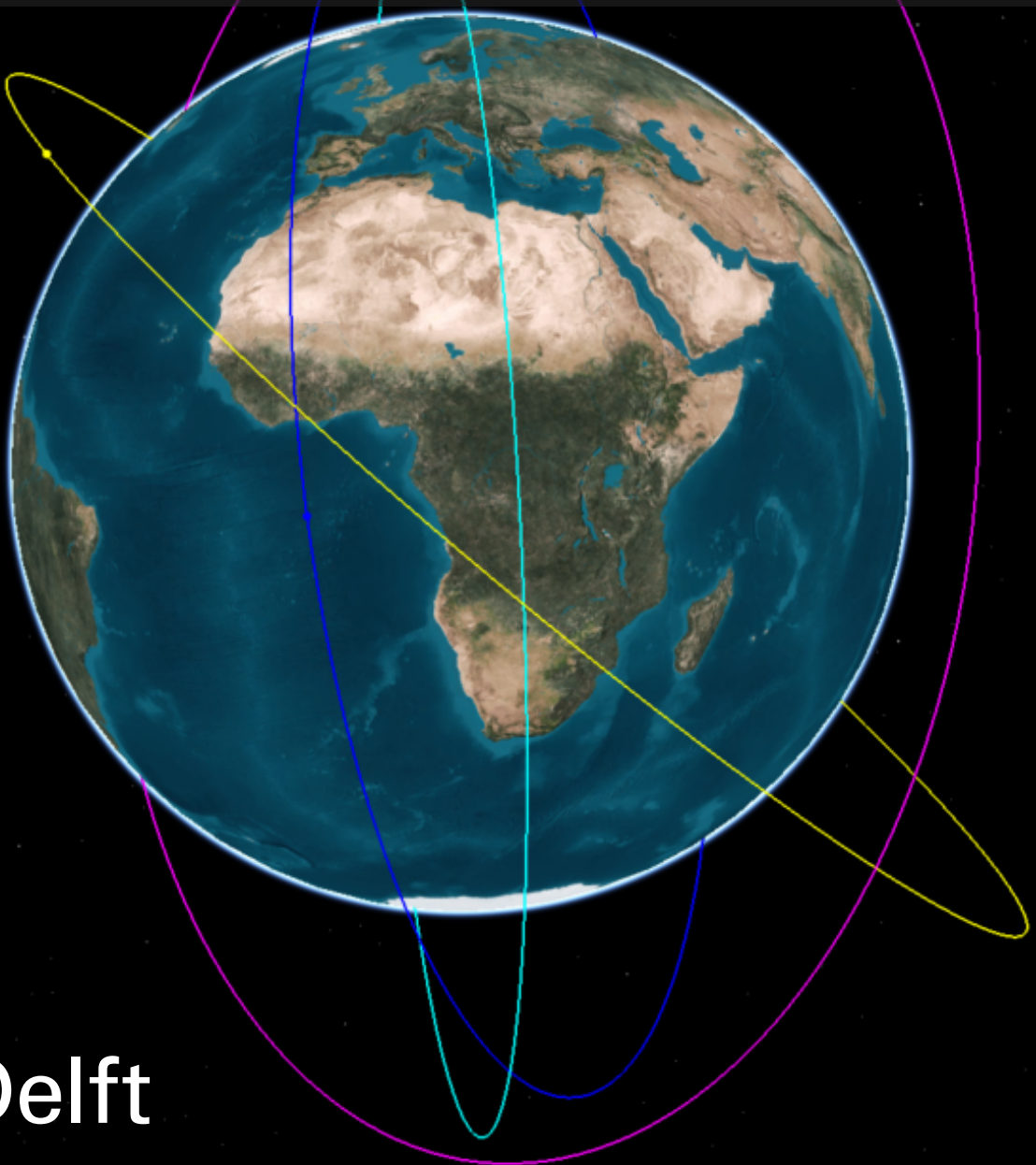
Constellation Optimization Using a Novel Coverage Analysis Method

MSc Thesis

Loek Sanders



Delft University of Technology

TU Delft

Constellation Optimization Using a Novel Coverage Analysis Method

by

Loek Sanders

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on TBD

Chairman:	Dr.ir. J. Bouwmeester	TU Delft
Supervisors:	Dr. J. Guo	TU Delft
	J. Campoy	OHB System AG
Independent examiner	Dr.ir. M.J. Heiligers	TU Delft
Faculty:	Faculty of Aerospace Engineering, Delft	

Cover: STK 12 visualization of an optimal satellite constellation.
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Preface

This thesis was made in collaboration with OHB System AG and TU Delft. In the summer of 2024, I was working as an intern for the predevelopment department of OHB. This is where I discussed the possibility of creating a new thesis topic of interest to the department, which is how the satellite constellation optimization topic came to be. The tool created during this thesis is aimed to provide early estimates of mission design for new proposals created by the predevelopment department.

I would like to thank all colleagues and friends I made at OHB for their support and amazing working environment the past year. In particular Javier Campoy and Stephan Holsten for their continued guidance and supervision from OHB side, and Jian Guo for his great tips and guidance from TU Delft.

Special thanks to Jaimy Thielen, Giorgio Balbo, Mateo Caruso, Luca Elbracht, and Noah Tarbah for being amazing friends throughout my whole university career, as well as to my family and friends back home for their unrelenting support.

Loek Sanders
Delft, September 2025

Summary

In the last decade, the number of Earth observation satellites in orbit have exponentially increased. These satellites are used for a wide range of applications, from environmental analysis and disaster monitoring to precision farming and even military surveillance. Due to these increasingly specialized demands of users, a rise has been seen in the number of Earth observation missions that make use of satellite constellations.

This development mandates the need for better and improved mission analysis tools, in particular for the design of satellite constellations. Since the launch of the first satellites, efforts have been made in developing these tools, and to find the most optimal configurations of satellite constellations. This thesis aims to add on to this research, by both providing a new, improved coverage analysis method as well as providing more research on the topic of satellite constellation optimization.

The analysis of the established research showed a clear path for improvements. Both the existing coverage analysis methods, as well as most optimizations did not consider either elliptical orbits or asymmetrical constellations, or both. Therefore the objective of the new coverage analysis method and constellation optimization was to be able to handle all types of orbits and all types of constellation configurations.

In constellation optimization, the coverage analysis method is used to evaluate candidate constellations based on the desired figures of merit and gives the candidate a score. These figures of merit were established as the revisit time and coverage of the constellation. As the evaluation function is called many times during optimization, the main objective of the coverage analysis method development was to make it as computationally fast as possible. The creation of a semi-analytical, grid-based method proved both fast in computation and accurate, as well as flexible in the type of orbits it can analyze.

