Report

Satellite Constellation Design Optimization for Maximum Surface Coverage and Minimum Revisit Time

Davide Basso [1], Nicolò Basso [2]

[1] Delft University of Technology dbasso@tudelft.nl
[2] Delft University of Technology nbasso@tudelft.nl

Abstract: The aim of this project is to optimize a constellation of three satellites, such that it maximizes the surface covered and minimizes the revisit time in a specific area and time frame. An initial simplified problem is considered, where the orbit of single satellite is optimized by adjusting its initial RAAN and inclination. Then, the state of a constellation of three satellites is considered, making using of different global optimization algorithms.

1. Statement of contribution

Topic	**Davide Basso**	**Nicolò Basso**
Problem definition	Together	
Physical model	Lead	Support
MATLAB implementation	Lead	Support
Simulated Annealing	Implemented	Reviewed
Preliminary analysis	Support	Lead
Sensitivity analysis	Support	Lead
Results interpretation	Support	Lead
Conclusions	Support	Lead
Report writing	Together (AI-assisted)	
GitHub Copilot (coding)	Used	Used

2. Introduction

Satellite orbits must be carefully planned in order to accomplish specific goals. Designing the satellites' orbits in order to maximise their effectiveness gets increasingly challenging as the number of satellites grows. Solving an optimisation problem is essential in such a scenario, given that space launches are costly and good mission planning may result in less satellites or improved coverage.

3. Problem

A small constellation comprising three satellites is considered in this study. The satellites operate in Low Earth Orbit (LEO). The primary objective is to maximize coverage over a designated target region, selected here as Italy and South Africa. A simulation time frame ΔT of ten hours is adopted, which is

Citation: Davide Basso; Nicolò Basso. Satellite Constellation Design Optimization for Maximum Surface Coverage and Minimum Revisit Time. *T.U. Delft* **2026,** 17, April.

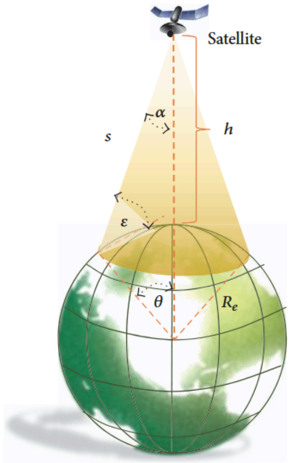


Figure 1: Satellite footprint projection (Savitri et al., 2017).

sufficiently long to capture the representative orbital evolution and coverage dynamics of the LEO constellation. A minimum elevation angle β_{min} of 30° is considered. We also consider an ideal revisit T_r time of 1 hour.

3.1. Optimization problem

The constellation configuration is determined by the Keplerian state of each one of its satellites at a specific time. Those combined states constitutes what will be called the state of the constellation. Each satellite initial state is subject to a set of constraints, described in next section.

The problem considered in this work is defined as: "Determine the orbital parameters of a three-satellite LEO constellation that maximise coverage over the target area while minimising revisit time". During the course of the research, the work of (Savitri et al., 2017) will be used as a reference.

3.1.1. Bounds & Constraints

The constraints originate from the following assumptions, both to have more realistic results and to simplify the model:

- Orbits must stay within the LEO boundaries;
- Given Italy and South Africa as the target regions, the orbital inclination is constrained to be at least equal to the latitude of the country's southernmost point, ensuring adequate ground coverage. In this case, 15 degrees were chosen to guarantee a proper margin.

Parameter	Value
Semimajor axis	6578 to 7,178 km
Eccentricity	0 - 0.03
Inclination	15 - 165 degrees
RAAN	0 - 360 degrees
Argument of periapsis	0 - 360 degrees
Orbit height	$h_{\text{min}} = a(1-e) - R_{\oplus} \geq 200$ km
LEO	$h_{\text{max}} = a(1+e) - R_{\oplus} \leq 800$ km

Table 1: Constraints.

3.1.2. Design Variables

The initial state of each satellite can be fully described by its Cartesian state with respect to Earth:

$$[x \;\; y \;\; z \;\; \dot{x} \;\; \dot{y} \;\; \dot{z}]^{\mathrm{T}} \tag{1}$$

Where the coordinates x, y, z and their derivatives $\dot{x}, \dot{y}, \dot{z}$ are to be taken in the Earth Centered Inertial (ECI) reference frame.

Alternatively, more conveniently, a Keplerian set of six orbital parameters can be used (Wakker, 2015):

$$[a \;\; e \;\; i \;\; \Omega \;\; \omega \;\; \theta]^{\mathrm{T}} \tag{2}$$

Where each element of the set represents, respectively, the semi-major axis, eccentricity, inclination, right ascension of the ascending node (RAAN) and true anomaly of the orbit along its instantaneous orbit.

Given we are considering a constellation of three satellites, the variables of the optimization problem are 18.

3.1.3. Objective Function

The orbits of the satellites, with their combined initial states constituting the design vector of the problem $\boldsymbol{x}$, are used to evaluate the achieved coverage of the constellation.

A single satellite j of the constellation traces an orbit descrived by the set of positions:

$$\boldsymbol{r}_j(t) \qquad t_i \leq t \leq t_f \tag{3}$$

With t_i and t_f being the initial and final times of the propagation.

Each infinitesimal area $\mathrm{d}A$ of the surface is approximated by a point p at position $\boldsymbol{r}_p$. We study whether this point is covered by the a satellite in position $\boldsymbol{r}_S$. This is true if the elevation angle with respect to the satellite β, as seen from an observer in $\boldsymbol{r}_p$, is larger than the minimum elevation angle $\beta_{\min}$.

However, we can also relate the minimum elevation angle to a maximum slant range $d_{\max}$, which is calculated as:

$$d_{\max} = \sqrt{r^2 + (r+h)^2 - 2r(r+h)\sin\left(\beta_{\min} + \arcsin\left(\frac{r}{r+h}\cos(\beta_{\min})\right)\right)} \tag{4}$$

Thus

$$c(\boldsymbol{r}_S, \boldsymbol{r}_p) = \begin{cases} 1 & \text{if } d \leq d_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Mathematically, it can be written as:

$$d = \|\boldsymbol{r}_p - \boldsymbol{r}_S\| \leq d_{\max} \tag{6}$$

For performance purposes, a $\mathscr{K}$d-tree is used, implemented in MATLAB through the `rangesearch` function.

