AAS 25-293

SATELLITE SERVICING IN CISLUNAR NETWORKS WITH DESIGN AND SCHEDULING USING A GENETIC ALGORITHM

Cody Waldecker* **and Kathleen Howell†**

Amid the increasing interest in In-Orbit Servicing, Assembly, and Manufacturing (ISAM) and the renewed focus on lunar exploration exemplified by the Artemis missions, the need for a robust ISAM design framework in cislunar space has become evident. This investigation addresses critical considerations and strategies for designing networks of servicer and customer satellites within the Circular Restricted Three-Body Problem (CR3BP), with validation in an Earth-Moon-Sun Ephemeris model. This paper introduces a two-part Genetic Algorithm (GA) scheme designed to optimize the scheduling and transfer design for on-orbit servicing networks in the cislunar vicinity. While employing a one-dimensional chromosome representation, the GA effectively incorporates servicer and customer orbits, phasing revolutions, and optional orbits for a comprehensive trade space analysis. The algorithm leverages selection, crossover operators, and elitism to determine optimal servicer satellite schedules across diverse network configurations. In a case study involving L_1 and L_2 Lyapunov orbits, the GA significantly outperforms a traditional Auction Algorithm by addressing downrange phasing requirements. The GA is further enhanced with a multi-objective fitness score that considers total network ΔV, the number of servicers, and customer orbit benefits. This fitness evaluation reveals that orbit symmetry, particularly in terms of orbit family and an energy gap (Jacobi constant), is the most significant impact on ΔV requirements for network traversal. The robustness of the proposed GA is demonstrated through its application to a diverse set of networks, including planar, non-planar, and resonant orbits. The algorithm proves versatile and effective in optimizing a wide range of orbital configurations; thus, it serves as a very useful tool for the design and initial selection of orbits for on-orbit servicing networks in cislunar space.

INTRODUCTION

Recently, interest in satellite servicing has increased significantly with the investments in early-stage technologies to advance the development of commercial refueling and servicing satellites. The ability to refuel, service, and upgrade satellites that have already been deployed presents a strategic advantage and presents the opportunity for commercial satellite operators to extend satellite lifetimes at a lower cost than deploying new spacecraft. Most rising satellite servicing/refueling companies including OrbitFab, Astroscale, and Northrop Grumman's MEV program are targeting customer satellites in Geosynchronous orbits (GEO) due to high asset values. However, the translation of satellite servicing capabilities from GEO to the cislunar vicinity is essential due to the increasing interest in lunar exploration and development as well as staging location for missions beyond lunar orbit. As NASA prepares for lunar missions and the establishment the lunar gateway facility, the ability to service, assemble, and maintain satellites in the lunar vicinity is an important step towards sustainability. In GEO, satellite servicing is a cost-effective solution for extending the life of satellites, enhancing their capabilities, and ensuring operational continuity. Applying these capabilities in the cislunar region will support a wide range of activities, including scientific research, communication infrastructure, and navigation systems, thereby facilitating activities in cislunar space. This investigation focuses on the optimal design of servicing schedules for predefined servicer/customer orbits in the cislunar vicinity and the optimal selection process when orbits are not preselected.

*PhD Student, School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47906, USA.
†Hsu Lo Distinguished Professor of Aeronautics and Astronautics, School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN 47906, USA.

The servicing framework consists of servicing hubs placed in various orbits in the lunar vicinity that are tasked with servicing customer satellites in various other orbits. Each servicing hub acts as a home base to another, more agile satellite, termed the servicer. In a mission scenario, the servicer spacecraft departs the hub, transfers to any number of the customer satellites, accomplishes desired services, and returns to the original servicing hub for resupply/refuel.

This work focuses on the optimal scheduling of servicers and does not delve extensively into the details of the transfer design process as they are detailed in Waldecker and Howell.[1] The original Traveling Salesman Problem (TSP)[2] is a well-researched combinatorial optimization problem where given a set of cities and the distances between each pair of cities, the goal is to determine the shortest possible route that visits each city exactly once and returns to the starting city. When the distance between two targets varies with time, as in the case with satellite servicing, the problem transforms into a Dynamic Traveling Salesman Problem (DTSP).[3] Numerous studies have been conducted in this area, focusing on methods for generating optimal solutions and evaluating the upper and lower bounds.[4–7] Similar problems exist, outside of a servicing framework, where one or more satellites seek to visit different target orbits, such as the design of the Voyager 2 mission to visit Io, Europa, and Ganymede;[8] such is regarded as the Orbit Traveling Salesman Problem (OTSP).

Due to the significant variation in phase between different orbits at different visit times, producing an optimal solution by directly formulating an optimization model with the order, time, and control law as unknown variables is challenging. The approach here decomposes the problem into two steps, a transfer generation step that uses the invariant manifolds for orbits in the Circular Restricted Three-Body Problem (CR3BP) and a Genetic Algorithm (GA) step that involves a linear interpolation across those solutions to determine a locally optimal servicing schedule. Several authors have presented similar multi-step methods for decomposing the OTSP for use in LEO and GEO,[9–12] but application to the lunar regime presents a unique set of challenges. Unlike other authors who make two-body assumptions or use orbit energy level to estimate fixed transfer costs, the employed strategy leverages dynamical structures in the region to accurately model transfer costs and account for phasing requirements.

BACKGROUND

Dynamics Model

The Circular Restricted Three-Body Problem (CR3BP) is a classical problem in celestial mechanics that models the motion of a small body (such as a satellite) under the gravitational influence of two larger bodies (such as the Earth and the Moon), assumed to be in circular orbits around their common center of mass. In this model, illustrated in Fig. 1, the two spherically symmetric gravitational bodies, assumed to be the Earth (P1) and the Moon (P2) in this investigation, form the primary system as they move in circular orbits around their common barycenter, B. The relative sizes of the primaries are represented by the mass ratio $\mu = \frac{m_2}{m_1+m_2}$. The equations of motion are nondimentionalized for numerical stability in application. The characteristic length, l^*, represents the average distance between the Earth and the Moon, the characteristic time, t^*, is defined such that the nondimensional gravitational constant is equal to one, and the characteristic mass, m^*, denotes the sum of the masses of the primaries. The nondimensional position and velocity vectors of P_3 with respect to the barycenter comprise the six-dimensional vector $\bar{r} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$ in a reference frame that rotates with the motion of P_2, denoted by the dashed line frame in Fig. 1. The scalar CR3BP equations of motion in the rotating frame are

$$\frac{\delta U}{\delta x} = \ddot{x} - 2\dot{y}, \ \frac{\delta U}{\delta y} = \ddot{y} + 2\dot{x}, \ \frac{\delta U}{\delta z} = \ddot{z} \tag{1}$$

where the pseudo-potential U is defined as,

$$U = \frac{1}{2}(x^2 + y^2) + \frac{1-\mu}{d} + \frac{\mu}{r} \tag{2}$$

The distances between P_3 and first and second primary body are defined as $d = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$, respectively.

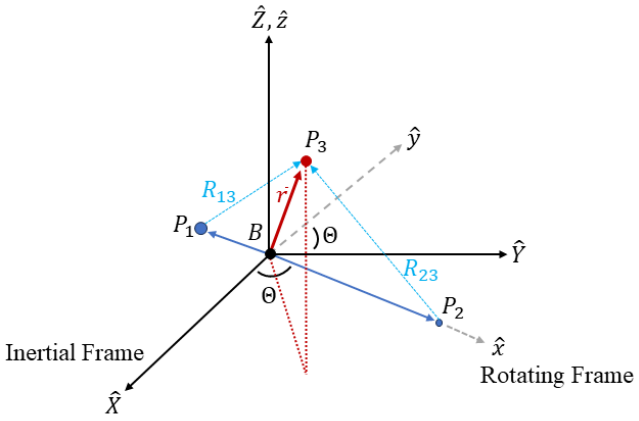


Figure 1: Three-Body Rotating Frame

The CR3BP equations of motion yield one integral of motion, i.e., the Jacobi Constant (JC). The JC is related to orbital energy as observed in the rotating frame and remains constant along any ballistic arc. The JC is comprised of a pseudo-potential term and the square of the rotating velocity, evaluated as

$$JC = 2U - (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) \tag{3}$$

The Jacobi Constant offers insight into the energy level for the orbits, providing a foundation for comparing transfer costs and bounded regions in space. Finally, the CR3BP gives rise to the Lagrange points or libration points, $L_1 - L_5$, that are point solutions where the velocity and acceleration of P_3 relative to the rotating frame are equal to zero.

Transfer Generation

Transfer design throughout the servicing network relies on the use of manifold arc trajectories that asymptotically depart from and arrive to unstable periodic orbits. In the CR3BP, manifold arc trajectories refer to paths that are naturally associated with the stable and unstable manifolds of periodic orbits. The manifolds are key tools for understanding the dynamics near periodic orbits and are useful for designing efficient transfer trajectories between orbits. The stable manifold associated with a periodic orbit consists of trajectories that asymptotically approaches the periodic orbit as time progresses forward. A spacecraft on a stable manifold trajectory naturally moves towards the periodic orbit without requiring additional propulsion. Conversely, the unstable manifold consists of trajectories that move away from the periodic orbit as time progresses forward. A spacecraft on an unstable manifold trajectory naturally departs from the periodic orbit. The segments along the stable and unstable manifolds that extend from or toward a periodic orbit are manifold arc trajectories. These arcs are leveraged to design transfers between different orbits or between a periodic orbit and another destination in space.

