A Comprehensive Approach to Optimized Cislunar Architecture Design Utilizing Capacity

Justin Kim
BAE Systems, Inc.
Naomi Owens-Fahrner
BAE Systems, Inc.

ABSTRACT

Cislunar Space Domain Awareness (SDA) has gained significant attention due to growing interest from government, scientific, and commercial sectors. However the complex dynamics, large design space, and lack of standard design methods present a significant challenge to generating effective architectures. Prior work has focused on optimizing architectures for observability (access), but did not evaluate other critical measures of performance. This paper presents a framework and method to simultaneously optimize observability and capacity. The resulting sensor tasking-related metrics more realistically assess mission performance. This study formulates the architecture optimization problem as a Markov Decision Process (MDP) that is solved with a multi-objective Monte Carlo Tree Search (MCTS). The capacity optimization problem is formulated as a grid search MDP in which the sensor explores the cislunar volume with an ε-greedy algorithm. By expanding the optimization problem with critical objectives and constraints, the simulation method provides a more comprehensive and streamlined approach to architecture design for cislunar SDA. In this work we present a comprehensive, viable, and extensible framework for the high-level conceptual design and optimization of cislunar SDA architectures.

1. INTRODUCTION

Ground-based networks for near-Earth SDA are ill-equipped to perform cislunar SDA due to the large volumes, complex dynamics, and observation constraints in the cislunar environment. An effective cislunar SDA architecture will likely require some combination of collaborative in-space assets [1]; however, the large design space makes this a challenging problem to address. In particular, the widely varying properties of orbit families in the cislunar regime makes orbit selection far from obvious [2]. While the desire and need for SDA solutions in the cislunar region is clear, at present there is no standard approach to design, evaluate, and optimize such architectures [3].

Prior work has formulated the architecture design problem as a multi-objective optimization problem that attempts to maximize observability of a volume [4] [5] [3]. Observability is the theoretical Field of Regard (FOR), subject to geometric and radiometric constraints. In the context of this problem, it is the percentage of the volume that is visible to the architecture over time, and has been considered the standard metric for optimizing cislunar surveillance networks [6]. Solutions to this problem have been demonstrated with Monte Carlo Tree Search, branch & bound, particle swarm, simulated annealing, and other evolutionary algorithms [5] [7]. However, we posit that observability alone is not sufficient to assess mission performance and realistically evaluate architectures.

Capacity is a measure of performance defined by the percentage of the volume that is observed over time by the architecture subject to sensor FOV, agility, and tasking constraints, in addition to the observability constraints. Fig. 1 illustrates the difference between observability and capacity. For observability, the entire cardioid FOR is observable except for an angular keep-out zone around the Earth. For capacity, coverage is determined by the sensor's slew path over time, which is constrained by sensor performance parameters such as agility, integration time, and FOV. Incorporating these factors into the architecture optimization problem is critical to realistically assess mission performance. A capacity-based cislunar SDA solution is explored by Owens-Fahrner et al. [8], in which the authors initially optimize and select architectures for observability, and then perform a capacity analysis on the best-in-class architectures. However, this process requires extensive domain knowledge and human-in-the-loop effort to generate and analyze the architectures.

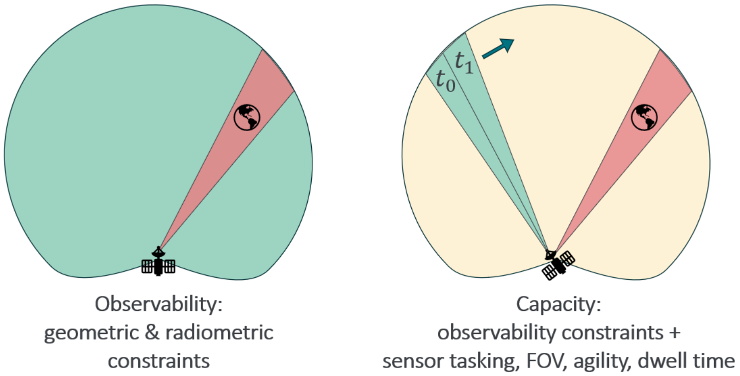
t_0
t_1
Observability:
geometric & radiometric
constraints
Capacity:
observability constraints +
sensor tasking, FOV, agility, dwell time

Fig. 1: Coverage as a function of observability vs capacity

This paper expands on such work by simultaneously optimizing for observability and capacity. Additional constraints and objectives are imposed on the optimization problem such as orbit stability, scheduling metrics, and a higher fidelity radiometric model. Challenges in this problem include the three-body orbital dynamics, line-of-sight constraints, large volumes, and low SNR [9] [10]. We address such challenges with a comprehensive simulation method that informs some of the trades that must be considered in optimizing an architecture for cislunar SDA.

As a mixed-integer, multi-objective, non-linear optimization problem, the architecture optimization problem is formulated as a Markov Decision Process (MDP) as in [5]. A Monte Carlo Tree Search solution was demonstrated, which we expand upon by including sensor tasking optimization and related objectives. Past studies have explored some these objectives as isolated pieces, while in this work the objective space of the problem is expanded to include capacity optimization and sensor tasking analysis, which provide metrics and constraints that are critical for assessing architectures and evaluating mission performance. Simultaneously optimizing for observability and capacity reduces the effort and mission knowledge required to evaluate cislunar architectures. With this streamlined method, we have developed a comprehensive framework that more accurately reflects the complex dynamics and expansive trade space of the problem.

2. BACKGROUND

2.1 Markov Decision Process

Both the architecture optimization and capacity optimization problems are formulated as Markov Decision Processes (MDP). An MDP is a discrete-time stochastic control process and mathematical framework used to model decision making. It is represented by the 5-tuple (S, A, T, R, γ) with state space S, action space A, state transition function T, reward function R, and discount factor γ. MDPs are memory-less processes, meaning that evolution of the state is dependent only on the current state, and none of the preceding states. In general an actor, or agent, at state s may perform an action that induces a transition to a new state s' according to the transition function. The state-action pair and associated new state produce an associated reward. The goal of an MDP is to discover a policy $\pi(s)$ that maps states to actions such that the cumulative future reward of the process is maximized. A number of different algorithms can be used to solve an MDP - in this work the architecture optimization problem is solved via Monte-Carlo Tree Search, and the capacity optimization problem is solved via ε-greedy search.

