Robust Cislunar Architecture Design for Cooperative Agents

Michael Klonowski
University of Colorado at Boulder
Naomi Owens-Fahrner
Ball Aerospace

Casey Heidrich
University of Colorado at Boulder

Marcus J. Holzinger
University of Colorado at Boulder

ABSTRACT

As launch capabilities and interest in Cislunar missions expand for commercial, government, and military actors alike, so grows the necessity for a robust architecture in the Cislunar region for conducting Space Domain Awareness (SDA). Among the numerous objectives associated with constructing a robust Cislunar SDA architecture, of particular interest is creating an architecture that is cost effective for both architecture users and maintainers. This paper builds on previous work of using reinforcement learning methods to solve the multi-objective Cislunar architecture design problem by modeling a cooperative architecture user that seeks to maximize availability to the architecture while minimizing total delta-v needed. Through analysis of the interactions between the user and the architecture, we can maximize utility in the architecture design process while gaining insights into the fundamental qualities of a robust Cislunar architecture.

1. INTRODUCTION

Critical to the success of missions within the Cislunar domain is the development of an effective and low cost Cislunar Space Domain Awareness (SDA) architecture. With the increasing density of the LEO, MEO, GEO domains, ground-based SDA networks are heavily subscribed to the near-Earth domain, and lack the technological capabilities to conduct SDA in the Cislunar regime. To perform effective Cislunar SDA, a distributed network of sensors within the region are required that can perform the tasks needed within the highly complex and chaotic dynamics system of Cislunar space. This paper builds on previous work [16] and employs a novel technique to generate sets of Pareto-optimal solutions to the Cislunar SDA architecture problem that are optimized for various objectives critical to the success of future Cislunar missions.

In 2022, the White House Office of Science and Technology Policy released its first document outlining national strategic policies and goals for development of the Cislunar regime [4]. The document places strategic emphasis on the development of Cislunar SDA architecture, and since publication there has been a growing volume of academic work in support of this effort. A significant amount of research has been conducted analysing specific families of Cislunar orbits for the objective of maximizing coverage of targets and volumes in Cislunar space, [26, 7, 15, 1, 27, 13, 16, 9, 14]. Due to the complex and highly non-linear dynamics governing Cislunar space, many of these studies involve geometric and photometric analyses of families of periodic orbits.

Conclusion from works such as [26, 13, 15, 1] find that certain characteristics of orbits may provide better relative viewing geometries with respect to the solar phase angle for electro-optical (EO) observers. Works such as [16, 27] further abstract these conclusions and pose the architecture design problem as a multi-objective optimization problem (MOOP). In these, architectures are generated iteratively, consisting of combinations of multiple observers in various orbits across the Cislunar regime. Pareto-optimal sets of Cislunar SDA architectures are then obtained, representing the

trade spaces of objectives such as cost and performance. Results from these analyses reinforce the intuitive result that combinations of multiple observers in periodic orbits approximately evenly distributed across the Cislunar region of interest can provide sufficient coverage of said areas. As expected, nuances appear in the relative phasing of observers within the architecture and with the solar phase angle. In [16], the authors find that the selection of initial phasing within each observers' orbit can result in significant performance deviations over multiple years. Specifically, the authors find that with a single Earth-Moon transfer trajectory, architecture performance greatly varies over time.

As has been discussed at length within the literature, while fundamental solutions exist within low-fidelity dynamical models, motion within Cislunar space is highly non-linear and chaotic. Operating a mission within the region requires generation of a transfer trajectory, often to one of the five Lagrange points as defined within the Circular-Restricted Three Body Problem (CR3BP) - a nontrivial task. As the number of Cislunar missions increase, so will the diversity of transfer geometries and mission objectives. A critical aspect of an effective Cislunar SDA architecture is that cooperative agents operating in the region are able to use said architecture in a cost effective and efficient manner, while satisfying other high level mission objectives. In these scenarios, designing an architecture with the objective of maximizing accessibility of a specific volume or coverage of a specific trajectory may not suffice, as it does not necessarily consider the needs of the architecture user.

In this work, we build on [16] by introducing a novel method to generate Cislunar SDA architectures that minimize the delta-v of a cooperative agent (CA), while minimizing architecture cost, and maximizing coverage of the CA. This novel objective is implemented through the modelling of a CA which iteratively generates optimal transfer trajectories that minimize its delta-v while maximizing its photometric detection threshold based on visual apparent magnitude as seen by the architecture. The resultant MOOP is optimized via a hybrid multi-objective Monte-Carlo Tree Search (MO-MCTS)/NSGA II algorithm [28, 8]. We then use well known analysis techniques [3] as well as introduce new methods to aid in analysis of the resultant trajectories generated by the CA. These tools help to better understand the underlying dynamics of the Cislunar SDA architecture problem, and help to differentiate various types of architectures based on their iteractions with the CA.

2. MULTI-OBJECTIVE OPTIMIZATION

Multi-objective optimization (MOO) is the process of optimizing a collection of objective functions subject to some pre-defined constraints. The solutions of a MOOP thus lie along a Pareto front, representing the trade space of optimality. Pareto-optimality and its associated definitions, which will be used interchangably throughout this paper, is defined as follows, taken from [19, 16]:

Definition 2.1 (Pareto-optimal) *Given a MOOP with objective function* $\mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}), F_2(\mathbf{x}), ..., F_k(\mathbf{x})]^T$ *a solution* $\mathbf{x}^*$ *is* ***Pareto-optimal*** *if and only if there does not exist another point* $\mathbf{x}$*, such that* $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$*, and* $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$ *for at least one* $F_i \in \mathbf{F}$*. The set of Pareto-optimal solutions is referred to as the* ***Pareto set*** *(P), and* $\mathbf{x}^*$ *is said to be* ***non-dominated*** *in P. Furthermore,* $\mathbf{x}^*$ ***dominates*** $\mathbf{x}$*.*

2.1 Markov Decision Processes

A Markov Decision Process (MDP) is a discrete-time stochastic process in which the probability of each state in the process depends only on the previous state. An MDP is defined by the tuple (S, A, T, R, γ), with state space S, action space A, state transition probability function T, reward function R, and discount factor γ. The goal of an MDP is to find a policy function $\pi(s) : S \rightarrow A$ that maximizes some function of the cumulative future rewards. MDPs are by definition fully observable, and thus can be solved using a variety of algorithms when no time constraint is given. However, for MDPs with large state and action spaces, solving for the optimal policy, $\pi^*(s)$, quickly becomes intractable due to the so-called "curse of dimensionality". Numerous methods have been developed that use random sampling combined with heuristics to find sufficiently good approximations to these problems.

2.2 Multi-Objective Monte Carlo Tree Search

Monte Carlo Tree Search is a tree search algorithm that has found great success across a variety of domains, and in general may be applied to any sequential decision process. In its most common form, it is combined with the bandit based selection policy Upper Confidence Bounds Applied to Trees (UCT) [17] which balances exploration of new states with exploiting previously visited promising states. MCTS continues to be successfully applied across a wide variety of Cislunar SDA specific problems [9, 25, 22, 16]. For brevity, we provide a brief discussion on MCTS here.

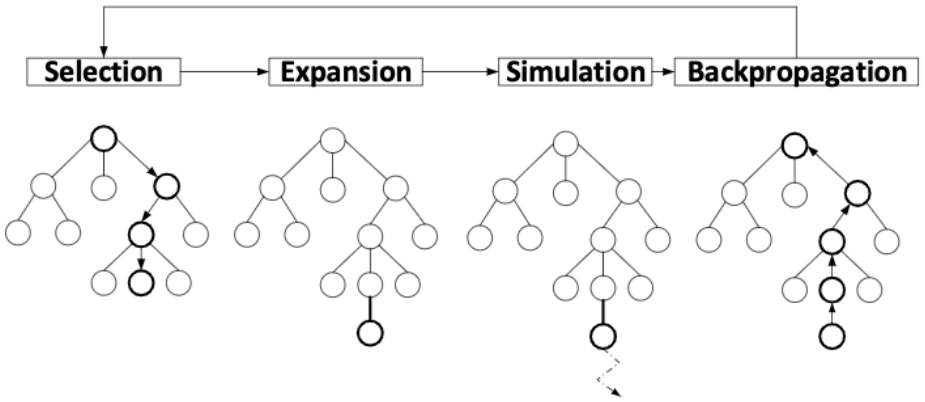


Fig. 1: Visualization of the steps of Monte Carlo Tree Search from [24].

MCTS works in four distinct steps, all of which serve to iteratively generate a tree whose nodes represent states in the underlying MDP. These steps are visualized in Figure 1 and are described as follows:

1. **Selection** - Starting from the root node, the algorithm traverses the tree using UCT until it reaches a node that has not yet been fully expanded (a leaf node).
2. **Expansion** - Unless the leaf node is a terminal state, from the leaf node a child node is created from a chosen action.
3. **Simulation** - From the new child node, a Monte-Carlo simulation is run using a rollout policy until a terminal state or some other condition is reached. At this stage, a reward is obtained through the MDP.
4. **Backpropagation** - The resultant outcome of this simulation is used to update the statistics of all the nodes visited during the tree traversal. These statistics include the node's estimated value and visit counts.