With the coverage analysis method established, the optimizer framework could be developed. Here, the biggest variable to choose is the optimization algorithms. In this thesis, six algorithms were implemented and consequently compared. These algorithms included both single objective and multi objective optimization algorithms. The algorithms were compared on the most optimal constellation found, as well as the computation time of the full optimization. It was concluded that the genetic algorithm and the covariance matrix adaptation evolution strategy performed the best for single objective optimization. The non-dominated sorting genetic algorithm II was analyzed as a multi objective optimization and proved to be a good alternative to single objective optimization.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
AOP	Argument of perigee
CMA-ES	Covariance matrix adaptation evolution strategy
DE	Differential evolution
DEAP	Distributed evolutionary algorithms in Python
DOP	Dilution of precision
ECEF	Earth centered, Earth fixed
EO	Earth observation
FoV	Field of view
GA	Genetic algorithm
LEO	Low Earth orbit
NSGA-II	Non-dominated sorting genetic algorithm II
PSO	Particle swarm optimization
RAAN	Right ascension of the ascending node
RGT	Repeating ground track
SA	Simulated annealing
SoC	Streets-of-coverage
STK 12	System ToolKit 12
TA	True anomaly

Symbols

Symbol	Definition	Unit
a	Semi-major axis	[m]
C	Total cost	[-]
d	Distance	[m]
EA	Eccentric anomaly	[rad]
e	Eccentricity	[-]
el	Elevation angle	[rad]
h	Specific angular momentum	$[m^2/s]$
h_s	Altitude of the satellite	[m]
i	Inclination	[rad]
j	Orbit number	[-]
MA	Mean anomaly	[rad]
MA_ϕ	Mean anomaly at latitude	[rad]
n	Mean motion	[rad/s]
n_{sat}	Number of satellites	[-]
$\vec{n}$	Normal vector of the plane	[m]
$\vec{P}$	Position vector of the grid point	[m]
P_n	Nodal period	[s]
p	Semilatus rectum	[m]
R_e	Spherical radius of Earth	[m]
S	Number of satellites	[-]
R	Latitude belt	[-]

Symbol	Definition	Unit
R_ϕ	Latitude belt bound parallels	[-]
r	Radius	[m]
T	Orbital period	[s]
t	Observation time since epoch	[s]
U	Uniform distribution	[-]
u	Argument latitude	[rad]
w_{cvg}	Coverage weight	[-]
w_{rt}	Revisit time weight	[-]
w_{sat}	Number of satellites weight	[-]
ΔL	Visibility longitude range for non-rotating Earth	[rad]
Δt_{REV}	Revisit time	[s]
Δt_{REV}^{LOW}	Lower bound of revisit time	[s]
Δt_{REV}^{*}	Estimated revisit time	[s]
$\Delta\lambda$	Drift in longitude of successive passes	[rad]
$\Delta\Omega$	Visibility longitude range	[rad]
$\delta\Omega$	Rate of nodal regression	[rad]
$\tilde{\eta}_{FOV}$	Field of view of the sensor	[rad]
θ	Geocentric angular swath width	[rad]
Λ	Coverage angle	[rad]
λ	Longitude	[rad]
μ	Gravitational parameter of Earth	[m^3/s^2]
ν_ϕ	True anomaly at latitude	[rad]
τ	Required (maximum) revisit time	[s]
ϕ	Latitude	[rad]
Ω	Right ascension of the ascending node	[rad]
$\dot{\Omega}$	Right ascension of the ascending node change rate	[rad/s]
ω	Argument of perigee	[rad]
ω_e	Rotational rate of Earth	[rad/s]

Part I

Background

1

Introduction

In the past decade, the number of Earth observation satellites has increased exponentially. From 2013 to 2023 the number of satellites has seen a growth of 714.37% [65]. Furthermore, 86% of all Earth observation satellites is part of a satellite constellation [65]. This number is only expected to grow, with the majority of newly proposed satellite constellations being for Earth observation purposes [8].

Earth observation satellite constellations are used for a wide variety of objectives. These can range from environmental analysis [47] and disaster monitoring [9] to precision farming [53] and even military surveillance [48]. The rise in satellite constellation can also be related to more specialized demands from users. New developments in technology allow for more precise definitions of the purpose of the satellite constellations, as well as the narrowing or broadening of the area of interest.

These developments mandate the need for better and improved mission analysis tools, specifically for the design of the constellations. Ever since the launch of satellites, researchers have developed various ways of finding the most optimal constellation. One of the earliest studies was done in 1961 by Luders [35]. In this study, the streets-of-coverage satellite constellation is proposed, aimed to provide continuous global coverage. A decade later, in 1971, Walker introduced the famous Walker constellations. Like Luders' streets-of-coverage constellation, the Walker constellation is designed for continuous global coverage. More recently, in 2004, another well-known constellation was proposed by Mortari, namely the flower constellation [41]. Contrary to the constellations proposed by Luders and Walker, the flower constellation makes use of elliptical orbits in a symmetrical pattern instead of only circular ones.

These standard constellation designs formed the backbone for the development of constellation optimization. Further, the increase of low Earth orbit constellation designs lead to increased attention to discontinuous coverage designs [51]. Discontinuous coverage introduces a number of new metrics for which satellite constellations can be optimized. The two most widely used are the total coverage of Earth and the (maximum) revisit time of the constellation. However, to evaluate satellite constellations based on these figures of merit, a so called coverage analysis has to be performed.

Many different coverage analysis methods have been proposed. Both numerical methods [63] that aim to provide an as accurate as possible analysis, as well as analytical methods [46] [59] that focus more on a fast computation, have been proposed. Additionally semi-analytical methods have been developed [6] [33] that aim to strike a balance between accuracy and computational speed.

To find an optimal constellation, the coverage analysis is combined with a numerical optimization algorithm to perform an optimization of the desired figures of merit. Many studies have been performed on the optimization of satellite constellations using numerical optimization. These often differ in objectives with some optimizing the dilution of precision [12] [54], some the number of satellites [68], and

others the coverage and revisit time [52]. Furthermore, the multiple objectives allow for a variety of different optimization algorithms to be applied to the problem. Single-objective optimization have been performed using algorithms such as Nelder-Mead optimization [12], differential evolution [68], and linear programming [36] among others. Multi-objective optimization is also possible as performed by Savitri [52] using the non-dominated sorting genetic algorithm II. While many different optimizations have been performed, no real comparison has been made to determine which optimization framework performs best, or if there even is a best optimization technique for the constellation optimization use case.

The aim of this thesis is to create a general satellite constellation optimization tool. This tool should be easy to use, run in a timely manner, and provide accurate results. Thus, the general objective of the thesis can be stated as:

Create a general, fast and accurate satellite constellation optimizer.

After the literature study in chapter 2, this objective is expanded using a variety of research questions. These can be found in chapter 3.

The layout of this thesis is as follows. Firstly, in chapter 2 a more in-depth look is taken into the existing literature on coverage analysis methods, optimization algorithms and satellite constellation optimization. This is followed by the identification of the research gap and the statement of the research questions in chapter 3. In Part II the methodology of the research is stated. Here, chapter 4 gives an overview of the development of the novel coverage analysis method and chapter 5 states the implementation of the optimization framework. Next, in Part III, the results of the final tool are stated and discussed. chapter 6 analyzes the coverage analysis method while chapter 7 evaluates the different optimization results. Finally, the conclusions of this thesis are given in chapter 8 and recommendations are given in chapter 9.