2.2 Monte-Carlo Tree Search

Monte-Carlo Tree Search is a heuristic search algorithm that can generally be used to solve sequential decision processes; its application to this problem is described in [5] and is briefly described here. The state of the architecture optimization problem is defined by the architecture parameters - namely, the number of observers and their orbital parameters. An architecture is composed of a set of observers as shown in Eq. 1, with each observer described by a tuple in Eq. 2 that describes the observer's orbit. In this work sensor parameters are homogeneous across the architecture and are not represented in the state space of the problem.

There are only two possible actions in the action space: add another observer, or modify an existing observer's orbit.

Rewards are calculated in the simulation step of Fig. 3, and are an output of the capacity optimization process. These rewards are used to assess the quality of an architecture and its membership in the Pareto-optimal set of solutions. Nodes of the tree are architecture states, which describe the constituent observer orbits. Branches between nodes represent the actions and transitions between states. There are four main steps in the Monte-Carlo Tree Search, described below:

$$X = [Obs_1, Obs_2, ...] \tag{1}$$

$$Obs_i = [Orbit\,Family, Member, Phase] \tag{2}$$

1. Selection Starting from a root node, traverse the tree via some exploration vs exploitation policy until an unexpanded node is found.

2. Expansion If the unexpanded node is not terminal, expand it by creating a child node with a chosen action.

3. Simulation Simulate the node (architecture) and compute the associated reward. This is where capacity optimization is performed, with reward metrics output from the sensor tasking optimization.

4. Backpropagation Update the estimated node value and visit count on the visited branch.

In this framework the root node can be selected by the decision maker, randomly set, or initialized as an "empty" architecture. The exploration vs exploitation policy used in this work is the Upper Confidence Bounds for Trees (UCT) [11]. This policy assesses the value of a node along with the number of times it has been visited, and attempts to balance the exploration of new states with exploitation of known promising states. It is defined in Eq. 3 where $Q(s,a)$ is the 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 action a has been taken at state s, and C is a constant. The first term represents the exploitation of known states, while the second state represents the preference for exploring states that have not frequently been visited.

$$Q(s,a) + C\sqrt{\frac{lnN(s)}{N(s,a)}} \tag{3}$$

Because of the large action space this tree is said to have a high branching factor, which may result in the tree search being "shallow". Expansion of a node is constrained by Progressive Widening (PW) [12], which determines whether a node is expanded or not by limiting the number of actions that can be taken from a node before it is considered "expanded". This encourages the tree search to reach states deep within the tree early on, while incrementally expanding the tree horizontally [13]. Hyperparameters in the UCT and PW may be tuned to favor depth or breadth in the tree search, i.e. exploitation vs exploration.

2.3 Pareto-optimal Set Evaluation

As a multi-objective optimization problem, determining a single optimal solution is typically not possible. Due to competing objectives, any number of solutions could be considered optimal depending on the decision maker's priorities. The complete set of optimal solutions is known as the Pareto frontier. The frontier could potentially be infinite in size, so we identify a subset of the Pareto frontier known as the Pareto-optimal set of solutions. The solutions in the set, also known as non-dominated solutions, are typically determined by their proximity to the Pareto frontier in addition to their diversity and spread in the objective space.

In this problem the Pareto-optimal set is determined by a set-quality indicator known as the Hypervolume Indicator (HVI) as defined in [14]. The HVI maps a point set to the region of the objective space that is weakly dominated by that set, and bounded below from a given reference point (assuming maximization). In simpler terms it can be thought of as the size of the space covered by the set in objective space. The HVI is widely used as a set-quality indicator due to its simplicity and strict monotonicity with respect to set dominance, and its ability to capture set characteristics in a single real value [14]. Fig. 2 shows a notional HVI in two-dimensional objective space - the HVI (in this case an area) is upper-bounded by the frontier of non-dominated solutions, and lower-bounded by an arbitrary reference point.

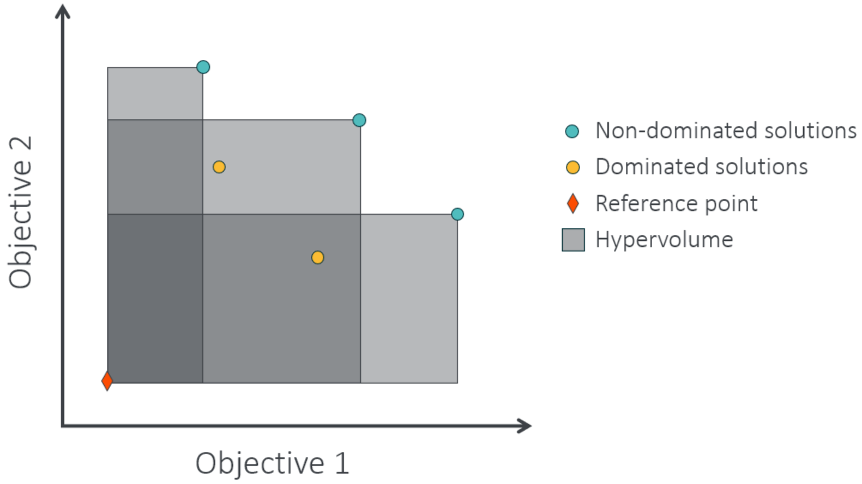

Fig. 2: HVI in two-dimensional objective space

As the tree search progresses, new solutions are evaluated and mapped onto the objective space - if a new solution marginally contributes to and increases the size of the HVI, then it is considered Pareto-optimal. Solutions that do not contribute to the HVI are considered dominated and are not included in the Pareto-optimal set. Due to its simplicity, the HVI extends with trivial ease to n-dimensional optimization problems. As described in [5] the HVI is incorporated in the UCT shown in Eq. 4, so that the policy optimizes the *set* of best solutions, as opposed to any single solution. Now that the state-action value estimate includes the HVI, the selection policy is compelled to act such that the hypervolume of the union of the Pareto-optimal set P with the state-action value estimate is maximized.

$$HV(Q(s,a) \cup P) + C\sqrt{\frac{lnN(s)}{N(s,a)}} \tag{4}$$

3. METHOD

The central problem being addressed in this work is the optimal design of architectures for cislunar SDA. The prevailing standard for optimizing cislunar surveillance networks has been to use observability to assess architectures [6], and a general method to address the optimization problem with this metric is described in the previous section. In this work we demonstrate a critical enhancement to the method, which is to include capacity optimization and sensor tasking metrics to the architecture optimization problem, thus providing critical performance metrics that are needed to realistically evaluate mission architectures. This section will detail the augmented method that includes the capacity optimization routine and sensor tasking metrics.