By carefully selecting the initial conditions along an unstable manifold of one periodic orbit and a corresponding point on the stable manifold trajectory of another orbit, it is possible to design a transfer trajectory that connects the two orbits. This technique leverages the natural dynamics in the CR3BP, potentially minimizing propellant consumption. Using manifold arcs can significantly reduce the ΔV (velocity change) required for orbital transfers, as the spacecraft exploits the natural gravitational dynamics. In an ideal transfer scenario, there exists an unstable manifold trajectory arc that departs from the original orbit and perfectly links with a stable manifold trajectory arc flowing into a desired target orbit. In this scenario, assuming that the original and target orbits exist at the same value of JC, a free transfer, termed a heteroclinic connection, exists between the two orbits.[13]

As described in more detail by Waldecker and Howell,[1] the first step in the network optimization process involves estimating the ΔV and time of flight (TOF) requirements to transfer between orbits throughout the network. Using the selected initial orbits, transfers across the network are efficiently identified by generating manifold arc trajectories and recording the state values at each Poincaré map crossing. This process is repeated for orbits near the original energy level, allowing the velocity discontinuities to be summarized as a heatmap. Each heatmap offers an estimate of the ΔV required to transfer between two orbits, highlighting the initial geometries that may enable the most efficient ΔV transfers. With these preliminary estimates, the actual transfer trajectories are refined and converged in the CR3BP using a differential corrections scheme, ensuring accuracy and feasibility. As visualized in Fig. 2, each cell of the heatmap represents various transfer options between two desired orbits. Then, each of the initial guesses from the heatmap are continued

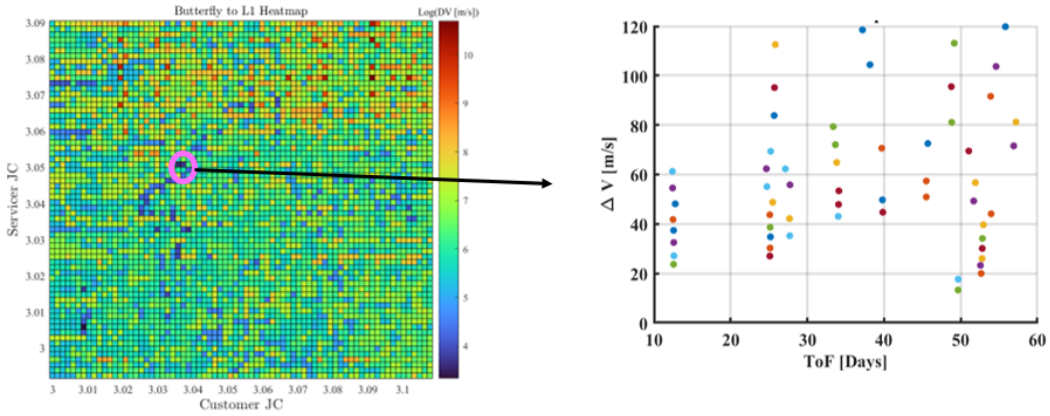


Figure 2: Example heatmap of transfers from a butterfly orbit to an L_1 halo orbit and ΔV vs ToF transfer options for selected energy configuration.

into a family of transfers that spans a wide ToF window, providing one strategy for overcoming the strict phasing requirements during ISAM operations. Convergence of a wide family of transfers is a numerically intensive process, therefore, a linear estimate of the ΔV required for a fully phased transfer is employed. For each desired transfer, the ΔV from the heatmap and the transfer time are combined to construct the cost for spacecraft i to transfer to target j, e.g.,

$$b_{ij} = \Delta V_{ij} + .15\Delta t_{error} \tag{4}$$

As opposed to other approaches that accommodate only the ΔV or the transfer time, this scheme incorporates the phasing error between the initial transfer and the target spacecraft. The non-dimensional timing error is multiplied by a weighting value equal to $.15\frac{km}{s}$ to weight the cost of altering the original transfer time in terms of ΔV, therefore providing a total cost b_{ij} in $\frac{km}{s}$. The weighting value of .15 is selected because it aligns well with previous observations, but it may be altered to match the needs of the designer.

GENETIC ALGORITHM

Overview

A Genetic Algorithm (GA) is an optimization technique inspired by the principles of natural selection and genetics, often used to solve complex problems with large search spaces. The algorithm initiates by randomly generating an initial population of candidate solutions, labelled individuals or chromosomes. Each chromosome represents a possible solution to the problem, typically encoded as a binary string. A fitness function is then defined to evaluate each chromosome for solving the problem, with the fitness score indicating the quality of the solution.

Once the population is initialized, the algorithm enters a loop of selection, crossover, and mutation to evolve the population over successive generations. During the selection phase, individuals are identified based on their fitness scores, those with higher fitness more likely to be selected. The selection process ensures that the best solutions offer a higher chance of passing their genetic material to the next generation. Common selection methods include roulette wheel selection, tournament selection, and rank-based selection.

In the crossover phase, selected individuals (parents) are paired to produce new offspring through a recombination process. During crossover, segments of the parents' chromosomes are exchanged to create new combinations of genes, that represent potential solutions. Various crossover techniques, such as single-point crossover, multi-point crossover, and uniform crossover, are options depending on the problem requirements. Mutation then introduces small random changes to individual chromosomes, ensuring genetic diversity within the population and allowing the algorithm to explore new areas of the solution space. The mutation rate is usually kept low to maintain a balance between exploration and exploitation.

The newly created offspring, along with some of the best individuals from the previous generation, form the next generation of the population. This process repeats for a set number of generations or until a satisfactory solution emerges. Over time, the population is expected to converge towards an optimal or near-optimal solution as the genetic algorithm explores the search space. The key components of the GA include population initialization, fitness evaluation, selection, crossover, and mutation that work together to iteratively improve the quality of solutions,rendering a tool for solving complex optimization problems.

Implementation

As opposed to the original Traveling Salesman Problem (TSP), the Orbit Traveling Salesman Problem (OTSP) when implemented near the Moon, presents a distinctive set of challenges. Considering N nodes on a graph, each pair of nodes is defined with distance $d_{i,j}$ between them. In a standard TSP, the goal is to determine an ordering of the nodes π that minimizes the total traveled distance

$$minJ_{\pi} = \sum_{i=1}^{N-1} d_{\pi_i,\pi_{i+1}} \tag{5}$$

by a single salesman. When implementing the OTSP, many of the standard assumptions in the TSP are broken. The most impactful difference between the TSP and OTSP is the time dependency inherent in transferring between orbits, especially in the case of ISAM where rendezvous is required. The time dependency directly impacts the ΔV requirements for a single transfer, where the ΔV required to transfer between two orbits at one epoch may or may not be the same at a different epoch,

$$\Delta V_{i,j}(t_1) \stackrel{?}{=} \Delta V_{i,j}(t_2) \tag{6}$$

This time dependency leads to breaking the path dependency assumption in the TSP. Specifically, the cost to transfer to a customer orbit, may not necessarily be more expensive when an intermediate stop is included, i.e.,

$$\Delta V_{1,2} + \Delta V_{2,3} \lessgtr \Delta V_{1,3} \tag{7}$$

In addition, there are also an infinite number of transfer paths that exist between two orbits given an initial epoch. To simplify the optimization process, a two-step solution is introduced comprised of one step that determines the minimum cost transfer between two orbits and a second GA process that determines the servicing order. As visualized in Fig. 3, given an initial set of servicer and customer orbits, the GA initializes the first generation of n random chromosomes representing various servicing schedules. Then, each member of the generation is individually evaluated for its fitness score. The chromosome is divided into each pair of genes representing orbits and the required ΔV is estimated using previously determined transfer options. The transfer option that minimizes the ΔV consistent with Eqn.(4) is determined for the given epoch and passed back to the evaluation process. Once the fitness score for all the chromosomes in the generation is determined, several crossover operations are applied to manipulate the next generation. This evaluation and crossover process is repeated for m generations or until a satisfactorily optimal solution is determined.

