Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Generalized Framework for Cislunar Satellite Placement and Operations Optimization

Malav Patel
Georgia Institute of Technology
Anay Naik
Georgia Institute of Technology
Koki Ho
kokiho@gatech.edu

Georgia Institute of Technology

Research Article

Keywords: Optimization, Satellite, Gradient, Linear Programming

Posted Date: April 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9076163/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Generalized Framework for Cislunar Satellite Placement and Operations Optimization*

Malav Patel[1], Anay Naik[1], Koki Ho[1*]

[1]Aerospace Engineering, Georgia Institute of Technology, 270 Ferst Drive, Atlanta, 30332, Georgia, USA.

*Corresponding author(s). E-mail(s): kokiho@gatech.edu;
Contributing authors: mpatel636@gatech.edu; anaik24@gatech.edu;

Abstract

Expanding infrastructure and traffic in the region between the Earth and Moon necessitates commensurate commitments in monitoring and surveillance architectures. This work presents a formulation for optimizing the placement of space assets via a heirarchal optimization scheme. The outer stage leverages first order methods to optimize asset placement in a continuous space, while the inner stage optimizes operation-level decisions; the gradient information generated from the inner loop is used in the outer loop for iterations. We show that this formulation, unlike previous approaches, is modular, allowing researchers to optimize a constellation's phasing while remaining agnostic to the underlying optimzation signal. We show its applicability on two application problems, namely the sensor tasking and communication packet routing problems. This formulation can be easily generalized to any dynamical system with periodic orbits in state space, which yields potential application outside of the space context.

Keywords: Optimization, Satellite, Gradient, Linear Programming

1 Introduction

Recently, there has been a renewed interest in space exploration, particularly towards expanding a more permanent human presence towards the Moon and beyond. NASA's

*This paper is a substantially improved version of Paper AAS 25-626 presented at the AAS/AIAA Astrodynamics Specialist Conference, Boston, MA, in Aug. 2025. The added content includes new methods, analysis, and results.

Artemis missions, the Lunar Gateway, Commerical Lunar Payload Service (CLPS), etc. present directives towards this end. In consequence, the region of space surrounding the Earth-Moon system, termed *cislunar* space, is anticipated to see increasing traffic in the coming decades. Thus, there is a growing need to establish infrastructure in the region to facilitate sufficient awareness of activity. This need is often given the name *space situational awareness* (SSA).

Space Situational Awareness (SSA) in the cislunar regime relies on a combination of sensing modalities and carefully chosen orbital architectures to ensure persistent and reliable coverage. SSA is typically achieved with ground-based sensors, space-based sensors, or a combination of both in order to monitor regions of interest. However, ground-based sensor often suffer from cloud cover, horizon-induced occlusion, and solar interference [1]. Space-based observers, although more difficult to launch and maintain, are mostly free from the aforementioned pitfalls and offer competetive observation capability, especially when coordinated [2, 3]. However, cislunar space is quite large and difficult to observe completely. Fortunately, traffic is likely to be concentrated in pockets near the Earth, Moon, and their associated L1-L5 Lagrange points. This effectively places a non-uniform prior over the space of realizable constellation architectures. Specific orbit families in these regions of interest have been shown to exhibit desirable properties, supporting arguments for their use in future missions [4–8]. These families have been used to optimize satellite constellations under the observability criterion [9–12] and on-orbit servicing networks [13]. Further investigations with higher-fidelity dynamics models [14, 15] show that cislunar space has a promising set of candidate orbits for future missions supporting SSA.

Identifying suitable orbits for asset placement addresses half of the SSA problem. The natural next step is to task these satellites in support of a mission objective. Two simple missions we explore in this work are sensor tasking for target tracking and communication packet routing within the asset network. Several approaches for the sensor tasking problem have already been studied. An example includes information-based [16–18] tasking where a constellation is tasked with minimizing filter uncertainty of a set of targets. Other examples solve online [19, 20] and decentralized [21] tasking problems for target tracking. The communication problem has also been studied. Examples include utilizing evolutionary approaches for the design task [22], deep learning based approaches [23] for secure and robust networks, and multi-hop relay networks for timely communication [24].

A second aspect that is equally important is optimizing the placement of satellites along their orbits. This is done either dicretely into "slots" or continuously via some parameterization of motion along the orbit. Discretization lends itself well to rigorous optimization formulations as placing assets into slots can be represented by integer variables [25–28]. Instead of slots, one can optimize the placement of assets in a *continuous* space, seen as the limit of infinitesimal discretization. It is clear that increasingly finer dicretization will grow the problem size, so integer programming will not scale practically. Fortunately, approaches sidestepping this issue have been investigated. An example outside of the SSA domain is the work by Choi et al. [29] who optimize the placement of orbital refueling depots in Earth orbit by using gradient-based methods on the vehicle routing problem. Patel et al. [30] formulate a heirarchal problem

leveraging finite-difference gradient-based methods in conjunction with integer programming for sensor tasking [16] to optimize the placement of space assets. However, their formulation was restricted in scope to an information-based tasking objective. Furthermore, finite-difference approximations to the gradient can be inaccurate in regions of high-curvature.

In this work, we continue to exploit the trend of merging gradient-based methods with standard programming formulations to design optimal sattellite constellations. However, in contrast with previous works, our contribution is modular enough to allow application outside of the SSA domain. The formulation presented can be regarded as a outer-loop phasing optimization process where the optimization signal is provided by an arbitrary inner-loop that represents the operation-level decisions. In the proposed approach, the inner-loop is typically chosen to be a separate optimization problem of interest to a constellation designer. This approach has a couple of advantages over previous works. First, the inner loop is free from a particular form of the inner-loop objective, providing generality. For example, one can combine the signal from more than one inner-loop processes to inform the phasing optimization process. Second, as a consequence of generality, one can leverage analytic gradients for optimization instead of finite differences (although both are viable).

The rest of this section is organized as follows. Section 2 presents the formulation along with two use cases of inner-loop operation-level decisions in the context of space application. The first is a sensor tasking application while the second is a packet routing application. Section 3 shows experiments with the two cases and investigates improvements to the algorithm. Section 4 concludes with a summary of the work and future research directions.

2 Method

At the highest level, the space asset placement problem for a set $\mathcal{A}$ of assets on periodic orbits requires a signal for optimization. Suppose $\mathcal{A}$ contains a set of state vectors $\{\mathbf{x}_i\}_{i=1}^{|A|}$ describing each asset's initial state. The most ubiquitous signal for optimization is the gradient: $\mathbf{x}_i \leftarrow \mathbf{x}_i + \alpha \nabla_{\mathbf{x}_i}(\cdot)$, where $\alpha \in \mathbb{R}$ is a step size and $\nabla_{\mathbf{x}_i}(\cdot)$ is an oracle providing the optimization signal.