The first step is to define the scenario. Any search volume may be used that is described by a set of static points in cislunar space, as in Fig. 6. The volume of interest, point placement, and point density are easily modified and up to the user to determine. The simulation time is defined and describes temporal parameters for the volume search and capacity optimization, such as the starting epoch, volume search time, and time step. Viewing constraints for observability are defined here, and typically include geometric and radiometric constraints. Sensor and target parameters are used to determine visual magnitude (mv), SNR, and agility for use in the capacity optimization. This work has incorporated higher-fidelity BAE radiometric models in case detailed SNR calculations are required; for the purpose of this work, a simple mv constraint is used to model detectability.

The next step is to define the architecture MDP parameters. The design space space of the architecture problem is determined by the number of observers and orbits that can populate an architecture. This is arbitrarily limited according to the user's preferences. Objective metrics must also be defined for the optimization problem to calculate rewards in the MDP. Relevant metrics were determined and included for the cislunar SDA problem (shown in Table 1), but the MDP formulation is flexible to include any number of additional metrics as long as they are quantifiable.

Weights are then applied to the metrics to represent a decision maker's priority.

Parameters for the Monte-Carlo Tree Search are then defined which include the number of iterations and hyperparameters that define the exploration vs exploitation policy. Inputs from steps 1-3 are fed into the MCTS optimization as depicted in Fig. 3. The critical enhancement in this work is the inclusion of a sensor tasking routine, which performs the capacity optimization, as described in the next section. The orange highlights in Fig. 3 indicate where the capacity optimization occurs. The final output of the method is a Pareto-optimal set of solutions, which is presented to the decision maker to investigate further, or extract insight from results as shown in Section 4.3.

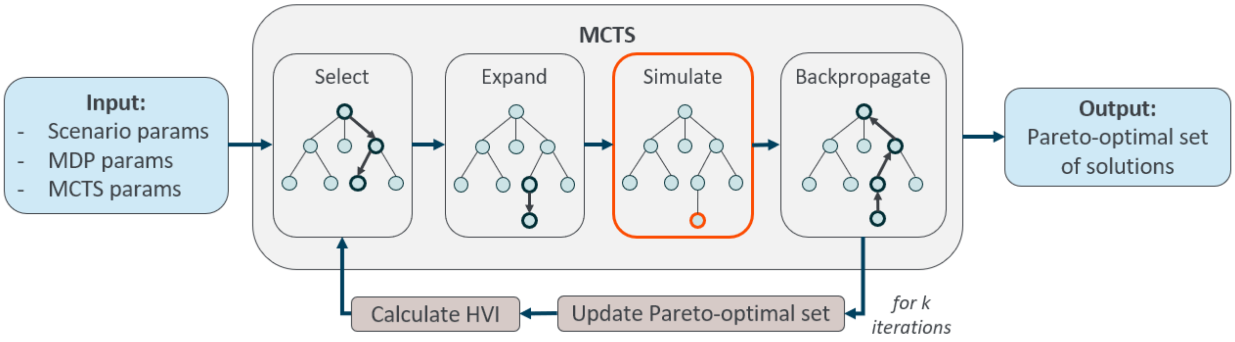

Fig. 3: Multi-objective MCTS flow diagram

3.1 Capacity Optimization

This section generally describes the volume search approach to capacity optimization, where the output is an optimized slew plan for each sensor. Observability must first be determined in order to evaluate capacity. The set of observers is instantiated with orbit and sensor parameters, and observer orbits are propagated for the simulation length. The time-history of observability is then calculated for every observer-target pair in the scenario, subject to the viewing constraints. Capacity is evaluated with a cooperative volume search among the instantiated observers. Using the observability data, each observer performs an ε-greedy search of the space. The search is collaborative, meaning that the observers share information on which targets were observed and when they were observed, so as to not perform redundant observations.

The volume search subroutine is formulated as a grid search MDP: each observer has a local environment defined as a Right Ascension / Declination (RA-dec) sphere, centered on the observer in the Earth-Moon Barycentric Rotating (EMBR) frame shown in Fig. 4. The sphere is discretized according to the sensor's FOV, with coordinates on the grid corresponding to sensor pointing angles in the aforementioned local environment.

Figure 5 depicts a notional subsection of the RA-dec coordinate system in two dimensions. The actor's state is composed of its position in the RA-dec coordinate system, last observed targets, currently observed targets, a global history of target observations, simulation time, and cumulative reward. The position on the grid in state s_i represents the observer's current pointing position. The action space consists of four cardinal directions in the RA-dec coordinate system - the actor can slew up, down, left, or right. From each state s_i and action a_i, there is an associated reward r_i. In this problem the reward is a linear function of the number of observable targets in that angular position, and the time since each target in that position was last observed. Note that this is modifiable, and any priority function can be used with the associated state data.

A simple ε-greedy algorithm is used by each actor to explore the space. At each time step, an actor will sample all possible actions and determine the associated reward. It will greedily pick the action with the highest reward, with a small probability (ε) of choosing a different random action. The ε parameter is essentially an exploration factor versus the greedy exploitation. The greedy algorithm serves not only as the transition function, T, but also as the policy $\pi(s,a)$. While it is not necessarily optimal, it was selected for its speed and simplicity. Note that other MDP solutions can be applied such as policy iteration, value iteration, and even MCTS.