Monte Carlo Tree Search typically operates on problems with that return single scalar reward, but has been extended to problems with multi-objective reward vectors via the hyper-volume indicator (HVI) [28]. The HVI is a measure of the hyper-volume of the objective space covered by a set of solutions with respect to a reference point. It takes the vectorial action-value estimate from MCTS and converts it to a scalar value that can be used within UCT. The inclusion of the HVI affects the UCT algorithm in the selection phase of MCTS, where now an action is selected that maximizes

$$U(s,a) = HV(Q(s,a) \cup P) + C\sqrt{\frac{\ln N(s)}{N(s,a)}} \tag{1}$$

where $Q(s,a)$ is the action value estimate of taking action a at state s, $N(s)$ is the number of times s has been visited, $N(s,a)$ is the number of times a has been selected at s, C is a constant typically chosen as $\sqrt{2}$ for rewards defined in the interval $[0,1]$, $HV(\cdot)$ is the hyper-volume indicator function from [2], and P is the current estimate of the Pareto set. Including the HVI here allows multi-objective Monte Carlo Tree Search (MO-MCTS) to consider the effect selecting an action at a node might have on all objectives and works to maximize the total HVI of the Pareto set. MO-MCTS has been shown to be an effective algorithm for providing Pareto optimal solutions to large scale, highly-nonlinear MOOPs [16].

In many applications, the size of the MDP's state and action spaces can hinder MCTS in its ability to reach states deep within the tree. The tree is said to have a large branching factor in these situations and methods can be implemented to control the expansion and allow deeper states to be reached by the algorithm. Here, we use the Progressive Widening (PW) algorithm, which constrains the amount of allowable actions at each node in the search tree [6]. Namely, with constants $C > 0$ and $\alpha \in (0,1]$ and node visits $j = 1,2,3,...$, the number of allowable actions at a node k increases as

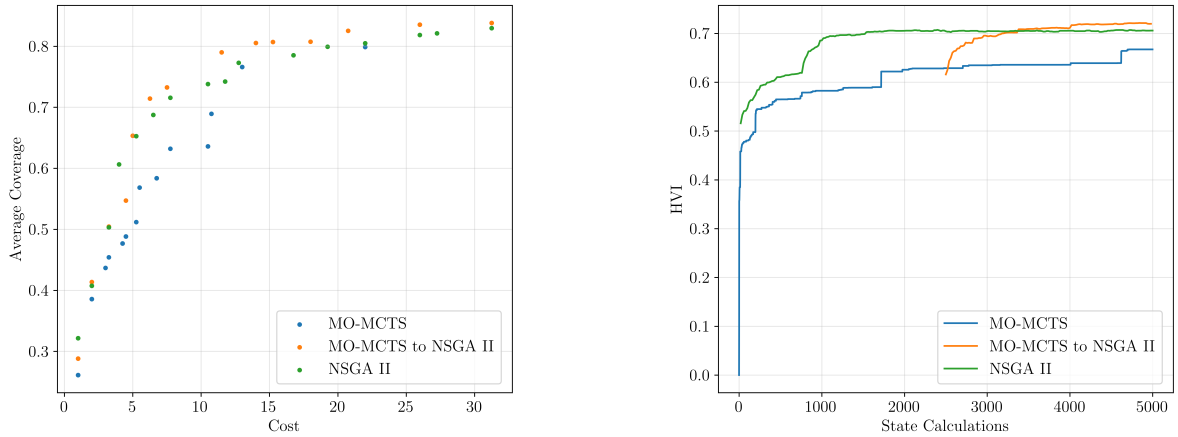
$$k = \lceil C j^{\alpha} \rceil. \tag{2}$$

The use of this heuristic method forces MCTS to reach states deep within the search tree early on, while incrementally expanding the tree horizontally.

While used frequently in two player games, MO-MCTS at its core works by efficiently searching the state space of a sequential decision problem. In an architecture design problem dealing with the placement of assets in specific locations, the order of actions taken is not typically of utmost importance. Furthermore, the actions themselves are not as important as the resultant state representing a layout. Architecture layouts are typically deterministic, as in their immediate value is apparent through evaluation of the reward function. So, while in two player game problems, Pareto-optimal actions are the desired output, in single player architecture design problems, the goal is to determine the set of Pareto-optimal states from all states visited. In our implementation, after each iteration we reevaluate the set of Pareto-optimal states, which is then used within the selection step and is ultimately our final solution set.

2.3 NSGA-II

Nondominated Sorting Genetic Algorithm II (NSGA-II), first introduced in [8], is a popular multi-objective evolutionary algorithm that implements sorting of individuals based on their levels of nondominance. It also utilizes elitism, or the process by which best performing individuals remain unaltered, while others go through the processes of crossover and mutation. NSGA-II has been shown to perform well with highly nonlinear MOOPs. In this work, we utilize the Deap Python package [10] for the implementation of NSGA-II.



(a) Pareto front comparison.

(b) Hyper-volume indicator history comparison.

Fig. 2: Optimization performance comparison between MO-MCTS, NSGA-II, and our hybrid implementation on the trajectory-oriented Cislunar architecture design problem from [16].

2.4 Hybrid MO-MCTS NSGA II

In our work, we have found that the HVI of solutions found in MO-MCTS tend to plateau rather quickly in some specific problems, and has difficulty finding more diverse solutions as iterations go on. This is likely due to the algorithm becoming stuck in localized regions in which major additions or changes to the states are unlikely to occur. On the other hand, NSGA II seems to suffer from the opposite issue. Due to its purely stochastic population initialization, NSGA II can become stuck iterating on non-optimal states, evolution out of which is difficult. We find that in the problems explored here a hybrid approach that combines the strengths of both algorithms tends to perform better long term. Therefore, for the optimization problems that follow, we first generate a set of Pareto-optimal states within MO-MCTS and after some number of iterations we pass that set off to NSGA II as its initial population. Through this, we see increased performance results when compared to both NSGA II and MO-MCTS on their own with the same amount of state evaluations. Figure 2 compares the HVI history per state calculation for equal number of state calculations between pure NSGA II, pure MO-MCTS, and the hybrid approach. The problem used for reference is the trajectory-oriented Cislunar architecture optimization from [16].

3. CISLUNAR ARCHITECTURE DESIGN THEORY

The Cislunar SDA architecture design problem is inherently a mixed-integer non-linear programming (MI-NLP) problem that becomes more complex as new objectives are added and higher levels of fidelity are simulated. For analysis purposes, it can be modeled as a multi-objective optimization problem (MOOP). Difficulties in analysis, however, arise due to the fact that no closed-form solutions exist for this problem. This necessitates the use of algorithms to generate solutions and novel methods for dynamical analysis.

Our previous work included two objectives: total architecture cost as determined by a non-linear scale and total average coverage maintained over multiple epochs with varying solar phase angle. Other objectives for consideration in analysis of Cislunar architectures include a relative angular rate objective [11], the linear orbit stability metric [27], the common logarithm of the local observability gramian [12], periodic orbit encoding angle [14], and many others explored in the literature. Many of these objectives seek to quantify the orbit determination capabilities and effective performance of an architecture, core objectives of Cislunar SDA.

While choice of objectives within a problem formulation drives the solution set of the MOOP, there is little discussion in the fundamental qualities differentiating various Cislunar SDA architectures. Due to the highly complex non-linear dynamics of the problem, there is a lack of clarity in the underlying characteristics of architecture performance. Thus, new and novel methods are needed to help gain insights into these hidden dynamics.

3.1 Cooperative Agent Using an SDA Architecture

In this work, we augment our simple performance and cost objectives with a third: minimum delta-v required by a cooperative agent using the architecture for its own goals. At a high level, this objective aims to encapsulate the goals of an architecture user by ensuring that a Cislunar SDA architecture is constructed such that it is realistically cost effective for said user with respect to its mission objectives. Obviously, each Cislunar mission has varying objectives, whether those be fuel usage, transfer time, or target requirements. With that said, it can be reasonably assumed that all these objectives are critical, perhaps with varying weights, and total delta-v can be considered to sufficiently capture an overall mission cost. Furthermore, a CA model acting optimally and in concert with an architecture acts as a useful tool for analysis, as will be discussed later.

The first step in creating these minimum delta-v trajectories is to develop a CA model that is able to interact with a Cislunar architecture within a simulation, and define its parameters and intentions. We assume that the CA model has full knowledge of a given architecture layout (state of each observers and their capabilities). The goal of the cooperative agent is perform a transfer from GEO to the L1 Lagrange point in the Earth-Moon CR3BP. The CA actuates an optimal control thrust profile to minimize delta-v usage, while also minimizing its cumulative apparent magnitude with respect to each observer in the architecture.

