in a circular Earth orbit of radius 6678 km (300 km altitude) to L_4 and L_5. Their results conclude that an optimal trajectory that minimizes total mission Δv can be achieved if the phasing orbit is initiated at an Earth true anomaly of 350.6° for the case of L_4 and 4.4° for the case of L_5. The optimal Δv's calculated for each trajectory, respectively, are 5.25 km/s and 4.65 km/s.

The goal of this research is to design optimal trajectories to L_4 and L_5 using a lunar gravity assist that will require smaller Δv's than those predicted by co-orbital rendezvous. These optimal trajectories are determined using two heuristic algorithms: differential evolution (DE) and covariance matrix adaptation (CMA).

1.3 Differential Evolution

Differential Evolution is a real coded algorithm developed for solving problems in continuous spaces by R. Storn & K. Price [13]. DE uses selection, crossover, and mutation operators to form new solutions from a randomly generated initial seed population. The mutation operator takes a target vector and uses it to generate a new vector by taking the weighted difference of two other randomly selected vectors and adding it to a third. The selection of the target vector is specified by the strategy used. Strategies for DE are displayed in the form of DE/rand/2/bin. The first factor, DE, is the algorithm name. The second factor indicates whether the mutation target vector is either randomly selected (rand) or the vector that produces the current best objective value (best). If the best members are used, then the strategy is called elitist. The third factor is the number of vectors to be mutated, and the last controls the crossover method, either binomial or exponential. Crossover serves to increase population diversity by taking the mutated vector and randomly replacing the value of one of the variables with the value from the target vector. Then selection operator compares the value of the objective function obtained the trial vector with the vector corresponding to the minimum objective value from the previous generation. The vector which has the lowest value of the objective function then survives into the next generation.

In a similar problem optimizing the Δv required for an interplanetary trajectory to Jupiter using gravity assists from Venus and Earth, Bessette found that DE preformed better than another evolutionary algorithm known as particle swarm optimization[14]. Both heuristics converged to the same solution, but DE required a fewer number of function evaluations.

1.4 Covariance Matrix Adaptation

The CMA-ES (Covariance Matrix Adaptation Evolution Strategy) algorithm was chosen for its ability to solve non-linear non-convex optimizations problems with a continuous domain [16]. The algorithm also works well on bounded constraint optimization problems with dimensions between three and one hundred [16]. The search space of the problem being solved is expected to be non-linear and non-convex which falls into the strengths of the CMA-ES algorithm. The domain is defined to be continuous and bounded by a set of constraints. Documentation on the CMA-ES algorithm leaves little reason to doubt the CMA-ES algorithm ability to solve this problem based on the problem domain, dimension, and boundary conditions.

2. PROBLEM FORMULATION

The trajectories being designed are divided into three phases: Hohmann transfer phase, lunar gravity assist phase, and mid-course correction phase. The satellites being transferred to L_4 and L_5 are assumed to initially be in a circular Earth orbit of radius 6678 km (300 km altitude).

2.1 Hohmann Transfer Phase

Given a mission commencement time t since the beginning of the year 2010, initial position and velocity vectors for the Moon with respect to the Earth in Earth-centered inertial coordinates and the Earth with respect to the Sun in Sun-centered inertial coordinates are determined [8].

$$t \to \begin{cases} \vec{r}_{m/e}, \vec{v}_{m/e} \\ \vec{r}_e, \vec{v}_e \end{cases} \tag{2.1}$$

The satellite is placed on a trans-lunar trajectory using a Hohmann transfer orbit. The Hohmann transfer semimajor axis is given by,

$$a_H = \frac{\left(r_c + \left|\vec{r}_{m/e}\right| + r\right)}{2} \tag{2.2}$$

where r_c is the initial satellite circular parking orbit radius and r is the transfer target radius above the center of the moon. The transfer orbit eccentricity is given by,

$$e_H = 1 - \frac{r_c}{a_H} \tag{2.3}$$

and the semilatus rectum is calculated by,

$$p_H = a_H\left(1 - e_H^2\right) \tag{2.4}$$

The initial impulsive velocity change needed to place the satellite on a Hohmann transfer to the moon is determined by,

$$\Delta v_1 = \left|\sqrt{\frac{(2a_H - r_c)\mu_e}{a_H r_c}} - \sqrt{\frac{\mu_e}{r_c}}\right| \tag{2.5}$$

where μ_e is the gravitational parameter of the Earth. Satellite position and velocity vectors with respect to the Earth at the time when the Hohmann transfer is commenced are determined using the lunar position and velocity vectors.

$$\vec{r}_{o/e} = \frac{-r_c \vec{r}_{m/e}}{\left|\vec{r}_{m/e}\right|} \tag{2.6}$$

$$\vec{v}_{o/e} = \sqrt{\frac{(2a_H - r_c)\mu_e}{a_H r_c}} \frac{\vec{r}_{m/e} \times \left(\vec{r}_{m/e} \times \vec{v}_{m/e}\right)}{\left|\vec{r}_{m/e} \times \left(\vec{r}_{m/e} \times \vec{v}_{m/e}\right)\right|} \tag{2.7}$$

