J of Astronaut Sci (2016) 63:81–102
DOI 10.1007/s40295-016-0084-2

Early Mission Design of Transfers to Halo Orbits via Particle Swarm Optimization

Andrew J. Abraham[1] · David B. Spencer[2] · Terry J. Hart[1]

Published online: 16 March 2016

Abstract Particle Swarm Optimization (PSO) is used to prune the search space of a low-thrust trajectory transfer from a high-altitude, Earth orbit to a Lagrange point orbit in the Earth-Moon system. Unlike a gradient based approach, this evolutionary PSO algorithm is capable of avoiding undesirable local minima. The PSO method is extended to a "local" version and uses a two dimensional search space that is capable of reducing the computation run-time by an order of magnitude when compared with published work. A technique for choosing appropriate PSO parameters is demonstrated and an example of an optimized trajectory is discussed.

Keywords Particle swarm optimization · Evolutionary algorithm · Transfer orbit · Halo orbit · Lagrange point orbit · Stochastic optimization

Introduction

Transfers to Lagrange point orbits in the Earth-Moon system have recently become popular amongst spacecraft mission planners [1, 2]. The ARTEMIS mission,

Presented at the 24th AAS/AIAA Space Flight Mechanics Meeting, January 2014, Santa Fe, NM, No. 14-315

✉ Andrew J. Abraham
andrewabraham1@gmail.com

[1] Mechanical Engineering, Lehigh University, 19 Memorial Dr. W., Bethlehem, PA 18015, USA

[2] Aerospace Engineering, Penn State University, 229 Hammond Building, University Park, PA 16802, USA

designed to study the distant regions of Earth's magnetotail, is the most recent example of a mission to the Earth-Moon L_1 and L_2 Lagrange points. Other applications of Lagrange point orbits include communications [3], observation and surveillance [4], navigation, and human operations [5, 6]. While the majority of mission design has focused on impulsive transfers to these Lagrange points, or transfers between Lagrange point orbits [7–9], a small but growing body of research concerning low-thrust transfers is beginning to develop [10–13].

Mingotti et al. [14–16] used direct transcription and collocation to develop a low-thrust trajectory beginning with an initial guess of a simple, two-body, low-thrust spiral. His method, although quite rigorous, was built on gradient-based algorithms that are prone to becoming trapped within local minima and are very dependent on a "good" initial guess. The goal of this paper is to identify a useful method capable of quickly searching a global searchspace and identifiying a "good" inital guess solution that could then be refined further utilizing a more focused, gradient-based method such as the one discussed previously. Abraham et al. [17] used an evolutionary algorithm called Particle Swarm Optimization (PSO) to "prune" the search space and identify the vicinity of the best local minima (assumed, but not proven, to be the global minimum) as well as an adequate guess solution for Mingotti's method. This PSO algorithm was global in nature, optimized in only one dimension, and required an enormous amount of computational run-time (about 24 h). The work presented here is an extension of Abraham's original work. The PSO algorithm is converted into a "local," two dimensional form that is capable of interrogating multiple local minima simultaneously, thus greatly reducing computation run-time due to an additional degree of freedom. As an example of the capabilities of this method, this study will demonstrate the use of PSO to optimize a low-thrust transfer from a high-altitude, Earth orbit to an Earth-Moon, L_1, northern halo orbit via stable manifolds.

Problem Statement

Three-body, perturbative effects of the Moon become increasingly dominant in regions above low Earth orbit. While low-thrust trajectories have been studied extensively for low Earth orbits [18–21] only a small body of work exists [14–17] for low-thrust trajectories above low or medium Earth orbit. As with Abraham et al. [17] an optimal trajectory is desired that takes a spacecraft from a high-altitude, Earth orbit to a desired Earth-Moon, Lagrange point orbit via a low-thrust arc that terminates on the invariant stable manifold of the nominal (i.e. desired) Lagrange point orbit. The ratio of the Moon's gravitational force to the Earth's gravitational force is around 10^{-4} at 35,000 km in altitude and rapidly increases above that altitude. Since the main objective of this study is to focus on a regime of motion where three-body effects are significant, any low-thrust trajectory less than the orbital energy of of the high-altitude, Earth orbit will not be considered. Instead, for the sake of demonstration, ease of computational resources, and justifiably neglecting drag and geopotential forces the spacecraft's initial Earth orbit will begin with a Jacobi energy of a high Earth orbit and will increase as a function of time and thrust. Of course, the

initial state of the high-altitude, Earth orbit is controlled entirely through the introduction of a cost function that is problem-dependent. In some situations it is appropriate to have a cost function that is highly constrained by containing information capable of defining six orbital states of the original, high-altitude, Earth orbit. Some early mission design scenarios, however, are not nearly so constrained and would allow for a great deal of freedom in one or more orbital elements. This is a situation where early mission design using the PSO method truly excels.

In two-body dynamics it is well known that the most time and fuel-efficient continuous low-thrust control law is that of tangential thrust [14, 15]. Since continuous tangential low-thrust will not significantly alter the eccentricity of a circular orbit, it is assumed that the eccentricity of the initial high-altitude, Earth orbit should (ideally) remain low. This goal is incorporated into the fitness function described in later sections of this paper. The concept of utilizing the stable manifold is not arbitrary as a spacecraft that is placed on the manifold will be ballistically delivered to the nominal Lagrange point orbit after some finite amount of time. Unfortunately, the stable manifolds in the Earth-moon system never come very close to the Earth, as is evidenced by Fig. 3. It, therefore, becomes unavoidable for a low-thrust mission to operate an appreciable amount of time in a regime that includes three-body effects.

System Dynamics

The Circular Restricted Three Body Problem (CR3BP) is used to model the motion of the spacecraft under the simultaneous gravitational acceleration of two massive primaries. Referring to Fig. 1, which uses a synodic reference frame with its origin located at the barycenter of the two primaries, the equations of motion can be expressed as

$$\begin{aligned} \ddot{x} - 2\dot{y} &= \Omega_{\mathrm{x}} + u_{\mathrm{x}}(t), \\ \ddot{y} + 2\dot{x} &= \Omega_{\mathrm{y}} + u_{\mathrm{y}}(t), \\ z &= \Omega_{\mathrm{z}} + u_{\mathrm{z}}(t) \\ \dot{m}(t) &= -\frac{\|\mathrm{T}(t)\|}{I_{sp} g_o} \end{aligned} \tag{1}$$

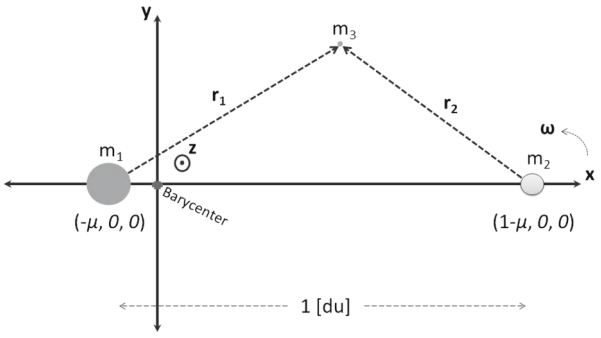

Fig. 1 Coordinate system of CR3BP

with the subscripts of the pseudopotential

$$\Omega(x, y, z) = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} \tag{2}$$

indicating partial derivatives with respect to the sub-scripted variable. T represents the thrust of the low-thrust engine, I_{sp} is the specific impulse, m the spacecraft's instantaneous mass, and g_o is the Earth's gravitational acceleration at sea level. The distances $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x-(1-\mu))^2 + y^2 + z^2}$ represent the normalized distance between the spacecraft and the smaller primary (Moon) and larger primary (Earth), respectively. Note that Eq. 1 are written in dimensionless units, that is to say that the angular velocity of the larger and smaller primary (ω), the sum of the masses of each primary, and the distance between each primary are all intentionally set to unity. The unit of distance, [du], is the distance from the larger primary to the smaller primary, while the unit of time, [tu], is equal to the synodic period of the primaries divided by 2π . The velocity unit, [vu], is simply equal to [du]/[tu]. The mass parameter is defined as

$$\mu = \frac{m_2}{m_1 + m_2} \tag{3}$$

which represents the ratio of the mass of the smaller primary divided by the total mass of the system. In the case of the Earth-Moon system, this mass parameter is $\mu = 0.0121506683$. The vector $\mathbf{u(t)} = [u_x(t), u_y(t), u_z(t)]^T = \left[\frac{T_x(t)}{m(t)}, \frac{T_y(t)}{m(t)}, \frac{T_z(t)}{m(t)}\right]$ represents the acceleration of the third body due to the low-thrust engine in the synodic frame.