Since each asset evolves periodically in state space, the oracle must capture the asset's performance along the entire orbit in order to be representative. A simple way to achieve this is as follows. Instead of computing the oracle with respect to the initial asset state, we sample the oracle along the asset's orbit and aggregate the signals:

$$\nabla_{\mathbf{x}}(\cdot) \sim \text{Agg}\{\nabla_{\mathbf{s}}\big|_{\mathbf{s}=\mathbf{s}_1}(\cdot),\ \nabla_{\mathbf{s}}\big|_{\mathbf{s}=\mathbf{s}_2}(\cdot),\ \ldots\} \tag{1}$$

where by a slight abuse of notation $\{\mathbf{s}_i\}$ is a set of asset states along the asset orbit. This yields a signal that is "averaged" over the asset orbit. Note that the "Agg" operation is general and can be chosen as any reduction over the set of vectors it operates on. Simple but nonexhaustive choices for "Agg" include average, sum, or random sampling. Pseudocode for asset placement is shown below. Note that the

gradient must be projected along the asset orbit before the update step. It is omitted in Line 9 for brevity.

Algorithm 1 Space Asset Placement Optimization

1: **repeat**
2: $\mathcal{G} \leftarrow \{\}$
3: **for** t = 1 ... T **do**
4: $\mathbf{s}_t \leftarrow$ asset state at time t
5: $\mathbf{g} \leftarrow \nabla_{\mathbf{s}}\big|_{\mathbf{s}=\mathbf{s}_t}(\cdot)$
6: $\mathcal{G} \leftarrow \mathcal{G} \cup \{\mathbf{g}\}$
7: **end for**
8: $\mathbf{g} \leftarrow \text{Agg}(\mathcal{G})$
9: $\mathbf{x} \leftarrow \mathbf{x} + \alpha\mathbf{g}$
10: **until** converged
11: **return** $\mathbf{x}$

Note that the choice of oracle is general. One can solve a subproblem for operation-level decisions inside the oracle and pass the gradient up to the asset location optimization loop. We explore two such subproblems below. Note that when a gradient becomes difficult to write in closed-form, we utilize automatic differentiation via the JAX library [31].

2.1 Sensor Tasking

Given a set $\mathcal{A}$ of assets, a natural next step is to task these assets to a set of targets $\mathcal{B}$. Specifically, we discretize the time evolution of assets and targets and sample their states at these timesteps. At each timestep, we must allocate $|\mathcal{A}|$ space assets to $|\mathcal{B}|$ targets. This problem can be formulated as an *assignment* problem at each timestep t:

$$\text{LP}_t := \underset{x}{\text{minimize}} \quad \sum_{(i,j)\in\mathcal{A}\times\mathcal{B}} w_{ij}x_{ij} \tag{2a}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{B}} x_{ij} = 1 \quad \forall i \in \mathcal{A} \tag{2b}$$

$$\sum_{i\in A} x_{ij} = 1 \quad \forall j \in \mathcal{B} \tag{2c}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \in \mathcal{A},\mathcal{B} \tag{2d}$$

Here, w_{ij} represents the cost of assigning space asset i to target j. Constraint (2b) ensures that each asset is assigned to 1 target. Constraint (2c) ensures that each target is assigned to 1 asset. Equation (2) is sometimes called the *1-to-1* assignment problem since 1 asset is tasked to exactly 1 target. In the case that the number of assets

exceeds the number targets or vice versa, constraint (2b) or (2c) becomes an inequality ($\leq$) constraint, respectively. This is often referred to as the *generalized assignment problem*, which will be used in this work. Note that Constraints (2d) ensure the x_{ij} are binary integer variables. However, as the constraint matrix is totally unimodular, the linear program relaxation of this integer program has integral solutions. Thus, the assignment problem is amenable to standard LP solvers. We employ Gurobi [32] in this work.

The oracle for the sensor tasking problem is simply the gradient of the optimum of the linear program:

$$\nabla_{\mathbf{s}}\big|_{\mathbf{s}=\mathbf{s}_t}(\cdot) := \nabla_{\mathbf{s}}\big|_{\mathbf{s}=\mathbf{s}_t}(\mathrm{LP}_t) \tag{3}$$

$$= \sum_{(i,j)\in\mathcal{A}\times\mathcal{B}} x^*_{ij} \frac{\partial w_{ij}}{\partial \mathbf{s}}\bigg|_{\mathbf{s}=\mathbf{s}_t} \tag{4}$$

Note that in the above expression, $\mathbf{s}$ is a variable representing the state vector of an asset. The gradient is taken with respect to $\mathbf{s}$ and evaluated at a particular state along the asset's orbit $\mathbf{s}_t$. Furthermore, it is reasonable to assume that w_{ij} does not depend on assets $k \neq i$, implying that $\frac{\partial w_{ij}}{\partial \mathbf{s}} = 0$ if $\mathbf{s}$ describes the state vector for asset $k \neq i$.

A more formal argument for this definition of the gradient is as follows. Let $\phi(w, x) = w^\top x$. The value function defined by a linear program can be written as

$$f(w) = \max_{x\in\mathcal{X}} \phi(w, x)$$

where $\mathcal{X}$ is a compact feasible set (we ignore LPs where the feasible set is unbounded). The value function is convex and piecewise linear in w. The piecewise property arises from the fact that an optimal solution $x^*(w)$ is, in general, piecewise constant with respect to w. In other words, perturbations in w either leave x^* unchanged or induce a step change to another extreme point in $\mathcal{X}$. Since ϕ is differentiable with respect to w for all $x \in \mathcal{X}$ and $\frac{\partial \phi}{\partial w}$ is continuous with respect to x for all w, we are permitted [33, 34] to write the subdifferential of f:

$$\partial f(w) = \mathrm{conv}\left\{ \frac{\partial \phi(w, x)}{\partial w} : x \in X^*(w) \right\}$$

where $X^*(w)$ is the set of maximizers and conv is the convex hull operation. The right-hand-side of Equation (4) is always an element of this convex hull. Note that when the convex hull has only one element (when the LP has a unique optimizer), f is differentiable with derivative equal to that element. In this case Equation (4) defines the gradient in the formal sense.

2.2 Shortest Path Packet Routing

Given a set $\mathcal{A}$ of assets, another useful problem involves routing information through the network defined by $\mathcal{A}$. Formally, let u, v be a source-target pair and G be a complete graph with vertices $\mathcal{A} \cup \{u, v\}$ and edges $\mathcal{E}$. The set $\mathcal{E}$ of edges is associated with a

weighted adjacency matrix w with w_{ij} the nonnegative cost of edge $i \to j$. The problem is to find a path $u \to \ldots \to v$ through the network at time t minimizing the total incurred cost. The shortest path problem can also be formulated with mathematical programming:

$$\begin{aligned}
\mathrm{LP}_t := \underset{x}{\text{minimize}} \quad & \sum_{i,j} w_{ij}\, x_{ij} && \text{(5a)}\\
\text{subject to} \quad & \sum_j x_{uj} - \sum_j x_{ju} = 1 && \text{(5b)}\\
& \sum_j x_{vj} - \sum_j x_{jv} = -1 && \text{(5c)}\\
& \sum_j x_{ij} - \sum_j x_{ji} = 0 \quad \forall i \notin \{u, v\} && \text{(5d)}\\
& x_{ij} \in \{0, 1\} \quad \forall i, j && \text{(5e)}
\end{aligned}$$

Constraints (5b) and (5c) enforce one unit of flow leaving the source and entering the target, respectively. Constraint (5d) enforces any unit of flow entering an intermediate node to leave that node while Constraint (5e) forces the decision variables x_{ij} to be binary. Like the assignment problem, the constraint matrix in the shortest path problem is totally unimodular, yielding a mathematically equivalent LP relaxation.