The transfer orbit true anomaly where the satellite intersects the lunar sphere of influence is calculated by

$$f_H = \cos^{-1}\left(\frac{-\xi_2 - \sqrt{\xi_2^2 - 4\xi_1\xi_3}}{2\xi_1}\right) \quad (2.8)$$

where the terms ζ_1, ζ_2, ζ_3 are defined as

$$\xi_1 = e_H^2\Psi^2 - e_H^2\left|\vec{r}_{m/e}\right|^2 - 2e_H p_H\left|\vec{r}_{m/e}\right| \quad (2.9)$$

$$\xi_2 = 2e_H\Psi^2 - 2e_H\left|\vec{r}_{m/e}\right|^2 - 2p_H\left|\vec{r}_{m/e}\right| \quad (2.10)$$

$$\xi_3 = \Psi^2 - \left|\vec{r}_{m/e}\right|^2 - p_H^2 \quad (2.11)$$

and Ψ is the radius of the lunar sphere of influence (assumed to be 66200 km). Satellite position and velocity vectors with respect to the Earth at the point where the satellite intersects the lunar sphere of influence are given by,

$$\vec{r}_{1/e} = F_H\vec{r}_{o/e} + G_H\vec{v}_{o/e} \quad (2.12)$$

$$\vec{v}_{1/e} = \dot{F}_H\vec{r}_{o/e} + \dot{G}_H\vec{v}_{o/e} \quad (2.13)$$

where F_H and G_H are defined as the Hohmann transfer Lagrange coefficients,

$$F_H = 1 - \frac{\mu_e\Psi[1-\cos(f_H)]}{\left|\vec{r}_{o/e}\times\vec{v}_{o/e}\right|^2} \quad (2.14)$$

$$G_H = \frac{\Psi\left|\vec{r}_{o/e}\right|\sin(f_H)}{\left|\vec{r}_{o/e}\times\vec{v}_{o/e}\right|} \quad (2.15)$$

$$\dot{F}_H = \frac{\mu_e[1-\cos(f_H)]}{\sin(f_H)\left|\vec{r}_{o/e}\times\vec{v}_{o/e}\right|}\left\{\frac{\mu_e[1-\cos(f_H)]}{\left|\vec{r}_{o/e}\times\vec{v}_{o/e}\right|^2} - \frac{1}{\left|\vec{r}_{o/e}\right|} - \frac{1}{\Psi}\right\} \quad (2.16)$$

$$\dot{G}_H = 1 - \frac{\mu_e\left|\vec{r}_{o/e}\right|[1-\cos(f_H)]}{\left|\vec{r}_{o/e}\times\vec{v}_{o/e}\right|^2} \quad (2.17)$$

The time between mission commencement and lunar sphere of influence intersection is calculated by

$$\Delta t_H = \frac{E_H - e_H\sin(E_H)}{n_H} \quad (2.18)$$

where E_H is the satellite's eccentric anomaly at the time of lunar sphere of influence intersection

$$E_H = 2\tan^{-1}\left[\sqrt{\frac{1-e_H}{1+e_H}}\tan\left(\frac{f_H}{2}\right)\right] \quad (2.19)$$

and n_H is the Hohmann transfer mean motion,

$$n_H = \sqrt{\frac{\mu_e}{a_H^3}} \quad (2.20)$$

2.2 Lunar Gravity Assist Phase

The lunar gravity assist phase lasts from the time the satellite enters the lunar sphere of influence (a hyperbolic trajectory) to the time it leaves the sphere. As the satellite enters the lunar sphere of influence, position and velocity vectors with respect to the moon are determined by,

$$\vec{r}_{1/m} = \vec{r}_{1/e} - \vec{r}_{m/e} \quad (2.21)$$

$$\vec{v}_{1/m} = \vec{v}_{1/e} - \vec{v}_{m/e} \quad (2.22)$$

The hyperbolic trajectory specific mechanical energy is given by,

$$\varepsilon_G = \frac{\left|\vec{v}_{1/m}\right|^2}{2} - \frac{\mu_m}{\left|\vec{r}_{1/m}\right|} \quad (2.23)$$

where μ_m is the lunar gravitational parameter. The trajectory's specific angular momentum, semilatus rectum, and eccentricity are calculated by,

$$h_G = \left|\vec{r}_{1/m}\times\vec{v}_{1,m}\right| \quad (2.24)$$

$$p_G = \frac{h_G^2}{\mu_m} \quad (2.25)$$

$$e_G = \sqrt{1 + \frac{2\varepsilon_G h_G^2}{\mu_m^2}} \quad (2.26)$$

The hyperbolic trajectory's true anomaly where the satellite enters the lunar sphere of influence is given by,

$$f_G = \cos^{-1}\left(\frac{p_G - \left|\vec{r}_{1/m}\right|}{e_G\left|\vec{r}_{1/m}\right|}\right) \quad (2.27)$$