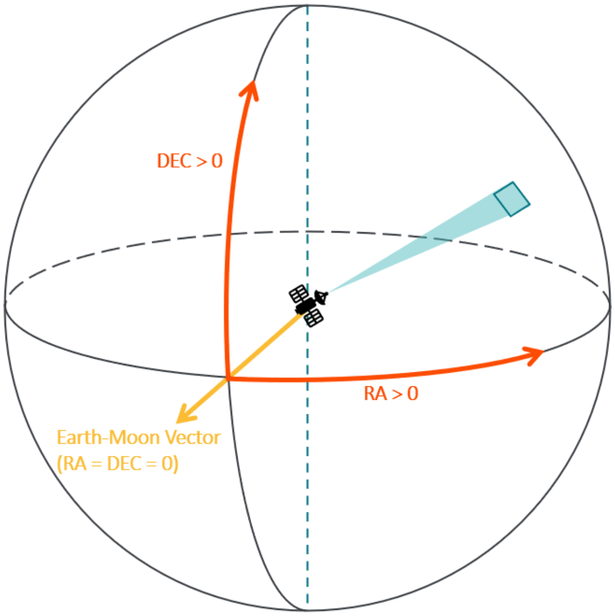

Fig. 4: RA-dec spherical coordinate system

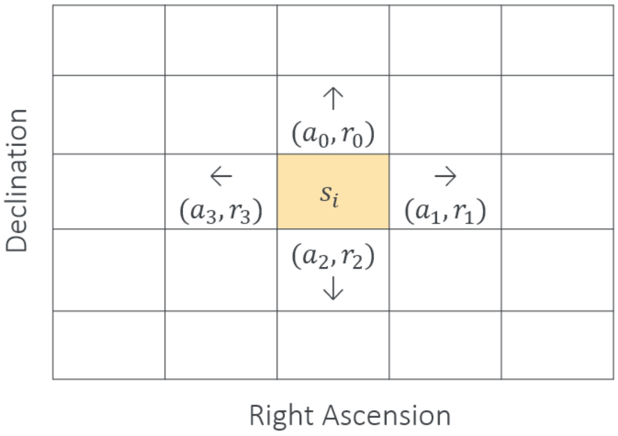

Fig. 5: RA-dec grid discretized by sensor FOV

The result of the capacity optimization is a slew plan and observation history. The ε-greedy search provides a time history of where each sensor is pointing, what targets are observed, and a cumulative reward generated throughout the simulation. These results are used to calculated the evaluation metrics which are detailed in the next section.

3.2 Evaluation Metrics

Architectures are evaluated according to the eight metrics in Table 1, which also comprise the objective space of the architecture optimization problem. The metrics *# Observers* and *Average orbit stability* depend only on the architecture instantiation - the remaining six metrics are outputs of the volume search capacity optimization. All values are normalized to $[0,1]$ in the optimization problem and are formulated such that maximization is ideal. A decision maker may apply weights to each metric to represent their priorities, which are then used to calculated the HVI and membership in the Pareto-optimal set.

Table 1: Optimization metrics

Maximum observability [%]	Cumulative percentage of the volume that is observable
Maximum capacity [%]	Cumulative percentage of the volume that is observed
Time to observe 90% of the volume [s]	Time duration to observe 90% of the volume
Mean Time To Access (MTTA) [s]	Median MTTA of targets in the volume
Average gap time [s]	Median average gap time of targets in the volume
Observations per target [*N*]	Median # of observations per target in the volume
# Observers [*N*]	Count of observers
Average orbit stability [-]	Average of each observer's orbit stability

Maximum observability and *Maximum capacity* are both measures of cumulative coverage of the volume, subject to the different constraints as explained in Section 1. *Time to observe 90% of the volume* captures the objective of observing the full volume in the shortest amount of time.

Mean time to access is a measure of the response time of the system, and is defined as how long it takes, on average, for a given target to be observed. *Average gap time* is the average gap duration in observability for a given target over the simulation time. *Observations per target* reflects the objective of frequent revisits in the volume along with *Average gap time*. In case a decision maker prioritizes frequent revisits, they can then assign higher weights to these metrics. Note that these three metrics are calculated for every target, and the median value over the entire target deck is used in the optimization problem.

# Observers is a proxy for cost, as an ideal architecture will use the fewest number of observers to sufficiently observe the volume. *Orbit stability* is also considered a proxy for cost due to station-keeping and delta-V requirements that are not modeled in this work. Stability values are taken from the JPL Three-body Orbit Catalog [15] where a value of 1 indicates a perfectly stable orbit and larger values indicate greater instability. Orbits with stability index greater than 2 are filtered out of the design space.

4. RESULTS

4.1 Scenario Description

The cislunar SDA scenario for this work is presented as a volume search in cislunar space. In these results we define the volume as a 4x GEO shell with a 1x GEO inner radius - note that any static volume in the EMBR frame can be used. All scenario plots are presented in the EMBR frame with dimensionless units of length (LU) as shown in Fig. 6. The search time for the capacity optimization routine is set to one hour. The additional viewing constraints in the scenario require us to consider temporal affects on observability i.e. the position of the sun will affect viewing geometry and target detectability. Each instance of architecture simulation is performed at 4 different epochs, which are evenly spaced across one Earth-Moon synodic period - this accounts for the varying position of the Sun in the EMBR frame. The final computed objective values during evaluation are average values over the *Epochs per simulation*.

Assuming a visual sensor we impose the following viewing constraints: solar exclusion is defined as an angular keep-out zone from the vector between the observer and the Sun, and is set to 30°. Due to stray light interference, similar constraints are applied for Earth and Lunar exclusion zones where the angular keep-out is defined from the vector between the observer and the limb of the body. Both the Earth-limb and Lunar-limb exclusion angles are set to 5°. A visual sensor with absolute visual magnitude sensitivity of 20 is used for this analysis. A sunlit constraint is applied to the targets in order to be observable, and a range-by-phase constraint is applied to the sensor. The sensor's range is a function of its visual magnitude sensitivity, solar phase angle, and target parameters as in [16].

The architecture optimization MDP reward space has eight dimensions, which is the number of optimization objectives included in the HVI. For the purpose of demonstration, the architecture design space was limited to 4 observers, 8 phases per orbit, 35 orbit families, and 10 members per orbit family. This is an artificial constraint to reduce a technically infinite design space to an amenable number. Weightings are applied to the objective metrics, with higher priority placed on *Maximum capacity*, and *# Observers* as a proxy for cost. The Monte-Carlo Tree Search was run for a total of 1,000 iterations, with every iteration producing a unique architecture.