The contribution $S_{\hat{p},t}$ of every point $\hat{p}$ covered by the satellite to a global score S is:

$$S_{\hat{p},t} = \min\bigl(t - t_{\hat{p},\text{last visit}}, T_\text{r}\bigr) \tag{7}$$

Where T_r is the ideal revisit time, and $t_{p,\text{last visit}}$ the last time at which that point was visited. This time is updated during the score simulation.

The final score is computed as:

$$S = \sum_{t_i} \sum_{\hat{p}} S_{\hat{p},t_i} \cdot c(\boldsymbol{r}, \boldsymbol{r}_A) \tag{8}$$

The maximum achievable score is:

$$S_\text{max} = (\Delta T + T_\text{r}) \times \#p \tag{9}$$

With $\#p$ being the number of points used for the simulation.

We can now define the cost C of the objective function as:

$$C = S_\text{max} - S \tag{10}$$

3.1.4. Optimization Problem Formulation

The optimization problem can then be formulated as:

$$\begin{aligned} \min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad & \boldsymbol{x} \subseteq \mathbb{R}^{18} \\ \text{s.t.} \quad & \underline{\boldsymbol{x}} \leq \boldsymbol{x} \leq \overline{\boldsymbol{x}}, \\ & g(\boldsymbol{x}) \leq 0 \end{aligned} \tag{11}$$

Where, after defining the constraints for a single satellite of the constellation, following data in Table 1:

$$\underline{\boldsymbol{x}}_j = \begin{bmatrix} \underline{a}_j \\ \underline{e}_j \\ \underline{i}_j \\ \underline{\Omega}_j \\ \underline{\omega}_j \\ \underline{\theta}_j \end{bmatrix} = \begin{bmatrix} 6578 \text{ km} \\ 0 \\ 15 \text{ deg} \\ 0 \text{ deg} \\ 0 \text{ deg} \\ 0 \text{ deg} \end{bmatrix}, \quad \overline{\boldsymbol{x}}_j = \begin{bmatrix} \overline{a}_j \\ \overline{e}_j \\ \overline{i}_j \\ \overline{\Omega}_j \\ \overline{\omega}_j \\ \overline{\theta}_j \end{bmatrix} = \begin{bmatrix} 7178 \text{ km} \\ 0.03 \\ 165 \text{ deg} \\ 360 \text{ deg} \\ 360 \text{ deg} \\ 360 \text{ deg} \end{bmatrix} \tag{12}$$

$$\boldsymbol{g}_j(\boldsymbol{x}_j) = \begin{bmatrix} \bigl(a_j(1-e_j) - R_\oplus\bigr) - 200 \text{ km} \\ 800 \text{ km} - \bigl(a_j(1+e_j) - R_\oplus\bigr) \end{bmatrix} \tag{13}$$

We can combine them to obtain the constellation equivalents:

$$\underline{\boldsymbol{x}} = \begin{bmatrix} \underline{\boldsymbol{x}}_1 \\ \underline{\boldsymbol{x}}_2 \\ \underline{\boldsymbol{x}}_3 \end{bmatrix}, \quad \overline{\boldsymbol{x}} = \begin{bmatrix} \overline{\boldsymbol{x}}_1 \\ \overline{\boldsymbol{x}}_2 \\ \overline{\boldsymbol{x}}_3 \end{bmatrix}, \quad \boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{g}_1(\boldsymbol{x}_1) \\ \boldsymbol{g}_2(\boldsymbol{x}_2) \\ \boldsymbol{g}_3(\boldsymbol{x}_3) \end{bmatrix} \tag{14}$$

3.2. Modelling Aspects

The formulation of the optimization problem relies on a set of modelling assumptions aimed at reducing computational complexity while preserving the essential physical characteristics of the system. Given the preliminary nature of the design problem, a simplified but representative modelling framework is adopted. The following sections detail the underlying dynamical model and the assumptions used in the evaluation of coverage.

3.2.1. Dynamical Model

The orbital propagation is based on the classical two-body problem, including the effect of Earth's oblateness through the second zonal harmonic (J_2). Higher-order spherical harmonics and other perturbations, including solar radiation pressure and drag, are neglected. Equation 15 shows the final acceleration model used. (Wakker, 2015)

$$\frac{d^2\boldsymbol{r}}{dt^2} = \frac{\mu}{r^2}\hat{\boldsymbol{r}} - \frac{3}{2}\mu J_2 \frac{R^2}{r^5} x \begin{bmatrix} 1 - 5\frac{z^2}{r^2} \\ 1 - 5\frac{z^2}{r^2} \\ 3 - 5\frac{z^2}{r^2} \end{bmatrix} \tag{15}$$

In the following block the MATLAB implementation of the J_2 perturbation is presented. This perturbation is summed to the two-body acceleration.

matlab

```matlab
r = norm(position);
f = -(1.5 * j2 * mu * R ^ 2) / r ^ 5;

acceleration = f * position .* [
    1 - (5 * position(3) ^ 2) / r ^ 2;
    1 - (5 * position(3) ^ 2) / r ^ 2;
    3 - (5 * position(3) ^ 2) / r ^ 2
    ];
```

3.2.2. Ground Sampling

In order to evaluate the coverage performance of the satellite constellation, a set of points uniformly distributed over the target region was required to represent the ground surface. This was achieved using a Fibonacci lattice method, also known as the golden spiral, which generates equally spaced points on the surface of a sphere (Saff & Kuijlaars, 1997). By applying this method to the Earth's spherical model, a set of unit vectors was obtained representing the ground sample points within Italy.

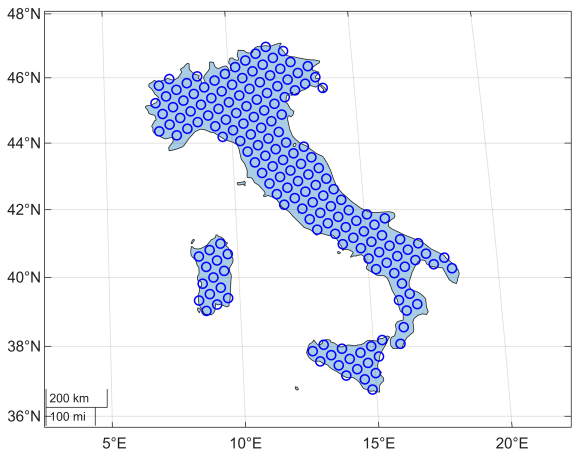


Figure 2: Example of ground sampling inside Italy.