Satellite position and velocity vectors with respect to the moon as it leaves the lunar sphere of influence are determined by,

$$\vec{r}_{2/m} = F_G\vec{r}_{1,m} + G_G\vec{v}_{1,m} \quad (2.28)$$

$$\vec{v}_{2/m} = \dot{F}_G\vec{r}_{1/m} + \dot{G}_G\vec{v}_{1/m} \quad (2.29)$$

where F_G and G_G are defined as the hyperbolic trajectory's Lagrange coefficients,

$$F_G = 1 - \frac{\mu_m\Psi[1-\cos(2f_G)]}{h_G^2} \quad (2.30)$$

$$G_G = \frac{\Psi\left|\vec{r}_{1/m}\right|\sin(2f_G)}{h_G} \quad (2.31)$$

$$\dot{F}_G = \frac{\mu_m[1-\cos(2f_G)]}{h_G\sin(2f_G)}\left\{\frac{\mu_m[1-\cos(2f_G)]}{h_G^2} - \frac{1}{\left|\vec{r}_{1/m}\right|} - \frac{1}{\Psi}\right\} \quad (2.32)$$

$$\dot{G}_G = 1 - \frac{\mu_m\left|\vec{r}_{1/m}\right|[1-\cos(2f_G)]}{h_G^2} \quad (2.33)$$

Satellite position and velocity vectors with respect to the Earth as it leaves the lunar sphere of influence are given by,

$$\vec{r}_{2/e} = \vec{r}_{2/m} + \vec{r}_{m/e} \quad (2.34)$$

$$\vec{v}_{2/e} = \vec{v}_{2/m} + \vec{v}_{m/e} \quad (2.35)$$

The time between lunar sphere of influence entrance and departure is determined by,

$$\Delta t_G = \frac{2h_G^3\left[e_G \sinh(H_G) - H_G\right]}{\mu_m^2\left(e_G^2 - 1\right)^{3/2}} \tag{2.36}$$

where H_G is the hyperbolic trajectory eccentric anomaly as the satellite leaves the lunar sphere of influence,

$$H_G = \sinh^{-1}\left[\frac{\sqrt{e_G^2 - 1}\sin(f_G)}{1 + e_G\cos(f_G)}\right] \tag{2.37}$$

2.3 Midcourse Correction Phase

The midcourse correction phase lasts from the moment the satellite leaves the lunar sphere of influence to the time when it rendezvous with the L_4 or L_5 Lagrange point. As the satellite leaves the lunar sphere of influence, its position and velocity vectors with respect to the Earth are transformed to Sun-centered inertial coordinates,

$$\vec{r}_2 = T\vec{r}_{2/e} + \vec{r}_e \tag{2.38}$$

$$\vec{v}_2 = T\vec{v}_{2/e} + \vec{v}_e \tag{2.39}$$

where the transformation matrix T is given by,

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\left(\frac{23\pi}{180}\right) & \sin\left(\frac{23\pi}{180}\right) \\ 0 & -\sin\left(\frac{23\pi}{180}\right) & \cos\left(\frac{23\pi}{180}\right) \end{bmatrix} \tag{2.40}$$

Given a correction time of flight for the triangular Lagrange point destination, the total mission time of flight is computed by,

$$\Delta t = \Delta t_H + \Delta t_G + \Delta t_C \tag{2.41}$$

Final triangular Lagrange point position and velocity vectors with respect to the Sun are determined by using the Earth's classical orbital elements and solving Kepler's equation [9],

$$M_o = E_o - e\sin(E_o) + \Delta t\sqrt{\frac{\mu_s}{a^3}} \tag{2.42}$$

$$E_f = e\sin(E_f) + M_o \tag{2.43}$$

$$f = 2\tan^{-1}\left[\sqrt{\frac{1+e}{1-e}}\tan\left(\frac{E_f}{2}\right)\right] \tag{2.44}$$

$$\begin{aligned} L_4 &: \left\{a \quad e \quad i \quad \Omega \quad \omega \quad f + \frac{\pi}{3}\right\} \to \vec{r}_f; \vec{v}_f \\ L_5 &: \left\{a \quad e \quad i \quad \Omega \quad \omega \quad f - \frac{\pi}{3}\right\} \to \vec{r}_f; \vec{v}_f \end{aligned} \tag{2.45}$$

where M_o and E_o are the Earth's mean anomaly and eccentric anomaly at mission commencement, respectively; μ_s is the solar gravitational parameter; E_f is the Earth's eccentric anomaly when the satellite rendezvous with the triangular Lagrange point; and a, e, i, Ω, ω, and f are the Earth's classical orbital elements when the satellite intercepts the triangular Lagrange point. The midcourse correction transfer's initial and final velocity vectors are determined by solving Lambert's boundary value problem [10, 11, and 12],

$$L\{\vec{r}_2; \vec{r}_f; \Delta t; \mu_s\} \to \vec{v}_3; \vec{v}_4 \tag{2.46}$$