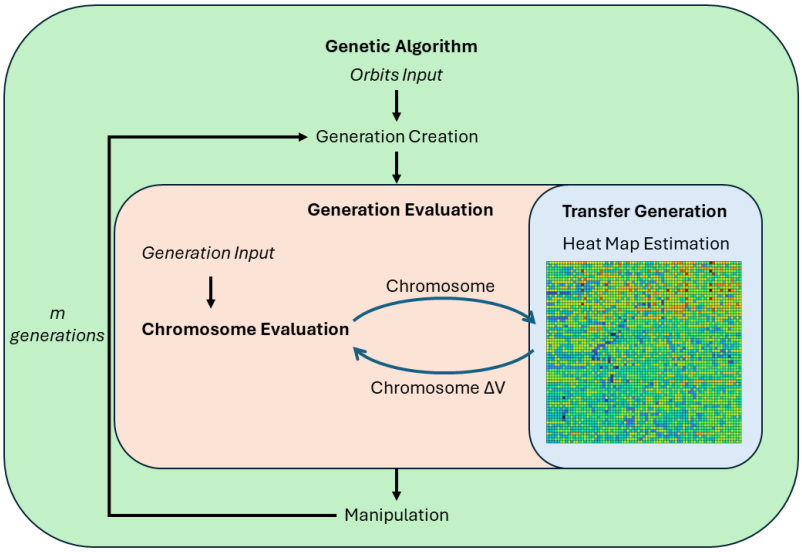


Figure 3: GA systems with heatmap transfer evaluation

Chromosome Representation To apply a Genetic Algorithm (GA) to any optimization problem, it is essential to represent solutions as valid chromosomes, ensuring that the crossover operation yields valid chromosomes as well. For the Multi-Objective Traveling Salesman Problem (MTSP), three main chromosome representation methods are commonly used: the one-chromosome method,[14] the two-chromosome method,[15] and the two-part chromosome method.[16,17] The representation here involves one-chromosome with artificial depots, similar to that of Al-Furhud,[18] but includes the addition of three components that allow for the incorporation of optional customer orbits and phasing revolutions. A sample chromosome for two servicer spacecraft and six customer satellites, $n_{serv} = 2$ and $n_{cust} = 6$, is shown in Fig. 4. In the sample: each integer value less than n_{cust}, but greater than zero, represents a customer orbit (Blue); each value greater than n_{cust}, but less than $n_{cust} + n_{serv} + 1$, represents a servicer (Red); each zero (Black) represents the servicer remaining in the previous orbit for one additional revolution to alter the phasing with future orbits; each value greater than $n_{cust} + n_{serv}$ represents an optional customer orbit that may or may not be included in the final solution (Purple); and all values following the termination genome, -1, are not included in the schedule (Gray). Should a chromosome represent *not* using/servicing the optional orbit, it is placed after this termination genome. In this chromosome, [7, 2, 3, 0, 6, 8, 0, 9, 5, 4, 0, 1], there are 9 genes including

7	2	3	0	6	8	0	9	5	4	0	1	-1	0	0

Figure 4: Example Chromosome: Red - servicer start, Blue - customer orbit, Black - phasing rev, Purple - optional customer, Grey - terminated

an artificial depot, 8, and an optional orbit 9. This sequence represents the tour of two servicers, 7 and 8, following the path of $7 \to 2 \to 3 \to 6 \to 7$ and $8 \to 9 \to 5 \to 4 \to 1 \to 8$ respectively. Servicer 1 starts in orbit 7, visits customers 2 and 3, conducts one phasing revolution in orbit 3, visits customer 6, and then returns to the original servicing hub in orbit 7. Servicer 2 starts in orbit 8, immediately remains there for one phasing revolution, visits customers 9, 5, and 4, conducts a phasing revolution in orbit 4, visits customer 1, and then returns to its original servicing hub in orbit 8. This chromosome representation enables the ability to model complex sequences involving optional orbits and phasing revolutions better than a typical binary representation. It also ensures that crossover methods produce valid chromosomes.

Chromosome Manipulation After evaluating a generation in the Genetic Algorithm (GA), several manipulation techniques are applied to produce the next generation of solutions. The first step is selection, where individuals from the current population are chosen based on their fitness scores. This process favors individuals with higher fitness, giving them a better chance of contributing their genetic material to the offspring. Rank-based selection is employed to ensure that better-performing solutions are more likely to be selected. Following selection, the crossover, or recombination, operation is performed. Crossover combines the genetic information from two parent individuals to create new offspring, allowing the algorithm to explore new areas of the solution space. Mutation is another key manipulation technique, where small random changes are introduced to individual chromosomes. By altering one or more genes, mutation aids in maintaining genetic diversity within the population and prevents the algorithm from converging prematurely to suboptimal solutions. Finally, elitism is incorporated into the GA to preserve the best-performing individuals from the current generation. By carrying over these elite solutions directly to the next generation without any modifications, the algorithm ensures that the best traits are retained and not lost during the evolutionary process.

Selection: For the selection and ranking of the chromosomes, the fitness score for each individual is evaluated. Since the needs of each transfer designer are different, priorities on the weighting of fitness score components may also vary. For this investigation, the ΔV estimate from the transfer generation process plays a large role in the fitness score evaluation. The GA proposed here is used to design optimal servicer schedules and, later, for the identification of optimal orbits for servicing networks. Since the search space for potential servicer/customer orbits in the cislunar vicinity is vast, additional terms are added to the fitness score to compare overall network optimality. In addition to incorporating the total network ΔV requirements, a penalty is applied for each additional servicer assigned to a tour and a customer benefit is incorporated to encourage the inclusion of the "best" customers. While customer benefit is largely mission specific, three values are included to represent low deployment cost, low upkeep cost, and Lunar south pole coverage. Therefore, the fitness score for a chromosome is evaluated as,

$$score = \frac{1}{\Delta V_T} - \gamma(n_s - 1) + \sum(\gamma_{JC}\beta_{JC}, \gamma_{SP}\beta_{SP}, \gamma_{SK}\beta_{SK}) \tag{8}$$

The servicer penalty is included to discourage the use of additional servicers in the network. While many optional servicers may be input to the algorithm, the network that uses the fewest servicers, with reasonable ΔV impact, is preferred. The scaling factor γ is included to model the cost of new servicer deployment in terms of ΔV. This value relates to the cost to deploy a new servicing hub or the maximum ΔV allowed to a single hub before additional servicers should be added.

The customer benefit is comprised of the satellite deployment benefit, (β_{JC}), station keeping benefit, (β_{SK}), and its south pole coverage benefit, (β_{SP}). While ballistic lunar transfers are a low-cost option for satellite deployment in the lunar vicinity, the deployment costs are evaluated here as an energy differential between the orbit of interest and the L_1 and L_2 Lagrange points. A NASA plan to deploy satellites to the L_1 and L_2 Lagrange points as a part of a broader initiative is known as the Metro campaign. This campaign focuses on establishing a network of spacecraft to enhance communications and navigation in the vicinity of the Moon, particularly in support of the Artemis program and future lunar exploration missions. Therefore, the primary deployment cost beyond the Metro involves transferring from either L_1 or L_2 to the target orbit, a cost that is directly correlated to a change in energy level. Therefore, the deployment benefit value is constructed as,

$$\beta_{JC} = \frac{1}{1 + min(|[JC - JC_{L_1}, JC - JC_{L_2}]|)} \tag{9}$$

As the linear instability for an orbit increases, the estimated costs to conduct station keeping maneuvers also increases. While it is difficult to precisely predict the station keeping costs from the stability of an orbit, they are directly correlated, and can inversely represent the station keeping benefit. First, the time constant is introduced, an estimate of the number of revolutions along an orbit for a perturbation to grow by a factor of e, i.e.,

$$\tau = \frac{1}{|Re(ln(\lambda_{max}(\Phi(T, 0))))|} \tag{10}$$

where λ_{max} represents the largest eigenvalue from the monodromy matrix, the state transition matrix (STM) over one period of the orbit, $\Phi(T, 0)$.[19] Since τ approaches infinity for linearly stable orbits, when the monodromy matrix that possesses all eigenvalues equal to 1, the time constant is truncated to a maximum value of 5 when evaluating the station keeping benefit as,

$$\beta_{SK} = min([\tau, 5]) \tag{11}$$

The goals of each mission are unique so each customer orbit benefit may be altered at the discretion of the designer; for this analysis, the mission specific benefit includes Lunar south pole coverage. The metric for measuring the south pole coverage is the proportion of the orbit period for which the elevation angle from the Lunar south pole is greater than $15°$,[20] i.e.,

$$\beta_{SP} = \frac{1}{T}\int_0^T \delta dt \in [\delta > 15°] \tag{12}$$

Finally, the specific weighting values γ, are up to the designer. For this assessment, more emphasis is placed on the customer orbits with south pole coverage, so the weighting values selected are $[\gamma, \gamma_{JC}, \gamma_{SP}, \gamma_{SK}] = [1, 1, 1, 5]$.

Greedy Crossover: After the chromosomes are ranked in terms of their fitness scores, a greedy sequential crossover is implemented to create the next generation. Similar to that by Al-Furhud,[18] an "offspring" chromosome is generated from two "parent" chromosomes. Given the first servicing hub, intentionally the same first gene in each chromosome, the next gene selected for the offspring is selected by searching for the best valid gene that follows the previous one in both of the parents. If no valid genes follow the current gene in either of the parents, then the cheapest valid gene is selected from the set of remaining options. While more computationally cumbersome than typical crossover operators, this greedy crossover outperforms other manipulation methods[18] while maintaining valid chromosomes. To select which parent chromosomes create an offspring, the previous generation is ranked in terms of fitness score and are paired off in order of rank, i.e., the first two chromosomes are paired, then chromosomes three and four, and continuing.

As an example, suppose a greedy crossover is implemented on the following two parent chromosomes, $P_1 = [5, 3, 2, 1, 4]$ and $P_2 = [5, 4, 2, 3, 1]$. The child is initially assigned the first gene of 5, with the other genes unassigned, $C = [5, *, *, *, *]$. Then, the cost to transfer from orbit 5 to orbits 3 and 4 are constructed and the minimum solution is selected. Assume that the minimum next orbit is orbit 3, then the child is be composed of $C = [5, 3, *, *, *]$, and the process repeats. Then the cost to transfer from orbit 3 to orbits 2 and 1 are compared. Assuming, again, that orbit 1 represents the lower cost, the child then becomes $C = [5, 3, 1, *, *]$. At this point, there are no longer valid genes in both parents (since P_2 reached the end of its sequence), so the next gene is selected from the list of remaining legitimate orbits (orbits 2 and 4). This process repeats until the offspring chromosome is filled to the same length as the parents. While this example does not include the complications of zero genes representing phasing revs and the -1 termination gene, in practice, this process is extended to accommodate them. When only phasing revs and the termination gene are left in the set of legitimate remaining genes, the sequence is terminated and all phasing revs, $0s$, are placed after the -1 in the chromosome. While this implementation introduces a bias to reduce the number of phasing revs used in a schedule over the course of the GA, this bias is only impactful when shorter time of flight options also reduce the total ΔV.