2

Literature study

Constellation design has a long history. Many different forms of coverage analyses have been proposed for a variety of use cases. Furthermore, these analyses have since been applied in the field of constellation optimization. First, this chapter discusses the primary objectives for which a coverage analysis is to be performed for. Then, a variety of different existing coverage analysis methods are studied. These methods have been developed for widely differing objectives and scenarios, and have their own strengths and weaknesses. A good coverage analysis is of utmost importance when moving to constellation optimization, as it is the primary method to assign objective values to individuals in the optimization. Thus, after establishing the existing research on coverage analysis methods, a study is performed on the implementation of constellation optimization. Many different methods and optimization algorithms have already been applied on the satellite constellation scenario. Finally, a number of popular optimization algorithms that are to be compared in this thesis are more thoroughly explained.

2.1. Figures of merit

To perform any type of optimization, it is important to correctly identify and define the desired objectives. In satellite constellation design various figures of merit can be used depending on the intended usage of the satellites. For Earth observation missions two main figures of merit exist, namely revisit time [23] and coverage [2] [25]. Revisit time is defined as the time between two observations of the same point by the satellite constellation. Coverage is the fraction of the target area that is observed by the constellation during the analysis time frame. The revisit time and coverage can further be specified by a number of definitions, which are described by Wertz [64]:

- The maximum revisit time, or maximum coverage gap, is defined as the maximum amount of time between any two successive coverage periods of a point or area.
- The average revisit time, or mean coverage gap, is defined as the average length of time between coverage periods of a point or area.
- Percent coverage is defined as the number of times a point is covered by any satellite divided by the analysis period length. The percent coverage thus indicates the fraction of time that a point or area is covered by the constellation.

The revisit time metrics can further be expanded to global figures of merit, i.e. including all analyzed points:

- The maximum gap is the maximum of all maximum coverage gaps of all points.
- The mean maximum gap is the average of the maximum coverage gaps of all points.

2.2. Coverage analysis

With the revisit time and coverage figures of merit defined, a method to compute the objectives needs to be created. For Earth observation missions this is typically done by a coverage analysis. Over the

last fifty years many different methods have been suggested. In the beginning these methods were used to analyze specific types of constellations, and to help design standard constellations.

One of the earliest studies on coverage analysis was performed by Luders [35]. In this study the Streets-of-Coverage method was introduced, which was used to analyze a constellation of multiple satellites in circular polar orbits on different orbital planes. The goal of this constellation was to provide continuous global coverage. This means that every point of Earth should be visible to a satellite in the constellation at every epoch.

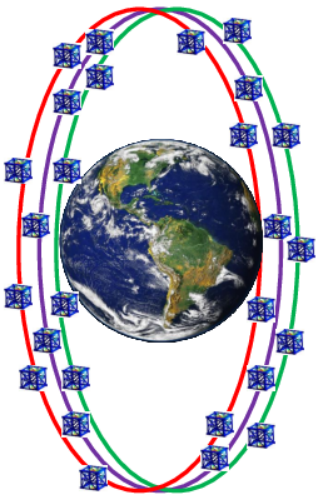


Figure 2.1: A visual example of a polar-inclined streets-of-coverage constellation. [50]

About a decade later, Walker [62] introduced a new constellation design, the Walker constellation. Contrary to Luders' Streets-of-Coverage constellation, the Walker constellation is symmetric. It is defined by three parameters: the number of total satellites, the number of orbital planes and the phasing parameter for satellites in adjacent planes. Like Luders' constellation, the Walker constellation was designed for continuous global coverage.

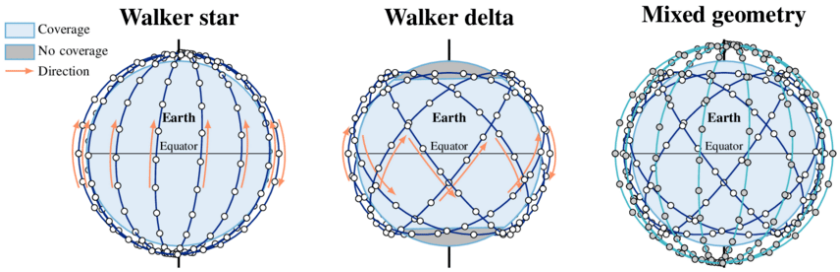


Figure 2.2: A visual example of a Walker star, Walker delta and mixed constellation. [31]

The last of the basic constellation designs is the Flower constellation from Mortari [41]. Also designed for continuous coverage, these constellations are characterized by eleven parameters and generally consist of repeating ground tracks. Like Walker constellations, Flower constellations are symmetric. All satellites in a Flower constellation have the same semi-major axis, eccentricity, inclination and argument of perigee.

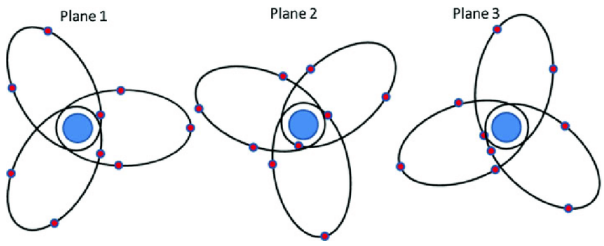


Figure 2.3: A visual example of a 3D-lattice flower constellation. [10]

2.2.1. Discontinuous coverage

In the earliest works on constellation design, only continuous coverage was considered. Due to increasing developments of satellite constellation operating in Low Earth Orbits (LEO), discontinuous coverage has gained more attention. [51] In contrast to continuous coverage, discontinuous coverage occurs when coverage gaps are present in the observation of a certain area. This leads to the discontinuous coverage problem having many differences with the continuous coverage problem which make it much more complex. According to Razoumny [45], there are three main differences between the two problems. Firstly, the rotation of the Earth does not influence the coverage characteristics of the continuous problem. On the contrary, it does influence the discontinuous coverage problem, and it is necessary to take it into account to solve the problem. Secondly, continuous coverage can be calculated on a stationary unit sphere without any errors in the coverage characteristics, whereas for the discontinuous case the specific altitude has to be taken into account when computing the coverage. Finally, the precession of the orbit has to be taken into account as well for the discontinuous case, and not for the continuous case.

Coverage geometry

Every discontinuous coverage analysis starts by defining the geometry of the field of view of one satellite. In Vallado [61] the basic geometry for the field of view of a nadir pointing satellite is given.

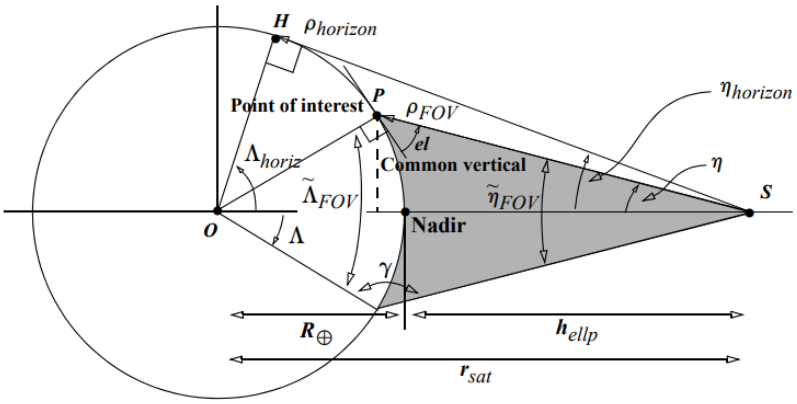


Figure 2.4: The basic geometry for a nadir pointing satellite [61].