The two impulsive maneuvers required by the midcourse correction maneuver are given by,

$$\Delta v_2 = \left|\vec{v}_3 - \vec{v}_2\right| \tag{2.47}$$

$$\Delta v_3 = \left|\vec{v}_f - \vec{v}_4\right| \tag{2.48}$$

Total mission velocity change is determined by,

$$\Delta v = \Delta v_1 + \Delta v_2 + \Delta v_3 \tag{2.49}$$

2.4 Optimization Problem

The objective of this research is to find an optimal trajectory required to reach L_4 and L_5. An optimal trajectory is defined as one that minimizes the total mission velocity change (Δv),

$$\min(\Delta v) = \min(\Delta v_1 + \Delta v_2 + \Delta v_3) \tag{2.50}$$

given the following constraints

$$0 \text{ days} \le t \le 365 \text{ days} \tag{2.51}$$

$$2000 \text{ km} \le r \le 60000 \text{ km} \tag{2.52}$$

$$\begin{aligned} L_4 &: 1 \text{ day} \le \Delta t_C \le 365 \text{ days} \\ L_5 &: 1 \text{ day} \le \Delta t_C \le 450 \text{ days} \end{aligned} \tag{2.53}$$

3. DIFFERENTIAL EVOLUTION

The DE source code used was the Matlab version freely available by Neumaier and Storn [13]. All simulations using this algorithm were run on a Pentium 4 3.06 GHz processor. The average time of a function evaluation was 0.1 seconds. The source code was modified to save the best value of each variable corresponding to the best Δv for each generation for later analysis. Constraints were handled in the form of static penalty functions by adding a constant penalty to Δv depending on how far out of the specified bounds the value was.

Since the value of the global optimum was not known, the value to reach of the algorithm was set to zero. Ten initial tests were run out to 900 generations for L4 to get a rough estimate what the number of function evaluations need might be and to get an sense of what minimum Δv might be.

3.1 Parameter Testing

To make the most effective use of the DE algorithm, the input parameters were tested to find the values of that generated the best performance. The DE algorithm has 4 user specified input parameters that effect its performance: strategy, population size

(*NP*), crossover ratio (*CR*), and step size (*F*). For each parameter, ten trials were run out to 250 generations. The value to reach was set at 6.1513km/s, the lowest Δv from the initial trials(Δv_{inital}) . The reference values of the parameters used were a population size of 30, a crossover ratio of 0.8, a step size of 0.8 and the non-elitist crossover strategy DE/rand/2/exp which allowed two random population members survive into the next generation. These values were taken were from suggestions of initial parameter guesses in source code documentation[13].

The final selection of parameter values to be used in the actual trials was based on balancing the greediness and the search of the DE algorithm. The main criterion used were the number of function evaluations required for convergence to Δv_{inital} since time was limited, although percentage of the trials that converged was also taken into consideration.

3.1.1 Strategies

The DE source code had ten possible strategies. From the tests, the best crossover strategy was DE/rand-to-best/1/bin. It was the only strategy to converge to Δv_{inital} all trials and in the fewest amount of function evaluations. This strategy employs a binomial crossover. Initially, it is an non elitist, but after the first generation, it becomes elitist, adding the single best member from the preceding generation to survive into the subsequent one. This result was surprising because it was beat out more elitist strategies.

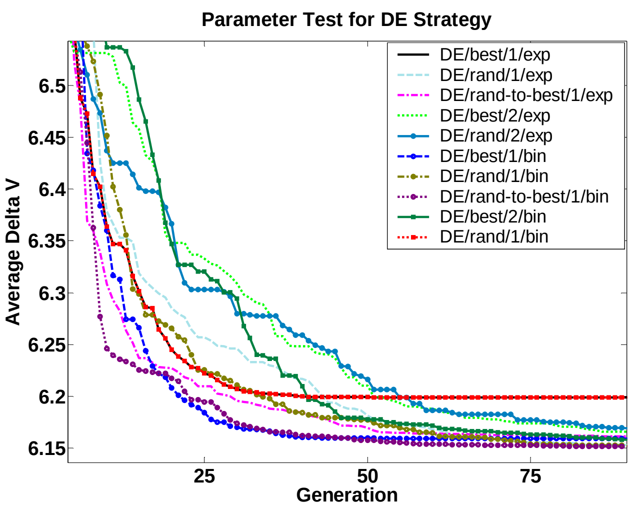

Figure 3.1: Comparison of average values of Δv for each crossover strategy.