When a point is inside the instantaneous footprint of at least one satellite, it is deemed covered. These sites are used as reference locations for coverage assessment. By preventing clustering close to the poles and guaranteeing a nearly uniform sample, the Fibonacci-based distribution offers a reliable and computationally effective depiction of the target region.

Following is the MATLAB function implementation of the method.

matlab

```matlab
function points = fibonacci_sphere(n)
    golden_ratio = (1 + sqrt(5)) / 2;

    i = (0:n - 1)'; % indices 0 ... n-1

    theta = acos(1 - 2 * (i + 0.5) / n); % polar angle
    phi = 2 * pi * i / golden_ratio; % azimuthal angl

    points = [sin(theta) .* cos(phi), ...
              sin(theta) .* sin(phi), ...
              cos(theta)]'; % 3 x n
end
```

The obtained points are then rescaled onto a geodetic model of the Earth as reference (National Geospatial-Intelligence Agency, 2014).

3.2.3. Model Limitations

The buildup of mistakes resulting from the neglected disturbances is the most prominent model limitation. The inaccuracy increases at a rate of kilometres each day. Considering the problem's initial scope, it is not regarded as a problem. Additional perturbations might be taken into consideration to fix this, but doing so would increase the model and computation time.

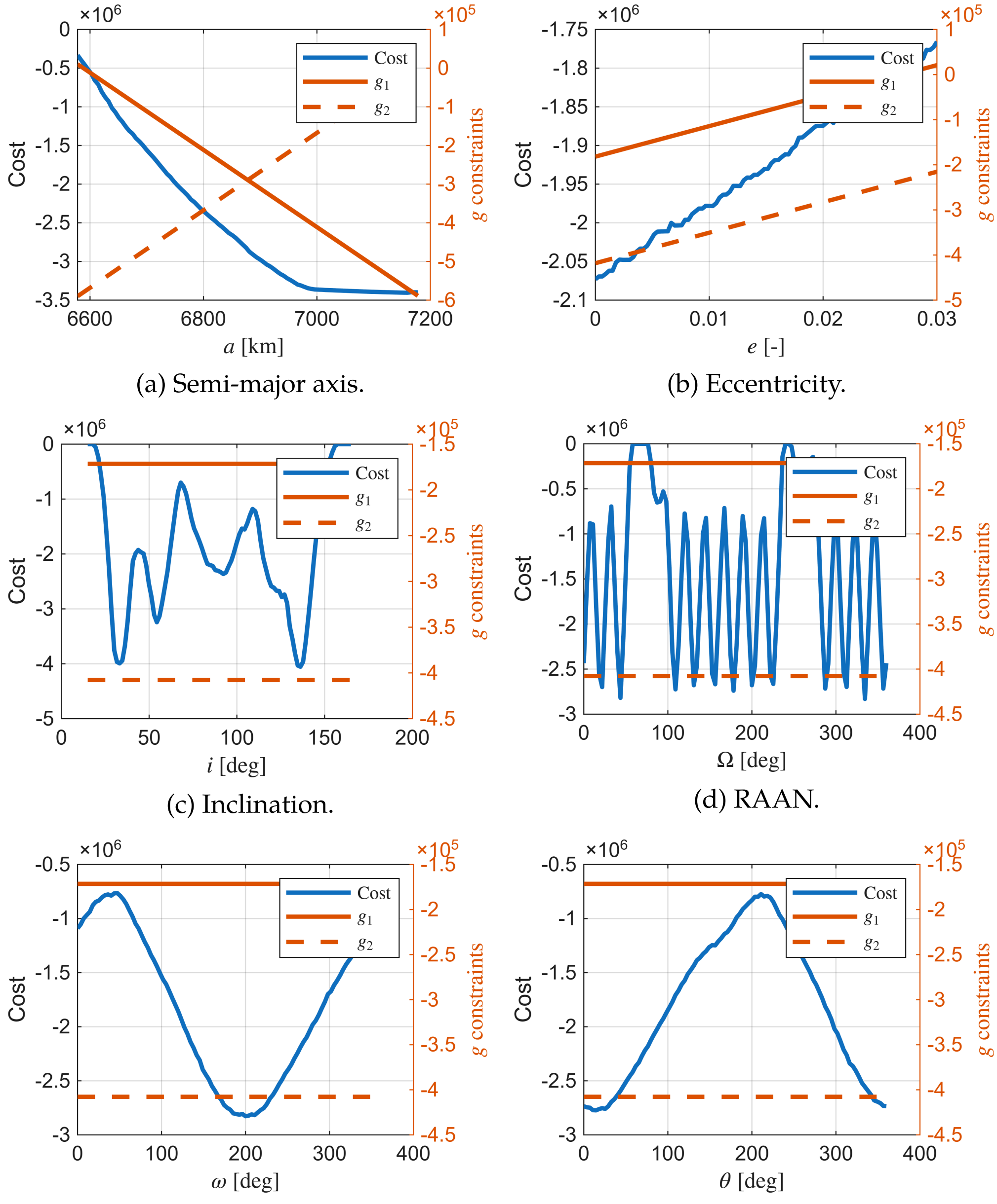
4. Initial problem investigation

Here a simplified proble with a single satellite is considered. Figure 3 illustrates a One Variable at a Time analysis for the six orbital elements. For this analysis, an orbit that passes over the target was first identified.

This choice was necessary because, if the orbit did not include at least one pass over the target, varying certain parameters (e.g., the semi-major axis) without an appropriate RAAN or inclination would result in a flat response within the considered timeframe, failing to capture their actual influence on system behavior.

The initial state used is:

$$\begin{bmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ \theta \end{bmatrix} = \begin{bmatrix} 6860 \text{ km} \\ 0.00054 \\ 97.5643^\circ \\ 182.719^\circ \\ 275.0914^\circ \\ 85.03242^\circ \end{bmatrix} \tag{16}$$



(a) Semi-major axis. (b) Eccentricity.
(c) Inclination. (d) RAAN.
(e) Argument of pericenter. (f) True anomaly.