The oracle for the shortest path problem is again the gradient of the optimum of the linear program:

$$\begin{aligned}
\nabla_{\mathbf{s}}\big|_{\mathbf{s}=\mathbf{s}_t}(\cdot) :&= \nabla_{\mathbf{s}}\big|_{\mathbf{s}=\mathbf{s}_t}(\mathrm{LP}_t) && \text{(6)}\\
&= \sum_{i,j} x^*_{ij} \frac{\partial w_{ij}}{\partial \mathbf{s}}\bigg|_{\mathbf{s}=\mathbf{s}_t} && \text{(7)}
\end{aligned}$$

Note that unlike the assignment problem, the cost coefficient w_{ij} is likely to depend on two asset state vectors: asset i and asset j. So in general, this implies $\frac{\partial w_{ij}}{\partial \mathbf{s}} = 0$ if $\mathbf{s}$ describes the state vector for asset $k \notin \{i, j\}$. Lastly, following the same argument in section 2.1, we may establish a formal understanding of the gradient in Equation (7).

2.3 Algorithm Considerations

In our experiments, we notice critical points on the objective landscape with asymptotically vanishing gradients approaching from one direction but finite gradients when approaching from another. Secondly, some objectives exhibit local minima at equidistant constellation phasing although this phasing is not globally optimal. To avoid settling in local minima caused by these points, we add two modifications to the gradient update, noise and "heavy ball" momentum:

White Noise	**Momentum**
$g \leftarrow g(1+\epsilon), \quad \epsilon \sim \mathcal{N}(0,\sigma)$	$m \leftarrow \gamma m + (1-\gamma)g$
$x \leftarrow x - \alpha g$	$x \leftarrow x - \alpha m$

Here σ is a scale parameter controlling the signal-to-noise ratio, $\gamma \in (0,1)$ is the momentum parameter, and α is the learning rate. Adding white noise to the gradient is thought to introduce local random walks to an otherwise deterministic trajectory, allowing more dense exploration of parameter space. The white noise form we adopt departs from the traditional Langevin update and was primarily chosen to randomly *scale* the gradient, as it was observed that the algorithm would converge prematurely to a local minimum in the presence of equal magnitude but oppositely signed gradients. Adding momentum to the gradient achieves a similar, but more calculated effect on the descent trajectory. Note that further modifications to the descent algorithm were made during experiments and will be discussed in a later section as appropriate.

3 Experiments

In this section we outline experiments into the two aforementioned problems. The first is a sensor tasking problem, where a set of observers is tasked with providing sufficient observational capability to maintain custody of target spacecraft. The second is a shortest path optimization problem for routing packets in a simple 4 node network consisting of two spacecraft and a source-sink pair.

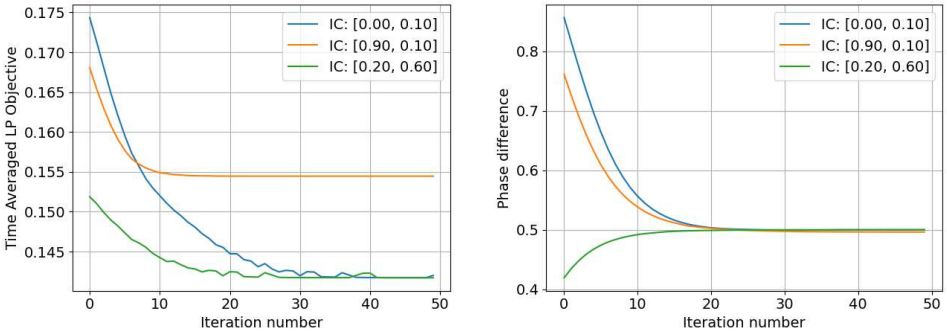
3.1 Sensor Tasking

The sensor tasking problem can be defined with a plethora of objectives. We investigate simple euclidean distance and information theoretic metrics for optimization.

3.1.1 Distance-Based Tasking

As an example scenario for the sensor tasking problem we consider a setup with 2 target satellites placed diametrically opposite each other on a libration point orbit (LPO) and a set of 2 observer satellites on a separate LPO whose initial conditions we aim to optimize for peak tasking performance. As done previously, we parameterize the decision vector as a vector of phase values $\boldsymbol{\phi} \in [0,1)^2$.

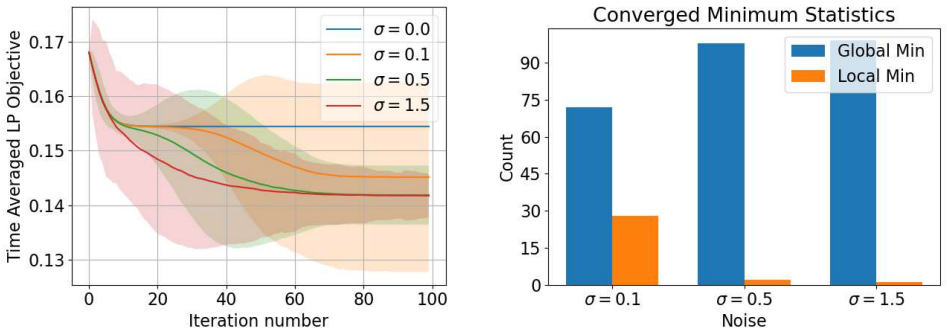
As a simple example consider using the squared Euclidean distance between observer-target pairs to identify "good" tasking potential. Figure 1 shows how standard gradient descent behaves for 3 initial conditions. As seen in Figure 1 (and more generally in this problem) we observed the set of initial conditions $\{[0.5+\epsilon, 0.5-\epsilon] \bmod 1 \cup [0+\epsilon, 0-\epsilon] \bmod 1\}$ converge to local minima. Members of this set are positioned equidistant from each of the global minima in this problem. When starting descent with initial conditions from this set, we observed gradients of equal magnitude but opposite sign. In a modulo 1 parameter space, this repels each observer's phase to the maximum phase difference (i.e. until a phase difference of 0.5). However, scaling the gradient with white noise removes this issue. Figure 2 shows the distribution of descent trajectories under several noise scales with initial condition $\boldsymbol{\phi}_0 = [0.9, 0.1]$.



(a) (b)

Fig. 1: Optimization history from the distance-based tasking case for (a) objective and (b) phase difference, under three initial conditions. Convergence is sensitive to choice of initial condition.

Generally, convergence speed scales with σ, but this behavior is expected to terminate once the noise scale is large enough to corrupt the gradient signal. Furthermore, small values of σ may not be sufficient to escape local minima. Large band widths around the inflection point occurring between iterations 20 and 40 reflect the fact that descent trajectories typically diverged at that point, either settling into the local minimum or decending further to the global minimum. These results reflect the importance of



(a) (b)

Fig. 2: Descent trajectories for the distance-based tasking case with modified descent. (a) Monte Carlo simulation of descent trajectories under several noise scales. Solid lines represent mean trajectories, and shaded regions represent 3-sigma variance. (b) Converged minimum bin counts under each noise scale.

incorporating sufficiently strong noise to the gradient to prevent convergence to this

class of local minima. We hypothesize more instances of these minima as the problem size scales. Lastly, these results validate the use of a standard descent algorithm. Convergence is relatively smooth and fast, within 100 iterations, with only first-order information. We also investigate the framework's behavior as the problem size scales.