A unique, optimal agent transfer trajectory from GEO to L1 is generated for every node architecture configuration visited in the MCTS tree. To facilitate rapid generation of a large number of optimal transfer trajectories, the CA is restricted to planar motion in the Earth-Moon plane. However, the architecture observers are computed from full 3-DOF motion. The optimal control objective for the agent is stated in integral form as

$$J = \int_{t_0}^{t_f} \|\mathbf{u}(t)\|^2 + \chi \sum_{i=1}^{n_{obs}} \text{apmag}(\mathbf{x}(t), \mathbf{x}_{Obs,i}(t), t)\, \mathrm{d}t \tag{3}$$

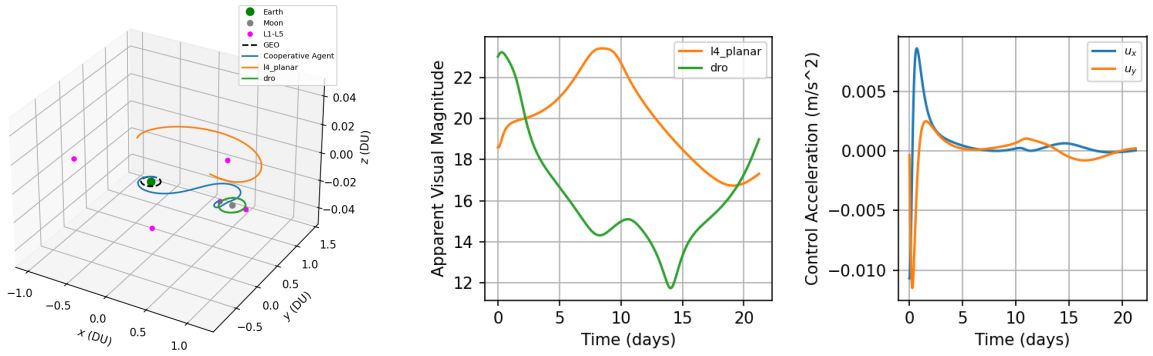
where $\mathbf{x}$ is the agent state, and $\mathbf{u}$ is the thrust acceleration input. The first term corresponds to an L^2-norm optimal control law, which reduces the overall control effort (and therefore delta-v) across the trajectory. The $\text{apmag}(\cdot)$ function computes the apparent magnitude of the agent relative to the i-th observer state $\mathbf{x}_{Obs,i}$ at time t given by [16]. The integrated total apparent magnitude over the trajectory is minimized in order to increase the likelihood of detection as seen by each observer. Note that this problem is non-autonomous, as the apparent magnitude is a direct function of time as it depends on the relative solar phase angle at each instant. Finally, a constant χ is multiplied into the objective for dimensional consistency. The value of this constant could be adjusted to place further emphasis on the control or apparent magnitude objectives.

The objective in Eqn. (3) is minimized subject to the dynamical constraints of the CR3BP system, which can be stated generally as

$$\dot{\mathbf{x}} - \mathbf{f}(\mathbf{x}, \mathbf{u}, t) = \mathbf{0} \tag{4}$$

Table 1: Summary of cooperative agent boundary conditions for generated optimal transfer trajectories (assumes planar x-y motion).

Boundary	x	y	$\dot{x}$	$\dot{y}$
Initial	$-\mu$	r_{GEO}	$-v_{\text{GEO}}$	0
Terminal	L1_x	L1_y	0	0



(a) Trajectory (b) Visual Magnitude (c) Control

Fig. 3: Sample optimal trajectory of cooperative agent for two observers (L4 Planar and DRO).

The transfer trajectory must also satisfy boundary conditions at the initial and terminal points, corresponding to GEO and L1, respectively.

$$\mathbf{c}(\mathbf{x}(t_0),\mathbf{x}(t_f),t_0,t_f) = \mathbf{0} \tag{5}$$

Boundary conditions for the transfer are given in Table 1.

A sample trajectory for the optimal Cislunar agent is illustrated in Figure 3. The transfer trajectory begins from GEO departure and ends with insertion (zero relative velocities) at L1. Both the apparent magnitude and control acceleration are also shown. Although the agent trajectory is assumed planar, incorporating full 3-DOF motion is unlikely to significantly affect apparent visual magnitude, which is assumed coincident with the x-y plane in the CR3BP system. These results illustrate a single potential architecture; however, the MOO approach must generate a new trajectory for every potential architecture visited in the tree.

3.2 Average Coverage Metric

Similar to our previous work, we use a metric that captures the ability of an architecture to maintain or obtain coverage of a trajectory within Cislunar space. We can say that a space object (SO) is accessible to an observer with an electro-optical sensor if it is visible to the observer (while considering pointing and occlusion constraints) and the observer obtains a photometric SNR of the SO greater than some predefined value. Within the architecture, we define the average coverage metric at the i^{th} simulation trajectory η_i as

$$\eta_i = \frac{n_{\text{access},i}}{N_{\text{steps},i}} \tag{6}$$

where $n_{\text{access},i}$ is the number of time steps along the SO's i^{th} trajectory that it is accessible to any observer in an architecture and $N_{\text{steps},i}$ is the total number of steps of its i^{th} trajectory. The total average coverage is then

$$\eta = \text{avg}(\{\eta_1,\eta_2,...,\eta_j\}) \tag{7}$$

for j simulation trajectories.

3.3 Architecture Cost Metric

The third metric considered in this analysis is a cost that increases with the number of observers in an architecture weighted by the size of their respective optics. An observer with a 300mm telescope is exponentially more expensive than one with a 200mm telescope, and this nonlinear cost metric reflects the nature of that increase. We weight the cost of a single observer with a telescope diameter 200mm, 300mm, or 500mm as

$$C_{200mm} = 1$$
$$C_{300mm} = 2.25$$
$$C_{500mm} = 6.25$$

The total cost of an architecture is then

$$C_{total} = C_{200mm}n_{200mm} + C_{300mm}n_{300mm} + C_{500mm}n_{500mm} \tag{8}$$

where n_{200mm}, n_{300mm}, and n_{500mm} are the number of observers with telescope apertures of 200mm, 300mm, and 500mm, respectively.

3.4 Cislunar Simulation Environment

To calculate the relevant metrics for a Cislunar architecture, we perform an in-depth simulation of the environment. In this section, we provide a brief overview of the simulation setup, constraints, and order of operations. The simulation environment consists of a few key objects. First are the observers within a given Cislunar architecture. Each observer employs an electro-optical (EO) sensor modeled after an off-the-shelf Finger Lakes Instruments Kepler KL4040 CMOS camera. When modeled with a telescope of a given diameter and optical properties, along with solar phase angle information, we can estimate the photometric signal-to-noise ratio (SNR) of a SO within Cislunar space as seen by an observer. We point the reader to [16] for further discussion on this modelling, as the steps for calculating SNR obtained from [5] are fairly involved. The calculation of SNR is performed at each time step along the SO's trajectory in the simulation, during which the average coverage metric is calculated.

Next is the generation of the SO or cooperative agent's (CA) transfer trajectory. Given an initial solar phase angle and an architecture with multiple observers in their various Cislunar periodic orbits, the CA object solves the optimal control problem as previously outlined and generates a trajectory. The resultant delta-v used for this transfer is obtained here as well.

Finally, as shown by [16], in architectures with multiple observers, the relative viewing geometry of a constellation combined with varying solar phase angles can have large impacts on performance. An architecture that provides near persistent coverage of a region at one epoch may perform significantly worse in a month's time. Furthermore, the total period of an architecture, or the time it takes for the relative positions of each observer to return to their initial orientation can be very long or hard to define due to the effects of perturbations in the environment. This is a continuing difficulty in Cislunar SDA, and any future Cislunar SDA architecture must take into account both near and long term performance. To account for this, we perform our analysis at multiple epochs over a long period of time, averaging the delta-v and coverage metrics. At each epoch within the simulation, a new CA trajectory is generated and evaluated within the architecture.

As the CA generates new trajectories, it also serves to provide us with a useful analysis tool, quantifying the existence of corridors within which varying levels of persistent detection is possible for an architecture. We expect that at any given simulation time, a Cislunar architecture can provide coverage to specific volumes of the space. A persistent detection corridor is represented by the existence of a trajectory that can realistically reside in these covered regions for the duration of its transit. Figure 4 displays this behavior with an arbitrary architecture. Along with a corresponding optimal trajectory, this analysis tool describes a time-varying volume of Cislunar space specific to an architecture layout and time. An architecture with many such corridors over time that are accessible within some range of delta-v's is one that may prove to be highly advantageous for Cislunar SDA.