3.1.2 Population size

Storm and Price suggest a population size between two and ten times the number of variables [12], so in the tests, NP for was varied from 5 to 120 in steps of 5. Between 6 and 42, *NP* was incremented by three. The results indicate that DE does seem to be sensitive to population size. As long as it was over 25, the algorithm converged every time to Δv_{inital}. Larger population sizes lead to convergence in a smaller but of generations, but took longer since the total number of function evaluations increases with population size. The setting for *NP* used in the actual trials was set at 45. At this setting, the algorithm converged in only twenty more generations than with a *NP* of 120 and required less than half the amount of function evaluations.

3.1.3 Step Size

The step size(*F*) parameter controls the weighting using in the mutation operator. According to the source code documentation says that DE is sensitive to step size [12]. For the parameter tests, *F* was varied from 0 to 0.5 and 1 and 2 in steps of 0.1. Between 0.5 and 1, *F* increased by 0.01 for each successive test. Results indicated that for most values of *F* tested under 0.97 converged. As the value of step size increased, so did the number of generations required for convergence. Storn and Price noted that mention that the results outside of the range of 0.5 to 1.0 tend not to be reliably replicable [12], so an *F* value of 0.54 was chosen to be used in the later trials.

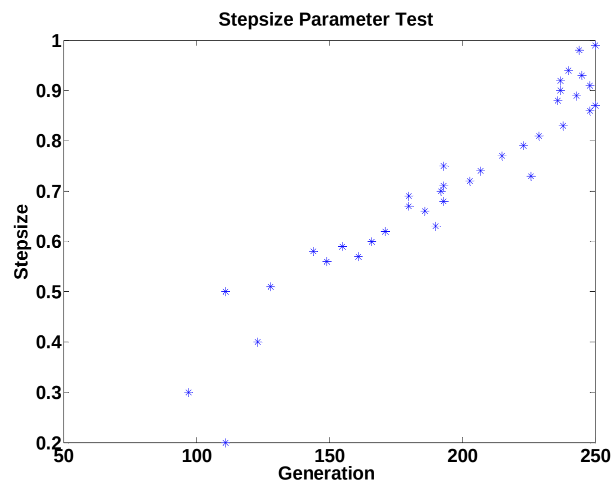

Figure 3.2: Plot of the trend of the number of generations required for converging *F* values

3.1.4 Crossover Ratio

The crossover ratio (*CR*) is a constant that controls how much crossover occurs. A larger *CR* allows for more crossover. For these tests, the value of *CR* was varied from 0 to 1 in steps of 0.02. The crossover ratio results converged to Δv_{inital} for only a small range: from 0.4 to 0.86. The number of generations required for convergence was a minimum $CR = 0.7$, so this was the value that was used.

3.2 Results

Using the input parameters taken from the testing described above, 100 trials were run for both L4 and L5 out to run 550 generations (74250 function evaluations). The number of generations was chosen to allow the maximum number of trials to be preformed while allowing DE ample space to search for global minimum. After these trials were completed, the results indicated that the algorithm converged by 200 generations; so to save on computation time, the maximum number of generations was then lowered to this value and another 50 trials for each point were run. Another set of trials, 25 for each point, were run seeded with values of *t, r,* and Δt_c from the previous trials corresponding to lowest Δv found to see if DE would find a lower Δv. However, the Δv for this set of trials immediately converged to the same value as Δv as before, so the results presented below exclude this last set of trials.

3.2.1 *L_4 Trajectory*

The lowest value of Δv found for a trajectory to L_4 was 6.1513 km/s. Using randomly generated values for the initial seeds, DE reached this minimum 83.3% of the time taking an average of 12960 function evaluations to converge. All the other trials converged to a slightly larger Δv of 6.2462 km/s. For the trials that reached this value, it was found that *t, r,* and Δt_c were all occur in a small range. This solution has and average mission commencement time was 21.312 ± 0.0167 days into 2010. The average flyby radius and correction time of flight were 59993.6 ± 4.36 km and 294.974 ± 0.053 days respectively. If analyzing values of *t, r,* and Δt_c at the end of 550 generations from first set of trials, their range is less. This suggests that if left to run to a larger number of function evaluations, the values of the variables for the best Δv for L_4 might all converge.

3.2.2 *L_5 Trajectory*

For the L_5 rendezvous, the lowest Δv was 5.0559 km/s achieved in an average of 15120 function evaluations. The DE algorithm was less reliable in this case, reaching this Δv only 51.3% of the time with initial random seeds. However, all other trials converged to a Δv of 5.0575 km/s which in the physical terms of the of the problem is not significantly different.

The minimum Δv was obtained with a mission commencing on day 363.705 ± 0.0121 of the year with an average *r* of 59993.1 km ± 3.913 km. This optimal trajectory requires a correction time of flight of 410.926 ± 0.057 days.