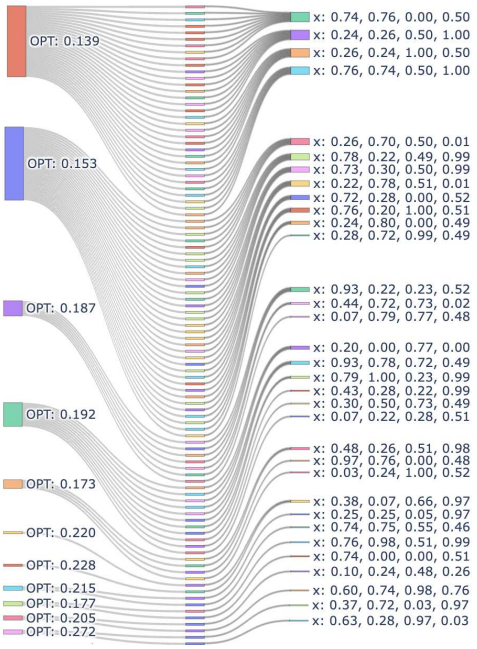


Fig. 3: Flow of 100 initial conditions for the distance-based tasking case. Each initial condition (middle) flows to an optimal point in parameter space (right) and achieves an objective value (left).

Figure 4 shows a practical constellation design problem where 4 observer satellites are placed on periodic orbits and are tasked with monitoring 2 targets on L1 Lyapunov and L2 Halo LPOs. Figure 3 illustrates the flow of 100 initial conditions towards their respective local minima.

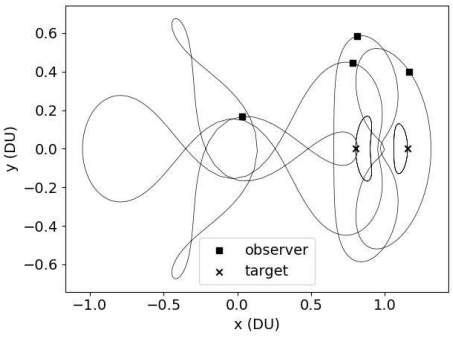


Fig. 4: Constellation of 4 observers and 2 targets. Observers are initialized randomly.

Approximately 33% of initial conditions flow to the best objective. There are four minimizers that achieve this objective value. These minimizers are all (roughly) permutations of the set $\{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. Of the four minimizers, two are truly geometrically unique while the remainder are reflections of the the former about the x-axis. Specifically, observers on the 2:1 and 3:1 resonant orbits are mirrored across the x-axis (see Table 5). This situation arises from geometric symmetries in observer orbits and an objective dependent only on observer-target geometry. This structured solution space continues through the second best achieved objective before devolving into unstructured local minima. We hypothesize that a similar behavior in problems with a geometric objective, such as the euclidean distance, and with symmetries in the orbits. The existence of many local minima presents a challenge to gradient based methods. However, since gradient descent is inexpensive to execute, a simple hedge against convergence to local minima involves sampling the parameter space several times and choosing the minimizer amongst all runs.

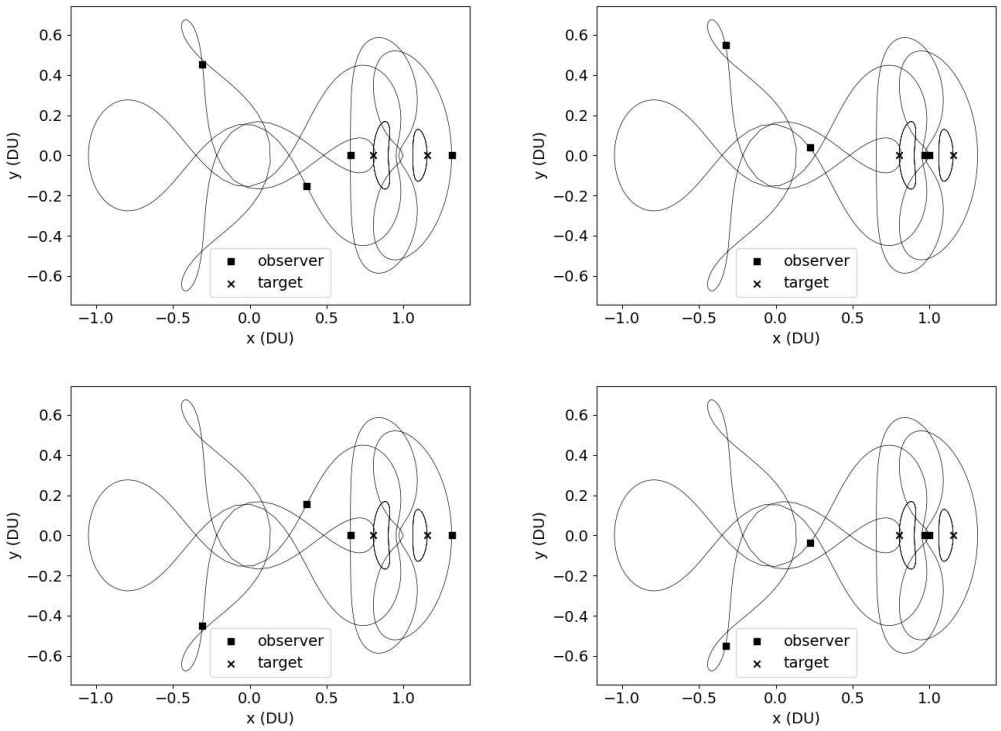


Fig. 5: Multiple optimal architectures for the distance-based tasking case. All configuration achieve the same objective. Top left / bottom left are mirrors accross the x-axis. Top right / bottom right are also mirrors across the x-axis.

3.1.2 Information-Theoretic Tasking

A second method to quantify tasking perfomance is via an information-theoretic measure, specifically the information gain due to measurements in a filtering algorithm. We quantify tasking performance with a suitable transformation of the information update added to the information matrix:

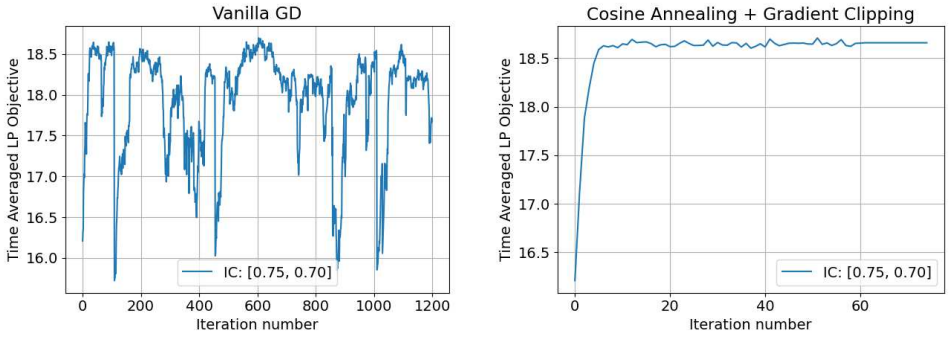
$$\mathbf{\Lambda} \leftarrow \mathbf{\Lambda} + \underbrace{\mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}}_{\text{to optimize}} \tag{8}$$

Here $\mathbf{H}$ and $\mathbf{R}$ are the measurement jacobian and covariance matrices, respectively. The trace and determinant are two metrics that capture the information spectrum in a single scalar. Control schedules optimized against these metrics are termed *trace-optimal* and *d-optimal* control, respectively. Since trace-optimal control can ignore low-information directions, we opt to use the d-optimality metric for optimization.