Figure 2.4 shows the basic geometry when assuming spherical Earth. There are two main parameters of interest to model the coverage of a single satellite. These are the altitude of the satellite and the field of view of the sensor ($\tilde{\eta}_{FOV}$). These parameters form a circular coverage area when projected on the (spherical) Earth. This geometry forms the basis of many coverage analysis methods.

2.2.2. Numerical methods for discontinuous coverage

Wertz [63] defined two numerical methods. The first makes use of the simple ground track of the mission geometry. This method is most useful for rapid mission analysis. The second method is a point coverage simulation. Here, a grid of points is created on Earth after which one or more satellites are flown over the grid. At the grid points the visibility characteristics are then evaluated.

The point coverage simulation method has been used by many researchers before, as it is a very straightforward method to implement. Furthermore, it can be applied to both continuous and discontinuous coverage problems to analyze a constellation's performance. Morrison [40] is one of the first to have applied this point coverage method. Morrison aimed to find the minimum number of satellites such that from any point on the surface of Earth three satellites are visible with an elevation angle of twenty degrees, and four satellites are visible with an elevation angle of ten degrees, at any time. Using numerical simulation, a sixteen satellite solution was found to meet all the objectives with the fewest amount of satellites. Hongliang [32] used the grid method to evaluate the coverage effectiveness of a remote sensing satellite. Similarly to point coverage, in the grid method an area is divided into many grid points. The constellation coverage performance is then analyzed for each grid point, from which statistical results, such as the average, maximum and minimum value, can be derived. Hongliang compared the obtained results to simulations using the Systems Tool Kit (STK). STK is a commonly used software tool that is also able to analyze satellite and constellation coverage using numerical methods. It is often used as a validation or comparison tool such as in Hongliang or Ulybyshev [58], acting as the baseline method.

The drawback of the numerical methods lies in its computationally intensive, almost brute force nature. Hongliang states that its grid method used a lot of memory and was time intensive, especially when increasing the amount of grid points. According to Crisp [6] numerical methods including STK can take a considerable amount of time to run when analyzing a large number of satellites or when considering many different constellation configurations. This is primarily caused by the larger amount of iterations that numerical methods perform to compute the results. Additionally, for STK specifically, numerous unrelated analyses are performed at the same time as the coverage analysis, further increasing the computational complexity.

2.2.3. Analytical methods for discontinuous coverage

Next to numerical methods, analytical coverage analysis methods have also been developed. In general, analytical methods make a number of assumptions to greatly increase the computational speed at the cost of slightly less accurate results. When using the coverage analysis in constellation optimization, its computation time must be as low as possible to keep the total optimization time manageable. This is due to the fact that the coverage analysis is called for every new solution generated by the optimizer, meaning that the number of calls can easily become more than thousands.

Various analytical methods for discontinuous coverage have been developed, each with their strengths and weaknesses. One of these methods is described by Ulybyshev [59] [60] which makes use of two-dimensional visibility maps. Firstly, a number of assumptions are made: 1) Earth is considered round; 2) All satellites have the same altitude, and each orbital plane has the same number of satellites; 3) All orbital planes have the same inclination. Using the coverage geometry (subsection 2.2.1), a formulation for the coverage angle Λ can be created:

$$\Lambda = cos^{-1}\left(\frac{R_e}{R_e + h_s} cos(el)\right) - el \tag{2.1}$$

Here h is the altitude of the satellite. Considering the simplest case, the coverage of a point by a single satellite, a two-dimensional map can be created, with the right ascension of the ascending node on the x-axis and time on the y-axis. Assuming the inclination, altitude and elevation angle are specified, the map is shown in Figure 2.5.

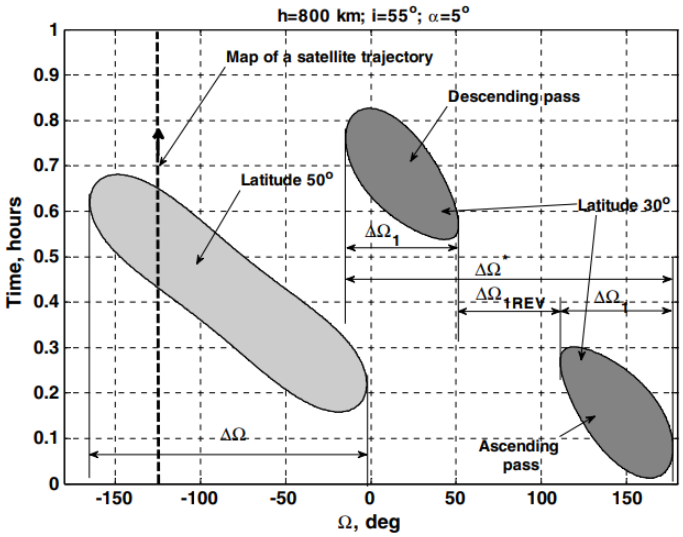


Figure 2.5: Satellite visibility maps [60]

The visibility conditions for the satellite are represented by the oval shapes in the figure. Further, the satellite trajectory can be shown as a vertical line. If the line intersects the visibility region, then there is a visibility interval. The visibility regions can have different shapes depending on the geographical latitude of interest ϕ, the coverage angle Λ and the inclination of the satellite i. Two categories can be defined. For $\phi < i - \Lambda$ there are two distinct ovals for the ascending and descending passes. For $i - \Lambda \leq \phi \leq i + \Lambda$ there is a single connected oval. On the next revolution, the maps should be shifted to the right by $\Delta\lambda_1 = \omega_e T$, where ω_e is the rotation rate of Earth and T the orbital period. It should be noted that for time intervals longer than several orbital periods, the regression of the ascending node should be considered. This would make the straight line slightly inclined to the y-axis.

The parameter of interest from the map is the visibility longitude range $\Delta\Omega$. The computation is done in two steps. First the visibility longitude range for non-rotating Earth ΔL is computed. This can be done analytically using trigonometry (Figure 2.6).

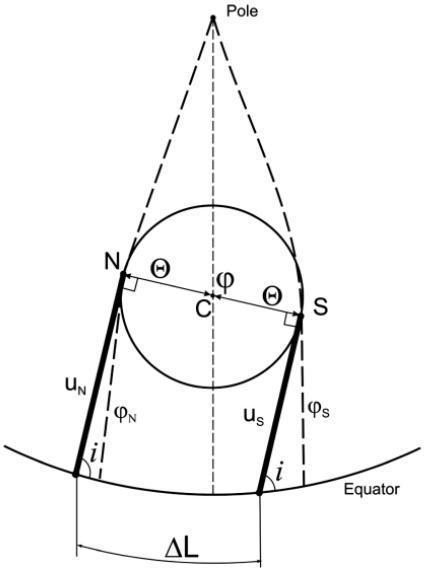


Figure 2.6: Ascending node boundaries. Note: the coverage angle is represented by Θ (in text: Λ) [60]

Then, the Earth's rotation needs to considered, resulting in:

$$\Delta\Omega = \Delta L(\phi, i, \Lambda) - \omega_e T[u_N(\phi, i, \Lambda) - u_S(\phi, i, \Lambda)]/2\pi \tag{2.2}$$

Here u_N and u_S are the argument latitudes of the corresponding boundary points (Figure 2.6). An interesting geometric pattern, a coverage belt, occurs when $\Delta\lambda_1 < \Delta\Omega$. This means that there are longitude ranges with visibility intervals at consecutive revolutions. Figure 2.7 shows both the cases when $\Delta\lambda_1$ is smaller (a)) and bigger (b)) then $\Delta\Omega$. In a) the coverage belts are formed.