Figure 3: Boundedness and Monotonicity Analysis.

The objective function shows a highly non-linear and non-convex behavior for some variables, characterized by significant oscillations. This is particularly true for inclination i and the RAAN Ω, while for ω and θ the figure shows a smoother landscape. This phenomena is due to the periodic natures of these four parameters, being them angles. The inclination and the RAAN plots are rich of spikes: intuitively, Ω and i influence the orbit the most and a small change could mean missing the target completely. It is clear from the figure how the system is well bounded with respect to these four parameters, that does not show a nature of monotonicity nor of convexity for these.

4.1. Numerical noise

Some noise is visible in the plots of ω and θ, due to the discretization of the terrain, as some passes are barely enough to cover some of the points. Some numerical noise is present when running very small steps in the objective function due to the integrator. This cause issues in the derivatives and first order algorithms, as can be seen in Figure 7. This could be solved by an appropriate selection of the step size, but it has not been done due to the choice of running zero order algorithms for the final problem.

4.2. Domain

We can trace a contour plot of the feasible domain, limited by the lower and upper bounds described in Section 3.1.1, and constrained by g_1 and g_2. Given that the constraints depend only on a and i, and that they are the same for all satellites, we can create a 2D plot. What we obtain is a convex domain. We also notice that the constraints are linear.

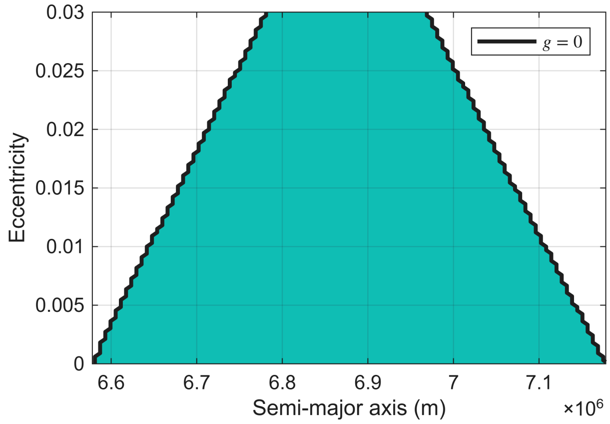


Figure 4: Contour plot of the feasible domain of the optimization problem.

4.3. Sensitivity analysis

A comprehensive sensitivity analysis was conducted, centered on the initial state defined in the preceding section. To ensure a robust characterization of the design space, one hundred discrete points were sampled for each individual variable while holding all other parameters constant.

The resulting logarithmic sensitivities, which illustrate the proportional relationship between parameter variations and the objective function, are presented in Figure 5.

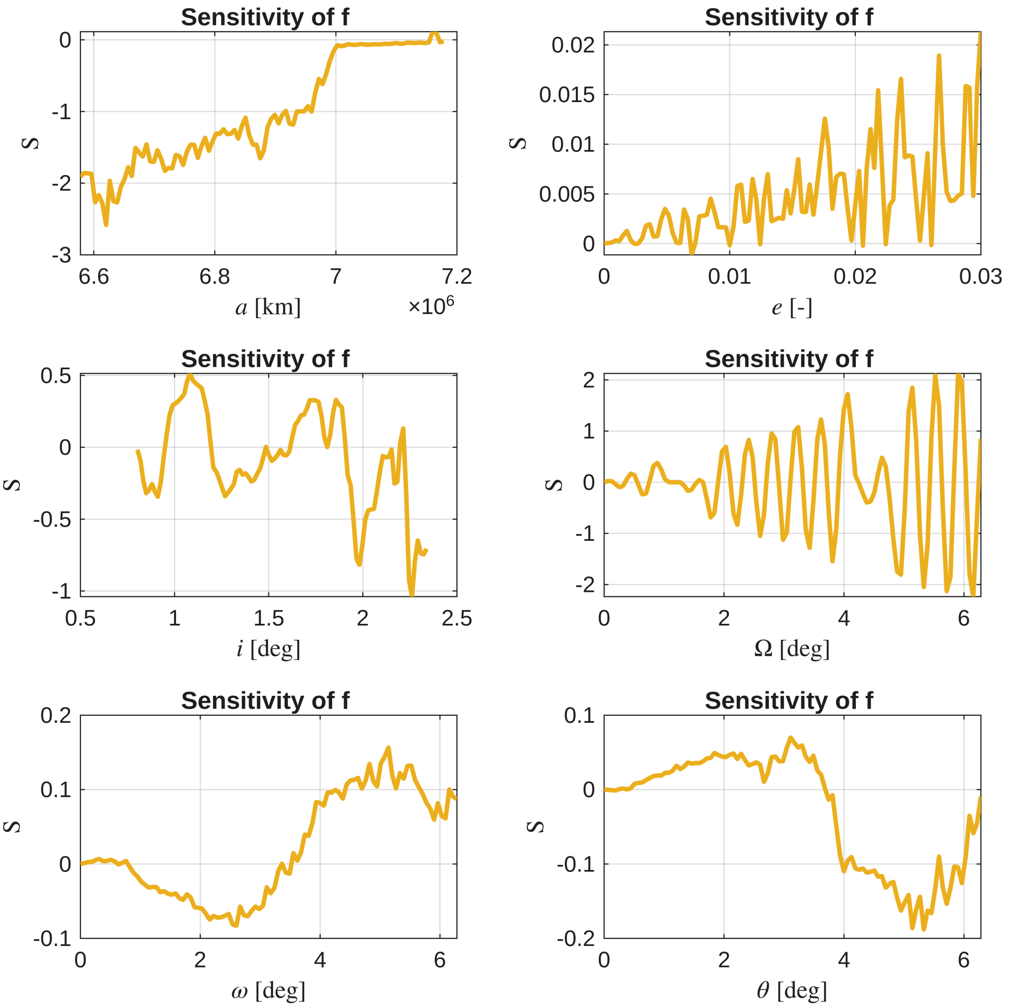


Figure 5: Sensitivity analyses for each design variable.