As an example, consider the optimization of a two-observer constellation tasked with observing two targets on an LPO orbit. At each timestep, we solve an instance of (2) with the cost coefficients derived from the information gain of the measurement update:

$$w_{ij} = \log\det \mathbf{H}_{ij}^{\top}\mathbf{R}_{j}^{-1}\mathbf{H}_{ij} \tag{9}$$

Details on the measurement model can be found in Appendix B. During our experiments, we observed that ocassionally large gradients disrupted the descent trajectory. To mitigate this, we employed a cosine annealed learning-rate schedule [35] and soft gradient clipping [36]. These modifications preserve the structure of the gradient signal while attenuating large peaks. Figure 6 shows the descent trajectory without and with these additions.

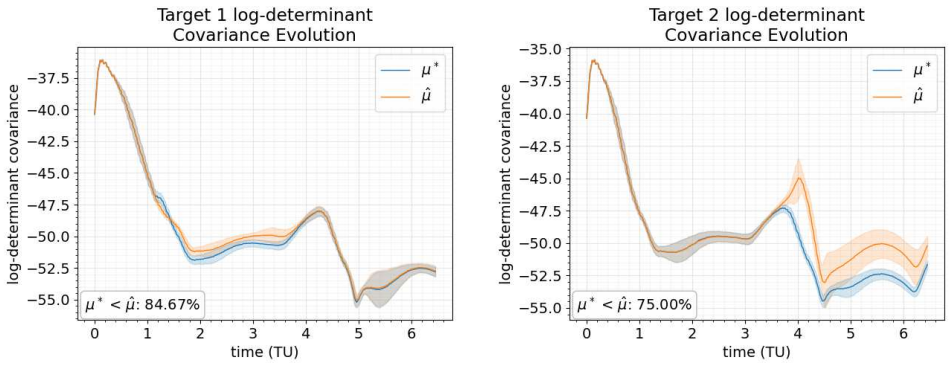


(a) (b)

Fig. 6: Descent trajectories for the information-theoretic tasking case under the log determinant information objective. (a) Vanilla gradient descent struggles to converge due to large gradient norms produced during optimization. (b) The same trajectory with a cosine annealed learning rate and soft gradient clipping. Clipping stabilizes convergence to an ϵ-ball near the optimum. Cosine annealing the learning rate helps the optimizer settle within the ball.

To determine whether an optimized control policy provides tangible benefit relative to a random policy, a Monte Carlo analysis was conducted. In each Monte Carlo trial we chose a random control schedule $\hat{\mathrm{u}}$ as well as a randomly generated process and measurement noise sequence. The optimal control schedule u^* sees the *same* process and measurement noise sequence within one Monte Carlo trial. Therefore, Monte Carlo simulation evaluates the performance of the optimal control schedule against fluctuations in process noise, measurement noise, *and* the control itself. Figure 7 shows the average time evolution for two different targets. A mean log-determinant covariance (μ^* and $\hat{\mu}$ under optimal and randomized control, respectively) is recorded across the simulation time along with 3σ bounds capturing the variance across Monte Carlo trials. For Target 1, μ^* lies below $\hat{\mu}$ for 84% of the simulation time. Similarly, for Target 2, μ^* lies below $\hat{\mu}$ for 75% of the time steps. For both targets the optimized tasking schedule yields more certain state estimates for a majority of the simulation time, with the largest separation happening in the latter half of Target 2's transit. Target 2 consistently shows separation between μ^* and $\hat{\mu}$ when compared to Target 1,

which has time intervals where $\hat{\mu}$ falls below μ^*. Additional statistics from the results of the Monte Carlo analysis are available in Appendix C.



(a) (b)

Fig. 7: Time evolution of state uncertainty for the information-theoretic tasking case for two targets (a) and (b). An ensemble of random policies (and associated process/measurement noise sequences) perform worse on average compared to an optimized tasking schedule

We note that the objective present in Equation (8) is myopic [16, 17], and that considering a forecasted information objective can lead to better long-term returns. Since the state transition matrix Φ used to forecast the information update is not dependent on the observer state, it is analytically differentiable and can be analyzed under this formulation. We leave this to future work.

3.2 Shortest Path Packet Routing

Motivated by previous works in cislunar communications networks, we present a simple study for shortest path packet routing. As an example scenario for the routing problem we consider a setup where the source-target pair is the Earth and Moon with a constellation of two satellites on the same L1 LPO. As satellites are power-constrained, power efficient signal transmission is a key optimization metric for packet routing in a network. Communication in this network occurs on the electromagnetic spectrum and the standard inverse square law for signal attenuation applies. To reflect this, edge weights are the squared separation distance between nodes in the network. Under this metric the gradient is analytic. Concretely, at some time t, let $\mathbf{r}_i$ be the position vector for node i in the network. Then the gradient is proportional to:

$$\nabla \sim \sum_{j \neq i} \underbrace{x^*_{ji}(\mathbf{r}_i - \mathbf{r}_j)}_{\text{incoming edge}} - \underbrace{x^*_{ij}(\mathbf{r}_j - \mathbf{r}_i)}_{\text{outgoing edge}} \tag{10}$$

We fix the initial condition of one asset and optimize for the phase $\phi \in [0, 1]$ of the second asset. Furthermore, to capture an example of non-uniform temporal demand

for packet routing during constellation operation, we employ a non-uniform weighting scheme during aggregation. Figure 8 compares the objective under a uniform (left) and non-uniform Gaussian (right) average weighting aggregation scheme.

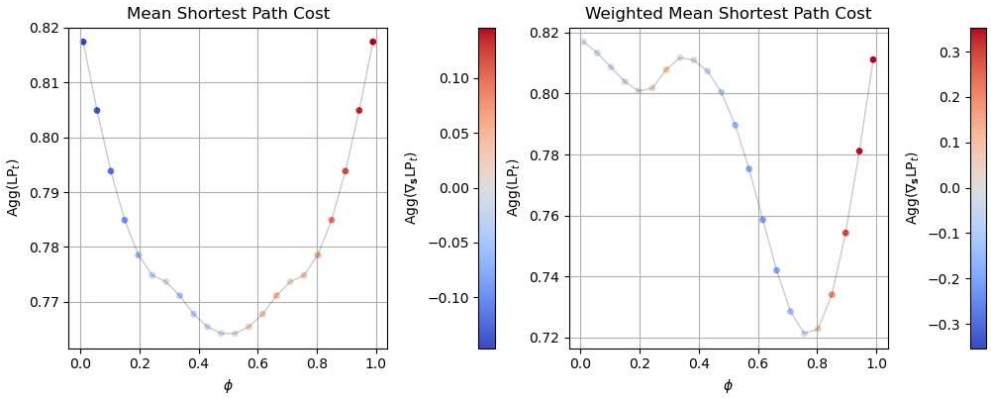


Fig. 8: Mean shortest path cost and computed gradient for uniform (left) and non-uniform (right) weighting schemes.