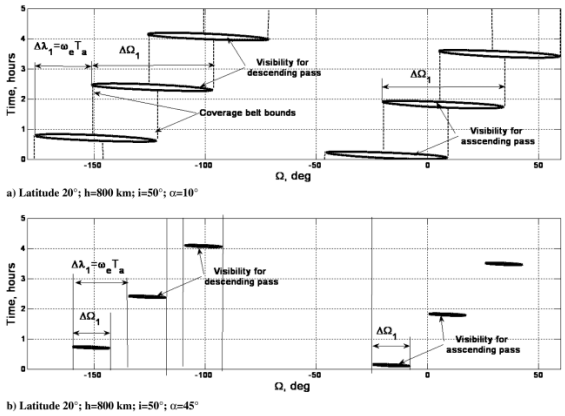


Figure 2.7: Multirevolution maps for visibility of a single satellite.

Based on if coverage belts can be formed or not, the revisit time can be estimated. If coverage belts can be formed, the revisit time can be estimated with an error no more than one orbital period:

$$\Delta t^*_{REV} \approx \lfloor \frac{2\pi}{2\Delta\Omega_1 - \Delta\lambda_1} \rfloor T \tag{2.3}$$

If coverage belts cannot be formed, only a lower bound for the maximum revisit time can be given:

$$\Delta t_{REV} \geq \Delta t^{LOW}_{REV} = \lfloor \frac{2\pi}{\Delta\Omega_1} \rfloor T \tag{2.4}$$

If there is visibility in ascending and descending passes (figure b)) then the double value of $\Delta\Omega_1$ should be used.

Extending the time interval to include many revolutions results in Figure 2.8. It can be seen that the coverage belts overlap such that the whole range of Ω is periodically covered. An algorithm can be applied to combine these coverage intervals to show where gaps appear.

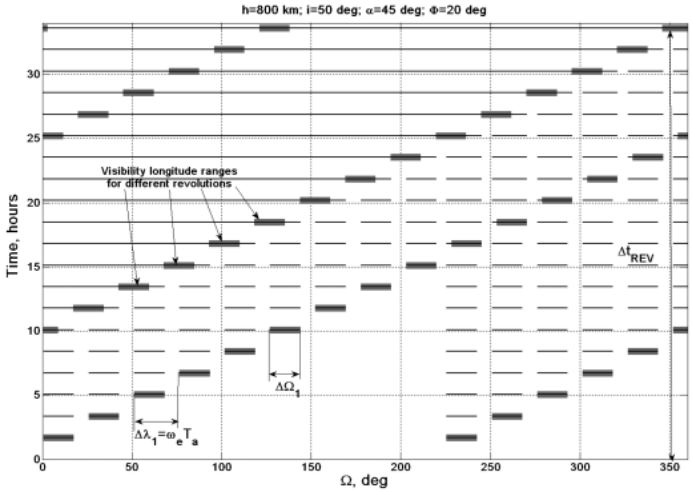


Figure 2.8: Extended multirevolution map [60]

Ulybyshev also proposes a simple algorithm to analyze homogeneous satellite constellations, namely Walker constellations. This algorithm can be used to make an optimal choice for the altitude and inclination to minimize the maximum revisit time. Finally, a note is also given on how to apply the algorithm to non-homogeneous constellations (differing altitudes and inclinations of satellites).

Contrary to Ulybyshev's ground-to-space method, Graziano [14] created a space-to-ground method using the Streets-of-Coverage method as a starting point. Three methods were considered, leading to both symmetric and asymmetric satellite patterns. The second method was found to perform the best. This method relies on horizontal streets of coverage generated by 2Λ-separated orbital planes. The minimum number of satellites guarantees a time separation of $T/S < \tau$. Here S is the number of satellites, T the orbital period and τ the required (maximum) revisit time.

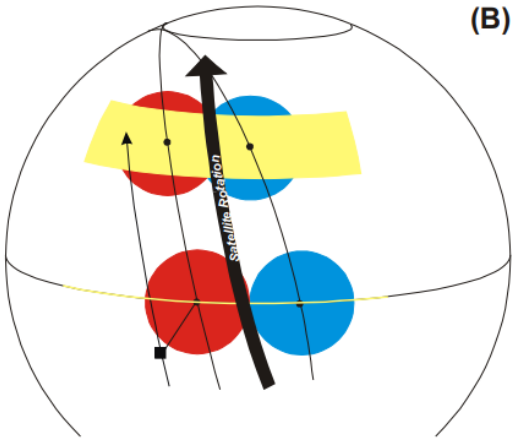


Figure 2.9: Principle of constellation design method [14]

The satellites are placed in different orbital planes with the same phase angle and move from pole to pole. Each of these "lines" of satellites create a horizontal street of coverage that cover half the globe. These lines sweep Earth, first ascending and then descending. The number of satellites per orbital plane is defined by the orbital period and the required revisit time:

$$S = \lceil \frac{T}{\tau} \rceil \tag{2.5}$$

Due to the rotation of the Earth, it could occur that points are only observed in the first ascending sweep, but not by the descending and following ascending sweep, due to lying close to the boundary. This condition must be prevented in order to comply to the revisit time requirement. This is done by considering the condition in the selection of the number of orbital planes. The number of orbital planes now considers three conditions (Figure 2.10): 1) Guarantee requested re-observation in the same phase only for the whole globe; 2) Guarantee for descending/ascending; 3) Guarantee for ascending/descending. It should be noted that the worst case is expected for ascending/descending at $\phi > 0$ and descending/ascending at $\phi < 0$.

$$P^B = max \begin{cases} \left\lceil \max_\varphi \left(\frac{\pi + 2\sin^{-1}\left(-\frac{\tan\varphi}{\tan i}\right) - 2\cos^{-1}\left(1 - \frac{1-\cos\theta}{\cos^2\varphi}\right)}{2\theta} + 1 \right) \right\rceil \\ \left\lceil \max_\varphi \left(\frac{\pi - 2\sin^{-1}\left(-\frac{\tan\varphi}{\tan i}\right) - 2\cos^{-1}\left(1 - \frac{1-\cos\theta}{\cos^2\varphi}\right) + \omega_\oplus \left\{ \frac{T}{S} - \left[\pi + 2\sin^{-1}\left(\frac{\sin\varphi}{\sin i}\right) - \left\lfloor \frac{\pi + 2\sin^{-1}\left(\frac{\sin\varphi}{\sin i}\right)}{2\pi/S} \right\rfloor \frac{2\pi}{s} \right] \frac{1}{\omega} \right\}}{2\theta} + 1 \right) \right\rceil \\ \left\lceil \max_\varphi \left(\frac{\pi + 2\sin^{-1}\left(-\frac{\tan\varphi}{\tan i}\right) - 2\cos^{-1}\left(1 - \frac{1-\cos\theta}{\cos^2\varphi}\right) + \omega_\oplus \left\{ \frac{T}{S} - \left[\pi - 2\sin^{-1}\left(\frac{\sin\varphi}{\sin i}\right) - \left\lfloor \frac{\pi - 2\sin^{-1}\left(\frac{\sin\varphi}{\sin i}\right)}{2\pi/S} \right\rfloor \frac{2\pi}{s} \right] \frac{1}{\omega} \right\}}{2\theta} + 1 \right) \right\rceil \end{cases}$$

Figure 2.10: Selection criteria for number of orbital planes [14]

The main drawback of this method is that it does not find the optimal constellation design to minimize the number of satellites. As it is based on the streets-of-coverage method, it suffers the same drawback that the coverage edges outside of the streets are not considered. Furthermore, for higher latitudes the satellites coverage regions suffer from a lot of overlap.