Mutation: The final operator included is a mutation operator. The mutation operator in a Genetic Algorithm (GA) is a key mechanism that introduces variability into the population of solutions. After the selection and crossover steps, mutation randomly alters some genes in the chromosomes of the offspring. This process mimics biological evolution where small random changes in genetic material can lead to new traits. The purpose of mutation is maintenance of genetic diversity within the population, aiding the avoidance of premature convergence to a local optimum by exploring a broader solution space. Mutation is implemented by randomly selecting two genes in the chromosome, excepting the first value and the termination gene, and swapping their locations in the vector, ensuring that the chromosome remains valid. One potential invalid mutation occurs when a mandatory customer gene is placed behind the termination gene, indicating that it

never gets serviced. Including this potential issue, the mutation operator diversifies the search space by swapping servicing orders, increasing or decreasing the number of phasing revs used, and including or excluding optional customer orbits.

Elitism: Elitism ensures that the best-performing individuals from one generation are preserved and carried over to the next generation without undergoing mutation or crossover. This approach guarantees that the optimal solutions thus far are not lost due to the randomness of genetic operations, thus maintaining or improving the overall quality of the population over successive generations. In practice, elitism involves selecting a few top-ranking individuals based on their fitness scores and directly copying them into the new generation. By safeguarding these elite individuals, elitism accelerates convergence towards an optimal solution, as it prevents the degradation of high-quality solutions that might otherwise occur through the GA's stochastic processes. This technique balances the exploration of new solutions with the exploitation of known good solutions, building a more robust and effective algorithm. Elitism here is implemented by selecting the best chromosomes from the set of original chromosomes, crossover offspring, and mutations until the desired population is reached. In the event that the three sets do not contain sufficient non-duplicated solutions to reach the required population size, randomly generated solutions are introduced.

Finally, immigration is implemented, that is, the introduction of new, randomly generated individuals into the population during the evolutionary process. This technique injects diverse genetic material into the population to prevent premature convergence to local optima and maintain diversity. By occasionally replacing some of the current population with new individuals, the algorithm explores new regions of the solution space not previously considered. Immigration is particularly useful in overcoming stagnation when the population becomes too homogeneous and with no significant progress toward better solutions. Immigration is implemented in different ways, e.g., replacing a certain percentage of the population at regular intervals or introducing new individuals when the algorithm detects that the population diversity has fallen below a certain threshold. Here, when the maximum score solution remains constant for forty generations, the bottom scoring half of the next generation is replaced with randomly selected chromosomes. By balancing exploration and exploitation, immigration ensures that the GA continues to search for better solutions while avoiding getting trapped in suboptimal regions of the solution space.

NUMERICAL RESULTS

Performance

Previously, Waldecker and Howell[1] proposed an auction algorithm (AA) for the assignment of servicer satellites to customer satellites in a servicing network. While the AA provides the advantage of significantly lower computational costs than the GA, it does not perform as well when designing schedules that involve sequential transfers. The AA is a class of algorithms to solve assignment problems, where the goal is to optimally match a set of agents to a set of tasks, often based on costs or profits associated with each assignment. These algorithms are inspired by auction processes, where customers field "bids" by servicers to maximize their own profit or minimize their cost. Since the problem of designing a servicing network is more similar to the TSP, naturally, the GA accommodates downstream dependencies better than the AA. In addition, the previously proposed AA does not add phasing revs or include optional orbits. While the AA is replaced with a GA for determining the servicer schedules, the transfer selection and design process used with the AA is replicated here.

To compare the results of servicer schedules determined by the AA and the GA, a servicing scenario is proposed. A network comprised of two servicing hubs and three customer satellites placed in Lyapunov orbits around the L_1 and L_2 Lagrange points is selected for this example. Visualized in Fig. 5, the customer satellites (Blue) are place in L_1 and L_2 Lyapunov orbits at the JC value $JC = [3.16, 3.14, 3.15]$ and the servicing hubs (Red) are placed in similar orbits at the JC value $JC = 3.15$. After the orbits are selected, the AA and GA are used to design schedules for the two servicers to service all customer spacecraft before returning to their respective hubs. The parameters used for the GA are summarized in Table 1.

Upon conclusion, the AA and GA produce different chronological orders for each customer satellite being serviced and different customer satellites assigned to each servicer spacecraft. A simplified schematic of the

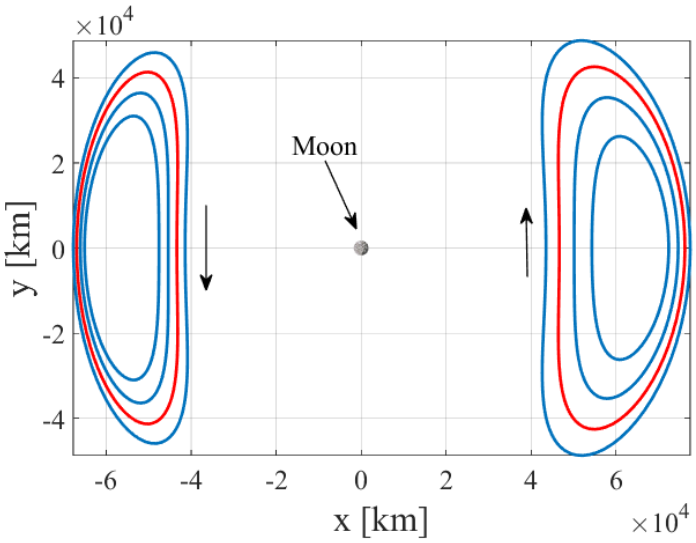
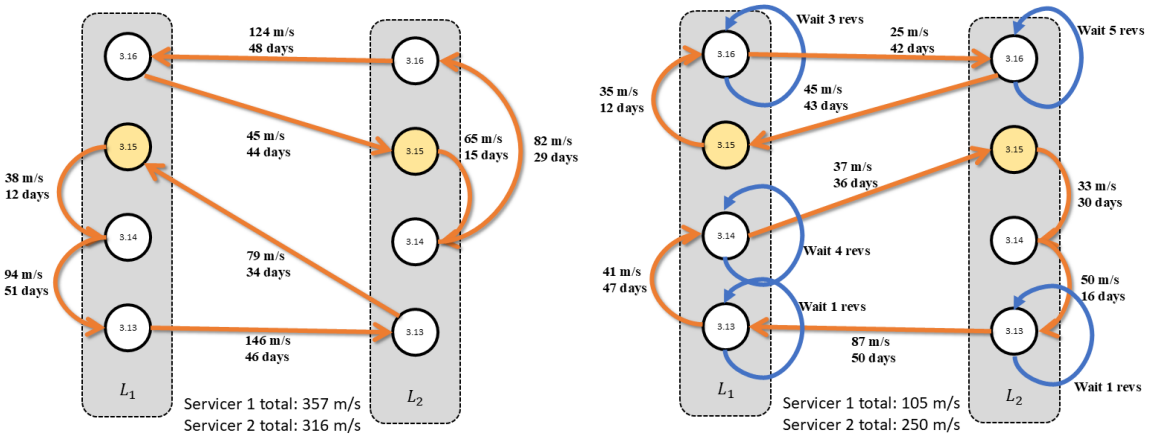


Figure 5: Example Lyapunov Network, moon-centered CR3BP. Red: servicing hub orbits; Blue: customer orbits

servicing order selected by both methods appears in Fig. 6 with the AA schedule on the left and the GA schedule on the right. In the figure, the ΔV is indicated for the fully converged network in the CR3BP. As



(a) AA Network Servicing Order

(b) GA Network Servicing Order

Figure 6: Servicing order determined by the AA (a) and the GA (b). Initial servicing hub orbits in orange and customer orbits in white. Orange arrow indicates transfer taken by servicer satellite.

is noticeable by examining the servicer ΔV totals at the bottom of each subfigure, the GA network requires significantly less ΔV, but also requires more time to service the entire network. The fitness scores for each generation of the GA along with the score associated with the AA are displayed in Fig. 7. It is clear from the figure that, after the first generation of the GA, the average score significantly outperforms that of the AA. In fact, even the best scoring chromosome of the first generation already outperforms the AA. While the GA was continued for all 500 generations, it is clear that the average value starts to plateau around the 150th generation, indicating that, although the maximum score slightly increases afterward, the GA could be stopped at this point for better computational efficiency. Also apparent are the eight sharp downward spikes

Table 1: GA Parameters

Parameter	Value
ΔV Weighting (eq. 4)	.15
γ	0
γ_{JC}	0
γ_{SK}	0
γ_{SP}	0
Generations	500
Population	30
Immigration Rate	40 Generations
Immigration Replacement Rate	60 %