As expected, the objective is symmetric about the global optimum when "Agg" is a uniform average, reflecting the observation that when constellation motion is averaged across time, equidistant phasing between assets is optimal. This behavior shifts when the network's routing capability becomes more critical at certain times than at others. Furthermore, the gradients $\mathrm{Agg}(\nabla_{\mathbf{s}}(\mathrm{LP}_t))$ are analytic and track the objective's first order information faithfully. We observe that under the uniform weighting, there are two local minima of the objective, placed symmetrically around $\phi = 0.5$. See Figure A1 in the appendix for a visualization of the objective around these points. These minima are close to discontinuities of the piecewise differentiable objective. Table 1 shows the behavior of a standard gradient descent algorithm on either side of the discontinuities.

ϕ_{disc}	ϕ_0	Update Type	ϕ^*	Gradient Norm
≈ 0.495	0.1	Vanilla	0.4949	-2.6e-15
		$\boldsymbol{\gamma = 0.2}$	**0.5000**	**1.1e-14**
		$\sigma = 1$	0.4949	2.7e-12
	0.4955	Vanilla	0.5000	-1.5e-15
		$\gamma = 0.2$	0.5000	2.8e-15
		$\sigma = 1$	0.5000	-8.9e-14
≈ 0.505	0.9	Vanilla	0.5051	4.1e-16
		$\boldsymbol{\gamma = 0.2}$	**0.5000**	**-1.1e-14**
		$\sigma = 1$	0.5000	-1.1e-10
	0.5045	Vanilla	0.5000	-3.6e-14
		$\gamma = 0.2$	0.5000	-3.3e-14
		$\sigma = 0.1$	0.5000	-8.4e-14

Table 1: Optimization Results Under a Standard Gradient Descent Algorithm

Note that adding momentum to the gradient update allows the trajectory to continue its descent to the global optimum. Although a white noise update also does the same, we note that across several runs, the algorithm converged to one of the three local minima. For this problem, the momentum update is a better solution for escaping the one-sided saddle.

4 Conclusion

In this work we present a formulation for the constellation design problem that jointly optimizes a constellation's placement and operations subject to an arbitrary optimization model. This is done via a hirarchical scheme. The inner-loop is responsible for optimizing the operation-level objectives for an objective and producing a gradient information for the optimization of the outer loop. Together, this formulation can handle concurrent optimization problem with nontrivial coupling between the two loops. Unlike previous approaches, the formulation is agnostic to the choice of objective and can theoretically be applied outside of the space context, highlighting its generalizability. We show its applicability in two different contexts, one for sensor tasking and the other for packet routing in space networks. Furthermore, the computational requirements for optimization fall strictly below enterprise-grade compute, with all experiments in this work run on desktop hardware.

This work has several promising future research directions. First, the algorithm exhibits sensitivity to initial condition, which arises from any gradient-based method. This problem becomes increasingly apparent as constellation size scales. These problems are not completely mitigated by the strategies employed in this work. Further experiments and ablation studies into these modifications can provide useful insight for constructing a robust algorithm. Furthermore, Algorithm 1 relied on the gradient from a linear program as the signal for outer-loop optimization. However, Algorithm 1 is general enough to allow for any pipeline that can produce a gradient, extending beyond linear programs. Examples can include pipelines implementing an objective via basis functions, neural networks, and quadratic programs.

Appendix A Shortest Path Problem

The shortest path problem investigated exhibited discontinuties placed symmetrically around the uniform weighting optimum. Noise and heavy-ball momentum are crucial for preventing premature convergence to this point. Figure A1 shows the behavior of the objective around the discontinuities.

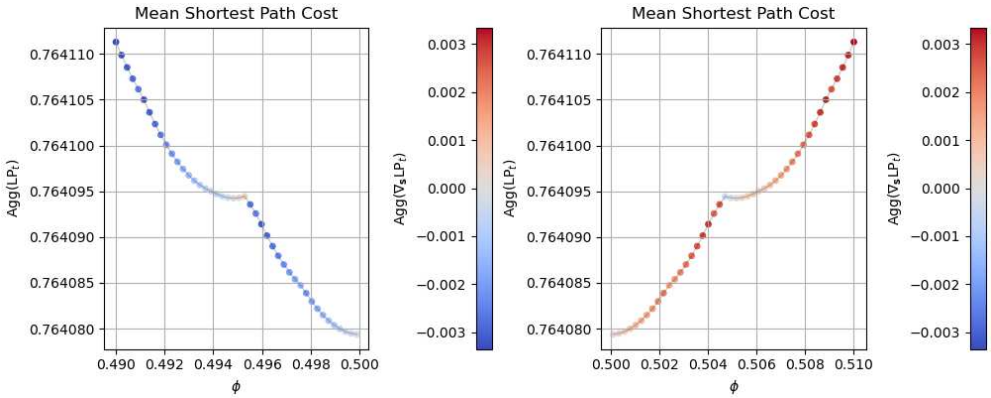


Fig. A1: The objective landscape of an example shortest path problem exhibits discontinuities, posing a challenge to a gradient-based optimizer if the initial condition is chosen unfortunately.

Appendix B Measurement Models

In this section we outline the measurement models used for optimization of a filter-based objective. Let $\mathbf{x} = \begin{bmatrix}\mathbf{r}_s & \mathbf{v}_s\end{bmatrix}^\top$ be the spacecraft state. Let $\mathbf{r}_o$ and $\mathbf{v}_o$ be the observer position and velocity. Let $\mathbf{r} = \mathbf{r}_s - \mathbf{r}_o$ and $\mathbf{v} = \mathbf{v}_s - \mathbf{r}_o$ be the relative position and velocity of the spacecraft. Modern cameras are often equipped with position sensing diodes, LIDAR, or other technology allowing the precise extraction of a line of sight vector from an image [37]. We assume our spacecraft has this capability and thus our ideal noise-free measurement $\mathbf{y}$ is the line-of-sight (LOS) vector and its rate:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix}\boldsymbol{\ell} \\ \dot{\boldsymbol{\ell}}\end{bmatrix}$$

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \dfrac{\mathbf{r}}{\|\mathbf{r}\|} \\ \dfrac{\mathbf{v}}{\|\mathbf{r}\|} - \dfrac{\mathbf{r}^\top\mathbf{v}}{\|\mathbf{r}\|^3}\,\mathbf{r} \end{bmatrix}$$