Setting $\mathbf{u} = \mathbf{T} = 0$ defines a ballistic trajectory in the CR3BP. Under ballistic conditions, exactly one constant of motion can be defined [22] for Eq. 1

$$C(x, y, z, \dot{x}, \dot{y}, \dot{z}) = 2\Omega(x, y, z) - (\dot{x}^2 + \dot{y}^2 + \dot{Z}^2) = \text{constant}, \tag{4}$$

and is known as the Jacobi energy. For a given spacecraft position and velocity, there exists exactly one Jacobi energy in the ballistic CR3BP. It is also possible for the spacecraft to locate itself at an equilibrium point of Eq. 1. Five equilibrium points, known as a Lagrange points, exist; with three collinear points (L_1, L_2 and L_3) and two triangular points (L_4 and L_5). It is possible to define complex, periodic trajectories in the vicinity of each collinear Lagrange point. These periodic trajectories are known as two-dimensional Lyapunov orbits, three-dimensional halo orbits, and quasi-periodic Lissajous trajectories.

Invariant Stable Manifolds

A Lagrange point orbit can be identified either via tabulated values [22], numeric computation that solves a two-point boundary value problem [13, 23, 24], or approximated using annalytic expressions [25, 26]. The Lagrange point orbit is then defined by an arbitrary state vector (anywhere along the orbit), $\mathbf{X_{orbit}}$, and the period of the

orbit, P. Under controlled CR3BP dynamics, the state vector for the spacecraft is expressed in terms of position, velocity, and spacecraft mass as

$$\mathbf{X} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ m \end{bmatrix} \tag{5}$$

with the controlled CR3BP expressed in terms of a first-order differential equation as

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ 2v_y + \Omega_x + u_x \\ -2v_x + \Omega_y + u_y \\ \Omega_z + u_z \\ -\frac{T}{I_{sp} g_o} \end{bmatrix} \tag{6}$$

According to Dynamical Systems Theory (DST) [27], numeric integration of Eq. 1 (with ballistic assumptions of $\mathbf{u} = \mathbf{T} = 0$ and $m = constant$) as well as the State Transition Matrix (STM) $\Phi_{(t,t_0)} = \frac{\partial \mathbf{X}(\mathbf{X}_0,t)}{\partial \mathbf{X0}}$, for one period, P, with the initial condition $\mathbf{X}_0 = \mathbf{X_{orbit}}$ enables the calculation of the monodromy matrix, $\mathbf{M} = \Phi_{(t=P,t_0)} = \frac{\partial \mathbf{X}(\mathbf{X}_0,t=P)}{\partial \mathbf{X0}}$ associated with the initial state, $\mathbf{X_{orbit}}$. The stable eigenvector, ν, of the monodromy matrix is multiplied by a very small number, ϵ, where $\epsilon = 10^{-10}$ in this study. A perturbed initial state, $X_{pert} = X_{orbit} \pm \epsilon\nu$, is then integrated backward in time using the ballistic version of Eq. 1. As in Abraham et al. [17] the integration is terminated when the spacecraft crosses the yz-plane from the negative x-direction. This defines a trajectory that is a member of the stable manifold of the nominal Lagrange point orbit. The nominal Lagrange point orbit can be discretized into N states defined as $\mathbf{X^{(k)}_{orbit}}$ with $k \in [1, N]$. Typically, the value of N is large enough to accurately construct the Lagrange point orbit with a high degree of fidelity (approximating a continuous variable) yet still small enough to avoid becoming computationally prohibitive (similar to finite element methods). The process is then repeated for other values of $\mathbf{X^{(k)}_{orbit}}$ which, in turn, constructs a stable manifold of N trajectories. If a spacecraft's state lies along a trajectory within this manifold then the ballistic flow forward in time will take it to the state $\mathbf{X^{(k)}_{orbit}}$ and the spacecraft will be automatically inserted into the nominal Lagrange point orbit. In this way, any state within the manifold, $\mathbf{X_{s.m.}}(\tau_{01}, k)$, can be expressed via two parameters:

1. An integer $k \in [1, N]$ that corresponds to a state on the nominal orbit $\mathbf{X^{(k)}_{orbit}}$ with N being the total number of states that represent a discretization of the orbit.
2. A time parameter τ that represents the time remaining for a ballistic flow of Eq. 1 to reach the state $\mathbf{X^{(k)}_{orbit}}$.

Feasible Trajectory Generation and Low-Thrust Control Law

Using Dynamical Systems Theory, the invariant stable manifold odiscretizationf the nominal Earth-Moon Lagrange point orbit is constructed as described in the preceding section. Next, a particular trajectory segment, k, is randomly selected from the stable manifold, W^s. From this trajectory segment a random state, $\mathbf{X_{s.m.}}(\tau_{01}, k)$, is chosen to be a "patch point." A low-thrust trajectory is then propagated backwards in time, according to Eq. 1, from this patch point until the Jacobi energy of the spacecraft is equal to the Jacobi energy of a chosen Earth orbit; then the integrator stops. The control law for the low-thrust acceleration during this propagation is quite straightforward:

$$\begin{aligned} \| \mathbf{u(t)} \| &= \tfrac{T_{max}}{m(t)} \\ \hat{\mathbf{u}}(t) &= \hat{\mathbf{v}}(t) \end{aligned} \tag{7}$$

with T_{max} being the magnitude of the maximum thrust of the engine and $\hat{\mathbf{v}}$(t) being a unit vector oriented in the same direction as the instantaneous, three-body velocity of the spacecraft. This control law was successfully used by Mingotti et al. [14] and was shown to be the most fuel and time efficient control law obtainable. A summary of this method for an are arbitrarily chosen trajectory and patch point is described below:

1. Create an invariant stable manifold, W^s, based on the nominal L-point orbit by using DST and backwards integrating the Eq. 1 of the CR3BP
2. Arbitrarily select a trajectory, k, from this manifold and then arbitrarily select a patch point, $\mathbf{X_{s.m.}}(\tau_{01}, k)$, from within this trajectory
3. Propagate the low-thrust trajectory from this patch point backwards in time until the Jacobi energy matches that of the chosen Earth orbit or orbit family.

It is now possible to describe a feasible continuous trajectory from an Earth orbit to a Lagrange point orbit given a patch point, $\mathbf{X_{s.m.}}$. It, therefore, becomes necessary to optimize this trajectory by identifying the optimal patch point, $\mathbf{X^*_{s.m.}}$. Thus, the search space for the PSO technique becomes all the possible points within the stable manifold.

Algorithm

Fitness Function

The fitness function used in this method is highly problem dependent but can be described in general as a function of a state on the stable manifold:

$$J(\mathbf{X_{s.m.}}) = c_1 f_1(\mathbf{X_{s.m.}}) + c_2 f_2(\mathbf{X_{s.m.}}) + c_3 f_3(\mathbf{X_{s.m.}}) + \ldots \tag{8}$$

where the terms $c_a f_a(\mathbf{X_{s.m.}})$ are selected by the researcher to satisfy two goals: (1) to optimize the transfer by minimizing some quantity (usually fuel mass or time of flight) and (2) to satisfy some initial conditions on the Earth orbit. The second goal

is accomplished by choosing a functional form that vanishes near the desired target for the constraint and is severely penalized elsewhere. In this way the inital conditions of the Earth orbit can be fully or partially defined, along with their respective tolerances, as the researcher sees fit. Keep in mind that the goal of this method is early mission planning with opportunities for fine-tuneing the solution reserved for follow-up, gradient-based methods.

Parametrization of the Search Space

The Particle Swarm Optimization (PSO) algorithm requires the a priori definition of a "search space" where it is permitted to search for an optimal solution. In this study, the search space is defined as all states within the invariant stable manifold, $\mathbf{X_{s.m.}}(\tau_{01}, k) \in W^s$ and within certain bounds. Each state is uniquely defined by exactly two parameters: k and τ. The parameter k represents an individual trajectory member of the stable manifold ($k \in W^s$) which is generated via the method outlined in the "Invariant Stable Manifolds" section of this paper. The parameter τ_{01} is the second parameter that defines $\mathbf{X_{s.m.}}$ and is defined relative to τ via a simple mapping function. The time of flight, τ, represents the amount of time required to get from an initial state to a state on the nominal Lagrange point orbit, as defined in the "Invariant Stable Manifolds" section. Unfortunately, it is impossible to define the entire stable manifold as a search space because it is infinite in nature and a search space (by definition) must be finite. The bounds of τ, therefore, are carefully chosen such that a wide swath of relevant manifold states are captured within the search space and irrelevant manifold states are excluded.