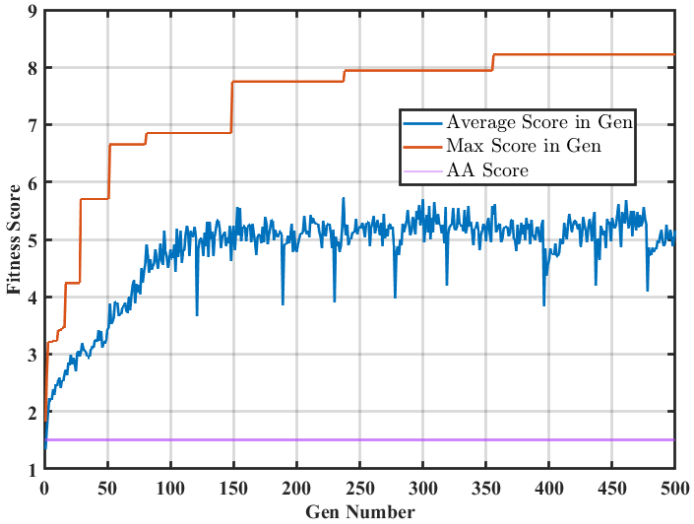
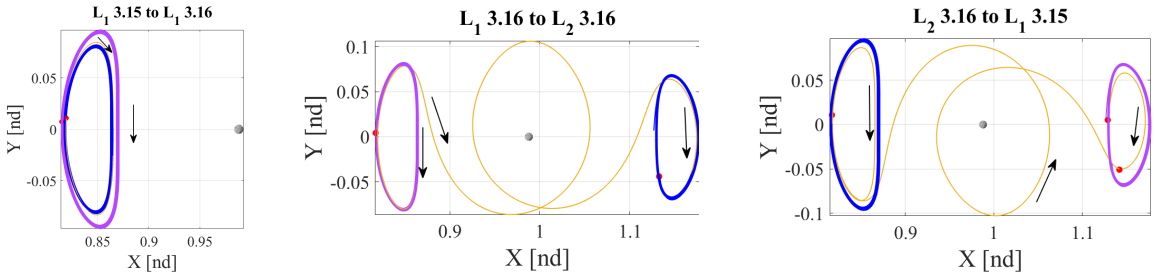


Figure 7: Evolution of GA fitness score and score of AA network. Blue: Average generation fitness score. Orange: Maximum generation fitness score. Purple: Fitness score of AA schedule.

in the average scores across the generations due to the implementation of immigration after 40 generations of the same maximum solution. In this example, the injection of random individuals sometimes led to the crossover result for a new maximum score, but not in all cases. After taking the previously converged GA solution and transitioning it to the ephemeris model, the required ΔV is further reduced. The final ephemeris solution for servicer 1 is plotted in Fig. 8 and servicer 2 in Fig. 9. The total ΔV and transfer time for each servicer is included in Table 2. In the table, comparisons between the CR3BP and ephemeris values are provided.

As noted in Waldecker and Howell,[1] the ephemeris transition process involves translation to the CR3BP barycenter and rotation to match the desired epoch of September 15, 2025. The underlying orbits are converged for a 1 year baseline solution and the transfers are constrained to apply a departure and arrival burn along the baseline orbits such that the departure and arrival locations match that of the previous and upcoming spacecraft respectively.



(a) Transfer 1: $\Delta V = 27\frac{m}{s}$, $ToF = 13$ days

(b) Transfer 2: $\Delta V = 3.1\frac{m}{s}$, $ToF = 43$ days

(c) Transfer 3: $\Delta V = 50\frac{m}{s}$, $ToF = 43$ days

Figure 8: Ephemeris transfers for servicer 1 starting in L_1 Lyapunov at JC = 3.15 visualized in the pulsating-rotating frame. Servicing order goes from left to right. Purple: initial orbit, Blue: target orbit, Yellow: transfer trajectory, Red Dots: ΔV locations

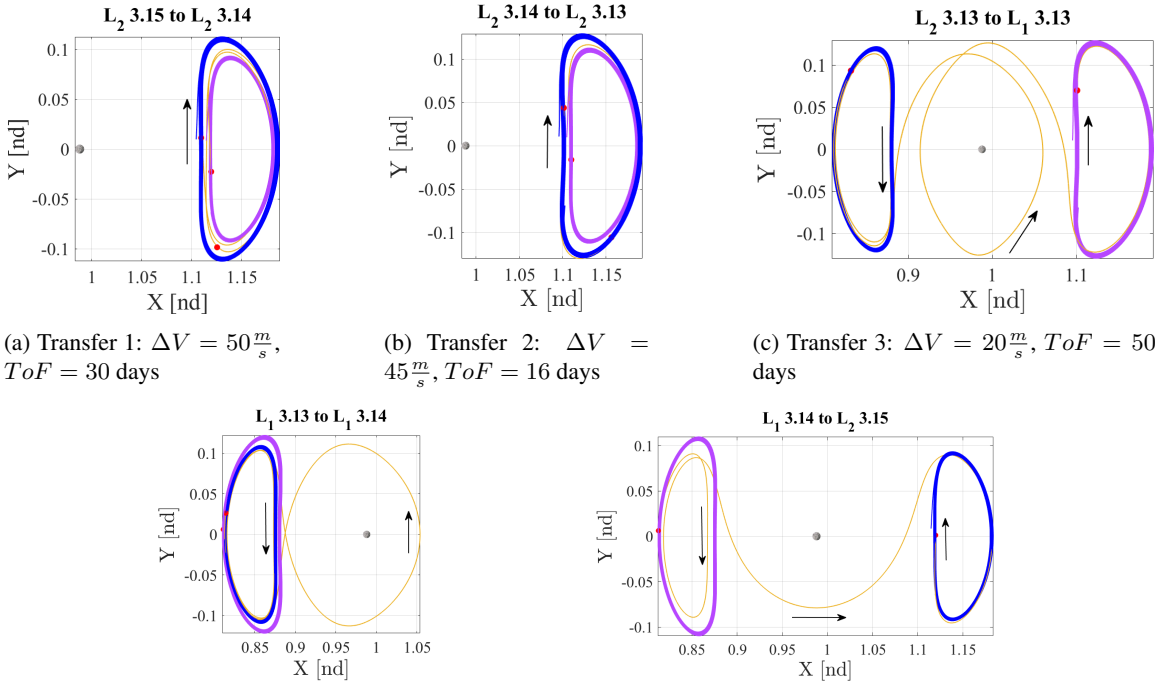
Table 2: Transfer Metrics

Servicer 1	CR3BP $\Delta V[\frac{m}{s}]$	CR3BP ToF [days]	Ephem. $\Delta V[\frac{m}{s}]$	Ephem. ToF [days]
L_1 3.15 to L_1 3.16	35.9	12.156	27.4	12.729
L_1 3.16 to L_2 3.16	25.4	42.356	3.2	43.159
L_2 3.16 to L_1 3.15	44.6	42.554	49.9	43.465
Servicer 2				
L_2 3.15 to L_2 3.14	33.2	29.912	50	30.100
L_2 3.14 to L_2 3.13	50.0	15.549	45.2	15.648
L_2 3.13 to L_1 3.13	87.0	47.329	20	50.476
L_1 3.13 to L_1 3.14	41.9	47.329	29.7	48.132
L_1 3.14 to L_2 3.15	37.9	35.616	40	35.625

Orbit Selection

In addition to selecting the optimal servicing schedule, by implementing optional orbits, the GA is adept at selecting initial servicer and customer orbits to best satisfy mission constraints. As previously posed in Waldecker and Howell,[1] the initial networks were designed based on a list of beneficial criteria that tends to reduce the ΔV required to transfer between orbits. The list of criteria included linear instability, similar JC values, "short" period orbits, near resonant periods throughout the network, symmetric orbits, and orbits near period multiplying bifurcations. Through the inclusion of optional orbits, multiple networks of diverse orbits are seeded into the GA to determine the relative priority of the originally posed beneficial criteria. Two simulations are conducted to identify priorities on this list, one set where customer satellites were restricted to L_1 Lyapunov orbits and another set where customers were restricted to L_1 halo orbits.

For the first example, ten different networks of orbits are selected to determine optimal pairings for servicer/customer orbits. To reduce the search space, customer orbits are restricted to L_1 Lyapunov orbits in the energy range $JC = [2.92, 3.18]$ and servicer orbits are restricted to L_1 orbits in the halo, Lyapunov, vertical, and axial families. For each of the ten simulations, two customers at random JC values and five servicers



(a) Transfer 1: $\Delta V = 50\frac{m}{s}$, $ToF = 30$ days

(b) Transfer 2: $\Delta V = 45\frac{m}{s}$, $ToF = 16$ days

(c) Transfer 3: $\Delta V = 20\frac{m}{s}$, $ToF = 50$ days

(d) Transfer 4: $\Delta V = 30\frac{m}{s}$, $ToF = 48$ days

(e) Transfer 5: $\Delta V = 40\frac{m}{s}$, $ToF = 36$ days

Figure 9: Ephemeris transfers for servicer 2 starting in L_2 Lyapunov at JC = 3.15 visualized in the pulsating-rotating frame. Servicing order goes from left to right. Purple: initial orbit, Blue: target orbit, Yellow: transfer trajectory, Red Dots: ΔV locations

from random orbital families and JC values are generated. All servicer orbits in each network are considered "optional", but all customer orbits are considered mandatory, assisting in the identification of the servicers that best minimize the overall ΔV. The results from the networks is plotted in Fig. 10 across six subplots. In the plots, 'x's represent servicer orbits, the solid colored dots represent customer orbits, and the black dots represent optional servicer orbits that were not implemented into the network. The colors are also maintained across the subplots, i.e., the network represented by one color in the "JC vs Time Constant" plot is the same network represented by the same color in the other subplots. Each of the subplots are as follows: a) each orbit JC vs Time Constant, b) JC vs ΔV in and out of each orbit, c) ΔJC vs ΔV for each transfer, d) Period vs JC for each orbit, e) Period vs ΔV in and out of each orbit, and f) ΔJC vs Δ Period for each transfer.