We see different behaviors between the variables:

- Semimajor axis: This parameter exhibits negative sensitivity across all values. This is explained by the fact that increasing the semimajor axis expands the coverage area, which subsequently decreases the objective function. However, due to operational constraints, selecting the maximum value is not a feasible option.
- Eccentricity: This variable has a significantly lower impact on the function compared to other parameters. Generally, increasing eccentricity leads to a highly unstable problem without yielding substantial benefits, as the sensitivity remains almost consistently positive.
- Inclination: This shows both positive and negative sensitivities with numerous zero-crossings. The objective function is clearly highly dependent on this parameter.
- RAAN: This parameter exhibits "spike-like" behavior. The sensitivities oscillate more aggressively as the RAAN increases; this is primarily an artifact of the logarithmic sensitivity scaling.
- Argument of pericenter (ω) and True anomaly (θ): These display similar patterns, each characterized by two zero-crossings. Given the low sensitivity magnitudes, this behavior is likely dictated by the position of the

apogee (for ω) and the satellite (for θ) relative to the propagation window. The two crossings indicate optimal solutions where the satellite spends the majority of its time at the apogee over the target regions.

The presence of high-frequency sensitivity and multiple local minima confirms that gradient-based methods (like Sequential Quadratic Programming) are unsuitable for this problem without an exceptionally good starting guess. This justifies the transition to population-based algorithms (Genetic Algorithm and Particle Swarm Optimization), which use stochastic search mechanisms to “ignore” local irregularities identified in this sensitivity study.

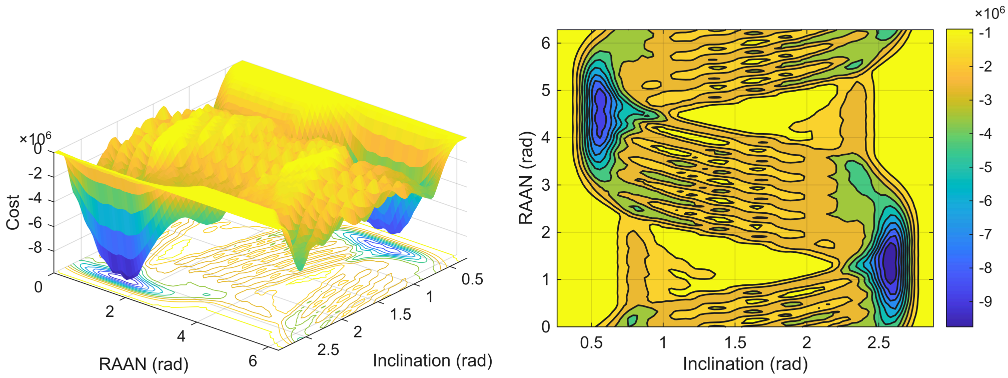
5. Initial optimization on simplified problem

As an initial optimization problem, we consider a simplified version with only one satellite, and where only two variables are being optimized: the inclination i and the RAAN Ω. The same upper and lower bounds described in Section 3.1.1 apply.

We note that both constraints do not depend on the variables of the simplified problem:

$$\frac{\mathrm{d}g_1}{\mathrm{d}i} = \frac{\mathrm{d}g_1}{\mathrm{d}\Omega} = \frac{\mathrm{d}g_2}{\mathrm{d}i} = \frac{\mathrm{d}g_2}{\mathrm{d}\Omega} = 0 \tag{17}$$

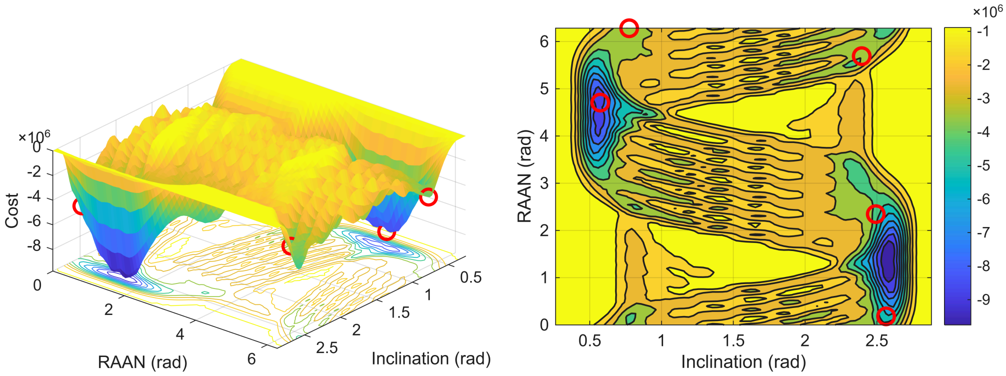
Given the limited dimensionality of the problem, we are able to perform to evaluate the objective function for a grid of points, obtaining a approximate surface representing the costs for all points in the subspace.



(a) Cost surface. (b) Cost isolines.

Figure 6: Cost function of the simplified problem.

What we see in Figure 6 is that the function presents many local minima. A single run of a Steepest Descent, Newton or Quasi-Newton algorithm will likely end on one of the local minima. Figure 7 shows the many local minima found using just five runs of SQP with random initial guesses.



(a) Cost surface. (b) Cost isolines.

Figure 7: Local minima found in 5 SQP runs with random initial guesses.

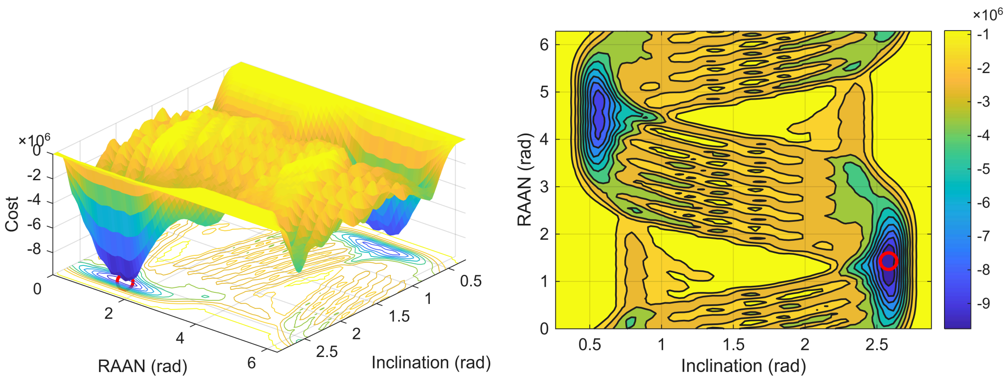
5.1. Motivation of optimization approach, choices

The objective function presents more than one local minima: the chosen algorithm must be able to escape one of those and find the global optimum. Another caution that has to be taken, regards the numerical noise and consequent derivatives issues that may arise. For this reasons, particle swarm algorithm and other order 0 global optimization algorithms would be a good fit for this problem.