In this study, the bounds of τ are $\tau_{L.B.} \leq \tau_{s.m.} \leq \tau_{U.B.}$ with:

- $\tau_{L.B.}$ being the time that trajectory k crosses the yz-plane from the positive x direction
- $\tau_{U.B.}$ being the time that trajectory k crosses the yz-plane located at $x = L_1$.

The $\tau_{U.B.}$ bound was chosen to ensure that the fitness function (8) can be evaluated at any point within the search space. If, for example, a patch point was located on the moon side of L_1, the control law (7) would cause the spacecraft to flow towards a lunar orbit instead of an Earth orbit; thus invalidating Eq. 8. The opposite bound, $\tau_{L.B.}$ was chosen as a matter of convenience and practicality. While it is true that the trajectories of W^s continue to flow for an infinite amount of time, one needs to cut off this flow after a finite amount of time due to the limitations of computing power. Since the run times of the PSO method can become quite large, a smaller search space is needed to adequately converge on an optimal solution. While this limited search space does preclude the possibility of lunar phasing maneuvers, such as those used by Mingotti et al. [14], it is more than adequate to address a fitness function that attempts to minimize time of flight. The values of τ are mapped to τ_{01} using the simple relationship that $\tau_{L.B.} = 1$ and $\tau_{U.B.} = 0$. Therefore, the values of τ for a given value of k are mapped to a normalized range of $\leq \tau_{01} \leq 1$. This mapping ensures that values of τ between $\tau_{L.B.}$ and $\tau_{U.B.}$ are treated equally, regardless of the value of k and the time of flight between the nominal Lagrange point orbit and the yz-plane located at $x = L_1$.

In a similar fashion, the values of k are mapped between $1 \leq k \leq N$ via a modulus function. In this study, $k = k_{desired}$ **mod** (N). This means, for example, that if $k_{desired} = N + x$ then $k = x$ assuming $0 \leq x \leq N$. Using this technique, no value of $k_{desired}$ is ever excluded from the search space but is instead looped back onto itself in k-space. In summary, any value of $\mathbf{X_{s.m.}}(\tau_{01}, k)$ can be uniquely parametrized, in $k\tau$-space, in terms of τ_{01} and k with the boundaries of these parameters being real numbers, $0 \leq \tau_{01} \leq 1$ and positive integers, $1 \leq k \leq N$.

Particle Swarm Optimization

The PSO algorithm is simple yet powerful [13]. It consists of N_p particles which are, initially (ie. $j = 1$), randomly distributed throughout the search space with a position, $\boldsymbol{\chi} = [\tau_{01}, k]^\mathrm{T}$, and velocity, $\boldsymbol{\omega} = [V_\tau, V_k]$. Note that the position maps to the state vector in the following manner:

$$\chi = [\tau_{01}, k]^\mathrm{T} \quad \Rightarrow$$

$$\mathbf{X_{s.m}}(\tau_{01}, k) = [x_{s.m.},\ y_{s.m.},\ z_{s.m.},\ x_{s.m.}, y_{s.m.}, Z_{s.m.}, m^T_{s.m.}. \tag{9}$$

Both the velocity and position of each particle in the search space is calculated by Eqs. 10 and 11, respectively:

$$\boldsymbol{\omega}_i^{(ij+1)} =$$

$$C_1(1+R_1(i,j))\boldsymbol{\omega}_i^{(j)} + C_\mathrm{C} R_2(i,j)\left(\boldsymbol{\psi}_i^{(j)} - \boldsymbol{\chi}_i^{(j)}\right) + C_S R_3(i,j)\left(\mathbf{Y}^{(j)} - \boldsymbol{\chi}_i^{(j)}\right) \tag{10}$$

$$\boldsymbol{\chi}_i^{(j+1)} = \boldsymbol{\chi}_i^{(j)} + \boldsymbol{\omega}_i^{(j+1)} \tag{11}$$

with the superscripts representing the j^{th} iteration $(1 \leq j \leq j_{max})$ of the PSO algorithm and the subscripts representing the i^{th} particle $(1 \leq i \leq N_p)$. Note that $R_{1,2,3}(i,j)$ represents a random number $0 \leq R_{1,2,3}(i,j) \leq 1$ following a uniform distribution, and the constants C_I, C_C, C_S, represent the "Inertial," "Cognitive," and "Social" weighting coefficients, respectively. The fitness function, Eq. 8, is evaluated for each particle and $\psi_i^{(j)}$ and $\mathbf{Y}^{(j)}$ are recorded. The "personal best" value, $\psi_i^{(j)} = \left[\tau_i^{(best)}, k_i^{(best)}\right]$, represents the best known value of the fitness function, recorded by particle i from iteration one to j. The "global best" value, $\mathbf{Y}^{(j)} = \left[\tau^{(best)}, k^{(best)}\right]$, represents the best known value of the fitness function, recorded by any particle in the swarm, from iteration 1 to j. In this way, the position and velocity of each particle can be calculated for iteration $j + 1$ based on the information contained in iteration j using Eqs. 10 and 11.

The inertial coefficient directs the particle's motion according to Newtonian mechanics (i.e. motion directed along the current velocity vector). The cognitive coefficient allows each particle to "remember" the best location it has visited and acts as an attractor to that location. Finally, the social coefficient allows each particle to "communicate" with the others in the swarm and attracts particle i to that location. The values of the coefficients used in this study have been inspired by the work of Pontani and Conway [28] and were only modestly modified, via trial and error,

by the author to achieve reasonable convergence while still identifying obvious local minima. They are summarized as follows:

$$\begin{aligned} \mathrm{C}_I &= 015 \\ \mathrm{C}_C &= 1.0\,. \\ \mathrm{C}_S &= 100 \end{aligned} \tag{12}$$

Local PSO The method outlined above is excellent for identifying local minima when only a few minima are present. Unfortunately, if a large number of local minima exist then the global PSO algorithm displays a tendency to converge on a non-optimal local minima instead of the best local minima discoverable. This algorithmic shortcoming exists because the global best solution, $\boldsymbol{Y}^{(j)}$, draws other particles away from what is oftentimes the vicinity of a better local minimum. This is especially true when the depth of multiple local minima are very similar. To avoid this problem a "local" version of the PSO algorithm has been developed and utilized in this research. This is accomplished by limiting the ability of the particles' to communicate over distances grater than some cutoff distance, $\mathbf{r}_{local}$. This mirrors conditions found in nature where collaborating swarms of animals have an inability (or a retarded ability) to communicate over vast distances, thus allowing more time to explore nearby local minima. In this study, $\boldsymbol{Y}^{(j)}$ is modified to $\boldsymbol{Y}^{local(i)}$ by utilizing the best value of a local swarm defined as all particles within radius, $\mathbf{r}_{local}$, of particle i. If particle i can not "see" a distant particle then that distant particle has no influence over the value of $\boldsymbol{Y}^{local(i)}$. Typically, $\mathbf{r}_{local} = \left[\frac{1}{20}, \frac{1}{16}N\right]^T$ in this study and is noted in the text otherwise.

As the PSO algorithm evolves over j iterations, the particles in the swarm begin to collect around various local minima. It becomes possible to define a convergence metric, γ, for the entire system. This metric is defined by

$$\gamma^{(j)} = \frac{N_C^{(j)}}{N_\mathrm{p}} \tag{13}$$

where $N_C^{(j)}$ represents the number of particles that have converged to the vicinity of $\boldsymbol{Y}^{j}_{local(i)}$. The vicinity of $\boldsymbol{Y}^{(j)}_{local(i)}$ is defined to be a circular area of $k\tau$-space, centered on $\boldsymbol{Y}^{(j)}_{local(i)}$, with a radius that is roughly 14 % the size of r_{local} and an area that is roughly 2 % the area of a circle of radius r_{local}. Using this definition of convergence it becomes possible to track the convergence of the PSO algorithm as a function of j and even terminate the algorithm early if a sufficient value of γ is reached.