The first significant result is apparent by zooming into the Period vs JC plot and identifying the individual families, as plotted in Fig. 11. In this first simulation, where all of the customer orbits are restricted to the L_1 Lyapunov family, the servicer orbits that minimize the ΔV are always in the same family (Lyapunov family) as indicated by the colored data points. While this result may not be surprising because the Lyapunov family is planar and the other families are not, some of the unselected halo options are very nearly planar with closer energy levels than the solutions selected, indicating potentially lower theoretical minimum ΔV requirements.

Next, examining subplots b) and e) in Fig. 10, it is noticeable that there is no strong correlation between the JC or Period and the ΔV requirements. In contrast to a previous hypothesis, the period of the orbits does not noticeably impact the ΔV required for phasing. Such is likely due to the implementation of phasing revs in the solution space. By allowing the servicer satellites to remain in a customer orbit after conducting servicing activities, the GA is able to identify suitable schedules and wait times to overcome the need to incorporate orbits with short periods (increasing the frequency of favorable configurations).

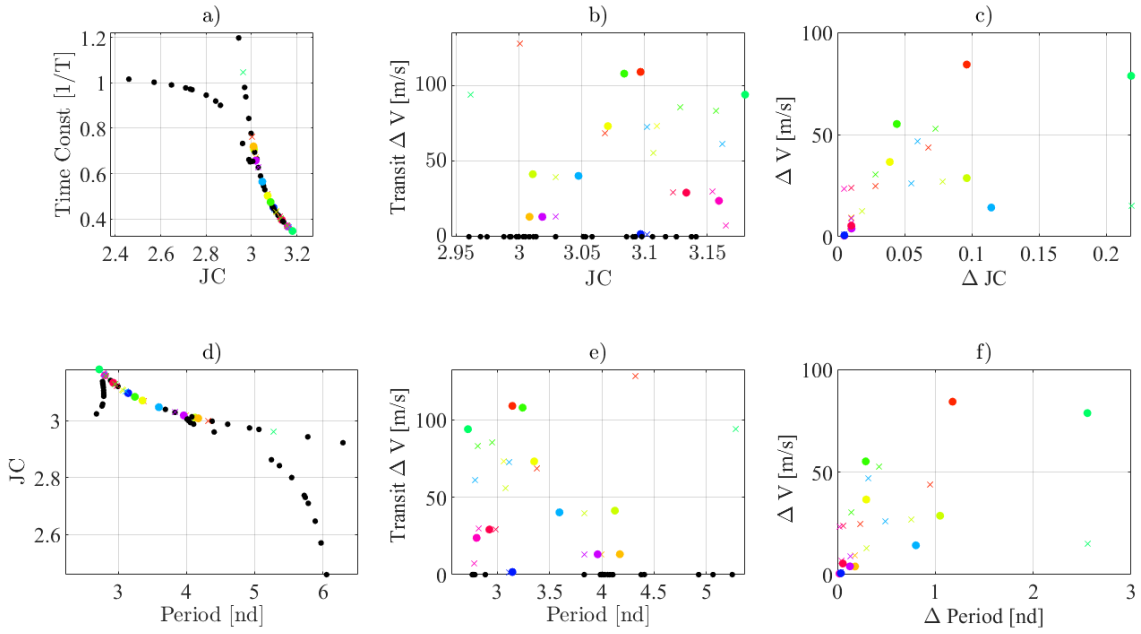


Figure 10: Lyapunov Customer Network Metrics.

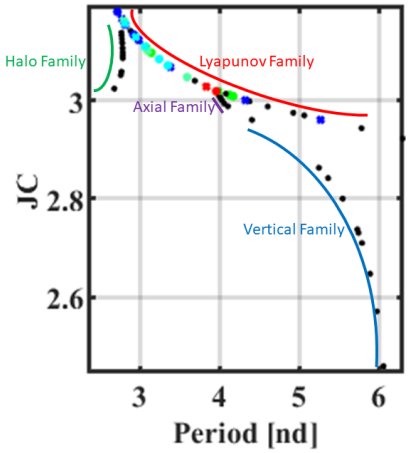


Figure 11: Lyapunov customer network JC vs Period with orbit families labeled.

Finally, subplots c) and f) indicate that there is positive correlation between the difference in JC or period and ΔV when designing the transfers. While there is a clear relationship between the JC and the period of the orbits, it is clear that the ΔJC offers a stronger positive correlation with the required ΔV. The second simulation is similar to the first, with the only alteration that all of the customer orbits are restricted to L_1 halo orbits. As before, the customer orbit energy levels are varied and the servicer orbits are randomly selected from the four families mentioned before. The six comparison subplots for the second simulation appear in Fig. 12. Examining the JC vs Period subplot a), it is noticeable that halo orbits were not the only servicers selected for the networks, in contrast to the Lyapunov network where only servicers in the same family were selected. However, by referencing the other subplots that represent the required ΔV, the networks that used servicers in foreign families also required the largest ΔV. The only networks that used foreign servicers and required less than $150\frac{m}{s}$ per transfer consisted of a servicing hub placed in a Lyapunov orbit and a halo

customer that was nearly planar. Similar to the Lyapunov simulation, the JC and period of the orbits does not have any noticeable impact on the ΔV requirements, but the ΔJC provides the strongest correlation to required ΔV.

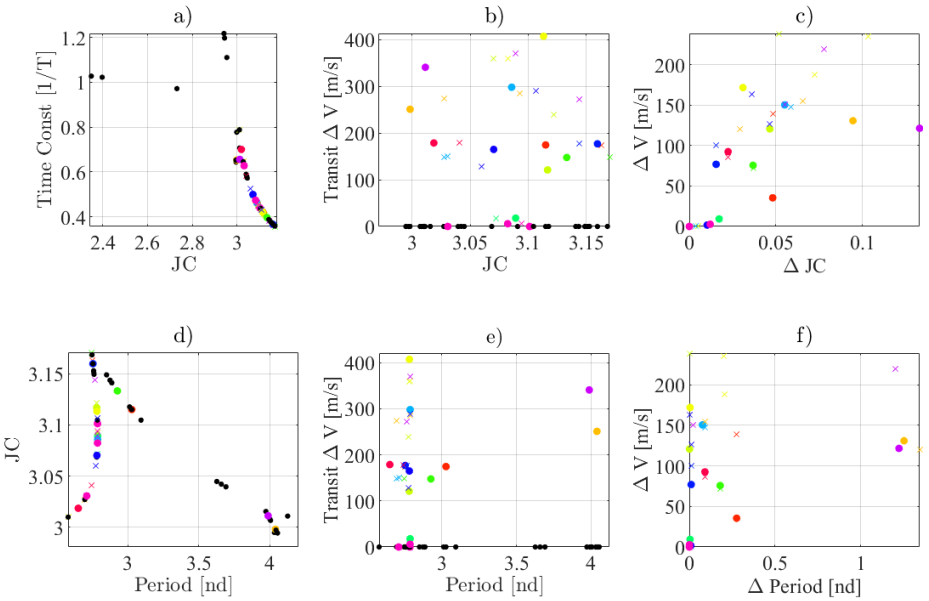


Figure 12: Halo Orbit Customer Network Metrics.

Finally, the impacts of the time constant on the transfer ΔV and transfer time is examined. Comparisons between the ΔV and transfer time versus orbit time constant for each individual transfer in both networks are plotted in Fig. 13. In the figure, the x-axis represents the time constant for the originating orbit for each transfer, with that respective transfers required ΔV and transfer time on the y-axis. From this plot, the impact of the time constant on transfer parameters is minimal. It is worth stressing that all orbits used in these simulations were linearly unstable with time constant values less than 1.2 revs. If stable or nearly stable orbits are incorporated, such as the Near Rectilinear Halo Orbits (NRHOs), the impacts of the time constant are expected to be much more impactful when designing transfers using the same schemes presented here.

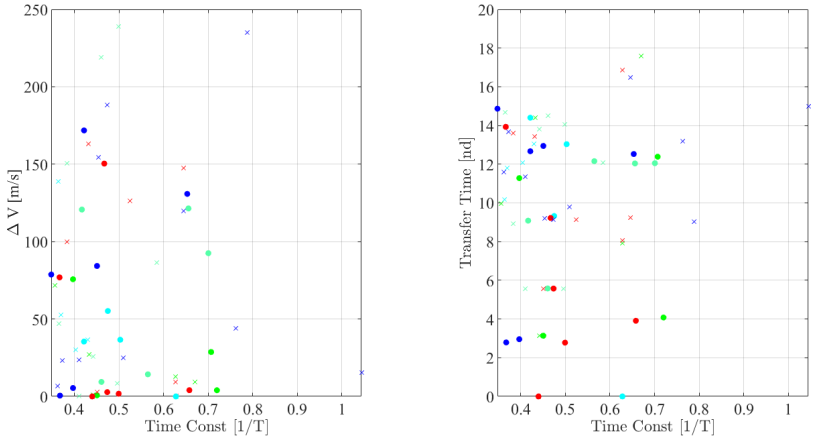


Figure 13: ΔV and Transfer Time vs Time Constant for transfers throughout both networks.