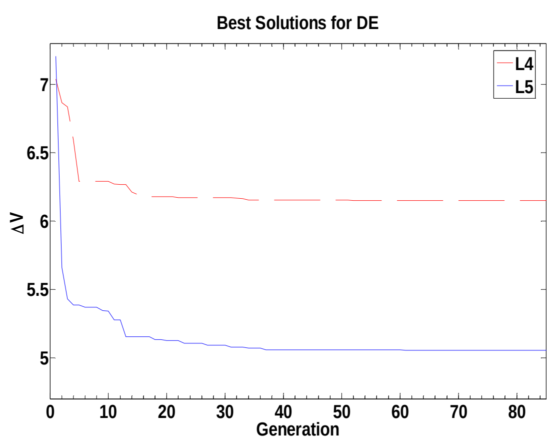

Figure 3.2: Plot of the solutions taking the fewest function evaluations for both L_4 and L_5

4. CMA-ES

The CMA-ES code was adapted to solve the problem at hand by adding additional lines of code that would support the boundary conditions required for the problem. Boundary conditions were handled as solid walls that could not be exceeded. Any values outside these boundaries were returned back to the nearest boundary (upper or lower bounds).

4.1 CMA-ES Self-Adaptation

A big advantage of the CMA-ES algorithm is its ability to self-adapt to the function that is being solved. With the use of a covariance matrix, successful mutations in the solution are tracked and used to update the covariance matrix every generation to reflect which mutations were good and which were bad. The goal is to increase the probability of successful mutations such that the individual solutions being computed continue to move towards an optimum. The covariance matrix helps determine the probability of which direction the mutations will move the population.

4.2 CMA-ES Parameters

The CMA-ES algorithm can be customized through several parameters. In most cases, the default values for the parameters have been chosen due to past experimentation with the algorithms authors. It is possible to adjust the population size per generation, set the maximum number of function evaluations to perform, the initial step size, number of parents, and fitness stopping criteria. For this problem, the minimum cost is unknown; therefore the fitness stopping criteria is left at zero, forcing the algorithm to continue until the maximum number of function evaluations has been reached. The CMA-ES tutorial documentation recommends that the population size and number of parents used for recombination remain at the default values which are set as a function of the number of dimensions of the problem. In this case it results in a population size of 7 and the use of 3 parents for recombination to generate the off-spring solutions. The final parameter is the initial step size. It is recommended by the literature that this value be determined through experimentation, though the default value of 0.50 is generally sufficient [17].

4.3 Step Size Optimization

The problem being solved is expected to have many local minimum that will share very similar values and close to the actual optimum for the problem. To assist in the early initial search while the covariance matrix has time to "learn" over the first several generations, selecting an optimal step size should help in finding the optimum. The initial step size can be defined to be between 0.0 and 1.0. Experimentation was used to determine the best value of the step size for this problem. To determine the value, the algorithm was run 40 times at 10 different step-size values from 0.05 to 0.95 in steps of 0.10. After the trials were finished, the 40 trials at each step size were averaged and plotted. Figure 4.1 shows the results of the trials where in general the smaller initial steps resulted in a slower convergence, but ultimately some of the lowest average costs. Based on these results, all additional runs of the CMA-ES algorithm will utilize a smaller initial step size of 0.25 as apposed to the default value of 0.50.

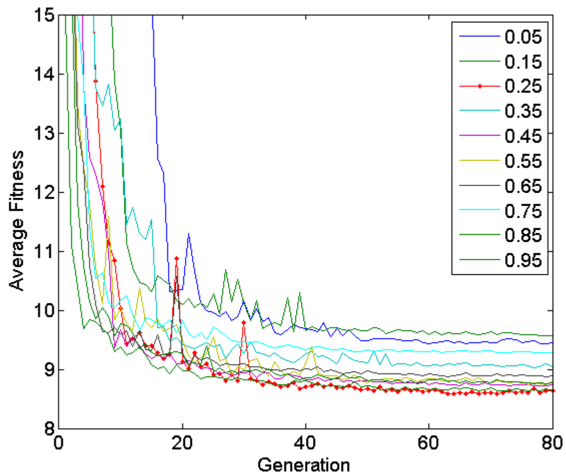

Figure 4.1: Initial Step Size Comparison

4.4 Results

Utilizing the results from the initial step sizing experiment, solutions for L_4 and L_5 were computed by running the CMA-ES algorithm 1,000 times for each trajectory. The CMA-ES algorithm performs 903 function evaluations at 129 generations per run. Run times vary, but an average of 85 runs of the CMA-ES can be completed per hour. At approximately 12 hours per trajectory optimization, the CMA-ES algorithm generates many very good solutions in a fraction of the time an exhaustive search would take.

4.4.1 L_4 Trajectory

The best trajectory computed for an L_4 rendezvous required a Delta V of 6.152 km/s. Figure 4.2 shows the plot of the minimum trajectory cost per generation. The trial that generated this minimum solution was the 16 trial of the 1000 trials run.