Search Space Boundary Conditions Occasionally, a particle attempts to exit the permissible search space to which it must be bound. In such a case, a series of rules is followed to gently guide the particle back into the search space and continue its search for the best local minima discoverable. One reason why a particle may attempt to exit the search space is because its velocity is too large. In this case, a saturation

limit is imposed on the velocity vector such that $\boldsymbol{\omega} \leq \boldsymbol{\omega}_{max} = [\pm\tau_{max}, \pm k_{max}]^T = \left[\pm\frac{1}{2}, \pm\frac{1}{2}N\right]^T$. In general, this velocity saturation limit is very high and only acts upon extremely unreasonable velocity values. As a consequence, some values of $\boldsymbol{\chi}$ still fall outside the search space. If this is the case then the τ_{01} component of position is bounded by the saturation limits

$$\tau_{01} = \begin{cases} \tau_{max} = 1 \;\; for \; \tau_{01} > \tau_{max} \\ \tau_{min} = 0 \;\; for \; \tau_{01} < \tau_{min} \end{cases} \tag{14}$$

and the k component is bounded by the modulus function $k = k_{requested}$ **mod** N. The velocity of this particle is also reset to $\omega = \mathbf{0}$ to prevent the particle from exiting the search space during the subsequent iteration.

Problem-Specific Boundary Conditions The boundary conditions of the initial and final states of the optimized trajectory are specified via three separate mechanisms of the optimization scheme. The terminal state of the trajectory (located on the Lagrange-point orbit) is specified via a list of states comprising the target Lagrange-point orbit. Therefore, the orbit itself (and the states it defines) constitute the terminal end of the boundary conditions. The inital state (in Earth orbit) is determined by two mechanisums. The first is the Jacobi constant which is used as a trigger to terminate the transfer trajectory propagation. The second is the way in which the fitness function is construted. Additional paramaters that describe the inital state (eccentricity, inclination, semi-major axis, etc.) can be easily incorporated into the fitness function in such a way as to have the terms vanish when the constraint is satisfied (or very nearly satisfied) while being be very large otherwise.

Example Problem and Results

The theory presented above is now applied to a specific demonstration problem. In this problem the trajectory propagation was selected to terminate at a Jacobi energy equal to that of a high-altitude, Earth orbit which was chosen arbitrarily. Note that a lower altitude orbit could have been selected but would require longer computation times and does not demonstrate significantly increased three-body influence. Other researchers have selected similar orbits for similar reasons [10, 12]. Also note that only the Jacobi energy and eccentricity of this orbit are fixed while the remaining orbital parameters are allowed to change (or are unconstrained) to allow for a thorough exploration of the trade space during early mission planning.

In this paper, low-thrust trajectories were integrated according to the control law found in Eq. 7. All numeric integration was performed using MATLAB's ODE113 solver; a variable order Adams-Bashforth-Moulton solver with a relative and absolute tolerance of 10^{-13}. This method was chosen because it is relatively efficient when using stringent tolerances as well as being efficient with computationally intensive functions. The computer utilized 2 parallel cores on a 3GHz Intel Core 2 Duo

processor. A low-thrust spacecraft was used with the following parameters in accordance with studies [14–16] conduced by Mingotti et al.:

- a dry mass of 1000 kg
- an I_{sp} of 3000 s
- a maximum thrust of 500 mN.

Destination Orbit and its Associated Manifold

All data in this study has been generated using an Earth-Moon, L_1, northern halo orbit as the nominal destination orbit. This orbit has a period of $P = 231339$[tu] (10 days) and an initial state of

$$\mathbf{X_0} = \begin{bmatrix} 0.866224052875085 \\ 0.011670195668094 \\ 0.186912185139037 \\ 0.013870554690931 \\ 0.245270168936540 \\ 0.021792775971957 \end{bmatrix} \tag{15}$$

expressed in [du] and [vu] and used in references [17, 29]. The orbit is displayed in Fig. 2 with the states being discretized into $k_{max} = N = 791$ points (which represents the total number of points produced by the integrator) with each point corresponding to a particular trajectory, k, on the invariant stable manifold. The points are more densley populated where the curvature of the orbit is large since the state is changing the most rapidly near these locations.

This orbit was arbitrarily chosen as an example of how the PSO technique can optimize a transfer trajectory to a complicated and highly three dimensional orbit.

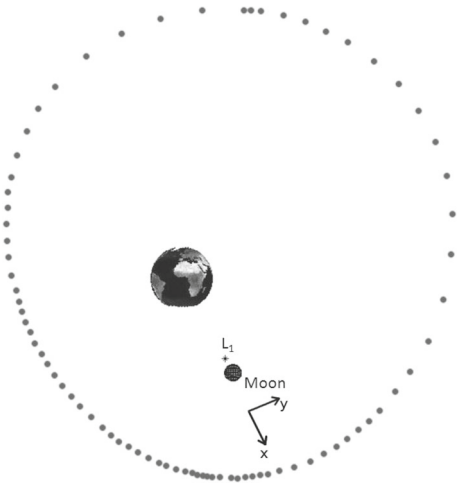

Fig. 2 The nominal Earth-Moon L_1 northern halo orbit. The orbit is broken into $N = 791$ points with every 10^{th} point displayed in this figure

According to Dynamical Systems Theory (DST) the invariant stable manifold of this northern halo orbit may be generated by the following procedure:

1. Select a point on the orbit and integrate the State Transition Matrix (STM) forward in time for one period.
2. Calculate the eigenvectors associated with the direction of the stable manifold at this point.
3. Multiply this eigenvector by a small number, ϵ, and add/subtract this small perturbation to the initial point.
4. Propagate this state vector backward in time until it crosses the x-axis from the $-x$ direction.

This procedure can be repeated numerous times; once for each point on the nominal orbit. All trajectories generated by points on the nominal orbit comprise the invariant stable manifold, W^s, of the nominal orbit. Figure 3 illustrates the invariant stable manifold of the Earth-moon, L_1, northern halo orbit.

Fitness Function

The fitness function used in this example problem can be expressed as a function of a state on the stable manifold:

$$J(\mathbf{X_{s,m}}) = c_1 \parallel e_{initial}(\mathbf{X_{s,m}}) - e_{desired} \parallel + c_2 \Delta m(\mathbf{X_{s,m}}) + c_3 \Delta T(\mathbf{X_{s,m}}) \quad (16)$$

where $e_{initial}$ is the eccentricity of the high-altitude-Earth-orbit, $e_{desired}$ is the desired eccentricity of the synchronous orbit, Δm is the fuel used by the low-thrust maneuver, and ΔT is the total Time of Flight (TOF) from the initial Earth orbit to the nominal Lagrange point orbit. The constants c_1, c_2, c_3 are all weighting constants

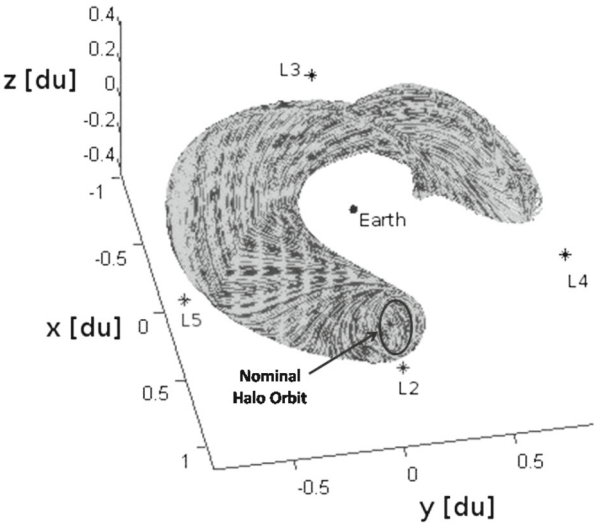

Fig. 3 Invariant stable manifold of the nominal, L_1 northern halo orbit. Note that the manifold never approaches the vicinity of Low Earth orbit

controlled by the researcher. In this study, the goal is to minimize the eccentricity of the original orbit and make it as circular as possible; therefore $e_{desired} = 0$. A circular orbit is desirable because many very low-thrust spirals begin and end with circular orbits. It is well known [14] that the most efficient control law for constant thrust (7) is the tangential thrust strategy, which will increase the specific energy of the spacecraft while leaving its eccentricity very near zero for most low-thrust trajectories. Thus, it is important that the target synchronous orbit also be near-circular if a mission planner desires to utilize a connecting low-thrust transfer from LEO as well.