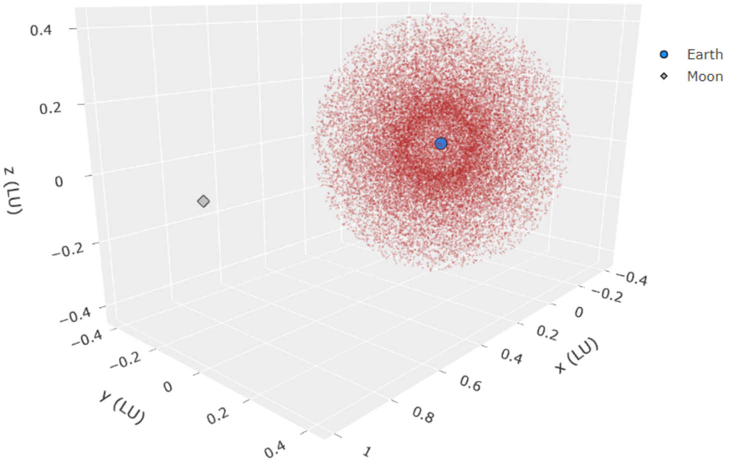

Fig. 6: EMBR frame and cislunar volume

4.2 Individual Architecture Results

This section will describe the outputs from the evaluation of a single optimized architecture. The outputs shown are from an architecture consisting of two observers - one at a 4x GEO circular orbit and one at a Distant Prograde Orbit (DPO). The results and plots in the section were from a single simulation at one epoch. The observability heatmap in Fig. 7 is a cumulative binary plot of observability over the simulation time. A cone of non-observable points (in red) is seen around the Earth due to the Earth-exclusion angle constraint.

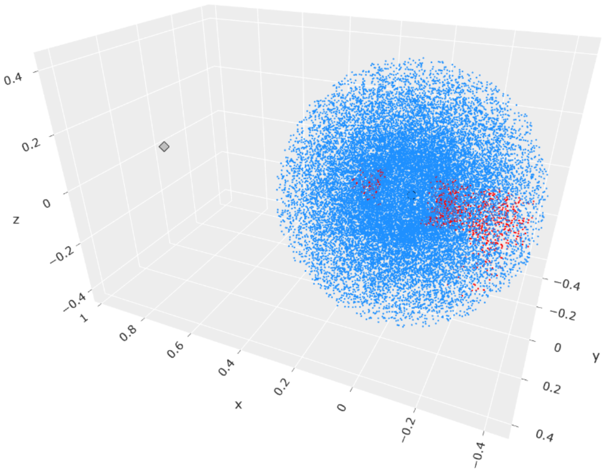

Fig. 7: Observability heatmap: 4xGEO + Distant Prograde Orbit observers

Volume search results from the capacity optimization can be seen in the observation count heatmap in Fig. 8. This figure shows the cumulative observation count per target over the simulation time. The DPO observer can be seen circling the moon, and the 4xGEO orbit observer can be seen on the green trace around the GEO shell. The resultant sensor slew plan can be seen in Fig. 9, which shows a visit count heatmap of each observer's local RA-dec coordinate system. This is a two-dimensional representation of each observer's pointing coordinates, centered on the respective observer. The heatmap is colored to indicate the frequency of visits on each grid space, showing where and how frequently the observer was pointing at a given angular coordinate. Gaps in each observer's search can be seen centered around the Earth which is consistent with the Earth-exclusion viewing constraint. The higher search frequency around the Earth is likely because the observer's priority function searches for regions of higher density in the volume.