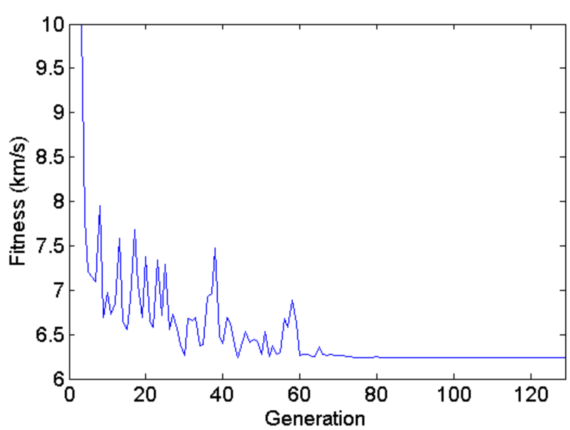

Figure 4.2: Minimum Delta V Solution

The minimum solution requires that the mission commence 21.3 days into the year, have a lunar fly-by radius of 59,497 km, and rendezvous with the L_4 Lagrange point 294.9 days after the lunar fly-by. Of the 1,000 trials tested, the results ranged between 6.152 km/s and 6.662 km/s with a mean of 6.348 km/s. 15.3% of all the solutions found were within 1% of the minimum found and 77.1% of all 1000 solutions were within 5% of the minimum solution found. The best trajectory was still 17.1% greater than a direct transfer to the L4 point indicating that a lunar flyby is not helpful for this mission.

4.4.2 L_5 Trajectory

The best solution found for the L_5 rendezvous was a trajectory that resulted in a required Delta V of 5.057 km/s. As expected, this solution was smaller than that of the L_4 rendezvous trajectory and consistent with the results of Ref. [2]. The solution of the minimum trajectory is shown in Figure 4.3 and shows a similar convergence of the L_4 minimum solution.

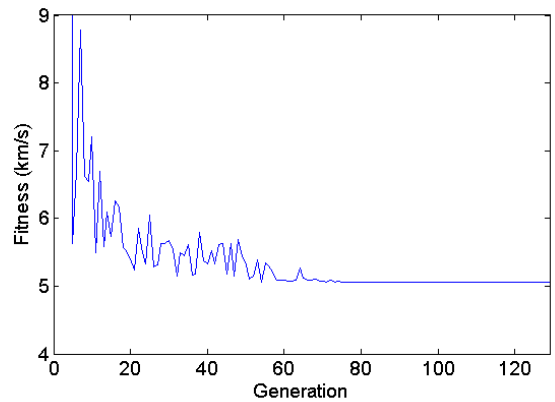

Figure 4.3: Minimum Delta V Solution

The best solution for the L5 trajectory has a mission commencement time of 4.3 days into the year, has a lunar fly-by radius of 59,916 km, and a rendezvous with the L5 point 410.6 days after the lunar fly-by. Like the solutions of the L4 trajectory, the solutions found by the CMA-ES algorithm were bunched close together with an average solution of 5.249 km/s. Of all of the solutions found 12.1% of the solutions were within 1% of the minimum solution found and 75.6% of the solutions were within 5% of the minimum solution found during the 1000 trial runs. Like the best L4 trajectory found by the CMA-ES algorithm, the best L5 solution was greater than the direct transfer solution found in Ref. [2] and was 8.7% greater.

5. OPTIMAL TRAJECTORIES

Tables 5.1 and 5.2 compare the best trajectory parameters found by the different evolutionary algorithms.

	Δv (km/s)	t (days)	r (km)	Δt_c (days)
DE	6.151	21.3	59,993	294.9
CMA-ES	6.152	21.3	59,497	294.9

Table 5.1: Best L_4 trajectory parameters

	Δv (km/s)	t (days)	r (km)	Δt_c (days)
DE	5.056	363.7	5,9993	410.9
CMA-ES	5.057	4.3	59,497	410.6

Table 5.2: Best L_5 trajectory parameters

Figure 5.1 and 5.2 show the optimal trajectories to both triangular points using the best results.