Noise in the measurement is modeled as a random small angle rotation matrix applied to $\boldsymbol{\ell}$ and $\dot{\boldsymbol{\ell}}$. The measurement jacobian and covariance are

$$\mathbf{H} = \begin{bmatrix} \dfrac{1}{\rho}\mathbf{I}_3 - \dfrac{\mathbf{r}\mathbf{r}^\top}{\rho^3} & 0 \\ -\dfrac{\mathbf{v}\mathbf{r}^\top + \mathbf{r}\mathbf{v}^\top + (\mathbf{r}^\top\mathbf{v})\mathbf{I}_3}{\rho^3} + \dfrac{3(\mathbf{r}^\top\mathbf{v})}{\rho^5}\mathbf{r}\mathbf{r}^\top & \dfrac{1}{\rho}\mathbf{I}_3 - \dfrac{\mathbf{r}\mathbf{r}^\top}{\rho^3} \end{bmatrix}, \qquad \rho = \|\mathbf{r}\|.$$

$$\mathbf{R} = \sigma_\phi^2 \begin{bmatrix} \mathbf{I}_3 & 0 \\ 0 & \frac{2}{\Delta t^2}\mathbf{I}_3 \end{bmatrix}$$

where σ_ϕ is the standard deviation of angle noise and Δt is the exposure time of the measurement. Note that if the LOS vector measurement is utilized in the finite difference measurement for $\dot{\boldsymbol{\ell}}$, independence is lost and a cross correlation term appears. The covariance becomes

$$\mathbf{R} = \sigma_\phi^2 \begin{bmatrix} \mathbf{I}_3 & \frac{1}{\sigma_\phi \Delta t}\mathbf{I}_3 \\ \frac{1}{\sigma_\phi \Delta t}\mathbf{I}_3 & \frac{2}{\Delta t^2}\mathbf{I}_3 \end{bmatrix}$$

Note that in our experiments we did not see a major difference in optimal constellation phasing between the independent and correlated measurement models.

Appendix C Monte Carlo Analysis

This section provides supplementary information on the Monte Carlo analysis comparing the optimized control tasking schedule to the random control tasking schedule. Table C1 shows the parameters of the Monte Carlo simulation. Target state estimates and uncertainty were provided by an Unscented Kalman Filter.

Parameters	Values
Monte Carlo trials N	100
Simulation Time (TU)	6.45
Control Timestep (TU)	0.015
Logging Timestep (TU)	0.0215

Table C1: Monte Carlo Parameters

The Monte Carlo simulation ran 100 trials. Time is measured in normalized CR3BP units (TU) and metrics were logged with a simulation time of 6.45 TU. Within each trial, one filtering run is completed for the optimal tasking schedule and another separate run for the random tasking schedule. To ensure that within each trial the constellation sees the same noise sequence, we reset the random number generator state between runs. Note that the random control policy was also varied between trials to evaluate the robustness of the optimal policy against a distribution of possible policies.

Figure 7 show that the optimized tasking schedule produces better state estimates over a majority of the simulation time. To quantify the performance gap, we can use the $\pm 3\sigma$ bounds of the random schedule. Let $\square^*$ denote a quantity related to the optimal tasking schedule and $\hat{\square}$ denote a quantity related to the randomized tasking schedule. The results show that μ^* lies below $\hat{\mu} - 3\hat{\sigma}$ for 33% of the time steps for Target 1 and 39% for Target 2. However, there are time intervals where the μ^* lies above $\hat{\mu}$. Table C2 shows that this happens for 15% of the simulation time for Target 1 and 25% of the simulation time for Target 2. This behavior may stem from a myopic

or greedy policy enforced by the optimization process. Monte Carlo results show that the optimized tasking schedule provides a more certain state estimate for a majority of the time steps taken in the experiment for both targets.

Condition	% of Sim Time Condition is True
$\mu^* < \hat{\mu}$	84.67% (254/300)
$\mu^* < \hat{\mu} - \hat{\sigma}$	38.67% (116/300)
$\mu^* < \hat{\mu} - 2\hat{\sigma}$	37.33% (112/300)
$\mu^* < \hat{\mu} - 3\hat{\sigma}$	33.33% (100/300)
$\mu^* > \hat{\mu} + \hat{\sigma}$	4.00% (12/300)
$\mu^* > \hat{\mu} + 2\hat{\sigma}$	3.67% (11/300)
$\mu^* > \hat{\mu} + 3\hat{\sigma}$	3.33% (10/300)
$\hat{\mu} < \mu^* - \sigma^*$	4.00% (12/300)
$\hat{\mu} < \mu^* - 2\sigma^*$	3.67% (11/300)
$\hat{\mu} < \mu^* - 3\sigma^*$	2.67% (8/300)
$\hat{\mu} > \mu^* + \sigma^*$	39.00% (117/300)
$\hat{\mu} > \mu^* + 2\sigma^*$	37.00% (111/300)
$\hat{\mu} > \mu^* + 3\sigma^*$	35.33% (106/300)

Target 1 MC Average Statistics

Condition	% of Sim Time Condition is True
$\mu^* < \hat{\mu}$	75.00% (225/300)
$\mu^* < \hat{\mu} - \hat{\sigma}$	45.33% (136/300)
$\mu^* < \hat{\mu} - 2\hat{\sigma}$	43.33% (130/300)
$\mu^* < \hat{\mu} - 3\hat{\sigma}$	39.33% (118/300)
$\mu^* > \hat{\mu} + \hat{\sigma}$	0.00% (0/300)
$\mu^* > \hat{\mu} + 2\hat{\sigma}$	0.00% (0/300)
$\mu^* > \hat{\mu} + 3\hat{\sigma}$	0.00% (0/300)
$\hat{\mu} < \mu^* - \sigma^*$	0.00% (0/300)
$\hat{\mu} < \mu^* - 2\sigma^*$	0.00% (0/300)
$\hat{\mu} < \mu^* - 3\sigma^*$	0.00% (0/300)
$\hat{\mu} > \mu^* + \sigma^*$	45.33% (136/300)
$\hat{\mu} > \mu^* + 2\sigma^*$	44.00% (132/300)
$\hat{\mu} > \mu^* + 3\sigma^*$	43.00% (129/300)

Target 2 MC Average Statistics

Table C2: Monte Carlo Average comparison for each of two targets. The statistics capture how the ensemble average performance of the optimal policy fares against the ensemble average performance of a random policy and vice versa.

Acknowledgment

The authors gratefully acknowledge Logan Feld for their insightful comments and discussion. Artificial intelligence technologies were used in this research for grammar checking.

Author Contribution Statement

The study conception and design were performed by M. P. and K. H. Coding and analysis were performed by M. P. and A.N., both under the supervision of K. H. The first manuscript was written by M. P., and all authors critically revised the previous versions of the manuscript for important intellectual content. All authors read and approved the final manuscript.

Data Availability

Not Applicable.

Declarations

Funding and Conflicts of Interests

The authors gratefully acknowledge support for this research from the Air Force Office of Scientific Research (AFOSR), as part of the Space University Research Initiative (SURI), grant FA9550-22-1-0092 (grant principal investigator: J. Crassidis from University at Buffalo, The State University of New York). There is no further conflict of interest.