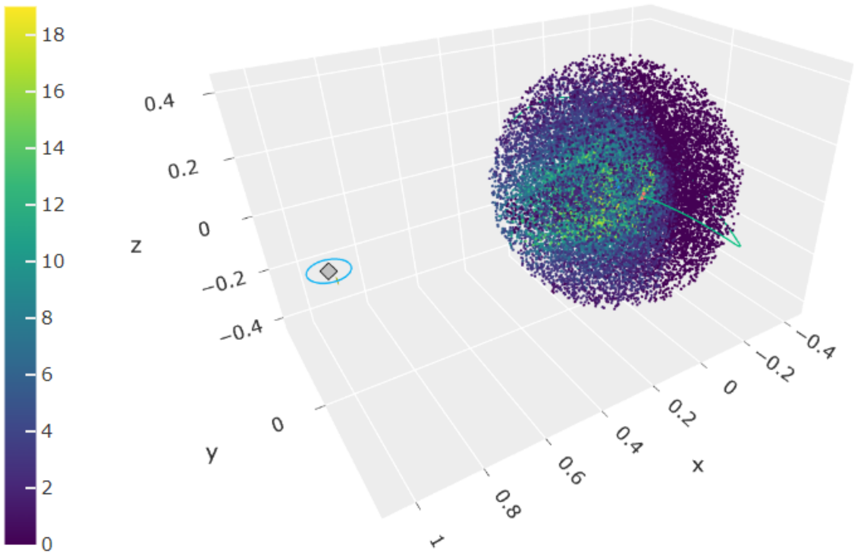

Fig. 8: Observation count heatmap: 4xGEO + Distant Prograde Orbit observers

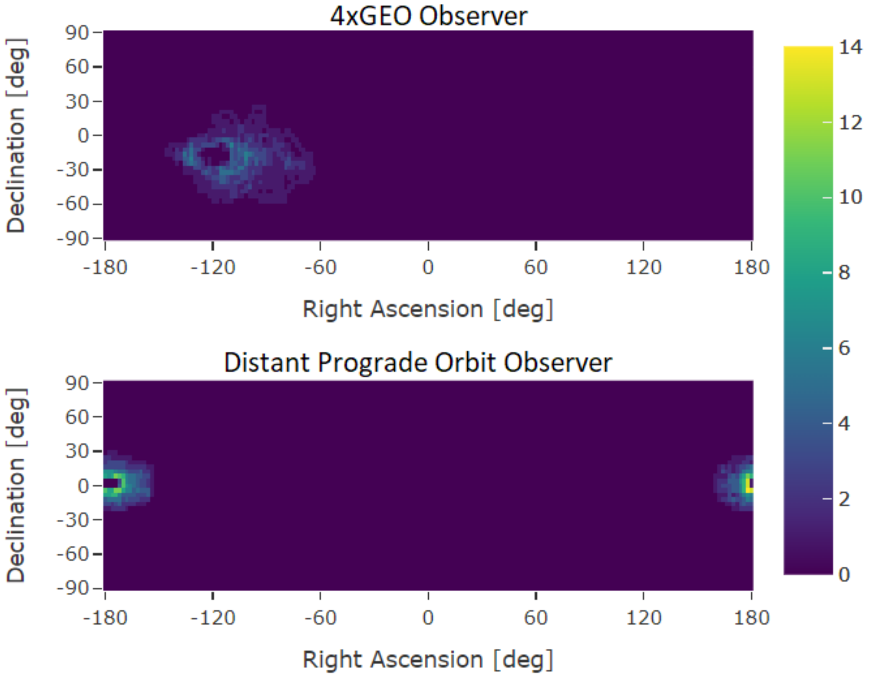

Fig. 9: Local environment visit count heatmap: 4xGEO + Distant Prograde Orbit observers

A time history of the observability and capacity coverage is shown in Fig. 10. The coverage depicted is a result of the collaborative search, meaning capacity contributions from all observers in the architecture. As expected, the observability is relatively constant and much greater than the capacity. Capacity, being a cumulative measure, continues to increase as the observer searches the volume. The results shown thus far are for a single epoch evaluation i.e. the architecture evaluated at only one starting time for a one hour search window.

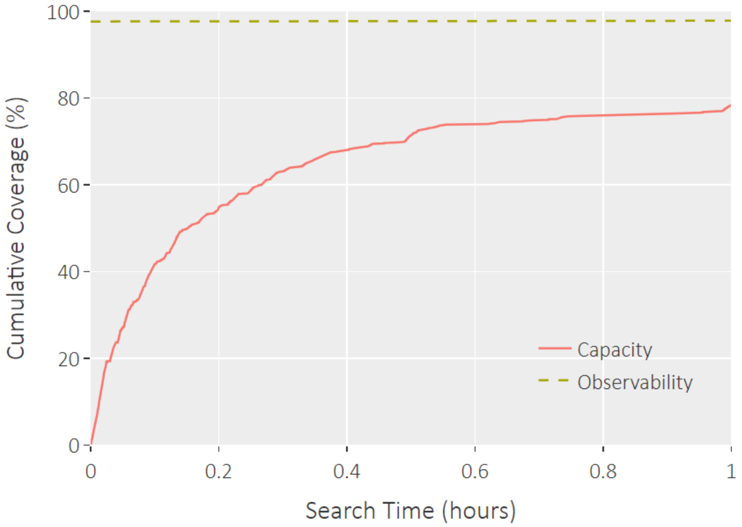

Fig. 10: Coverage vs time: 4xGEO + Distant Prograde Orbit observers

Aggregate results of the objective metrics are shown in Table 2, with true dimensioned values. As noted in Section 3.2, these values are averaged over several epochs to account for the varying position of the Sun. Note again that the *MTTA*, *Average gap time*, and *Observations per target* are median values of the entire target deck. As shown in Fig. 10 the architecture never achieves 90% capacity in the one hour search time, therefore the *Time to observe 90% of the volume* is the maximum simulation time of 3,600 seconds. Depending on its orbit phase around the moon, the DPO observer is able to see much of the volume, but is limited by range. The 4xGEO observer does not have such limitations; however, due to its proximity to the volume and narrow FOV, only small arc segments of the volume can be seen at any given moment which results in a slow search of the space. In this example, the observers are complementary to each other and achieve good coverage of the volume without requiring a large number of observers.

Table 2: Objective results - 4xxGEO + Distant Prograde Orbit

Maximum observability	98.80%
Maximum capacity	78.34%
# Observers	2
Average orbit stability	1.001
Time to observe 90% of the volume	3600 s
MTTA	368 s
Average gap time	713 s
Observations per target	4

4.3 Pareto-optimal Results

The overall result of the architecture optimization process is a Pareto-optimal set of architectures. This set, and the associated HVI, evolve as the tree search iterates as shown in Fig. 11. With enough iterations the MCTS is guaranteed to converge on the global optimum; this would however require a number of iterations approaching the number of total possible architectures. The advantage of MCTS is its ability to quickly explore breadth in the tree while also remembering and exploiting valuable solutions. Among the 28,000 possible architectures, several optimized solutions were found within the 1,000 architecture evaluations. In reality the design space and number of possible architectures is technically infinite, thus the possible architectures and number of iterations were limited to a tractable number for the purposes of demonstration.

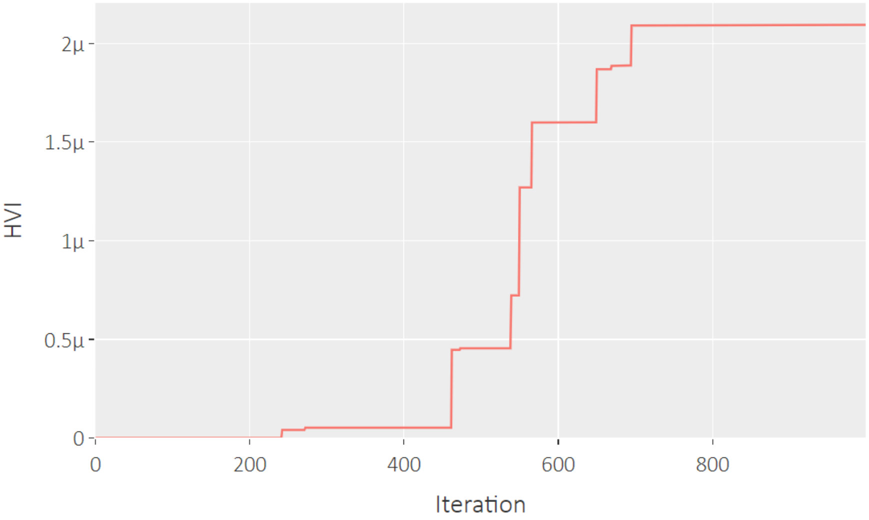

Fig. 11: HVI history

A Pareto plot of the objective space is shown in Fig. 12 for selected metrics. Each dot represents a unique architecture, with grey dots representing dominated solutions and colored dots representing the Pareto-optimal solutions. Note again that the objective values have been normalized between $[0,1]$ with weighting factors applied, and are formulated such that maximization is ideal. A roughly linear trend between observability and capacity can be seen, with a similar trend between capacity and gap time. In general, Pareto plots in the objective space can be useful to identify relationships between objectives that are not obviously correlated.