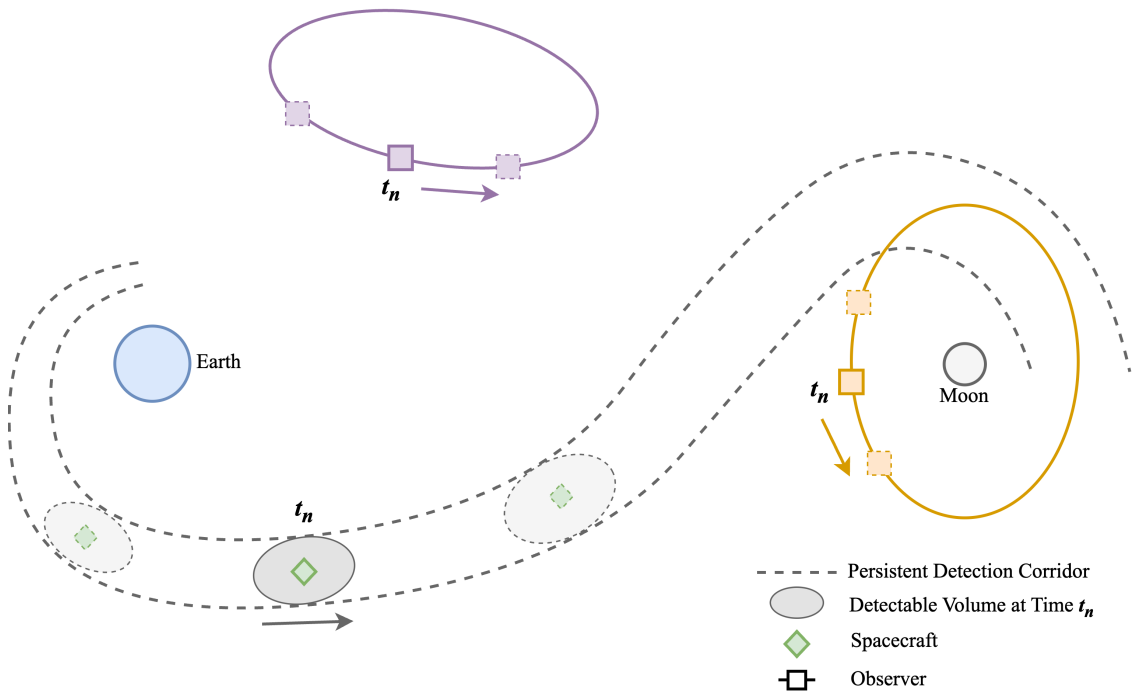


Fig. 4: A single theoretical persistent detection corridor for an arbitrary architecture. The detectable volume of space by the architecture varies with time along the spacecraft's trajectory. In reality, there may be multiple corridors each corresponding to various levels of delta-v required to remain within the corridor.

4. PROBLEM FORMULATION

4.1 Observer Orbits

While much work has been conducted analyzing the performance of individual orbits and their performance in Cisluanr SDA, here, we consider a wide variety of orbits in the Cislunar regime whose periods have approximate resonance with the Earth-Moon synodic period. Previous work has shown the SDA performance gains with EO sensors in these types of orbits when appropriately phased along their orbit, as with the solar phase angle they provide improved coverage to the region [26]. So, for this work we consider orbits that have 1:1, 1:2, 1:3, 1:4, 2:1, 3:1, 3:2, 4:3, 5:2, 5:3, 5:4, and 6:5 resonance with the synodic period. These candidate orbits include the following:

- L1 and L2 Lyapunov, Halo, Axial, and Vertical
- L4 and L5 Axial, Planar, and Long
- Distant Retrograde and Prograde, Low Prograde
- 1:2, 1:3, 2:3, 3:2 Resonant Orbits (resonance with the CR3BP period)
- Dragonfly and Butterfly Orbits
- GEO Orbit

4.2 MOOP Formulation

Here, we formally define the multi-objective optimization problem used throughout this work, as well as its implementation as an MDP. First, we rewrite each objective to be within the interval $[0, 1]$, and to be maximized. Average coverage already meets these requirements, so the cost and minimum delta-v metrics, respectively, are written as:

$$\Delta v_{normalized,i} = \max\left(1 - \frac{\Delta v_{CA}}{\Delta v_{max}}, 0\right) \tag{9}$$

$$C_{normalized} = 1 - \frac{C_{total}}{C_{max}} \tag{10}$$

where Δv_{max} is a pre-defined maximum theoretical delta-v allowed for the cooperative agent, and C_{max} is the maximum non-linear cost given a pre-defined maximum allowed number of observers within an architecture. Given that for a single architecture layout we consider multiple epochs at which to generate an optimal trajectory for the CA, we also take the average of all the normalized delta-v's for use in our optimization and call it ΔV_{total} where

$$\Delta V_{total} = \text{avg}(\{\Delta v_{normalized,1}, ..., \Delta v_{normalized,j}\}) \tag{11}$$

for j simulation epochs. Now, the optimization problem can be written as

$$\max_{\mathbf{X}} \left(C_{normalized}, \Delta V_{total}, \eta\right) \tag{12}$$

A state $\mathbf{X}$ represents an architecture, and is a tuple containing individual observers

$$\mathbf{X} = [Obs_1, Obs_2, ...] \tag{13}$$

where each observer Obs_i is a tuple containing information about the observer's position and capabilities

$$Obs_i = [Family,\ Index,\ Phasing,\ Telescope\ Diameter] \tag{14}$$

Here, *Family* indicates the type of orbit, *Index* refers to the specific orbit in the family, *Phasing* $\in [0,1]$ is the initial position of the orbit as a fraction of its period, and *Telescope Diameter* $\in [200mm, 300mm, 500mm]$ is the diameter of the observer telescope. Each observer's state can be evaluated at time t as $\mathbf{x}_{Obs_i}(t)$. *Phasing* is inherently a continuous decision variable, so for this work we discretize the variable for use within our optimization methodology. Finally, we set the maximum allowed number of observers in any given architecture to four to help alleviate computational intensity.

4.3 MDP Formulation

As in our previous work, we can formulate the optimal Cislunar architecture design problem as a sequential decision problem in the form of an MDP. First, we define the state in our MDP, which is the same as in Eqn. 13. Next are the actions. In this formulation, the actions within our MDP guide the construction of an iteration of a Cislunar SDA architecture. We define two distinct types of actions, namely, "add" actions and "change" actions. "Add" actions take an architecture and add a new randomly generated observer Obs_i to the layout. In our previous work, this type of action was quite rigid as once an observer was added to an architecture layout it could not be altered. Thus, here we include "change" actions, which act on existing observers in an architecture to alter *Phasing* or *Telescope Diameter*. The inclusion of these actions allow MO-MCTS to further refine the geometric qualities of an existing observer. During MO-MCTS, when sampling an action during the expansion step we weight the selection of "add" and "change" actions to be equal, but include logic to ensure that a single decision variable for an observer can only be changed once per architecture. This ensures that MO-MCTS does not waste time cyclically changing a single decision variable for an observer. Finally, the reward for our MDP is defined as the three dimensional vector $\mathbf{r} = [C_{normalized}, \Delta V_{total}, \eta]$ whose values are obtained for each architecture iteration from the core simulation.

4.4 Genetic Formulation

To formulate our MOOP for use within NSGA-II we introduce two distinct methods for navigating the state space, namely, crossover and mutation. Crossover involves taking two states within a population and randomly or heuristically combining them to form a new state containing decision variables from both parents. In this paper, we perform crossover on two states $\mathbf{X}_1$ and $\mathbf{X}_2$ as in Eqn. (13) by selecting a random number of individual observers from both

$\mathbf{X}_1$ and $\mathbf{X}_2$ such that the total number of observers in this set is less than our predefined maximum. The sampled observers then form a new architecture $\mathbf{X}_{child}$ whose parents are $\mathbf{X}_1$ and $\mathbf{X}_2$. Mutation works on a single state by randomly altering individual decision variables in an observer, *Phasing* and/or *TelescopeDiameter* as in Eqn. (14). The usage of crossover and mutation are dictated by both elitism within NSGA-II as well as user defined probabilities. We introduce another slight alteration wherein during mutation, we allow with some probability a completely new random architecture to be added to the NSGA-II population. In brief tests, this has been shown to encourage NSGA-II to find even more diverse solutions within our specific MOOP.

5. RESULTS

To start, we run our architecture optimization with the cooperative agent model within the hybrid algorithm for a total of roughly 1,000 state evaluations, or architecture iterations. We choose to generate CA trajectories at two epochs over approximately five years, or approximately 60 Earth-Moon synodic periods. This duration is chosen due to the fact that of all possible combinations of observer orbits in an architecture, the longest possible constellation period (total time for all observers to return to their initial relative position) is roughly 60 synodic periods. See [16] for further discussion. Due to the complexity of the CA's trajectory optimization, a single iteration can take upwards of several minutes to complete. Furthermore, the optimization performed by the CA object is highly sensitive to the dynamical environment. Future work will involve analyzing these sensitivities and reducing the total computation time of the CA trajectory generation. Regardless, successful trajectory generation is frequent and provides us with a diverse set of results.