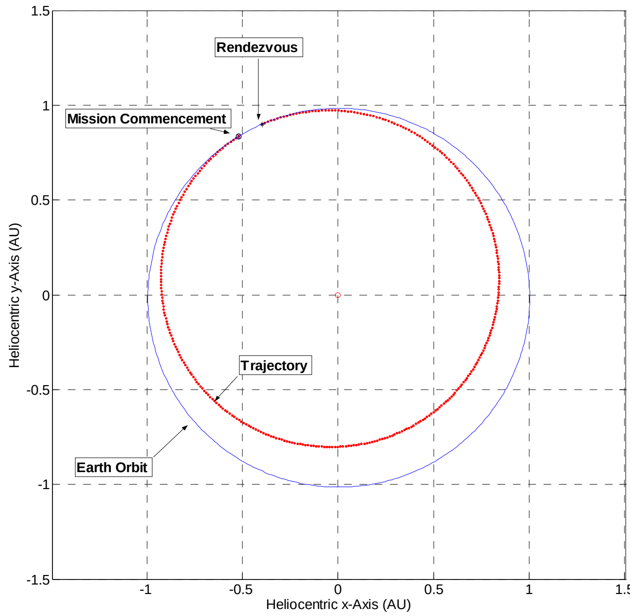

Figure 5.1: Best L_4 Trajectory

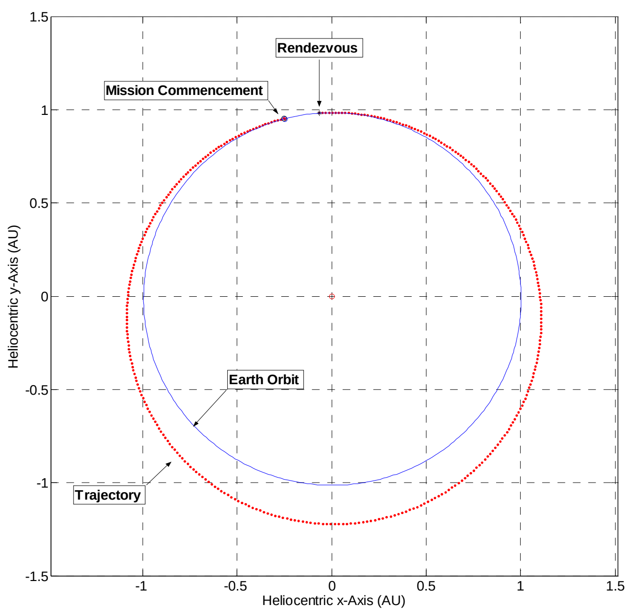

Figure 5.2: Best L_5 Trajectory

6. CONCLUSIONS

The results reveal that sending a satellite from Earth orbit to the Sun-Earth triangular Lagrange points using a lunar gravity assist will generally cost more than the co-orbital rendezvous results predicted by Benavides and Spencer [2]. The optimal trajectory parameters calculated for both triangular points by DE and CMA-ES agree closely.

7. REFERENCES

[1] Curtis, H. D. (2005): *Orbital Mechanics for Engineering Students*; Elsevier; pp. 89-96.

[2] Benavides, J.C.; Spencer, D.B. (2007): *Sun-Earth Triangular Lagrange Point Insertion and Satellite Station Keeping*; AAS/AIAA Space Flight Mechanics Conference; Sedona, Arizona, United States; January 28-February 1, 2007.

[3] *SOHO, 10 Years (1985-2005)*; http://soho.esac.esa.int; June 24, 2006.

[4] *Advanced Composition Explorer (ACE)*; http://www.srl.caltech.edu/ACE; June 24, 2006.

[5] *Wilkinson Microwave Anisotropy Probe*; http://www.map. gsfc.nasa.gov; June 24, 2006.

[6] Gomez, G.; Llibre, J.; Martinez, R.; Simo, C. (2001): *Dynamics and Mission Design Near Libration Points, Vol. II Fundamentals: The Case of Triangular Libration Points*; World Scientific

[7] Valdes, F.; Freitas, R. A. (1983): A Search *for Objects near the Earth-Moon Lagrangian Points*; Academic Press

[8] Curtis, H. D. (2005): *Orbital Mechanics for Engineering Students*; Elsevier; pp. 257-263.

[9] Vallado, D. A. (2001): *Fundamentals of Astrodynamics and Applications*; Microcosm Press; pp. 104-126.

[10] Prussing, J. E.; Conway, B. A. (1993): *Orbital Mechanics*; Oxford; pp. 62-79.

[11] Bate, R. R.; Mueller, D. D.; White, J. E. (1971): *Fundamentals of Astrodynamics*; Dover; Pages 227-271.

[12] Curtis, H. D. (2005): *Orbital Mechanics for Engineering Students*; Elsevier; pp. 202-213.

[13] Storn, R. & Price, K. (1997): *Differential Evolution- A simple and Efficient heuristic for Global Optimization over Continuous Spaces.* J. of Global Optimization, pp. 341-359.

[14] Differential Evolution Homepage; *http://www.icsi.berkeley.edu/~storn/code.html*; March 30, 2007

[15] Bessette, C. (2006): *Optimal Interplanetary Trajectories via Evolutionary Algorithms*. Masters Thesis, Penn State University

[16] The CMA Evolution Strategy; *http://www.bionik.tu-berlin.de/user/niko/cmaesintro.html*; April 15, 2007

[17] The CMA Evolution Strategy: A Tutorial; *http://www.bionik.tu-berlin.de/user/niko/cmatutorial.pdf*; April 15, 2007