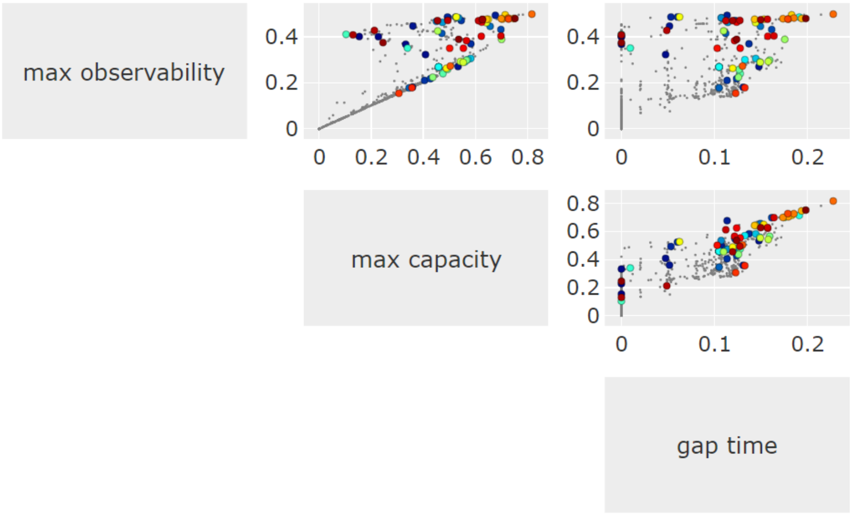

Fig. 12: Pareto plot of selected metrics in the objective space

A coverage versus time plot for selected architectures in the Pareto-optimal set is shown in Fig. 13. Individual results for the *4xGEO - DPO* architecture are shown in the previous section. While it did not achieve as high capacity as the other architectures, it achieved almost as much capacity with one fewer observer. If the *# Observers* metrics is weighted heavily such results may be expected. Fig. 14 shows a histogram of orbit families that appear in the Pareto-optimal set - depending on the scenario, this can vary and inform decision about candidate architectures. Because this example scenario uses a GEO shell volume, we may expect families of orbits that remain relatively close to the Earth to perform well, which is consistent with the high frequency of xGEO and Low Prograde Orbits seen in the Pareto-optimal set.