This algorithm works by creating initial particles with given initial velocities. It finds the lowest function value and the optimal location by evaluating the objective function at each particle location. Based on the current velocity, each particle's optimal location, and the optimal locations of their neighbors, it selects new velocities. The particle locations, velocities, and neighbors are then updated iteratively and, until the algorithm meets a stopping criterion, iterations continue (MathWorks, 2026).

For this problem, Genetic algorithms represented another possibility and were already used in similar cases (Savitri et al., 2017). However particle swarm represents a simpler solution with less parameters to tune and is simple to parallelize on MATLAB.

Figure 8 show the obtained global optimum using the chosen algorithm.



(a) Cost surface. (b) Cost isolines.

Figure 8: Obtained global optimum in the simplified problem.

5.2. Investigation of obtained optimum

The obtained minimum is at $[i, \Omega] = [2.5804, 1.4213]$ rad (with cost 2.56E7). The corresponding orbit is shown in Figure 9. From the figure, it can be see that the optimum is a retrograde orbit ($i > \frac{\pi}{2}$). The RAAN is such that the orbit apogee is over South Africa most of the time. This is intuitive and corresponds to a well-constrained solution: both countries cannot be covered within the selected propagation period with a single satellite. Rather, the algorithm found that covering South Africa exclusively results in a higher score because it has more points than Italy.

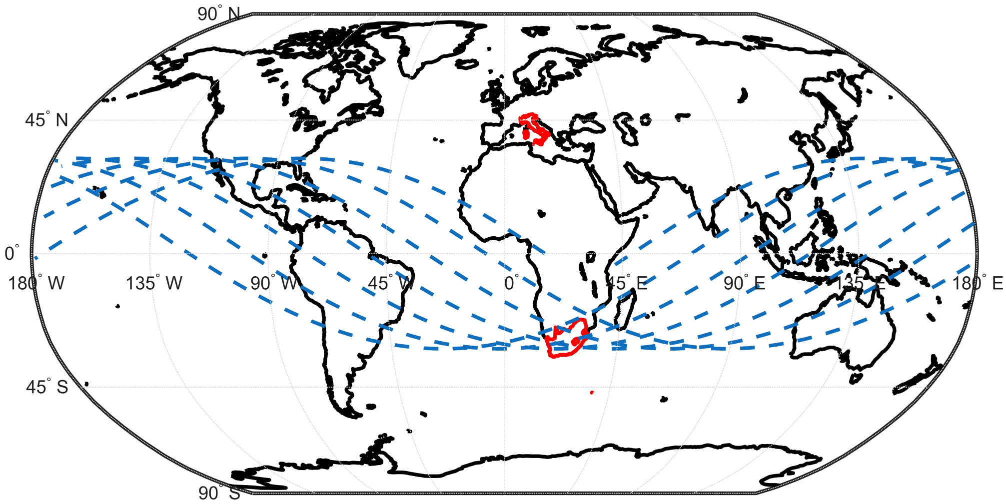


Figure 9: Ground track corresponding to global optimum found for the simplified problem.

5.3. Conclusions

The investigation into the simplified two variable problem (i and Ω for a single satellite) provided insights into the nature of the objective function. Analysis of the cost surface plots reveals a highly multimodal landscape characterized by at least two local minima across the feasible domain. This complexity was confirmed empirically by five independent runs using the SQP algorithm, that yielded a wide spread of different solutions.

These results demonstrate that gradient-based methods are insufficient for this problem. Because these algorithms rely on local descent, they inevitably converge to the local minimum nearest to their starting point rather than seeking the global optimum.

The simplified case confirms that the choice of algorithm that prioritize global is the most critical factor in solving this problem.

6. Optimization of actual problem

The complete, 18 dimensions design space, problem requires an approaches a constrained optimization algorithm. We have tested two different approaches:

1. **Simulated Annealing**, using our own implementation.
2. **Genetic Algorithm**, using MATLAB builtin `ga` implementation.

6.1. Motivation of optimization approach

Both approaches where chosen as they are able to approximate the global optimum of the problem. As experienced with the simplified problem, we expect the new 18-dimensional cost function to present many local minima. For this reason, quadratic methods were avoided.

Both approaches make use of a penalty system to transform the constrained problem to an unconstrained one. The penalty system is implemented as follows:

MATLAB

```matlab
f = @(y) f_unconstrained(y) + p * sum(max(0, g(y))) ^ 2;
```

The order of the penality `p` is chosen equal to 10. This provides sufficient penalties to solutions outside boundaries, and no problems were observed during the optimizations.

6.1.1. Simulated Annealing

The behavior of a Simulated Annealing run depends on several configuration parameters:

- Initial temperature T_i: the temperature at the start of the run;
- Alpha coefficient α: determines how fast the temperature decreases (in our implementation, we have $T = T_i \alpha^{k-1}$, with k the # of the run);
- The initial guess $\boldsymbol{x}_0$.

Our implementation also uses Gaussian noise for the determination of neighbors, and uses a reflection techniques of candidates found outside the boundaries. The step scale is adjusted based on the ratio between temperatures T/T_i.

Given we do not know the optimal parameters, as well as the stochastic nature of the algorithm, we perform 12 runs, with parameters randomly selected within the following intervals.

Parameter	**Min Value**	**Max Value**
Initial Temperature[1]	1E2	1E5
Alpha	0.99	0.999
Initial Guess	$\underline{\boldsymbol{x}}$	$\overline{\boldsymbol{x}}$

Table 2: Simulated Annealing parameters range.

The convergence of the 12 runs is visible in Figure 10.

[1]Random values sampled inside this range are distributed equally in logarithmic scale (base 10).