Sarno [51] takes the methods of Ulybyshev and Graziano and takes the positives of both methods to create a new method. This new method exploits the coverage regions concept of Ulybyshev and the asymmetric distribution of orbital planes around the equation from Graziano.

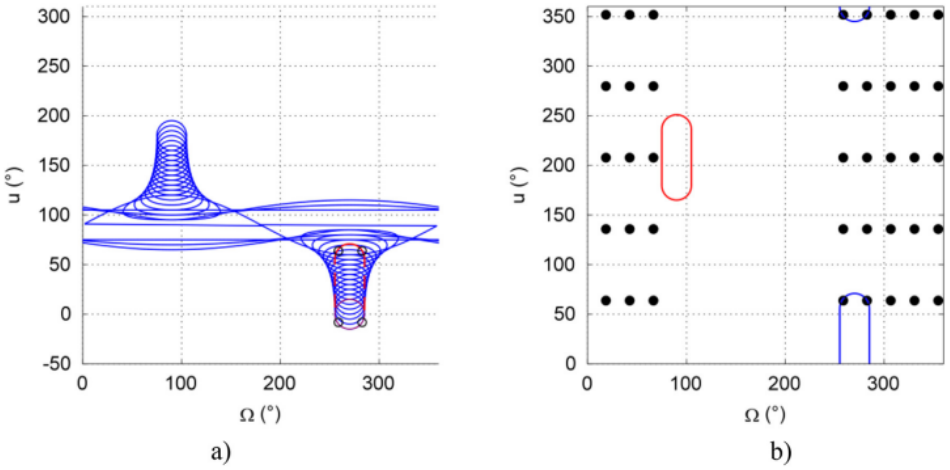


Figure 2.11: h = 600 km, i = 90°, t_{rev} = 15 min, T = 96.6 min, $\Delta\Omega$ = 24°, N/P = 40/8. a) shows coverage regions for latitude 0° to 90° with step 5°. b) represents the constellation pattern. [51]

Figure 2.11 shows a coverage map with various target latitudes. Note that the y-axis now represents the argument of latitude (u) instead of time, as in Ulybyshev. The red ascending lobe corresponding to the minimum latitude is selected and stretched along u to meet the temporal requirement. This is used to define the separation between satellites in the same orbital plane, and between adjacent orbital planes. Then, using the criteria from Graziano (Figure 2.10), the number of orbital planes is defined. This results in the constellation pattern shown in b) of Figure 2.11. Comparing this method with that of Graziano, Sarno shows that both the number of satellites per plane as well as the total number of

satellites is lower, especially for larger swath width. However, this method still does not guarantee the most optimal constellation.

Li [33] proposes another method which makes use of revisit orbits. The research is based on observation satellites with a very narrow swath, thus the constraint is added that a point on Earth is only revisited when it is on the nadir of the satellite. An orbit is defined as a revisit orbit if the satellite observes the target point at the ascending stage and at the descending stage alternately in a single cycle (Figure 2.12). This occurs when a satellite runs at a regressive orbit.

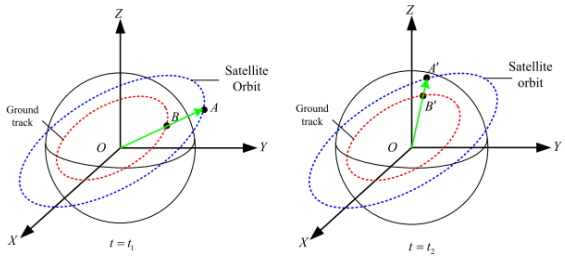


Figure 2.12: A revisit orbit [33]

Using spherical trigonometry, the mathematical model of the revisit orbit can be defined. It should be noted that revisit orbits are typically circular, thus only three orbital parameters need to be determined, namely the semi-major axis, the inclination and the right ascension of the ascending node. The mathematical relations between these parameters are stated both for round Earth as well as for including J2 perturbations. Finally, a constraint function is given that, when solved, provides the circular revisit orbit. Unfortunately, the constraint function can only be solved numerically.

The main benefit of the optimal circular revisit orbit for coverage analysis is the fact that the revisit time is smaller than that of traditional repeat ground track orbits. However, this method is not fully analytical. Furthermore, no mention is made of its application for satellite constellations.

Recognizing the shortcomings of the previous papers, Razoumny [45] proposes a method that is free from them. This method applies the route theory for satellite constellation design for discontinuous coverage. It considers (near-)circular orbits and J2 perturbations, as well as spherical Earth. The only constraint specified is that the satellites fly in repeating ground track orbits.

As every coverage analysis, Razoumny starts by defining the geometry of the coverage. It is assumed that the (spherical) Earth with radius R_m is covered by the swath of the satellite moving on a circular, repeating ground track orbit. This orbit is defined by a radius r_m, inclination i and a repetition factor m/n, where:

$$T_{tr} = mT_{nod} = nT_{ef} \tag{2.6}$$

Here, T_{tr} is the tracks repetition period, T_{nod} is the nodal period of the satellites orbit and T_{ef} is the efficient period of Earth rotation.

The latitude of interest is specified as the parallels R_ϕ in the latitude belt R. R is bounded by parallels R_ϕ on the latitudes ϕ_{min} and ϕ_{max}. A constraint is put on the inclination to make total coverage of the latitude belt possible:

$$i_0 + \theta > \phi_m, i_0 = min(i, \pi - i), \phi_m = max(|\phi_{min}|, |\phi_{max}|) \tag{2.7}$$

Here, θ is the geocentric angular satellite swath width on non-rotating Earth. Two cases can be considered: if $|\phi| < i_0 - \theta$ the latitude is specified as a lower latitude. For lower latitudes, satellites cross the parallel twice in a revolution; on the ascending latitudinal points A_j and the descending latitudinal nodes D_j. This means that in one track repetition period there are m points A_j and m points D_j. Each point

system is uniform, where the distance between neighboring points is equal to the inter-nodal distance ΔL:

$$\Delta L = 2\pi/m \tag{2.8}$$

The points A and D are characterized by angular distances. δ_R and δ_L (see Figure 2.13a)). Combining this right and left shift results in the inter-nodal distance. To determine the value of the shifts, the sweep angle of the points A and D $\Delta\psi$ is introduced. This designates the difference between the longitudes of A_k and D_k.

$$\Delta\psi = \psi_{BM} - \psi_{MN} \tag{2.9}$$

Here, ψ_{BM} is the longitude difference for the points of intersection between the orbit and the R_ϕ parallel on non-rotating Earth, and ψ_{MN} is the angle by which the Earth rotates in the time the satellite flies from B to M (see Figure 2.13b)).