Because no one term in Eq. 16 should dominate the others, the choice of the constants c_1, c_2, c_3 is of critical importance. In this study, the constants c_2 and c_3 are chosen such that the ratio

$$\frac{c_2 \Delta m(X_{s.m.})}{c_3 \Delta T(X_{s.m.})} \tag{17}$$

is does not allow either parameter to dominate the function. For example, if $\Delta m \approx 60$, $\Delta T \approx 30$ and the desired ratio of Eq. 17 was roughly 1:1 an appropriate choice of constants would be $c_2 = 0001$, $c_3 = 0002$. This selection method does not require *a priori* knowledge of the value of Δm or ΔT, but rather an *a priori* estimate of their relative magnitudes. Once c_2 has been determined, it is possible to set a value for c_1 by setting some target threshold for the first term in Eq. 16. Values below this threshold would be overwhelmed by the other terms in Eq. 16 while values above this threshold would dominate Eq. 16. In this study, a threshold value of 1 % seems reasonable because it represents a relatively small error in eccentricity. Here, the eccentricity represents an incorporation of a boundary condition into the fitness function while the remaining terms are intended to be optimized. Thus the ratio

$$\frac{c_2 \Delta m(X_{s.m.})}{c_1 \| e_{GEO}(X_{s.m.}) - e_{desired} \|} \tag{18}$$

should be roughly 1:1 assuming $c_2 = 0001$, $\Delta m \approx 60$, and the threshold value of $\| e_{initial}(X_{s.m.}) - e_{desired} \|$ is roughly 10^{-2}. This gives a value of $c_1 = 1$. In summary, following the procedure outlined in this example yields the values $c_1 = 1$, $c_2 = 0001$, $c_3 = 0002$ for the weighting constants. This setup ensures convergence on an optimal solution that rigidly enforces the eccentricity constraint while minimizing the fuel usage and time of flight.

Shape of the Fitness Function

The primary motivation for the application of evolutionary algorithms, in this study, is the lack of convexity of the fitness function. Figure 4 was generated via 150,000 random samples of the search space (with a 24 h computational run-time). The image to the left shows the raw data while the image to the right illustrates a curve fit using two dimensional, cubic interpolation from MATLAB's "Curve Fit" toolbox. While this fit is too crude to use in the optimization algorithm (and takes far too long to generate) it is helpful in visualizing the complexity of the fitness function. This function has a very large number of local maxima and minima with a wide variation in their values. To make matters worse, many of the extrema's values differ significantly with other points in their immediate vicinity. This yields nearly infinite gradients in the

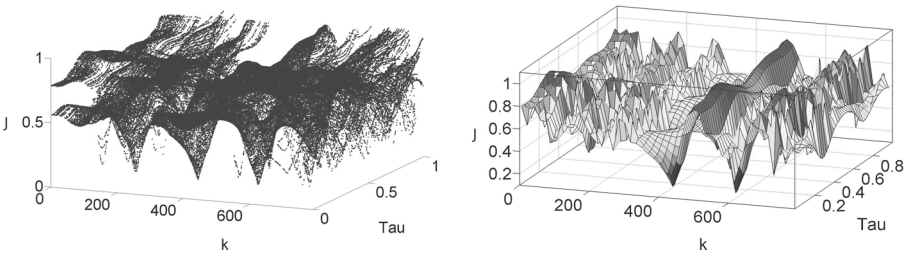

Fig. 4 *Left*: The fitness function is plotted in $k\tau$-space using roughly 150, 000 samples generated over a 24 h Monte Carlo trial. *Right*: A curve fit of this data was generated via MATLAB's "Curve Fit" toolbox to illustrate the non-convexity and irregular behavior of the fitness function

vicinity of these points and could lead astray even the most robust of gradient based optimization routines unless a near-perfect initial guess is given. Finally, a few of the best local minima have nearly identical values making the consistent identification of the global minimum extremely difficult.

Additional information regarding the calibration of the PSO method for this specific example problem can be found in the Appendix section of this paper as well as reference [29].

Optimal Low-Thrust Trajectory

Mindful of the results of this paper's preceding sections, a PSO algorithm was run with the following conditions: $N_p = 50$ particles, $\gamma_{stop} = 0.8$, $\boldsymbol{r}_{local} = \left[\frac{1}{20}, \frac{1}{16}N\right]^T$, and $j_{max} = 50$ iterations. This setup was repeated over 10 trials to obtain an average cost and standard deviation of $J_{best-median} = 0.1217 \pm 0.7$ % with a maximum value of $J_{best-max} = 0.1231$ and a minimum value of $J_{best-min} = 0.1201$. Table 1 shows a summary of this data with the values of the three terms of the fitness function displayed in addition to the value $j_{discovered}$ and $j_{\gamma=0.8}$. $j_{discovered}$ represents the iteration number when J_{best} was discovered where as $j_{\gamma=08}$ represents the iteration number when $\gamma = \gamma_{stop} = 0.8$. Note that, on occasion, $\gamma < \gamma_{stop}$ when $j = j_{max}$. In this case the PSO algorithm terminated before reaching γ_{stop}, as desired.

The 2-D PSO algorithm presented in this paper was relatively fast in comparison with other published results. Abraham et al. [17] published results that took approximately 8–16 h to achieve similar values of the fitness function using a 1-D version

Table 1 Summary of the maximum, minimum, and average values of the fitness function

	J_{best}	$e_{initial}$	Δm [kg]	TOF [tu]	$j_{discovered}$	$j_{\gamma=0.8}$	k	τ_{01}
$J_{best-max}$	0.1231	0.0017	64.85	28.27	15	37	141	0.3261
$J_{best-median}$	0.1217	0.0017	64.85	27.57	9	26	141	0.3257
$J_{best-min}$	0.1201	0.0019	64.66	26.73	4	N/A	743	0.0556

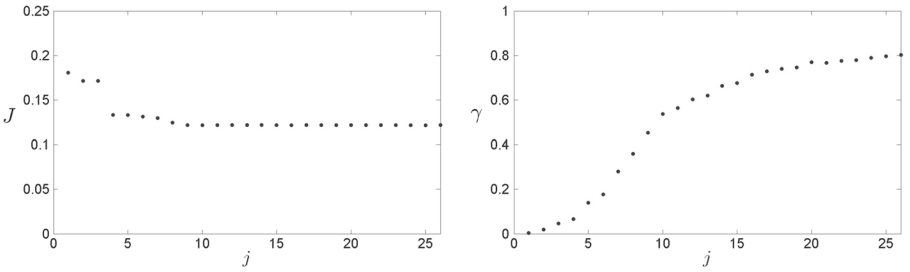

Fig. 5 *Left*: Cost vs. iteration number for $J_{best-median}$ case. *Right*: Convergence vs. iteration number for $J_{best-median}$ case

of the PSO algorithm. In this 2-D study, the average CPU run-time was 34 ± 9 min ($\pm 1\sigma$) with a maximum value of 46 min and a minimum value of 22 min. The majority of this variability is due to the fact that the PSO algorithm terminates in as little as $j = 26$ iterations and as many as $j = j_{max} = 50$ iterations. This 2-D PSO algorithm produces an order of magnitude decrease in the CPU run-time when compared with the 1-D result and represents a sizable boost in performance, usability, and practicality.

The evolution of J and γ as a function of j also agrees well with previous data. Referring to the left plot of Fig. 5 one will find a plot of J vs. j for the median case outlined in the center row of Table 1. Note that this data does not refute the veracity of Eq. A-2 although the variation and limited scope of the data limits a direct comparison. Conversely, the right plot of γ vs. j in Fig. 5 illustrates data with only minor variation that agrees well with the form predicted by Eq. A-1.