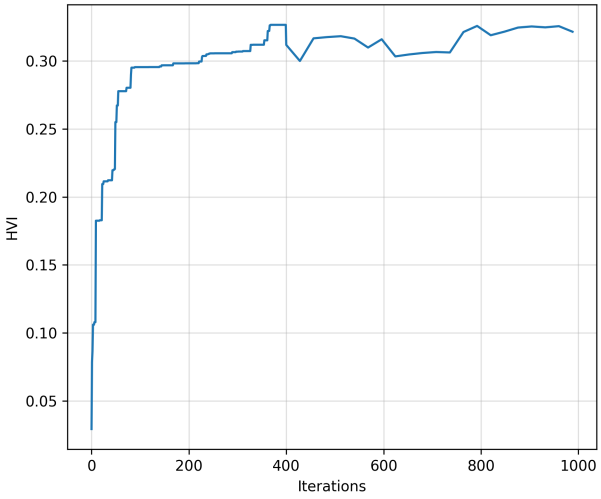
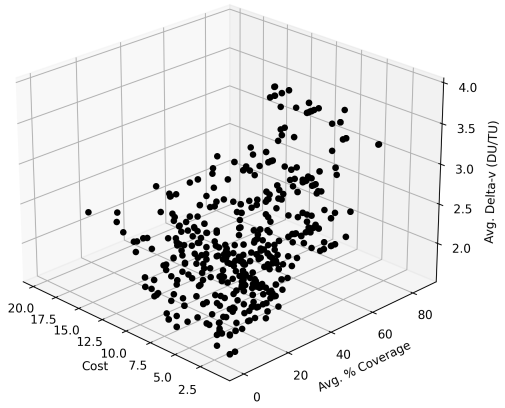


Fig. 5: Hyper-volume indicator per state evaluation/iteration.

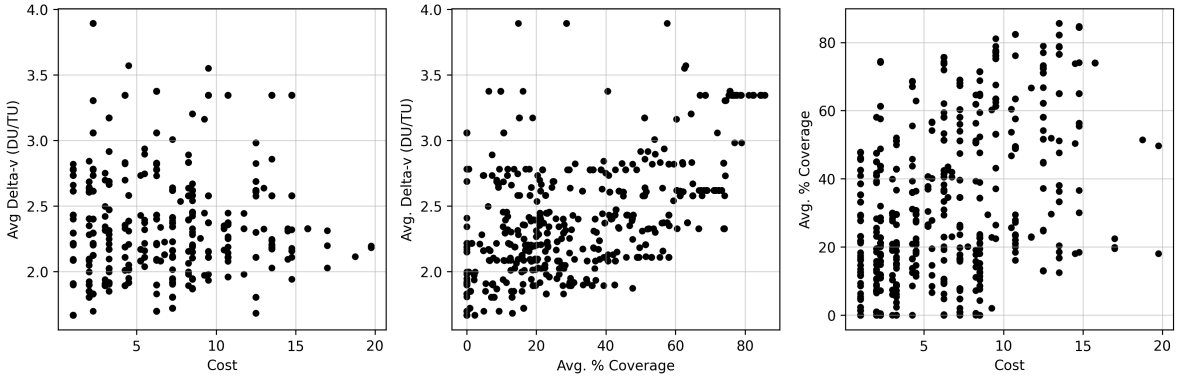
Figure 5 shows the hyper-volume history per state evaluation for this run. Of note here is the fact that the HVI seems to quickly plateau once NSGA-II begins its optimization around iteration 400 (when NSGA-II takes over). This is distinctly different than the behavior seen in Figure 2 where there is an immediate increase in the hybrid algorithm for the static trajectory-oriented Cislunar architecture design problem. This is likely due to the relatively small population size chosen for the hand-off from MO-MCTS to NSGA-II of only 28 individuals, selected due to computational intensity. Future work will include larger population sizes to reduce these fluctuations.

Figure 6 shows the diversity of points in objective space explored by our optimization run. As seen in the "Coverage vs Cost" projection plot in Figure 6b, there are a number of architectures that provide 40-80% coverage from costs of 1-10, but in no case does a single architecture of four observers provide 100% coverage. This indicates that the limit of four observers in our problem formulation may be limiting in coverage, and that the cooperative agent may be expending more delta-v to make up for the architectures' shortcomings. This behavior also helps to confirm that our CA model is acting as expected by maximizing detection by the given architecture.

The Pareto front is shown in Figure 7a and represents the trade space of solutions, with behavior we generally expect



(a) Objective history for all visited states in which a transfer trajectory was successfully generated.

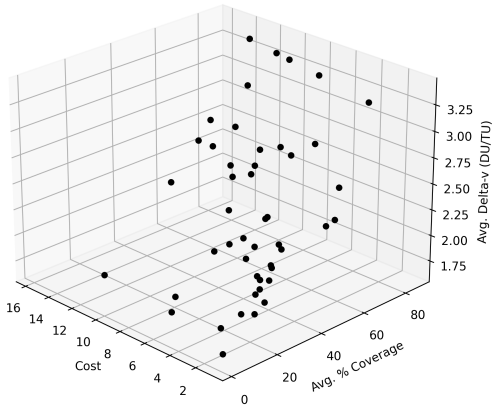


(b) Objective history projections.

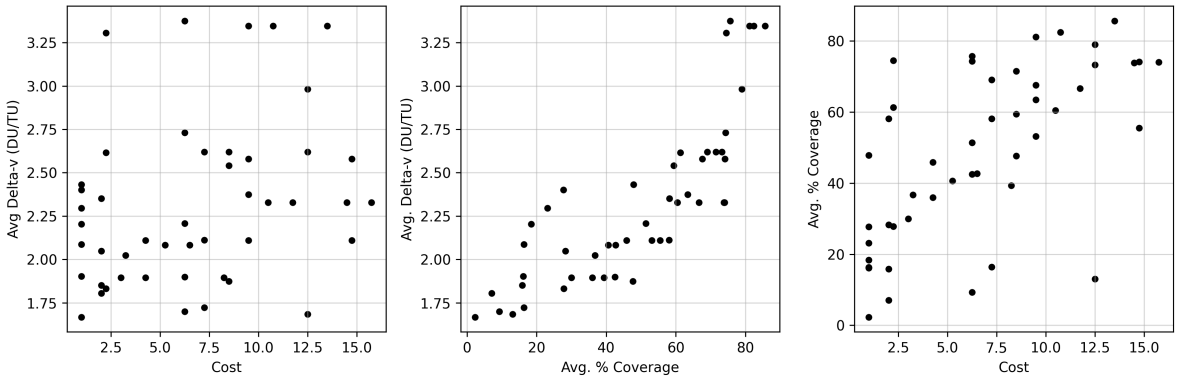
Fig. 6

from this problem. This Pareto front represents all the non-dominated solutions uncovered during the optimization process. We can dissect the trade spaces by looking at the projections of the Pareto front onto their respective axes as in Figure 7b. For example we can look at the "Coverage vs Cost" plane to see the behavior indicating that, in some cases, we may not have to incur a relatively large cost constructing an architecture to maintain near perfect coverage of a cooperative agent. However, as seen in the "Delta-V vs Cost" and "Delta-V vs Coverage" planes, such a low cost architecture while maintaining high coverage tends to require a significantly greater delta-v expenditure by the cooperative agent. On the other hand, if we wanted to reduce the total effort required by the cooperative agent while maintaining sufficient coverage we would have to construct a significantly more costly architecture. Even so, there are few architectures here that provide coverage over 75%. By increasing the maximum allowed number of observers per iteration up from four we may see even more improvements in CA delta-v usage, at the expense of a more costly architectures. Regardless, these results are intuitive, and further emphasize the importance of considering all agents' needs within a Cislunar SDA architecture. If we can construct an architecture that provides guaranteed coverage for a cooperative agent through ensuring the existence of persistent detection corridors then we can lower overall cost of our architecture.

Figure 9 shows a number of the resultant architectures from the Pareto front as well as the optimal trajectory generated by the cooperative agent at a single simulation epoch. As in [16], there seems to be a shared quality of these architectures where each observer is spread out in space. Most notably, however, are the diversities of transfer geometries created by the CA in its attempt to minimize it cost function in Eqn. (3). Most trajectories follow the same general "S" curve in their transfer to L1 in part due to the constraints on the initial position being in a GEO orbit. By varying this constraint in future work we may see a broader diversity in transfer geometry. Regardless, there are distinct differences among the trajectories recovered, and we can see that the cooperative agent expends significant effort in some cases to ensure detection by the observers. These transfers can potentially tell us a lot about the underlying dynamical interactions between observers in an architecture, the solar phase angle (in the case of electro-optical measurements), and the cooperative agent.



(a) Final Pareto-optimal solutions in objective space.



(b) Final Pareto front solutions in objective space with their projections onto each axis.