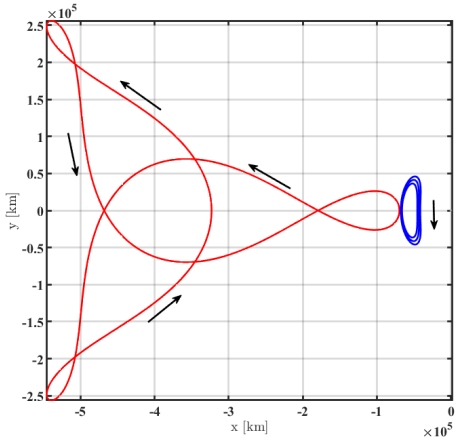
From the two simulations conducted, the importance of the previously identified criteria is identified. First, while symmetry is not directly measured, orbit similarity in the form of orbits being in the same family is the most important characteristic when selecting orbits for on-orbit servicing networks. While this requirement is significantly limiting, transfers between orbits belonging to the same family present the lowest ΔV requirements. Second, and not surprisingly, the difference in JC between two orbits provides the strongest relationship with ΔV costs. Especially notable for transfers through a single family, as the energy gap increases, so do the propulsive requirements to bridge that gap. The correlation between required ΔV and the energy gap also has a relationship with the difference in periods between the orbits, but due to the dependence of JC and orbit period, it is difficult to isolate the impact of Δperiod on transfer ΔV. Finally, the impact of orbit stability and individual periods of the orbits is not noticeable on the required ΔV and transfer time. Once again, these simulations are conducted with linearly unstable orbits and it is likely that there would be a noticeable relationship when replicating this work with stable or nearly stable orbits.

Additional Networks

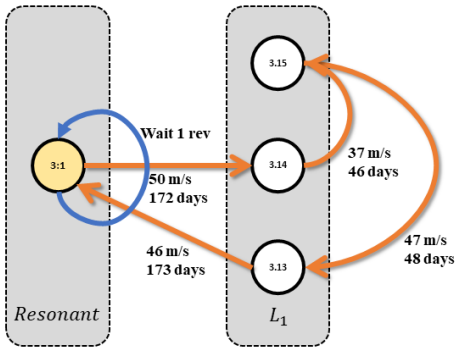
The application of the GA across various servicing networks is also explored, demonstrating its robustness and adaptability to different types of orbital configurations and the ability to select initial servicer/customer orbits that maximize network benefit. The flexibility of the GA in accommodating diverse orbital setups is a key element for optimizing satellite servicing networks, especially in the complex gravitational environment near the Moon. The GA consistently demonstrates its efficiency in minimizing the overall ΔV across multiple scenarios with distinct orbital characteristics. This adaptability highlights the GA's effectiveness in optimizing servicing schedules and its potential to be applied to a wide range of architectures.

3:1 Resonant Orbits and Lyapunov Orbits

The first additional network involves the placement of a servicer hub in a 3:1 resonant orbit with a $JC = 3.15$ and customer satellites placed in L_1 Lyapunov orbits at $JC = [3.14, 3.15, 3.16]$. Resonant orbits present particular challenges when they are incorporated into a servicing network due to the inherently long periods. While they offer numerous transfer options over the span of a single period, the resulting transfer times are frequently much longer than other orbits in the lunar vicinity. Representative orbits are plotted in the CR3BP in Fig. 14(a) and the network determined by the GA is displayed in Fig. 14(b). Even though the resonant orbit does not offer the most favorable phasing characteristics, the GA is capable of identifying a network for less than $200\frac{m}{s}$ total.



(a) Network orbits moon-centered CR3BP



(b) Servicing schedule

Figure 14: (a) Network with servicing hub in 3:1 resonant orbit and three customer satellites in L_1 Lyapunov orbits and (b) servicing schedule.

After converging the network in the ephemeris model, it is noticeable in Table 3 that the propulsive costs

departing from and arriving to the resonant orbit increase. This ΔV increase is likely due to the difficulty in converging the two-year baseline for the resonant orbit, creating large position discontinuities at the start of the convergence process. In addition, due to the vicinity of the 3:1 resonant orbit, an Earth-centered propagation and targeting scheme is exploited as opposed to the previously used Moon-centered scheme. The ephemeris transfers throughout the network appear in Fig. 15.

Table 3: 3:1 Network Transfer Metrics

Servicer 1	CR3BP $\Delta V[\frac{m}{s}]$	CR3BP ToF [days]	Ephem. $\Delta V[\frac{m}{s}]$	Ephem. ToF [days]
3:1 to 3.14	50	172.734	80.6	172.69
3.14 to 3.15	37.1	45.9	35.5	46.16
3.15 to 3.13	46.9	47.698	30	48.08
3.13 to 3:1	46.1	173.554	55.3	173.97

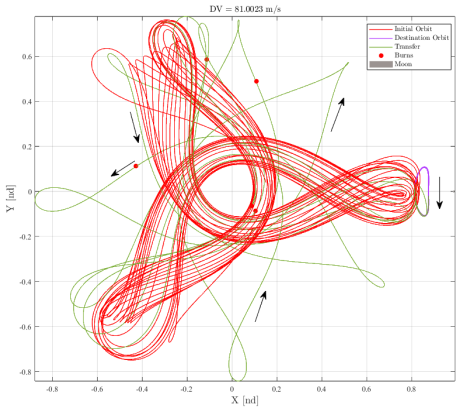
South Pole Optimized Network

Finally, a network of servicer hubs placed in a northern butterfly orbit at $JC = 3.05$ and an L_1 Lyapunov orbit at $JC = 3.06$ with customer satellites in: a northern butterfly orbit at $JC = 3.04$, a southern L_2 halo orbit at $JC = 3.05$, an L_1 Lyapunov orbit at $JC = 3.05$, and an L_2 Lyapunov orbit at $JC = 3.04$, are evaluated to demonstrate the versatility of the GA. This network of orbits is selected as an optimal network that heavily weights Lunar south pole (SP) coverage. The parameters for this network are displayed in Table 4. In the table, it is noticeable that the weighting for Lunar south pole coverage is significantly larger

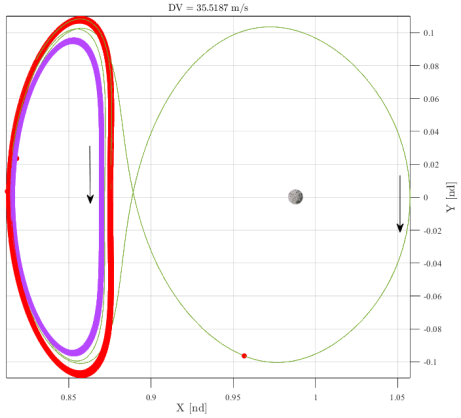
Table 4: GA Parameters

Parameter	Value
ΔV Weighting (Eqn. 4)	.15
γ	1
γ_{JC}	1
γ_{SK}	1
γ_{SP}	20
Generations	500
Population	30
Immigration Rate	40 Generations
Immigration Replacement Rate	60 %

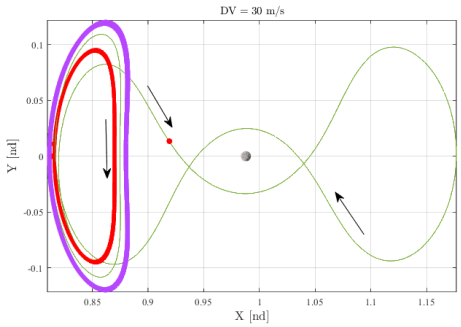
than the other values. This large value intentionally identifies the specific orbits that offer the most coverage, while also requiring the least amount of ΔV. Through a process of seeding various spatial and planar orbits in the lunar vicinity, the final network is selected out of 100 random seeds. Due to the nature of low ΔV transfers between halo orbits near the same Lagrange point and the energy level, the search space is pruned to remove networks solely comprised of similar halo orbits. The final network is displayed in Fig. 16 along with the converged ΔV requirements. Due to the heavy weighting of the Lunar south pole coverage, the ΔV requirements to transfer between the spatial orbits is significant. However, due to the large time interval over which the halo orbits maintain coverage of the Lunar south pole, the algorithm identified the additional transfer as a part of the optimal configuration. Against initial intuition, the GA also converged to a solution that included a servicer and customer satellites placed in Lyapunov orbits, even though they offer no Lunar



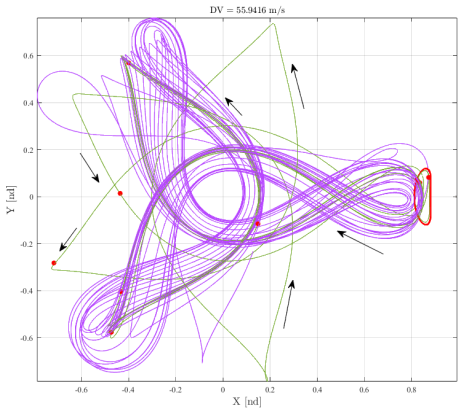
(a) Transfer from 3:1 Resonant to 3.14 Lyapunov Orbit



(b) Transfer from 3.14 Lyapunov Orbit to 3.15 Lyapunov Orbit



(c) Transfer from 3.15 Lyapunov Orbit to 3.13 Lyapunov Orbit



(d) Transfer from 3.13 Lyapunov Orbit to 3:1 Resonant Orbit

Figure 15: Ephemeris converged transfers throughout the 3:1 network plotted in the pulsating-rotating frame. In each sub figure: Red - Initial Orbit, Purple - Target Orbit, Green - Transfer Trajectory, Red Dots - ΔV Locations.

south pole coverage. While the orbits offer no benefit in terms of south pole coverage, the significantly lower ΔV required, low deployment costs, and low station keeping costs makes these orbits suitable candidates as part of the servicing network. Finally, while the mission objective for this example is to maximize Lunar south pole coverage, the GA fitness score may be altered to meet the needs of the designer.