The optimal trajectory corresponding to $J_{best-median}$ has been plotted in Fig. 6. This trajectory is representative of a typical optimized trajectory discovered by the

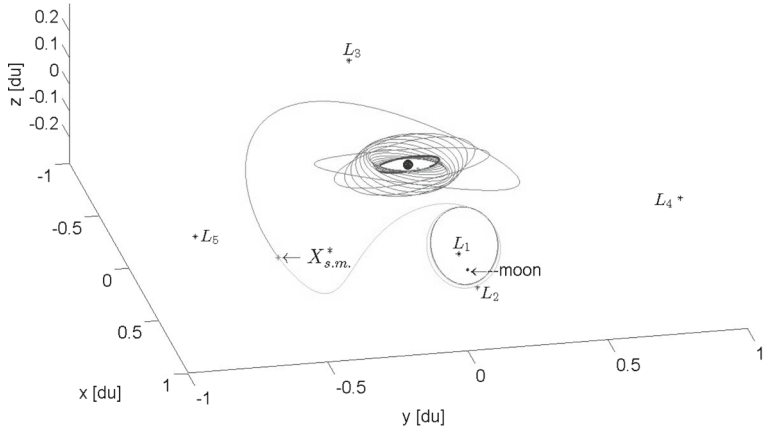

Fig. 6 Optimal Trajectory found using PSO. The spacecraft departs a high-altitude, Earth orbit (**in bold**) and follows a low thrust spiral until the manifold insertion point ($X^*_{s.m.}$) and low-thrust cutoff. Afterward, the spacecraft coasts in the manifold until insertion into the desired LPO

2-D PSO algorithm. Note the highly three dimensional nature of this transfer as the spacecraft becomes heavily influence by three-body dynamics. Figure 6 is plotted in the synodic, three-body reference system using the traditional non-dimensional units. The spacecraft begins its journey with the high-altitude, Earth orbit drawn in blue. The spacecraft then activates its low-thrust engine and tangentially thrusts itself into a spiral pattern of increasing altitude shown in red. The low-thrust spiral is greatly deformed as it enters a region of space that is clearly dominated by three-body effects. The spacecraft terminates this low-thrust arc at the optimized patch point, $\mathbf{X}^*_{s.m.}$, which is also a member of the stable manifold (and search space). The spacecraft ballistically coasts along the green trajectory and flows along the stable manifold towards the nominal halo orbit shown in blue. The spacecraft begins its journey with a wet mass of 1, 06485 [kg] and arrives at the nominal halo orbit 24 months later expending 64.85 [kg] of fuel and delivering 94 % of its original mass.

Conclusions

The preliminary optimization of a low-thrust transfer from an Earth orbit to an Earth-Moon, L_1 halo orbit has been demonstrated in this paper. Particle Swarm Optimization was used to prune a highly nonconvex search space that tends to trap traditional, gradient-based optimization algorithms in non-optimal, local minima. Indeed, since the topology of the fitness function is very rough, being filled with multiple extrema and discontinuities, a pruning method that is capable of obtaining a guess solution very near the optimal solution is warranted. Once the results of this PSO algorithm are obtained, they can be refined using a gradient based approach as demonstrated by Mingotti et al. [14–16].

It has been demonstrated, in this study, that utilizing a "local" version of the PSO algorithm can greatly improve its ability to simultaneously interrogate multiple minima without becoming trapped in a non-optimal, local minima. This result is in agreement with past research [13]. Based on the data gathered in this study, an appropriate choice of N_p is in the range of 200 to 400 particles and a suitable value of the convergence metric is $\gamma_{\text{stop}} \geq 075$. These values offer the best compromise between a through interrogation of the search space and prohibitive computational run-time. This study also demonstrated a vast improvement in computational run-time when utilizing the 2-D version of the PSO algorithm as opposed to the 1-D version [17] utilized elsewhere. Indeed, an order of magnitude improvement in the computational run-time has been demonstrated between the 2-D and 1-D versions of the PSO algorithm.

The optimal trajectory found using the 2-D, local PSO algorithm was highly three dimensional in nature—especially when three body effects are at their maximum. This trajectory began on a circular, high-altitude, Earth orbit. Three body effects are insignificant below this orbit so lower trajectories were not studied in this research. Instead, a connecting low-thrust spiral can be assumed. Overall, this technique is general enough to be applied to many other Lagrange point orbits that utilize a (initially) low-eccentricity, low-thrust transfer.

Appendix

This appendix describes the methods used to appropreately choose the various optimization paramaters of the PSO method as applied to the example problem. The same technique can be applied to other problems with little modification.

Sizing N_p

The PSO algorithm was run multiple times with $r_{\text{local}} = \left[\frac{1}{20}, \frac{1}{16}\right]^T$ and $j_{\max} = 15$. Each data point in Table 2 was averaged over 10 identical trials with only the difference being the initial χ and ω values which are chosen randomly. Referring to Fig. 7, one will note that the average value of the cost function, J_{best} decreases as a function of N_{p}, as expected. Initially, the decrease is substantial but the amount of increase diminishes as each additional particle is added to the system—especially for values of $N_{\text{p}} > 400$. Based on this data it seems reasonable to conclude that the ideal range of N_{p} lies somewhere between 200 and 400 particles. A PSO algorithm with less particles experiences a huge amount of variability and offers a relatively poor average J_{best} (i.e. the $N_{\text{p}} = 50$ data point). Conversely, a PSO algorithm utilizing more than 400 particles will consume copious amounts of CPU time without a significant decrease in the average J_{best} identified.

Sizing j_{max} via the Convergence Metric, γ

In the previous section, a method for determining the appropriate value of N_{p} was presented. This section develops a new metric that is used to determine an appropriate number of iterations to use when terminating the PSO algorithm. In general, there are two ways to accomplish this task. The first method is to simply select a value for j_{max} before running the PSO algorithm. An appropriate value of j_{max} is highly problem dependent and relies greatly on the experience and intuition of the researcher [13, 28]. A new metric, γ, is proposed, in this paper, to replace j_{max} as the trigger that terminates the PSO algorithm.

The top plot in Fig. 8 displays convergence (γ) as a function of PSO iteration (j) for a system of 50 particles. Each point is averaged over 10 (identical) data trials and

Table 2 Cost, convergence, and CPU time for $j_{max} = 15$ iterations and various number of particles (N_p)

N_p	Avg. best cost, J_{best}	Avg. final convergence, γ_{final}	Avg. CPU time [min]
50	$0.1357 \pm 88\ \%$	$0.71 \pm 14\ \%$	$9.0 \pm 6\ \%$
100	$0.1273 \pm 28\ \%$	$0.69 \pm 7\ \%$	$205 \pm 4\ \%$
200	$0.1260 \pm 28\ \%$	$0.60 \pm 10\ \%$	$428 \pm 3\ \%$
400	$0.1235 \pm 24\ \%$	$0.61 \pm 11\ \%$	$844 \pm 3\ \%$
800	$0.1221 \pm 14\ \%$	$0.61 \pm 11\ \%$	$160.0 \pm 4\ \%$

Each entry consists of the average of 10 identical trials with error bounds of $1 \pm \sigma$ standard deviation

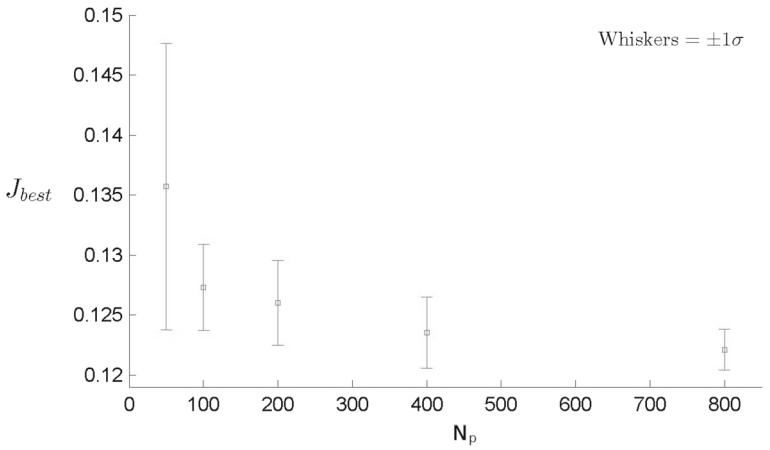

Fig. 7 Values of J_{best} as a function of N_p for a fixed value of $j_{max} = 15$ iterations. Data taken from Table 2

is plotted with $\pm 1\sigma$ error bars. Note that the convergence seems to initially increase exponentially with low values of j but then logarithmically approaches an asymptote for large values of j. This seems to indicate that, for large numbers of PSO iterations, an increasingly diminishing rate of marginal convergence is achieved. This relationship has been successfully fitted to the following curve using MATLAB's curve fitting toolbox:

$$c \exp\left(\frac{-b}{(j+a)^3}\right) \tag{A-1}$$

with the fitted values of the constants being $a = 7$, $b = 2421$, and $c = 08671$ with a "goodness of fit" measured as $R^2 = 09957$. In general, $\gamma_{max} \leq c \leq 1$ since the value of Eq. A-1 is c when $j = \infty$. Of course theory dictates that $\gamma_{max} \to 1$ as $j_{max} \to \infty$. In practice, however, this is not always the case since a small number of particles can become trapped in cycles that oscillate them between the neighborhood of one local minima and another. Also, on occasion, a particle will oscillate for a very long time between a personal best and global best value that are in opposite directions.