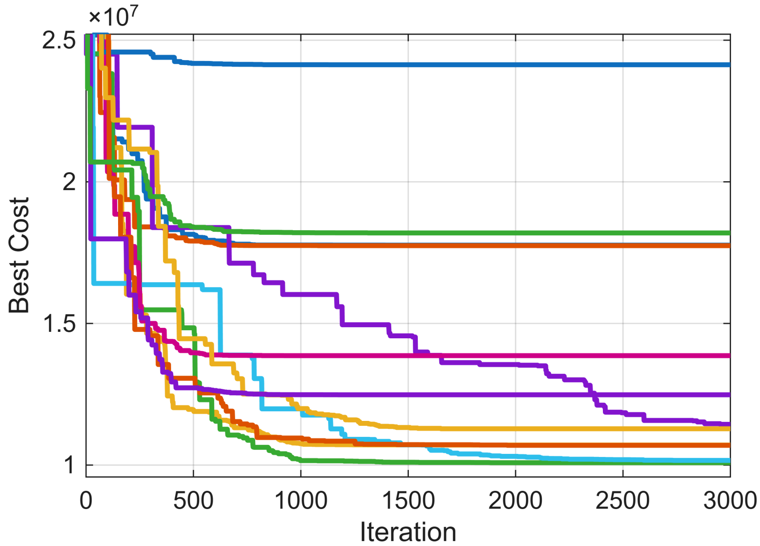


Figure 10: Best cost found per iteration for each run of Simulated Annealing.

The results for each run are reported in Table 3. The best solution found has cost 1.0085E7.

Run ID	Initial Temperature	Alpha	Best Evaluation
1	264.0	0.9918	1.7764E7
2	34649.0	0.9907	1.7747E7
3	599.0	0.9937	1.0707E7
4	136.0	0.9986	1.1439E7
5	7698.0	0.9952	1.0085E7
6	42541.0	0.9974	1.0165E7
7	62175.0	0.9902	1.3863E7
8	1003.0	0.9920	2.4134E7
9	4401.0	0.9963	1.0698E7
10	56666.0	0.9964	1.1283E7
11	889.0	0.9914	1.2482E7
12	20156.0	0.9914	1.8196E7

Table 3: Results for each run of Simulated Annealing.

6.1.2. Genetic Algorithm

We chose Genetic Algorithm as a reference to compare the previous results with, having also being used in related research (Savitri et al., 2017). We use MATLAB builtin implementation, with a `PopulationSize` of 300 and a `FunctionTolerance` equal to 1. The obtained optimum, after 51 generations, is 8.7814E+6.

6.2. Observations, interpretation of results, conclusions

In Figure 11 and Figure 12 are visible the constellation ground tracks corresponding to the solutions found by the Simulated Annealing and Genetic Algorithm approaches described above.

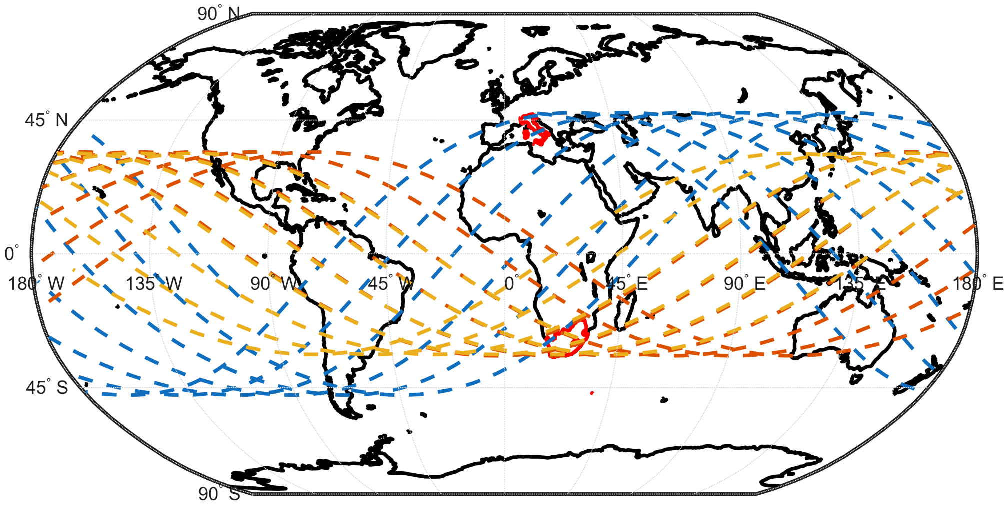


Figure 11: Ground track corresponding to optimum found with Simulated Annealing.

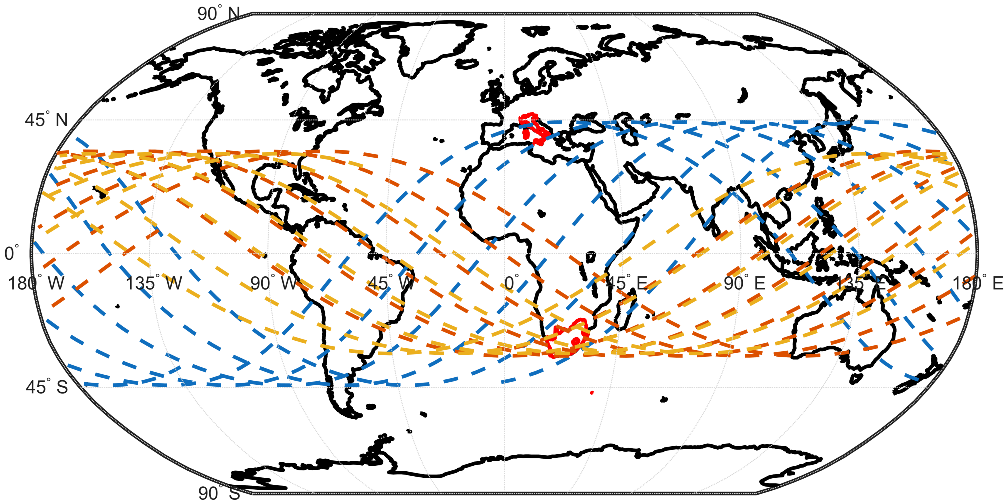
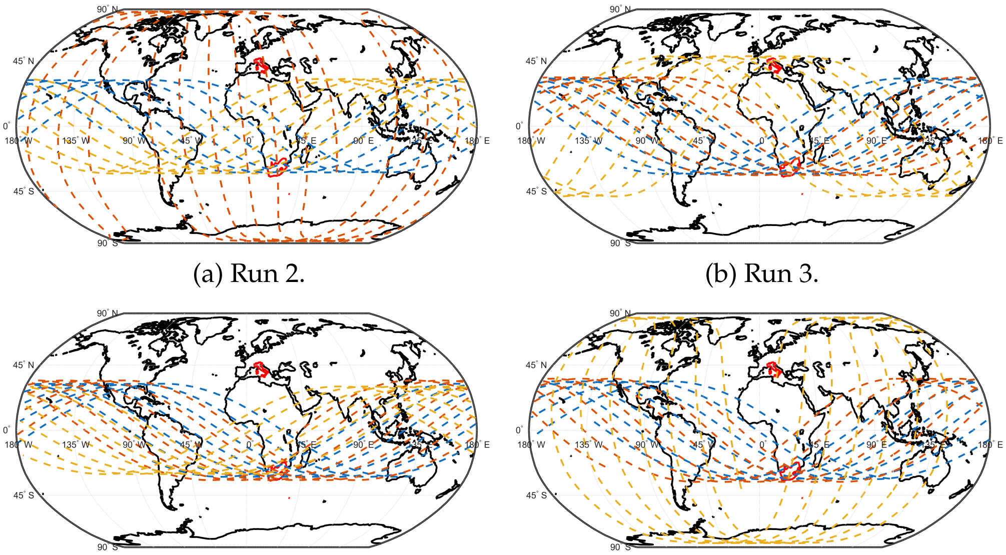