Fig. 7

While in [16] we saw frequent use of L1 and L2 Lyapunov orbits in the solution space, here we see frequent usage of L4 and L5 axial orbits as shown in Figure 8. These L4 and L5 orbits traverse a large volume of Cislunar space, allowing for the cooperative agent more freedom to traverse the region while uncovering unique persistent detection corridors. Furthermore, an observer not being restricted to the (x,y) plane while providing coverage to a planar trajectory likely reduces the impact of Earth, Moon, and Sun pointing constraints. It may also be that at the two simulation epochs

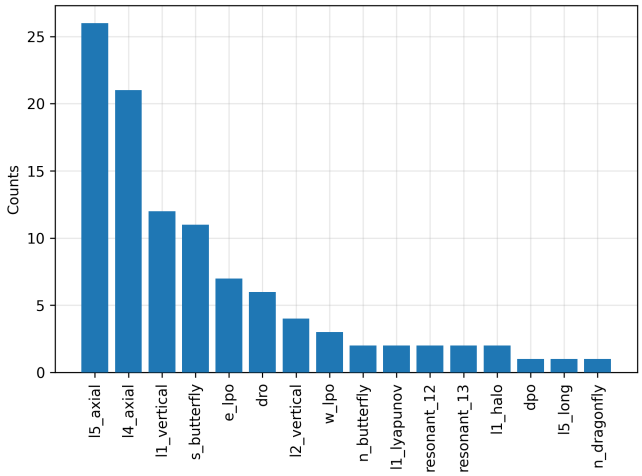


Fig. 8: Counts of orbit families used by observers within the Pareto-optimal architectures.

at which trajectory optimization was performed, the solar phase angle was oriented in such a way that the viewing geometry from near the L4 and L5 Lagrange points was more conducive to detection of the cooperative agent. In this sense, evaluating architectures over only two epochs is limiting in that we are not able to fully evaluate the relative geometry between the observers and the cooperative agent over a long period of time. This discrepancy of selected orbits will be explored in future work in part by evaluating architecture layouts at more epochs.

The collection of trajectories generated by the CA across all iterations is shown in Figure 10. Table 2 contains a summary of the resultant delta-v's from Pareto-optimal solutions. Based on the delta-v's and transfer times, these trajectories are reminiscent of direct lunar transfers [20] like those used in the Apollo missions and Artemis 1. These types of missions are expensive, and there is growing interest in low-energy transfers using ballistic lunar transfers such as the CAPSTONE mission, with a total duration of about 140 days. Furthermore, the trajectory optimization process performed by the cooperative agent is inherently multi-objective as well, and in this work we do not perform an analysis of the trade space between delta-v, transfer duration, and coverage when selecting a trajectory. In future work, we will continue to improve the trajectory optimization method based on the breadth of literature in this subject in order to create more representative architecture solutions [23, 21, 18].

Table 2: Summary of cooperative agent transfer delta-v's and durations. These correspond to direct, high-energy transfers similar to those used in the Apollo missions and Artemis 1.

	Δv	Duration
	km/s (DU/TU)	*days (TU)*
Min	1.709 (1.668)	5.250 (1.209)
Avg	2.487 (2.427)	12.057 (2.777)
Max	3.587 (3.501)	21.708 (4.999)

6. ANALYSIS

As seen in Figures 9 and 10, through the iterative process of constructing architectures, we uncover a diversity of transfer trajectories with varying levels of delta-v, each uniquely optimized for a specific architecture. We wish to explore methods that utilize these trajectory-architecture pairs to better understand the persistent detection corridors uncovered by the trajectories. These corridors are a result of the complex interactions between the cooperative agent, the architecture, the solar phase angle, and the underlying dynamics from the CR3BP. Here, we outline a clustering method inspired by [3] that will aid in categorizing Cislunar architectures based on the geometries of cooperative agents' trajectories as well as the solar phase angle and viewing constraints.

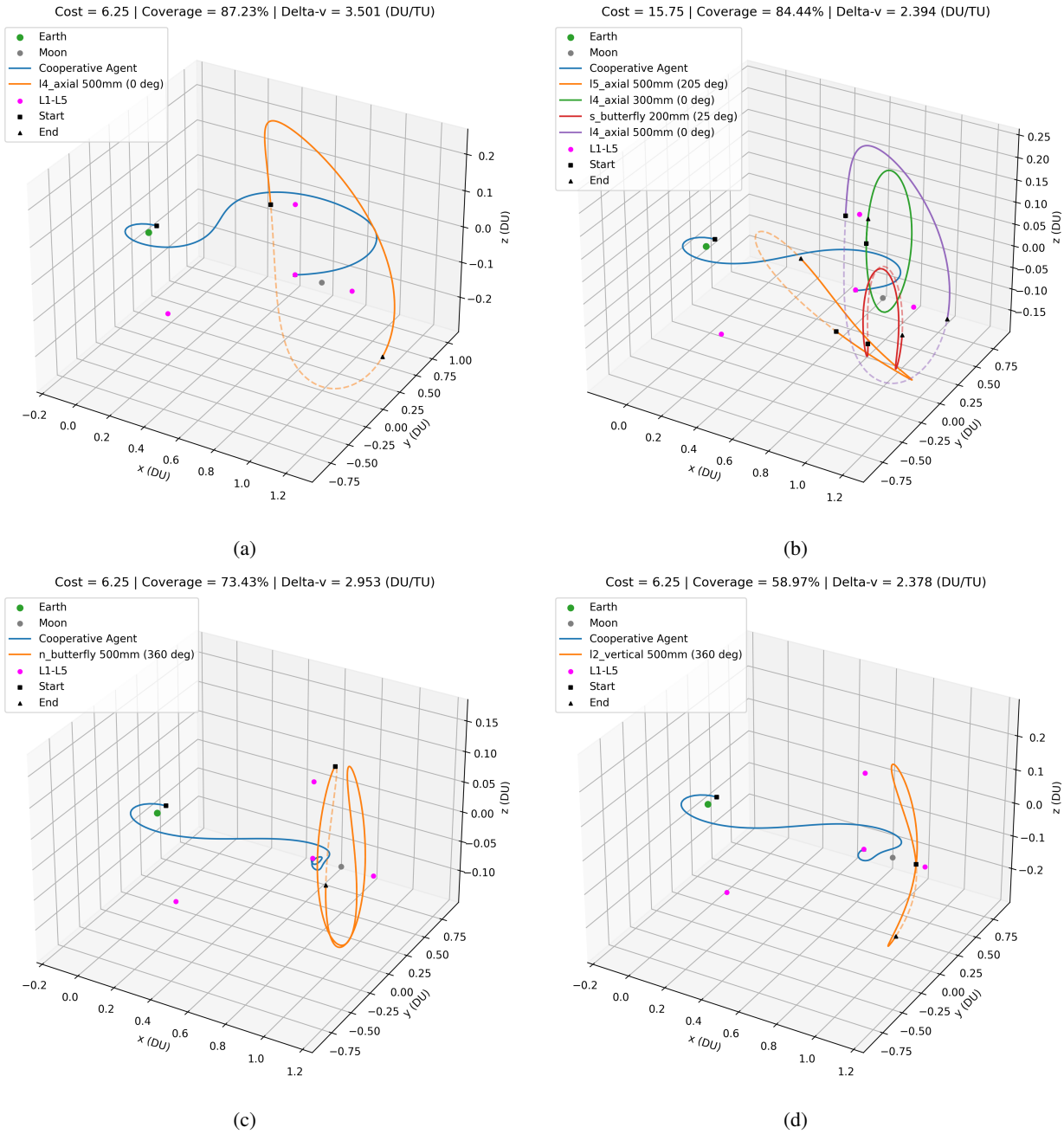
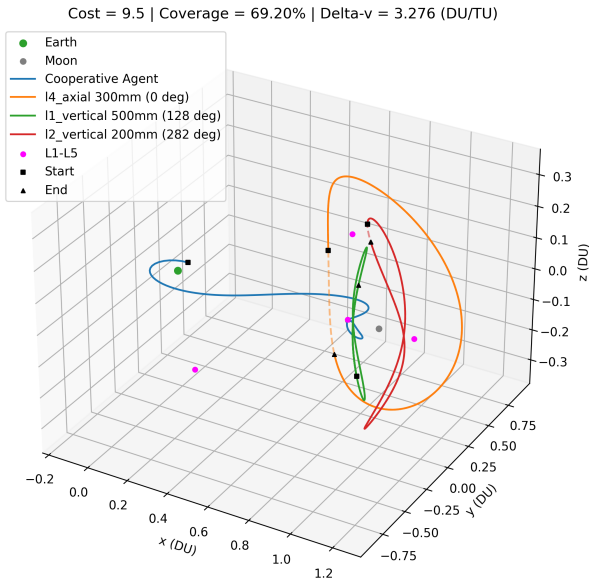
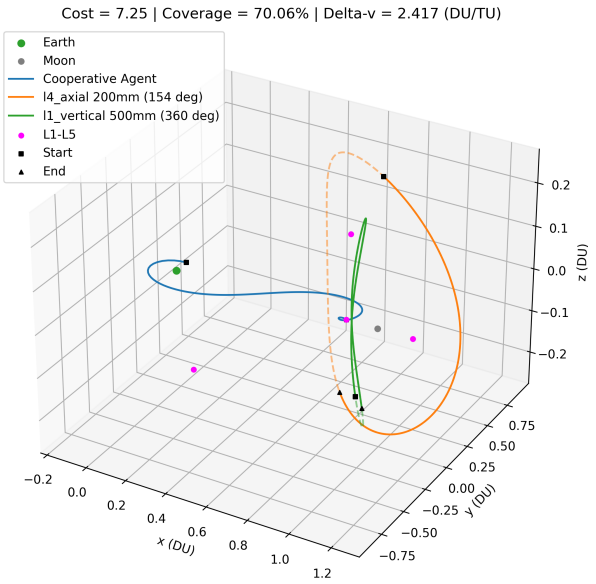