References

[1] Bloom, A., Wysack, J., Griesbach, J.D., Lawitzke, A.: Space and ground-based sda sensor performance comparisons. In: Advanced Maui Optical and Space Surveillance Technologies Conference, pp. 1–11 (2022). Maui Economic Development Board Maui, HW

[2] Block, J.M., Curtis, D.H., Bettinger, R.A., Wilmer, A.P.: Cislunar sda with low-fidelity sensors and observer uncertainty. sat **1000**, 1 (2022)

[3] Thornton, D., Little, B., Steward, B., Cobb, R., Colombi, J., Meyer, D.: Dim cislunar target tracking with geo-and heo-based optical sensors. Journal of Applied Remote Sensing **16**(4), 044511–044511 (2022)

[4] Wilmer, A., Bettinger, R.A., Little, B.: Cislunar periodic orbit constellation assessment for space domain awareness of l1 and l2 halo orbits. In: ASCEND 2021, p. 4191 (2021)

[5] Wilmer, A.P., Bettinger, R.A., Little, B.D.: Preliminary viability assessment of cislunar periodic orbits for space domain awareness. In: 2021 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) (2021)

[6] Dahlke, J.A., Wilmer, A.P., Bettinger, R.A.: Preliminary comparative assessment of l2 and l3 surveillance using select cislunar periodic orbits. In: AAS/AIAA Astrodynamics Specialist Conference, pp. 1–19 (2022)

[7] Gupta, M., Howell, K.C., Frueh, C.: Earth-moon multi-body orbits to facilitate cislunar surveillance activities. In: AIAA/AAS Astrodynamics Specialist Conference, pp. 17–18 (2021)

[8] Gupta, M., Howell, K.C.: Leveraging earth–moon resonant orbits for access throughout the cislunar region. The Journal of the Astronautical Sciences **72**(4), 42 (2025)

[9] Visonneau, L., Shimane, Y., Ho, K.: Optimizing multi-spacecraft cislunar space domain awareness systems via hidden-genes genetic algorithm. The Journal of the Astronautical Sciences **70**(4), 22 (2023)

[10] Harris, C., Thomas, D., Kadan, J., Schroeder, D., Black, J.: Expanding the space surveillance network with space-based sensors using metaheuristic optimization techniques. In: Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) (2021)

[11] Hartigan, M., Smith, D., Lightsey, E.G.: Optimization of early-phase cislunar navigation constellations for users near the lunar south pole. In: 33rd AAS/AIAA Space Flight Mechanics Meeting, Austin, TX (2023)

[12] Gupta, M., Howell, K.C., Frueh, C.: Constellation design to support cislunar surveillance leveraging sidereal resonant orbits. In: 33rd AAS/AIAA Space Flight Mechanics Meeting (2023)

[13] Waldecker, C., Howell, K.: On-orbit servicing networks in cislunar space: A framework for orbit selection, transfer design, and scheduling. The Journal of the Astronautical Sciences **72**(6), 65 (2025)

[14] Bhadauria, S., Frueh, C., Howell, K.: Cislunar space domain awareness using bicircular four body geometry. In: AAS/AIAA Astrodynamics Specialist Conference (2022)

[15] Bhadauria, S., Frueh, C.: Optical observation regions in cislunar space using the bi-circular restricted four body problem geometry. In: Proceedings of the 2022 Advanced Maui Optical and Space Surveillance Technologies Conference, pp. 28–30 (2022)

[16] Tomita, K., Shimane, Y., Ho, K.: Multi-spacecraft predictive sensor tasking for cislunar space situational awareness. In: Proceedings of the Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference, p. 22 (2023)

[17] Gualdoni, M.J., DeMars, K.J.: Impartial sensor tasking via forecasted information content quantification. Journal of Guidance, Control, and Dynamics **43**(11), 2031–2045 (2020)

[18] Adurthi, N., Singla, P., Majji, M.: Mutual information based sensor tasking with applications to space situational awareness. Journal of Guidance, Control, and Dynamics **43**(4), 767–789 (2020)

[19] Fedeler, S., Holzinger, M., Whitacre, W.: Sensor tasking in the cislunar regime using monte carlo tree search. Advances in Space Research **70**(3), 792–811 (2022)

[20] Siew, P.M., Jang, D., Roberts, T.G., Linares, R., Fletcher, J.: Cislunar space situational awareness sensor tasking using deep reinforcement learning agents. In: 2022 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), Maui, Hawaii (2022)

[21] Fedeler, S., Holzinger, M., Whitacre, W.: Decentralized decision making over

random graphs for space domain awareness. Advances in Space Research **73**(10), 5266–5283 (2024)

[22] Pasquale, A., Zanotti, G., Prinetto, J., Ceresoli, M., Lavagna, M.: Cislunar distributed architectures for communication and navigation services of lunar assets. Acta Astronautica **199**, 345–354 (2022) https://doi.org/10.1016/j.actaastro.2022.06.004

[23] Gecgel Cetin, S., Karabulut Kurt, G., Vazquez-Castro, Á.: Secure and robust communications for cislunar space networks. In: IET Conference Proceedings CP873, vol. 2023, pp. 45–52 (2023). IET

[24] Yuan, A., Hu, Z., Zhang, Q., Sun, Z., Yang, Z.: Toward the age in cislunar communication: An aoi-optimal multi-relay constellation with heterogeneous orbits. IEEE Journal on Selected Areas in Communications **42**(5), 1420–1435 (2024)

[25] Lee, H.W., Shimizu, S., Yoshikawa, S., Ho, K.: Satellite constellation pattern optimization for complex regional coverage. Journal of Spacecraft and Rockets **57**(6), 1309–1327 (2020)

[26] Patel, M., Shimane, Y., Lee, H.W., Ho, K.: Cislunar satellite constellation design via integer linear programming. The Journal of the Astronautical Sciences **71**(3), 26 (2024)

[27] Shimane, Y., Tomita, K., Ho, K.: Cislunar space situational awareness constellation design and planning with facility location problem. Journal of Spacecraft and Rockets, 1–23 (2025)

[28] Clareson, T.H., Fox, M.C., Amato, D.K., Lee, H.W.: Embedded state estimation for optimization of cislunar space domain awareness constellation design. Journal of Spacecraft and Rockets **62**(3), 898–914 (2025)

[29] Choi, E., Ho, K.: Orbital depot location optimization for satellite constellation servicing with low-thrust transfers. Journal of Spacecraft and Rockets, 1–10 (2025)

[30] Patel, M., Tomita, K., Ho, K.: Concurrent optimization of satellite phasing and tasking for cislunar space situational awareness. The Journal of the Astronautical Sciences **72**(5), 50 (2025)

[31] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., Zhang, Q.: JAX: Composable Transformations of Python+NumPy programs. http://github.com/jax-ml/jax

[32] Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2024). https://www.gurobi.com

[33] Bertsekas, D.P.: Nonlinear programming. Journal of the Operational Research Society **48**(3), 334–334 (1997)

[34] Danskin, J.M.: The Theory of Max-min and Its Application to Weapons Allocation Problems. Springer, ??? (2012)

[35] Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)

[36] Goodfellow, I.: Deep learning. MIT press (2016)

[37] Crassidis, J.L., Alonso, R., Junkins, J.L.: Optimal attitude and position determination from line-of-sight measurements. The Journal of the Astronautical Sciences **48**(2), 391–408 (2000)