CONCLUDING REMARKS

A two part scheme incorporating a genetic algorithm is introduced to determine the optimal scheduling and transfer design for on-orbit servicing networks in the cislunar vicinity. The one-dimensional representation of the chromosome enables the representation of servicer orbits, customer orbits, phasing revolutions, and the incorporation of optional orbits for trade space analysis. Selection, crossover operators, and elitism is employed to select optimal servicer satellite schedules for diverse network configurations. Using an example of L_1 and L_2 Lyapunov orbits, the GA outperforms an auction algorithm significantly by incorporating downrange phasing requirements.

The GA is implemented with a multi-objective fitness score that incorporates total network ΔV, servicer count, and customer orbit benefit. The customer benefit seeks to minimize deployment costs and station

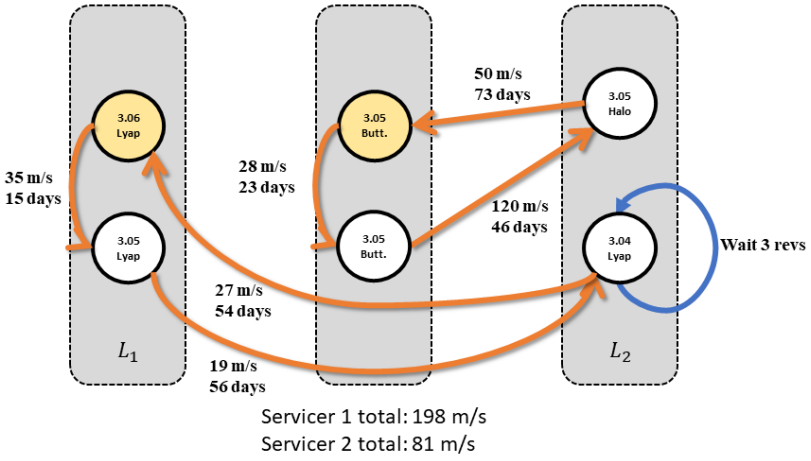


Figure 16: Lunar south pole coverage optimized network.

keeping requirements while maximizing coverage of the Lunar south pole. With this fitness score evaluation, the orbit symmetry in the form of two orbits belonging to the same family as well as the energy gap, i.e., JC, generally induce the largest impact on the ΔV requirements to traverse a network. The time constant and period of the orbits in the network has no significant impact on the ΔV requirements for linearly unstable orbits.

Finally, the GA is demonstrated with a diverse set of networks involving planar, spatial, and resonant orbits. This demonstration highlights the robustness of the proposed two-part solution and the applicability to various types of orbits. Additionally, it demonstrates the applicability for using the GA to identify optimal networks of servicer and customer satellites to achieve mission specific objectives such as Lunar south pole coverage.

ACKNOWLEDGMENT

The authors acknowledge support from the School of Aeronautics and Astronautics at Purdue University as well as the Honors College. Various colleagues offered many useful insights. Liam Fahey, Josiah Badiali, Ricardo Gomez, and other members of Purdue's Multi-Body Dynamics Research Group also provided valuable discussions that enriched the work.

REFERENCES

[1] C. Waldecker and K. Howell, "A Design Approach for Cislunar On-Orbit Servicing Networks," *AAS/AIAA Astrodynamics Specialist Conference*, Aug 2024.

[2] M. Bellmore and G. L. Nemhauser, "The Traveling Salesman Problem: A Survey," *Operations Research*, June 1968. Publisher: INFORMS, 10.1287/opre.16.3.538.

[3] M. Mavrovouniotis, F. M. Müller, and S. Yang, "Ant Colony Optimization With Local Search for Dynamic Traveling Salesman Problems," *IEEE Transactions on Cybernetics*, Vol. 47, July 2017, pp. 1743–1756. Conference Name: IEEE Transactions on Cybernetics, 10.1109/TCYB.2016.2556742.

[4] S. Lin, "Computer solutions of the traveling salesman problem," *The Bell System Technical Journal*, Vol. 44, No. 10, 1965, pp. 2245–2269, 10.1002/j.1538-7305.1965.tb04146.x.

[5] P. Larrañaga, C. Kuijpers, R. Murga, I. Inza, and S. Dizdarevic, "Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators," *Artificial Intelligence Review*, Vol. 13, Apr. 1999, pp. 129–170, 10.1023/A:1006529012972.

[6] R. Matai, S. Singh, M. L. Mittal, R. Matai, S. Singh, and M. L. Mittal, "Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches," *Traveling Salesman Problem, Theory and Applications*, IntechOpen, Dec. 2010, 10.5772/12909.

[7] M. Liśkiewicz and M. R. Schuster, "A new upper bound for the traveling salesman problem in cubic graphs," *Journal of Discrete Algorithms*, Vol. 27, 2014, pp. 1–20, https://doi.org/10.1016/j.jda.2014.02.001.

[8] B. A. Smith, L. A. Soderblom, R. Beebe, J. Boyce, G. Briggs, M. Carr, S. A. Collins, A. F. Cook, G. E. Danielson, M. E. Davies, G. E. Hunt, A. Ingersoll, T. V. Johnson, H. Masursky, J. McCauley, D. Morrison, T. Owen, C. Sagan, E. M. Shoemaker, R. Strom, V. E. Suomi, and J. Veverka, "The Galilean Satellites and Jupiter: Voyager 2 Imaging Science Results," *Science*, Vol. 206, No. 4421, 1979, pp. 927–950, 10.1126/science.206.4421.927.

[9] A.-y. Huang, Y.-z. Luo, and H.-n. Li, "Global Optimization of Multiple-Spacecraft Rendezvous Mission via Decomposition and Dynamics- Guide Evolution Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 45, Jan. 2022, pp. 171–178. Publisher: American Institute of Aeronautics and Astronautics, 10.2514/1.G006101.

[10] A.-y. Huang, Y.-z. Luo, H.-n. Li, and S.-g. Wu, "Optimization of low-thrust multi-debris removal mission via an efficient approximation model of orbit rendezvous," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 236, Nov. 2022, pp. 3045–3056. Publisher: IMECHE, 10.1177/09544100221074746.

[11] H. Li, S. Chen, and H. Baoyin, "J2-Perturbed Multitarget Rendezvous Optimization with Low Thrust," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 3, 2018, pp. 802–808, 10.2514/1.G002889.

[12] A. Petropoulos, D. Grebow, D. Jones, G. Lantoine, A. Nicholas, J. Roa, J. Senent, J. Stuart, N. Arora, T. Pavlak, T. Lam, T. McElrath, R. Roncoli, D. Garza, N. Bradley, D. Landau, Z. Tarzi, F. Laipert, E. Bonfiglio, M. Wallace, and J. Sims, "GTOC9: Results from the Jet Propulsion Laboratory (team JPL)," *Acta Futura*, Jan. 2018, pp. 25–35, 10.5281/zenodo.1139152.

[13] A. F. Haapala and K. C. Howell, "A Framework for Constructing Transfers Linking Periodic Libration Point Orbits in the Spatial Circular Restricted Three-Body Problem," *International Journal of Bifurcation and Chaos*, Vol. 26, May 2016, p. 1630013. Publisher: World Scientific Publishing Co., 10.1142/S0218127416300135.

[14] R. Matai, S. Singh, and M. L. Mittal, "Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches," *Traveling Salesman Problem* (D. Davendra, ed.), ch. 1, Rijeka: IntechOpen, 2010, 10.5772/12909.

[15] C. J. Malmborg, "A genetic algorithm for service level based vehicle scheduling," *European Journal of Operational Research*, Vol. 93, Aug. 1996, pp. 121–134, 10.1016/0377-2217(95)00185-9.

[16] E. Falkenauer, *Genetic Algorithms and Grouping Problems*. USA: John Wiley & Sons, Inc., 1998.

[17] S. Ross, "Introduction to Probability Models," *Introduction to Probability Models (Eleventh Edition)*, Boston: Academic Press, eleventh edition ed., 2014, https://doi.org/10.1016/B978-0-12-407948-9.00012-8.

[18] M. A. Al-Furhud and Z. Hussain, "Genetic Algorithms for the Multiple Travelling Salesman Problem," *International Journal of Advanced Computer Science and Applications*, Vol. 11, No. 7, 2020, 10.14569/IJACSA.2020.0110768.

[19] E. Spreen, K. Howell, and D. Davis, "Near Rectilinear Halo Orbits and Their Application in Cis-Lunar Space," *3rd IAA Conference on Dynamics and Controls of Space Systems*, 05 2017.

[20] D. J. Grebow, M. T. Ozimek, K. C. Howell, and D. C. Folta, "Multibody Orbit Architectures for Lunar South Pole Coverage," *Journal of Spacecraft and Rockets*, Vol. 45, No. 2, 2008, pp. 344–358, 10.2514/1.28738.