Figure 12: Ground track corresponding to optimum found with Genetic Algorithm.

Although the optimum found with the Genetic Algorithm approach has a lower cost, we can see how the two solutions are physically very similar: they both have two satellites flying exclusively above South Africa, with a difference in phase, and a third one with an higher inclination, covering both countries. The only substantial difference between the two seems to be relative phase between satellites' orbits.

In figure Figure 13 are shown alternative constellations corresponding to other Simulated Annealing run. We can see that the algorithm tried exploring very different solution approaches: some runs have the third satellite with very high inclination, others focus only on covering South Africa.

This result confirms that the Simulated Annealing implementation works well: the algorithm widely explored the solution space, and became occasionally trapped in different (we now know) local minima.



(a) Run 2. (b) Run 3.

(c) Run 6. (d) Run 11.

Figure 13: Ground track corresponding to alternatives optima found with Simulated Annealing.

7. Overall conclusions and recommendations

The project demonstrated the application of metaheuristic optimization techniques to the design of a three-satellite constellation. The primary goal was to maximize surface coverage over Italy and South Africa while minimizing revisit times. Initial sensitivity analysis revealed a very unstable and non-monotonic design space with significant numerical noise, which was attributed mostly to the discrete nature of the ground coverage grid. These characteristics confirmed that gradient-based methods were unsuitable for this specific problem, justifying the use of global search algorithms.

In the optimization of the 18-dimensional problem, the Genetic Algorithm identified a optimized configuration. The final design variables resulted in a constellation that favored multiple low-inclination orbits over polar ones to ensure more frequent passes over the target regions.

The implementation of the J2 perturbation model provided an appropriate balance between physical realism and computational speed. While the model did not include higher-order gravity terms or atmospheric drag, it successfully captured the nodal regression effects necessary to evaluate the stability of coverage over a ten-hour period.

8. Final Considerations and Future Work

Although the current results provide a viable orbital configuration, several areas for further investigation are recommended:

- Implementation of surrogate-based optimization: To address the high-frequency numerical noise found during sensitivity analysis, future studies could employ Kriging or other surrogate models. This would smooth the objective function and potentially allow for more efficient identification of the global optimum.

- High-fidelity force modeling: Long-term constellation stability should be evaluated by incorporating atmospheric drag and solar radiation pressure. These additions would provide a more accurate assessment of station-keeping requirements and the associated fuel costs.
- Ground grid: Investigating the impact of the grid point density in Italy and South Africa would help determine the minimum resolution required for accurate results while minimizing the computational cost per function evaluation.

References

MathWorks. (2026,). *How Particle Swarm Optimization Works*.

National Geospatial-Intelligence Agency. (2014). *Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems* (Technical Report No. NGA.STND.36_1.0.0_WGS84).

Saff, E. B., & Kuijlaars, A. B. J. (1997). Distributing many points on a sphere. *The Mathematical Intelligencer*, *19*(1), 5–11. https://doi.org/10.1007/bf03024331

Savitri, T., Kim, Y., Jo, S., & Bang, H. (2017). Satellite Constellation Orbit Design Optimization with Combined Genetic Algorithm and Semianalytical Approach. *International Journal of Aerospace Engineering*, *2017*, 1–17. https://doi.org/10.1155/2017/1235692

Wakker, K. (2015). *Fundamentals of Astrodynamics* (p.).

Appendix A: Matlab code

All the code used in this project is available at the following GitHub repository: https://github.com/BearToCode/constellation-optimization

Appendix B: Additional graphs/data

x Parameter	Value
x_1	6924538.679372338
x_2	0.02504923
x_3	2.310653363
x_4	5.423959259
x_5	6.135466156
x_6	6.089749088
x_7	6990477.358322447
x_8	0.016177615
x_9	2.54353293
x_{10}	1.728121761
x_{11}	6.054011189
x_{12}	3.078290056
x_{13}	7044388.30033035
x_{14}	0.018936984
x_{15}	2.552034749
x_{16}	0.73222706
x_{17}	5.86623512
x_{18}	5.685158285

Table 4: Simulated Annealing optimum.

x Parameter	Value
x_1	7102713.305461027
x_2	0.01017441
x_3	0.773839533
x_4	2.433352739
x_5	4.915166229
x_6	6.276070823
x_7	6971000.900729305
x_8	0.019668714
x_9	2.540680246
x_{10}	1.791005017
x_{11}	1.535096684
x_{12}	1.361946019
x_{13}	7144677.821079111
x_{14}	0.004165713
x_{15}	2.555087118
x_{16}	1.185805024
x_{17}	1.371533050
x_{18}	5.998713288

Table 5: Genetic Algorithm optimum.