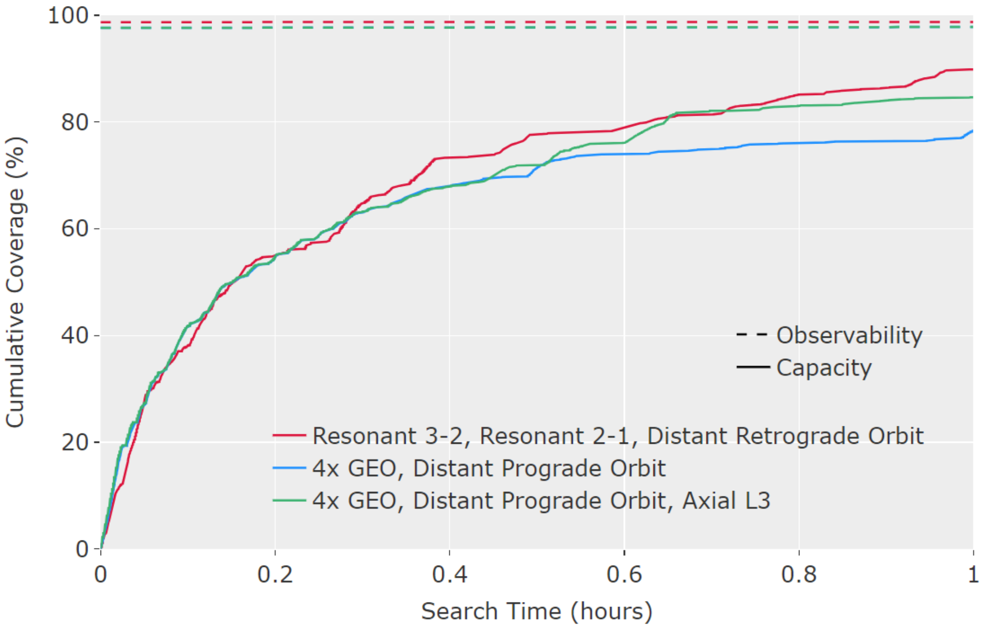

Fig. 13: Coverage vs time for selected Pareto-optimal solutions

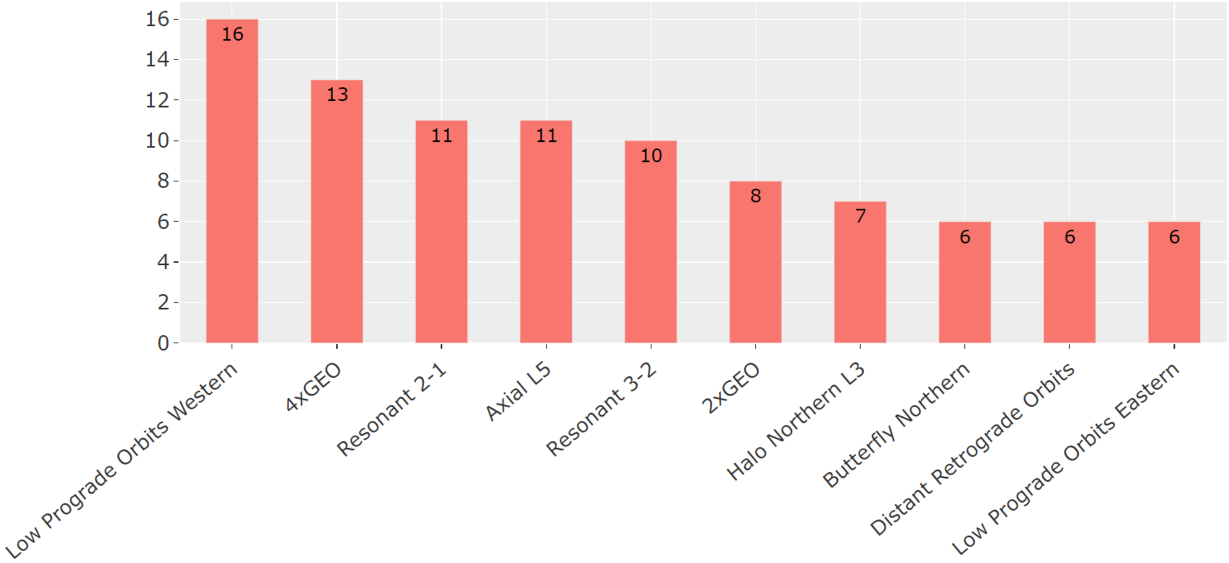

Fig. 14: Histogram of orbit families in the Pareto-optimal set

With information on the Pareto-optimal set, an analyst or decision maker can infer broad insights into the solution space of the problem. High-performing architectures and orbit families are identified for further investigation, and relationships can be identified between objective metrics. Additionally, determining a Pareto-optimal set of solutions allows one to identify and visualize trades in a high-dimensional objective space. The method and framework is designed to be flexible and extensible, and allow analysis on a wide variety of cislunar SDA problems. Different search volumes, optimization algorithms, objective metrics, and radiometric models can be implemented with ease.

5. CONCLUSION

While the need for cislunar SDA is present, the architecture design problem remains complex and challenging due to the complex dynamics and large design space. This work demonstrates a simulation framework that addresses the complexity of the problem with a streamlined and comprehensive approach to evaluating mission performance. The addition of capacity optimization in the problem provides critical measures of performance, allowing architectures to be evaluated with more realistic objectives and constraints. The approach shown greatly reduces the domain knowledge required to design cislunar missions, as well as the human-in-the-loop effort required to sustain such efforts. The demonstrated framework is extensible, and can be used to evaluate a variety of scenarios with different search volumes, optimization algorithms, sensor models, viewing constraints, and optimization metrics.

[1] Badura G, Shimane Y, Gregoire A, Patel R, Gilmartin M, Gangolli K, Visonneau L, Tysor J, Manojkumar J, Humphrey J, Valenta C, Blair R, Lourenco N, Hodkin J, Sudol A, Borowitz M, Gunter B, Christian J, , and Ho K. System design and analysis for cislunar space domain awareness through distributed sensors. Technical Report AAS-22-736, Georgia Tech Research Institue and Georgia Institue of Technology, Atlanta, GA, August 2022.

[2] Frueh C, Howell K, DeMars K, and Bhadauria S. Cislunar space situational awareness. Technical Report AAS-21-290, Purdue University, West Lafayette, IN, 2018.

[3] Duffy L. *Cislunar System of Systems Architecture Evaluation and Optimization*. PhD thesis, Colorado State University, Fort Collins, CO, 2023.

[4] Knister S. Evaluation framework for cislunar space domain awareness (sda) systems. Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, March 2020.

[5] Klonowski M, Holzinger M, and Owens-Fahrner N. Optimal cislunar architecture design using monte carlo tree search methods. In *2022 Advanced Maui Optical and Space Surveillance Technologies Conferences*, 2022.

[6] Badura G, Gilmartin M, Shimane Y, Crum S, Visonneau L, Valenta C, Steffens M, Cimtalay S, Humphrey F, Borowitz M, Gunter B, Christian J, and Ho k. Optimizing distributed space-based networks for cislunar space domain awareness in the context of operational cost metrics. In *2023 Advanced Maui Optical and Space Surveillance Technologies Conferences*, 2023.

[7] Di Marco S. Exploiting multi-body dynamics for distributed spacecraft architecture optimal design: the cislunar services case. Master's thesis, Politecnico Di Milano, Milano, IT, 2021.

[8] Owens-Fahrner N, Correa J, and Wysack J. Capacity-based cislunar space domain awareness architecture optimization. In *2022 Advanced Maui Optical and Space Surveillance Technologies Conferences*, 2022.

[9] Bolden M, Craychee T, and Griggs E. An evaluation of observing constellation orbit stability, low signal-to-noise, and the too-short-arc challenges in the cislunar domain. In *2020 Advanced Maui Optical and Space Surveillance Technologies Conferences*, 2020.

[10] Cunio P, Bever M, and Flewelling B. Payload and constellation design for a solar exclusion-avoiding cislunar ssa fleet. In *2020 Advanced Maui Optical and Space Surveillance Technologies Conferences*, 2020.

[11] Kocsis L, Szepesvari C, and Willemson J. Improved monte-carlo search. Technical Report 1:1-22, MTA SZTAKI and University of Tartu, Budapest, Hungary and Tartu, Estonia, 2006.

[12] Couetoux A, Hoock J, Sokolovska N, Teytaud O, and Bonnard N. Continuous upper confidence trees. In *5th International Conference on Learning and Intelligent Optimization*, 2011.

[13] Klonowski M, Owens-Fahrner N, Heidrich C, and Holzinger M. Robust cislunar architecture design for cooperative agents. In *2023 Advanced Maui Optical and Space Surveillance Technologies Conferences*, 2023.

[14] Guerreiro A, Fonseca C, and Paguete L. The hypervolume indicator: Computational problems and algorithms. *ACM Computing Surveys*, 2021.

[15] NASA JPL. Three-body periodic orbits. https://ssd.jpl.nasa.gov/tools/periodic_orbits.html, 2024. Accessed: 2024-08-07.

[16] Hejduk M D. Specular and diffuse components in spherical satellite photometric modeling. In *2011 Advanced Maui Optical and Space Surveillance Technologies Conferences*, 2011.