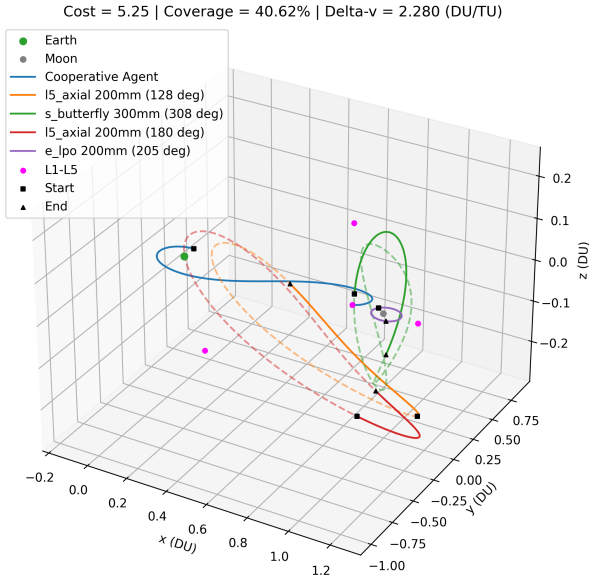
Fig. 9: Selection of trajectories and architecture layouts at a single epoch from the set of Pareto-optimal solutions. Coverage and delta-v values correspond to the individual epoch shown and not the over all epochs considered for optimization. Observer state histories during the transfer are shown in solid, with their full orbits shown in dashes. The relative phasing of observer orbits is given in the legend. Continued on next page.



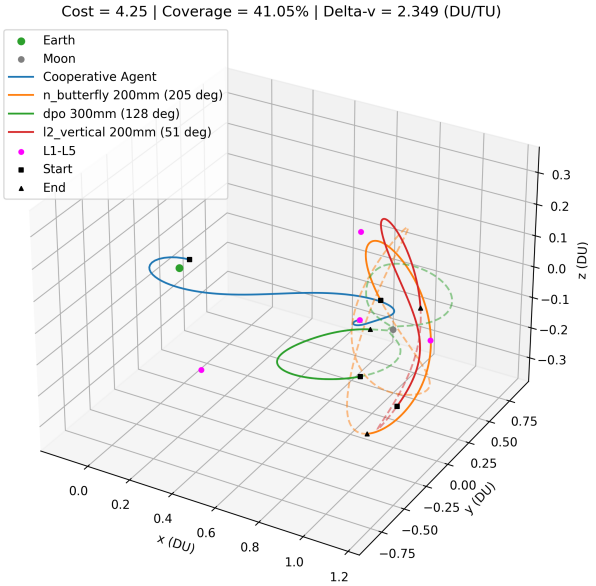
(e)



(f)



(g)



(h)

Fig. 9: Selection of trajectories and architecture layouts at a single epoch from the set of Pareto-optimal solutions. Coverage and delta-v values correspond to the individual epoch shown and not the over all epochs considered for optimization. Observer state histories during the transfer are shown in solid, with their full orbits shown in dashes. The relative phasing of observer orbits is given in the legend.

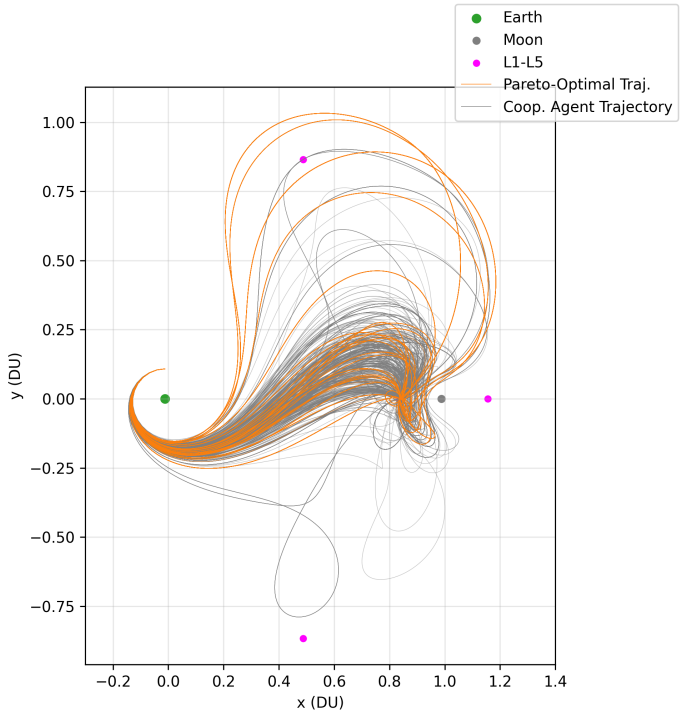


Fig. 10: All trajectories generated throughout the run with those corresponding to Pareto optimal solutions highlighted in orange.

Similar to [3], we begin analysis by generating the relevant data with which to eventually perform clustering. In [3] apse locations were found to be useful traits to help characterize the geometry of a purely ballistic trajectory in the CR3BP. Here, we propose analyzing the apoapses of our transfers and their relation to other qualities of a Cislunar architecture. An apoapsis occurs in a transfer given:

$$\begin{aligned} &(x - x_{body})\dot{x} + y\dot{y} + z\dot{z} = 0 \qquad \text{and} \\ &(x - x_{body})\ddot{x} + y\ddot{y} + z\ddot{z} + \dot{x}^2 + \dot{y}^2 + \dot{z}^2 < 0 \end{aligned} \tag{15}$$

An apogee occurs when $x_{body} = -\mu$ and an apolune when $x_{body} = 1 - \mu$, corresponding to an apoapsis about the Earth and the Moon, respectively. Since our trajectories rely on a control input at each step during integration, we use a numerical method to recover the approximate positions. The results of this calculation for all trajectories generated through optimization are shown in Figure 11. We note that not every trajectory generated has an apopasis with respect to the Earth, Moon, or both.

The location of an apoapsis helps define the general geometry of each transfer trajectory, but we must also look for metrics that captures the relationship of that location to the geometry of an architecture. To start, we may look to the time averaged mean position of an architecture over the duration of an individual transfer. From first principles, and without considering the solar phase angle, an architecture will generally perform better when a cooperative agent can minimize its distance from any observer at any given time. The time averaged mean architecture position for architecture $\mathbf{X}_k$ for epoch j at time t_j and transfer duration T_j with N_j steps is defined as the first three state elements (position) from

$$\Gamma_{k,j} = \frac{1}{N_j} \sum_{t=t_j}^{T_j} \frac{1}{N_{Obs}} \sum_{i=1}^{N_{Obs}} \mathbf{x}_{Obs_i}(t) \tag{16}$$

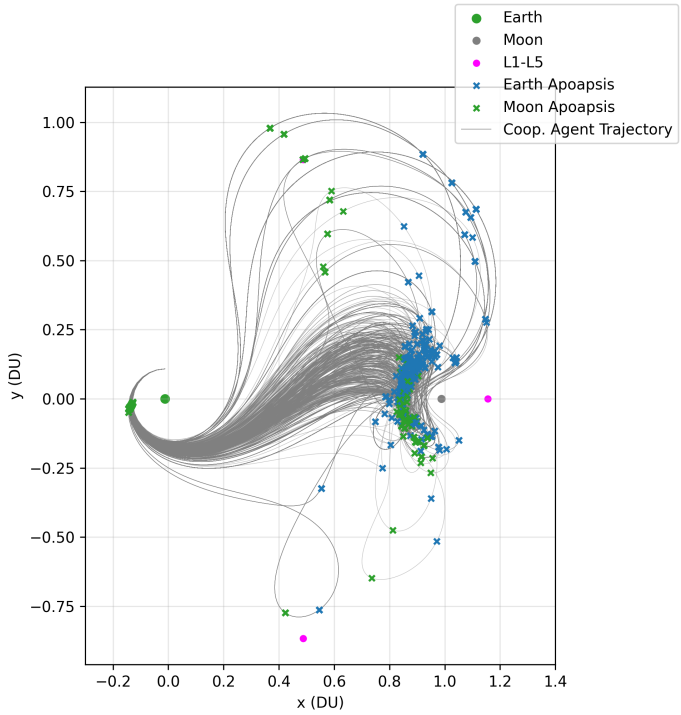


Fig. 11: Apoapses along cooperative agent trajectories. Only a single apoapsis for the Earth and Moon is shown for each trajectory, if one exists.

where N_{Obs} is the number of observers in the architecture, and the summations are performed element wise on the observer state. As we might expect, in Figure 12, we see a high density of time averaged mean positions around the L1 and L2 Lagrange points, with some spread in and out of plane.