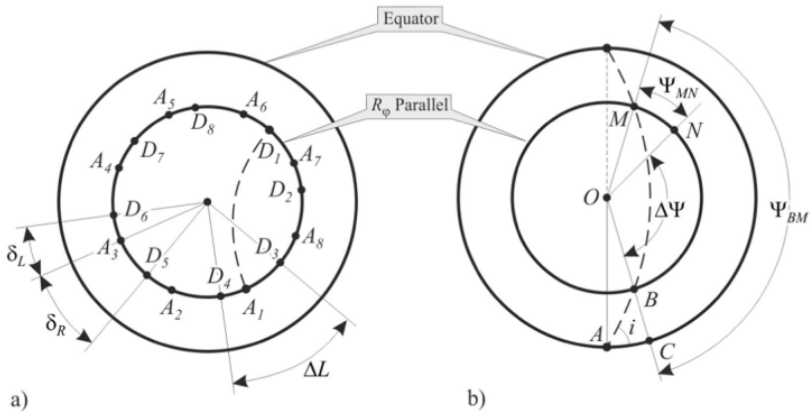


Figure 2.13: Arrangement of points A and D [45]

Using the geometry shown in Figure 2.13b) the sweep angle can be computed using trigonometry. From this, the right shift (and thus also the left shift) can now be computed:

$$\delta_R = \Delta\psi - \Delta L \lfloor \frac{\Delta\psi}{\Delta L} \rfloor \tag{2.10}$$

The second case to be considered is when $i_0 - \theta \leq |\phi| < i_0 + \theta$. These latitudes are called upper latitudes and are defined by only one crossing, the vertex point, instead of two. For one repetition track period, a satellite covering an upper latitude will have m vertex points V_j, each with a distance to neighboring nodes equal to the inter-nodal distance ΔL.

Next, the capture area, or instantaneous coverage area, of the satellite is to be defined. First, the lower latitude case is considered. The capture α is defined by a left and right bound, α_L and α_R, defined by points A and B of the swath on the R_ϕ parallel (see Figure 2.14). Point Q is the (ascending) latitudinal node.

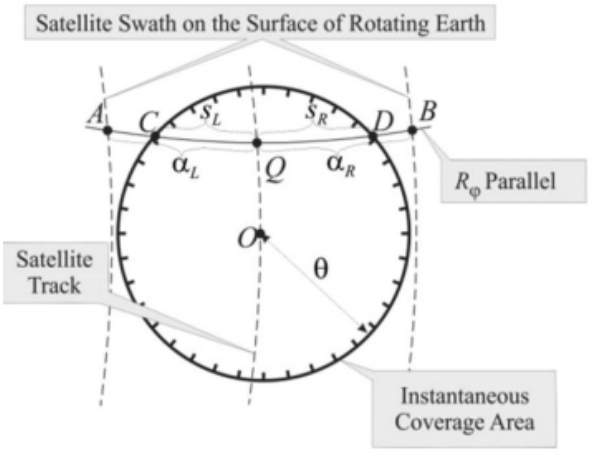


Figure 2.14: Swath capture for lower latitudes [45]

Assume that s_L and s_R are the distances from Q to the borders of the swath capture. Further, assume that u_I, u_F and u_ϕ are the arguments of latitude of the initial and final contacts of the coverage area and the parallel, and the moment of the satellite crossing the parallel. Then:

$$\alpha_L = \max_{u\in[u_I,u_F]} s_L(u), \alpha_R = \max_{u\in[u_\phi,u_F]} s_R(u) \tag{2.11}$$

where

$$u_I = sin^{-1}\frac{sin(\phi-\theta)}{sin(i)}, u_\phi - sin^{-1}\frac{sin(\phi)}{sin(i)}, u_F = sin^{-1}\frac{sin(\phi+\theta)}{sin(i)} \tag{2.12}$$

The equations for s_L and s_R, can be found using trigonometry. This can then be used to solve the equations for the capture bounds α_L and α_R.

A similar calculation is done for the upper latitude case. Here, it is noted that on a rotating Earth, the left-side capture is equal to half a full capture (see Figure 2.15):

$$\alpha = 2\alpha_L, \alpha_L = \max_{u\in[u_I,\pi/2]} s_L(u) \tag{2.13}$$

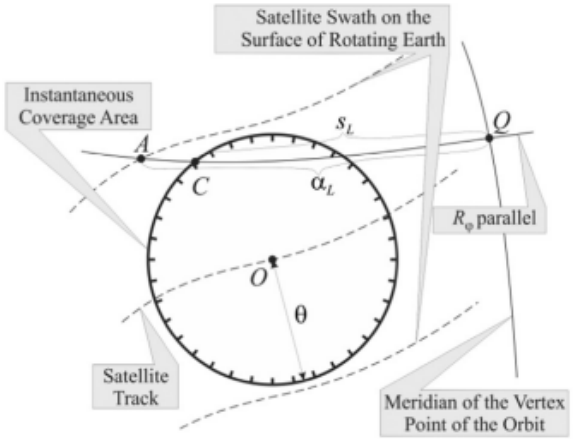


Figure 2.15: Swath capture for upper latitudes [45]

Again, the equation for s_L is found using trigonometry, allowing the computation of the capture bounds.

To analyze the coverage of a satellite, Razoumny considers the observability of an arbitrary point r that lies on the parallel R_ϕ. This point r is considered observed at moment t, if t is a moment of the satellite passing its latitudinal node. The observation time stream is defined as a monotonically increasing array of all moments in time when a point r is covered by a capture of a longitudinal point: $t_1, t_2, ..., t_j, t_{j+1}, ...$ The revisit times for a point r and the whole parallel are then:

$$\tau[r] = \max_j(t_{j+1} - t_j), \tau[R_\phi] = \max_\tau[r], r \in R_\phi \tag{2.14}$$

This means that the observation time stream depends on the swath width, as this is what defines the swath captures on the parallel. To compute the revisit time for r, knowing only the number of latitudinal nodes l that capture r is not enough. It is also necessary to know which specific nodes capture r. The manifolds of such nodes for various points r are defined as observation variants of these points and are associated with specific satellites. Further, an invariance sector of the observation time stream is defined as a continuous interval, with the property that for a given satellite or constellation, the same observation variant corresponds to all points r of this arc. This means that all the points of a single invariance sector are observed by the satellite with the same revisit time.

Razoumny finds that six different invariance sectors are possible: four variants for lower latitudes and two for upper latitudes. Each of these invariance sectors provide an analytical formulation of the revisit time. Expanding this analysis to a N-satellite constellation can be done by "fusing" single-satellite observation variants. It can be concluded that it has at most $4N$ different coverage variants from which the desired revisit time metrics can be determined. Further, to minimize the revisit time, the change in RAAN and argument of latitude for each satellite in the constellation can be computed analytically.

The work of Razoumny is very promising in the development of analytical coverage analysis methods. The main drawback is the fact that the Route theory method is only applicable for repeating ground track satellites. It should also be noted that the papers of Razoumny are mostly theoretical and that the method has not been applied or validated much.