The bottom plot in Fig. 8 displays the cost (J) as a function of PSO iteration (j) for the same system of 50 particles and 10 trials. The plot describes the evolution of the best value of the cost function. Note the exponential decay displayed in the cost function and fitted to the following curve:

$$ae^{-bj} + c \tag{A-2}$$

where the values of the fitted constants are $a = 0174$, $b = 02344$, and $c = 01314$ with $R^2 = 09879$. As $j \to \infty$ the limit of Eq. A-2

$$\lim_{j \to \infty} (ae^{-bj} + c) = c \tag{A-3}$$

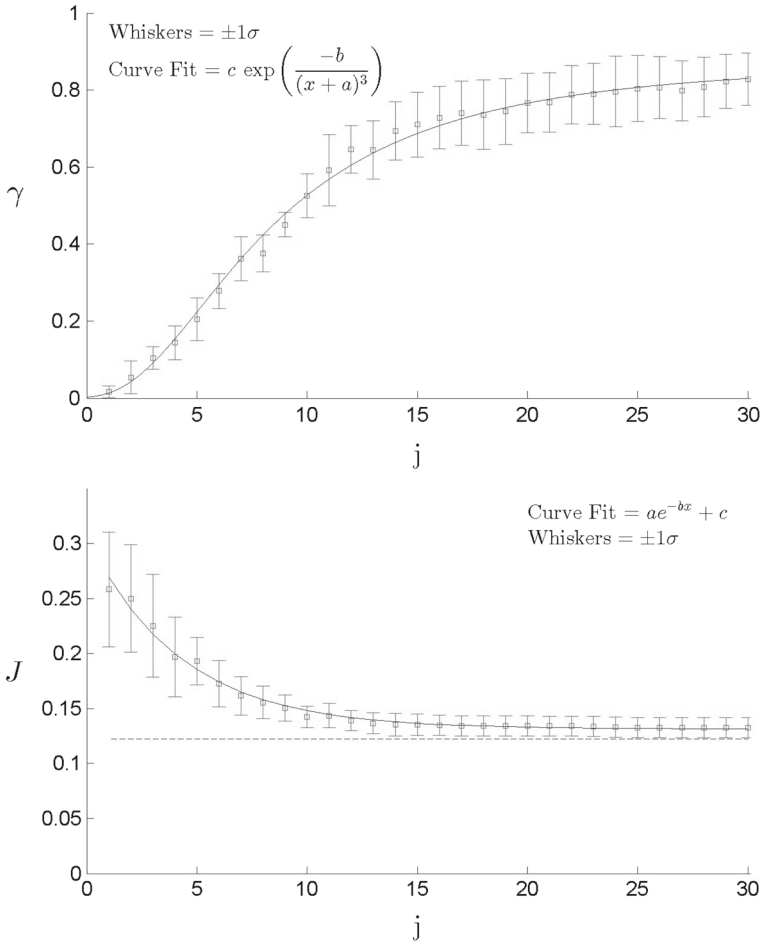

Fig. 8 *Top*: Convergence (γ) as a function of iteration number (j). *Bottom*: Cost (J) as a function of iteration number (j)

is equal to 01314 which is a greater value than $J_{best} = 01225$. J_{best} is indicated by the dashed line in the bottom plot of Fig. 8, where J_{best} represents the lowest value of the cost function encountered by any particle from any of the 10 trials. The difference between J_{best} and c is reflected in the variability of the last few data points in the plot where $J = 01327 \pm 00092(\pm 1\sigma)$ or a range of $J(j = 30)_{range} = [01225 \leftrightarrow 01520]$.

Based on the data found in Fig. 8 it seems reasonable to conclude that a value of $\gamma \geq 075$ is a suitable choice of terminal convergence to end the PSO algorithm. This roughly correlates to $j = [15 \leftrightarrow 20]$ iterations and places the value of the cost function very near its asymptotic limit without wasting CPU time on additional and unproductive iterations. It therefore seems reasonable to use a value of $\gamma_{stop} = 08$ as a conservative termination metric for the PSO algorithm. Inserting this value into Eq. A-1 and solving for j yields $j_{stop} = 24$ iterations on average.

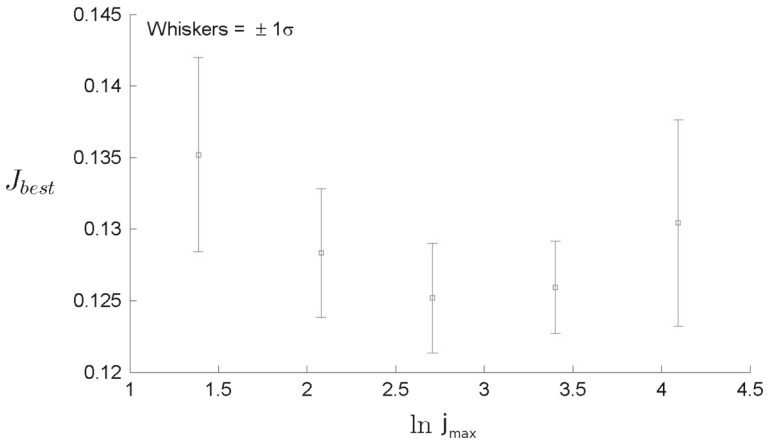

Fig. 9 Plot of J_{best} as a function of $\ln(j_{max})$ for a fixed value of 3,000 integrations per point. Each point was averaged over 10 identical trials with $\pm 1\sigma$ error bounds drawn

Effect of the Product $N_p j_{max}$ on the Optimal Solution

Since the most significant issue with PSO is the CPU run-time, and CPU run-time is proportional to the product $N_p j_{max}$, it is reasonable to investigate the trade between N_p and j_{max} while holding the number of integrated trajectories (and CPU run-time) fixed. This data is presented in Fig. 9 which plots J_{best} vs. the log of N_p (semi-log plot was chosen for illustrative purposes) given a fixed value of the product $N_p j_{max} = 3000$ evaluations (trajectories integrated according to Eq. 1). Values of j_{max} included [4, 8, 15, 30, 60]. Note that the lowest value of J_{best} was captured when $N_p = 200$ and $j_{max} = 15$. This roughly agrees with the recommended values of N_p and j_{max} (or γ_{stop}) developed in the two previous sections of this paper.

Effect of r_{local} on the Optimal Solution

The effect of the size of r_{local} on the convergence and cost is best captured in Fig. 10. These plots are produced by multiplying r_{local} by a scaling factor. The scaling factors

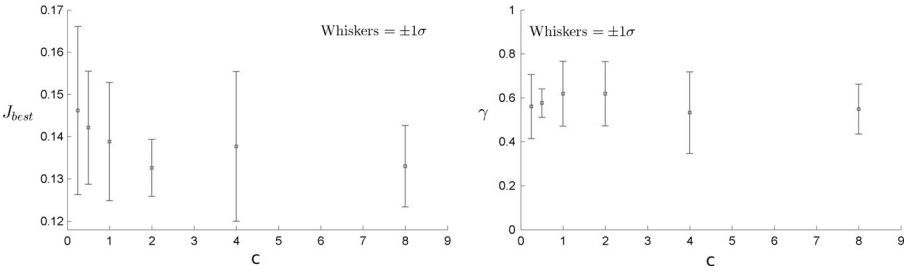

Fig. 10 *Left*: Cost as a function of radius. *Right*: Convergence as a function of radius