Along with the mean architecture positions and apoapsis locations, we might also consider the resultant coverage and delta-v per j^{th} trajectory corresponding to the k^{th} architecture iteration, $\eta_{k,j}$ and $\Delta v_{CA,k,j}$. By putting together these parameters, we can construct a vector $\mathbf{R}_k$ containing basic geometric information about an architecture, its performance, and the cooperative agent's transfer trajectory across all j simulation epochs:

$$\mathbf{R}_k = \left[\mathbf{x}_{apogee,k,1}, ..., \mathbf{x}_{apogee,k,j}, \mathbf{x}_{apolune,k,1}, ..., \mathbf{x}_{apolune,k,j}, \Gamma_{k,1}, ..., \Gamma_{k,j}, \eta_{k,1}, ..., \eta_{k,j}, \Delta v_{CA,k,1}, ..., \Delta v_{CA,k,j}\right] \tag{17}$$

where $\mathbf{x}_{apogee,k,j}$ and $\mathbf{x}_{apolune,k,j}$ are the locations of the apogee and apolune at the j^{th} epoch for the k^{th} architecture, respectively. In future work we will perform clustering on these data to attempt to categorize the various architectures and trajectories from these results. Significantly more effort is needed on this method of analysis, as well as further exploration into metrics that reflect the dynamics of the solar phase angle as it plays a fundamental role in accessibility in the Cislunar regime.

7. CONCLUSION

In this work, we presented a novel objective for the Cislunar SDA architecture design problem through the implementation of a cooperative agent generating transfer trajectories that optimize for its relative apparent magnitude as seen by observers within an architecture. Through iteratatively constructing architectures with the additional objective of minimizing the cooperative agent's delta-v, we recovered a set of Pareto-optimal architectures that not only provided persistent coverage of the CA at varying costs, but also took into account the relative cost a user would incur by relying on the architecture for mission objectives. Furthermore, we introduced the concept of persistent detection corridors uncovered by our cooperative agent model, within which an architecture can provide persistent coverage of a space

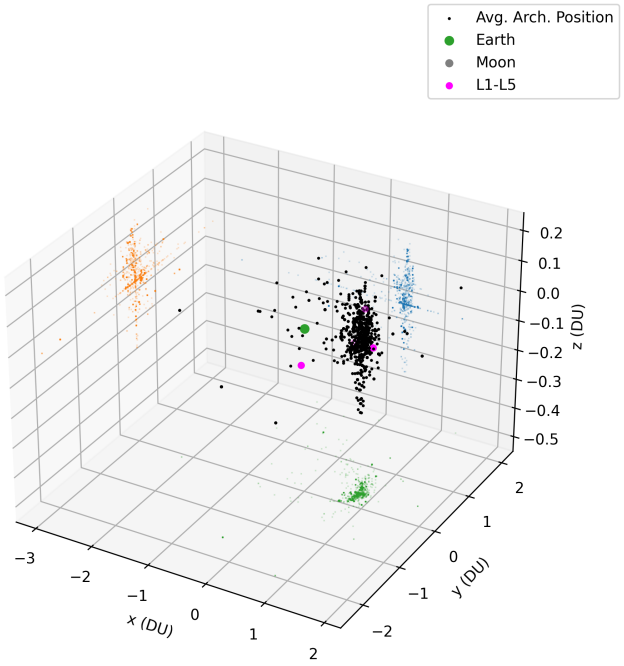


Fig. 12: Time averaged mean position of architectures at each simulation epoch for the duration of its corresponding cooperative agent trajectory with projections onto their respective axes.

object along a trajectory dictated by a required mission delta-v. Finally, we explored a selection of Pareto-optimal architecture designs and their optimal transfers, and laid the groundwork for future cluster-based analysis of Cislunar architectures. Through this work we have provided critical analysis of the complex relationships between competing objectives within the architecture design process, with an emphasis on the inclusion of cooperative agents and their goals within the Cislunar domain. Future work will include a continuation of this work, including deeper dives into the performance of our optimization methods, additional objectives, and refinement of the cluster-based analysis methodology for the categorization of architectures.

8. ACKNOWLEDGEMENTS

The authors would like to thank Ball Aerospace for the funding on this project, Casey Heidrich for his work on the cooperative agent model, as well as David Barker for his help in creating methods for storing data from long optimization runs.

9. REFERENCES

[1] Surabhi Bhadauria and Carolin Frueh. Optical observation regions in cislunar space using the bi-circular restricted four body problem geometry. 2022.

[2] J. Blank and K. Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[3] Natasha Bosanac. Data-mining approach to poincaré maps in multi-body trajectory design. *Journal of Guidance, Control, and Dynamics*, 43(6):1190–1200, 2020.

[4] Cislunar Technology Strategy Interagency Working Group. *National Cislunar Science and Technology Strategy*. National Science and Technology Council, 2022.

[5] Ryan D Coder and Marcus J Holzinger. Multi-objective design of optical systems for space situational awareness. *Acta Astronautica*, 128:669–684, 2016.

[6] Adrien Couëtoux, Jean-Baptiste Hoock, Nataliya Sokolovska, Olivier Teytaud, and Nicolas Bonnard. Continuous upper confidence trees. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5*, pages 433–445. Springer, 2011.

[7] Jacob A Dahlke, Adam P Wilmer, and Robert A Bettinger. Preliminary comparative assessment of l2 and l3 surveillance using select cislunar periodic orbits. In *AAS/AIAA Astrodynamics Specialist Conference*, pages 1–19, 2022.

[8] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[9] Samuel Fedeler, Marcus Holzinger, and William Whitacre. Sensor tasking in the cislunar regime using monte carlo tree search. *Advances in Space Research*, 70(3):792–811, 2022.

[10] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

[11] Erin E Fowler, Stella B Hurtt, and Derek A Paley. Orbit design for cislunar space domain awareness. In *2nd IAA Conference on Space Situational Awareness (ICSSA), Washington, District of Columbia*, 2020.

[12] Erin E Fowler and Derek A Paley. Observability metrics for space-based cislunar domain awareness. *The Journal of the Astronautical Sciences*, 70(2):10, 2023.

[13] C Frueh, K Howell, K DeMars, S Bhadauria, and M Gupta. Cislunar space traffic management: Surveillance through earth-moon resonance orbits. In *8th European Conference on Space Debris*, volume 8, 2021.

[14] Maaninee Gupta, Kathleen Howell, and Carolin Frueh. Constellation design to support cislunar surveillance leveraging sidereal resonant orbits. 01 2023.

[15] Maaninee Gupta, Kathleen C Howell, and Carolin Frueh. Long-term cislunar surveillance via multi-body resonant trajectories. In *AAS/AIAA Astrodynamics Specialist Conference*, 2022.

[16] Michael Klonowski, Marcus J Holzinger, and Naomi Owens Fahrner. Optimal cislunar architecture design using monte carlo tree search methods. *The Journal of the Astronautical Sciences*, 70(3):17, 2023.

[17] Levente Kocsis, Csaba Szepesvári, and Jan Willemson. Improved monte-carlo search. *Univ. Tartu, Estonia, Tech. Rep*, 1:1–22, 2006.

[18] Hanlun Lei and Bo Xu. Low-energy transfers to cislunar periodic orbits visiting triangular libration points. *Communications in Nonlinear Science and Numerical Simulation*, 54:466–481, 2018.

[19] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26:369–395, 2004.

[20] Jeffrey S Parker and Rodney L Anderson. *Low-energy lunar trajectory design*, volume 12. John Wiley & Sons, 2014.

[21] Robert E Pritchett, Emily Zimovan, and Kathleen Howell. Impulsive and low-thrust transfer design between stable and nearly-stable periodic orbits in the restricted problem. In *2018 Space Flight Mechanics Meeting*, page 1690, 2018.

[22] Peng Mun Siew, Daniel Jang, Thomas G Roberts, Richard Linares, and Justin Fletcher. Cislunar space situational awareness sensor tasking using deep reinforcement learning agents. 2022.

[23] Christopher John Sullivan, Natasha Bosanac, Alinda Kenyana Mashiku, and Rodney L Anderson. Multi-objective reinforcement learning for low-thrust transfer design between libration point orbits. In *2021 AAS/AIAA Astrodynamics Specialist Conference*, 2021.

[24] Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mańdziuk. Monte carlo tree search: A review of recent modifications and applications. *arXiv preprint arXiv:2103.04931*, 2021.

[25] Johnathan Tucker, Jackson Wagner, and Zachary Sunberg. Adaptive stress testing applied to space domain awareness systems. 2022.

[26] Jacob K Vendl and Marcus J Holzinger. Cislunar periodic orbit analysis for persistent space object detection capability. *Journal of Spacecraft and Rockets*, 58(4):1174–1185, 2021.

[27] Lois Visonneau, Yuri Shimane, and Koki Ho. Optimizing multi-spacecraft cislunar space domain awareness systems via hidden-genes genetic algorithm. *arXiv preprint arXiv:2302.09732*, 2023.

[28] Weijia Wang and Michele Sebag. Multi-objective monte-carlo tree search. In *Asian conference on machine learning*, pages 507–522. PMLR, 2012.