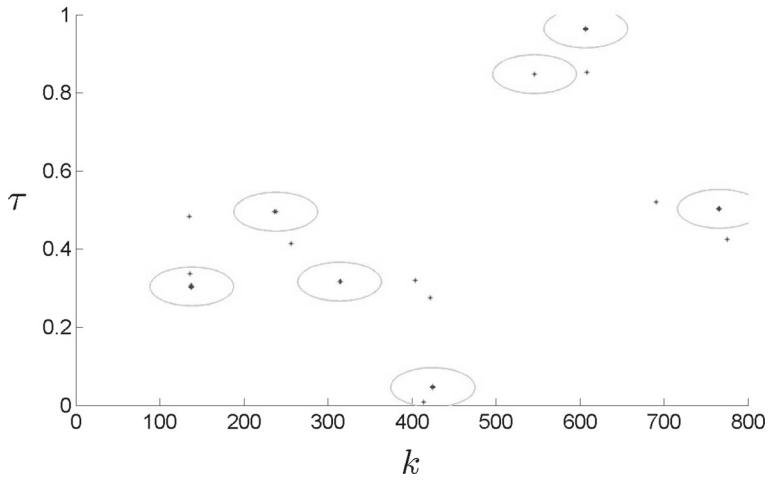

Fig. 11 A map of the final positions of 50 particles after 15 iterations. The ellipses are drawn around local swarms of two or more particles indicating a local minimum. The size of the ellipse reflects the dimensions of $r_{local} = \left[\frac{1}{20}, \frac{1}{16}N\right]$

used in this study are $\|r\| = \left[\frac{1}{4}, \frac{1}{2}, 1, 2, 4, 8\right]$ where $r_{resized} = \|r\| r_{local}$. Each data point was averaged over 10 identical trials and plotted with $\pm 1\sigma$ error bars using 50 particles and 15 iterations. Note that the convergence is not significantly affected by a change in the magnitude of r_{local}, as is evidenced by the fact that any data point lies within the error bars of any of the remaining points. Likewise, the cost typically is not affected by the magnitude of r_{local} except for extremely small values of $\|r\|$, as evidenced by the left plot in Fig. 10; in which case the size of a particle's local range of communication is so small it is effectively "blind and deaf."

Figure 11 displays a map of local minima identified by the PSO algorithm as plotted in $k\tau$-space. In this particular example $\|r\| = 1$ and seven local minima are identified. A green ellipse of size $r_{local} = \left[\frac{1}{20}, \frac{1}{16}N\right]_T$ is drawn around each local minima to illustrate the communication limit between a particle at the local minima and any nearby particle it can communicate with. This plot illustrates the ability of the local PSO algorithm to simultaneously interrogate multiple local minima instead of being limited to only one minimum using the traditional PSO algorithm.

References

1. Angelopoulos, V.: The ARTEMIS mission. Space Sci. Rev., 1–23 (2010). doi:10.1007/s11214-010-9687-2
2. Sweetser, T.H., Broschart, S.B., Angelopoulos, V., Whiffen, G.J., Folta, D.C., Chung, M.-K., Hatch, S.J., Woodard, M.A.: ARTEMIS mission design. Space Sci. Rev. **165**, 27–57 (2012). doi:10.1007/s11214-012-9869-1
3. Farquhar, R.W.: A halo-orbit lunar station. Astronaut. Aeronaut. **10**, 59–63 (1972)

4. Davis, E.: Transfers to the earth-moon l3 halo orbits. AAS/AIAA Astrodynamics Specialist Conference, Minneapolis, MN (2012). doi:10.2514/6.2012-4593
5. Martin, L.: Early human L2-farside missions, Tech. Rep., Lockheed Martin. http://www.lockheedmartin.com/content/dam/lockheed/data/space/documents/orion/LMFarsideWhitepaperFinal.pdf (2010)
6. Heppenheimer, T.A.: Steps toward space colonization: colony location and transfer trajectories. J. Spacecr. Rocket. **15**, 305–312 (1978). doi:10.2514/3.28013
7. Koon, W.S., Lo, M.W., Marsden, J.E., Ross, S.D.: Heteroclinic connections between periodic orbits and resonance transistions in celestial mechanics. Chaos **10**(2) (2000)
8. Vaquero, M., Howell, K.C.: Design of transfer trajectories between resonant orbits in the earth-moon restricted problem. Acta Astronaut. **94**, 302–317 (2014). doi:10.1016/j.actaastro.2013.05.006
9. Davis, K.E., Anderson, R.L., Scheeres, D.J., Born, G.H.: Optimal transfers between unstable periodic orbits using invariant manifolds. Celest. Mech. Dyn. Astron. **109**, 241–264 (2011). doi:10.1007/s10569-010-9327-x
10. Ozimek, M., Howell, K.: Low-thrust transfers in the earth-moon system, including applications to libration point orbits. J. Guid. Control. Dyn. **33** (2010). doi:10.2514/1.43179
11. Grebow, D.J., Ozimek, M.T., Howell, K.C.: Advanced modeling of optimal low-thrust lunar pole-sitter trajectories. Acta Astronautica (2010). doi:10.1016/j.actaastro.2010.01.024
12. Caillau, J., Daoud, B., Gergaud, J.: Minimum fuel control of the planar circular restricted three-body problem. Celest. Mech. Dyn. Astron., 138–150 (2012). doi:10.1007/s10569- 012-9443-x
13. Conway, B.A. (ed.): Spacecraft Trajectory Optimization. Cambridge (2010)
14. Mingotti, G., Topputo, F., Bernelli-Zazzera, F.: Combined Optimal Low-Thrust and Stable-Manifold Trajectories to the Earth-Moon Halo Orbits. Am. Inst. Phys. Conf. Proc. Belbruno, E. (ed). doi:10.1063/1.2710047 (2007)
15. Mingotti, G., Topputo, F., Bernelli-Zazzera, F.: Numerical Methods to Design Low-Energy, Low-Thrust Sun-Perturbed Transfers to the Moon. In: Proceedings of 49th Isreal Annual Conference on Aerospace Sciences, Tel Aviv-Haifa, Isreal, pp. 1–14 (2009)
16. Mingotti, G., Topputo, F., Bernelli-Zazzera, F.: Transfers to distant periodic orbits around the Moon via their invariant manifolds. Acta Astronaut. **79**, 20–32 (2012). doi:10.1016/j.actaastro.2012.04.022
17. Abraham, A.J., Spencer, D.B., Hart, T.J.: Optimization of Preliminary Low-Thrust Trajectories from GEO-Energy Orbits to Earth-Moon, L1 Lagrange Point Orbits Using Particle Swarm Optimization, No. 13-925, Hilton Head, SC, AAS Astrodynamics Specialist Conference (2013)
18. Scheel, W.A., Conway, B.A.: Optimization of very-low-thrust, many revolution spacecraft trajectories. J. Guid. Control. Dyn. **17** (1994). doi:10.2514/3.21331
19. Herman, A.L., Spencer, D.B.: Optimal, low-thrust earth-orbit transfers using higher-order collocation methods. J. Guid. Control. Dyn. **25** (2002)
20. Redding, D.C., Breakwell, J.V.: Optimal low-thrust transfers to synchronous orbit. J. Guid. Control. Dyn. **7**(2) (1984). doi:10.2514/3.8560
21. Redding, D.C.: Highly efficent, very low-thrust transfer to geosynchronous orbit: exact and approximate solutions. J. Guid. Control. Dyn. **7** (1984). doi:10.2514/3.8559
22. Szebehely, V.: Theory of Orbits: the Restricted Problem of Three Bodies. Academic Press Inc., New York (1967)
23. Pavlak, T.A.: Mission Design Applications in the Earth-Moon System: Transfer Trajectories and Stationkeeping, Master's thesis, School of Aeronautics and Astronautics. Purdue University, West Lafayette, Indiana (2010)
24. Gomez, G., Masdemont, J., Simo, C.: Quasihalo orbits associated with libration points. J. Astronaut. Sci. **46**(2), 135–176 (1998)
25. Farquhar, R.W., Kamel, A.A.: Quasi-periodic orbit about the translunar libration point. Celest. Mech. **7**, 458–473 (1973)
26. Farquhar, R.W.: Lunar communications with libration-point satellites. J. Spacecr. Rocket. **4**(10), 1383–1384 (1967). doi:10.2514/3.29095
27. Perko, L.: Equations, differential and dynamical systems. 175 Fifth Avenue, New York, NY, 10010, USA: Springer 3rd ed. (2001)
28. Pontani, M., Conway, B.A.: Particle swarm optimization applied to space trajectories. J. Guid. Control. Dyn. **33** (2010). doi:10.2514/1.48475
29. Abraham, A.J.: Particle Swarm Optimization of Low-Thrust, Geocentric-to-Halo-Orbit Transfers. PhD thesis. Lehigh University, Mechanical Engineering & Mechanics Department (2014)