Another (semi)analytical method was explored by Crisp [6]. Similarly to Razoumny, Crisp states that the calculation of the revisit time at a given latitude can be determined by correlating the longitude of all projected passes and the instantaneous coverage of the sensor over a period of analysis. In this method, oblate Earth and J2 perturbations are assumed.

As per usual, Crisp first looks at the sensor geometry to find the half-ground angle. This is followed by the calculation of the longitude of successive passes. The drift in longitude of successive passes can be calculated considering the rate of Earth rotation ω_E, the rate of nodal regression $\delta\Omega$, and the nodal period P_n:

$$\Delta\lambda = P_n(-\omega_E + \delta\Omega) \tag{2.15}$$

The longitude where the orbit ground track crosses the target latitude can be calculated using the true anomaly. As the satellite will cross the target latitude both ascending and descending, a pair of results will exist for both the true anomaly and longitude representing the location of these passes. The longitude of successive passes at the target latitude (up to a specified time) can be expressed as an arithmetic series in which the difference between consecutive terms is $\Delta\lambda$:

$$\{\lambda_j\} = \lambda_1 + (j-1)\Delta\lambda \;\; for \;\; j = 1, 2, 3...||\frac{T_{lim}}{P_n}|| \tag{2.16}$$

To consider the passes for a constellation, one can use the standard Walker notation of $i : t/p/f$, where i is inclination, t the total number of satellites, p the number of equally spaced planes and f the relative spacing between satellites in adjacent planes. From this the number of satellites per plane s can be determined. To determine all passes of the constellation, the longitude of passes of satellites in multiple planes $\{\lambda_p\}$ can be defined:

$$\{\lambda_p\} = \{\lambda_j\} + 2\pi m\left(\frac{1}{p} + \frac{f}{t}\right) \;\; for \;\; m = 1, 2, ..., (p-1) \tag{2.17}$$

Similarly, the longitude of corresponding passes of multiple satellites in each plane $\{\lambda_s\}$ can be defined:

$$\{\lambda_s\} = \{\lambda_j\} + \frac{l}{s}\Delta\lambda \quad for \quad l = 1, 2..., (s-1) \tag{2.18}$$

The total set of passes in an analysis period is thus:

$$\lambda_\phi = \lambda_j \cup \lambda_s \cup \lambda_p \tag{2.19}$$

For non-symmetric constellations, each plane can be considered individually with the possibility of variation in RAAN of each plane and in-plane spacing of the satellites.

To assess the revisit performance of the constellation, a discretised grid of longitudes $\{\Pi\}$ about the target latitude is defined. First, it should be indicated which longitudes in Π are visible by a pass of longitude λ_ϕ due to the angular range of the sensor Λ:

$$\lambda_\phi - \{\Pi\} \leq \Lambda \tag{2.20}$$

For the projection of the sensor footprint on the Earth surface, the rotation of the Earth and the angle between the orbit track and the target latitude should be considered. Crisp states that for target latitudes lower than 75 degrees the geometry of a simple ellipse can be used.

$$\frac{(\{\Pi\} - \lambda_\nu)^2}{\Lambda^2} + \frac{(\phi - \{\phi_\nu\})^2}{\theta^2} \tag{2.21}$$

Here the coordinates of the origin of the ellipse (λ_ν, ϕ_ν) are defined by the latitudinal and longitudinal coordinates of the ground track of the satellite over a range above and below the target latitude. By evaluation of the equations for the longitude of each pass at each latitude of interest, a list of accesses for each longitude on $\{\Pi\}$ is established. The maximum gap in time between two consecutive accesses represents the maximum revisit time.

Crisp validated the method using STK. Crisp's method seems to be an improvement of that of Razoumny in that it is able to consider satellites that do not have a repeating ground track. However, the validation was only performed on symmetric satellite constellations. Further, similarly to Razoumny, only circular orbits were considered.

He [18] [19] aimed to improve the field-mapping method to apply it to circular repeating ground track orbits. This means the orbits can be defined by their inclination, orbital radius and repetition factor m/n (see Equation 2.6). He also considers J2 perturbations.

The first improvement He made to the field-mapping method was to fix the coverage region on the map. This in turn changes the movement of the satellite to be several inclined lines on the map instead of vertical ones. Furthermore, the x-axis now becomes the longitude of ascending node, instead of the RAAN (see Figure 2.16).

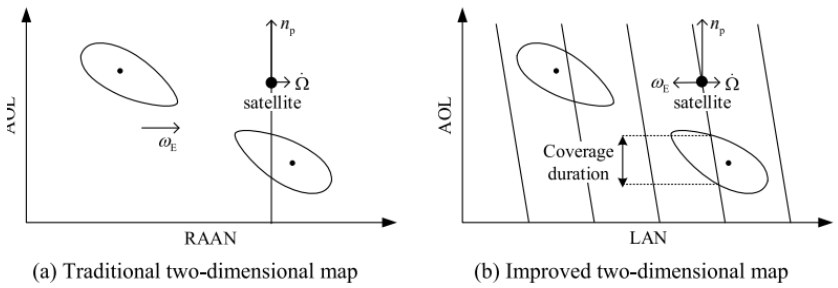


Figure 2.16: Field-mapping improvement [19]

For a repeating ground track orbit, the number of inclined lines is exactly m, i.e. the number of revolutions in one repetition period. The slope can be expressed as:

$$k_{sat} = -\frac{n_p}{\omega_E - \dot{\Omega}} = -\frac{m}{n} \tag{2.22}$$

Here, $\dot{\Omega}$ is the RAAN change rate. Furthermore, the distance between adjacent tracks is equal to the inter-nodal distance ΔL (see Equation 2.8).

To perform the coverage analysis, He takes inspiration from Razoumny. He applies the Route theory method and refines it using field-mapping techniques. When analyzing the coverage of the parallel (i.e. target latitude) by a single satellite, He uses the coverage variants and invariance sectors as defined by Razoumny. However, while the principles are the same, the values are slightly altered due to the improvements using field mapping. When considering a satellite constellation, the single-satellite coverage analyses can be combined using slight shifts to create the optimal coverage.

Contrary to Razoumny, He did apply the method and validated it using STK. It should be noted that only Walker constellations were used to obtain the results. The only drawbacks of this method are the constraints to circular and repeating ground track orbits.

2.2.4. Summary on coverage analysis methods

In this section, various coverage analysis methods have been described and studied. To evaluate what methods could be relevant for this thesis, a couple of distinctions can be made to group similar methods. Firstly, the coverage analysis methods can be split into methods to achieve continuous coverage and methods to achieve discontinuous coverage. Here continuous coverage relates to constellation designs that can observe the whole area of interest for every epoch. Examples of continuous coverage constellations are the Streets-of-Coverage constellation [35], the Walker constellation [62] and the Flower constellation [41].

Discontinuous coverage occurs when the area of interest is not continuously observed by the constellation. These types of constellations are characterized by a revisit time, i.e. the time between two observations of the same point or area on Earth. For Earth observation missions, continuous coverage is often not necessary, instead preferring discontinuous coverage.

Within the coverage analysis methods for discontinuous coverage, another distinction can be made, namely between numerical and analytical methods. In general, numerical methods provide more accurate results at the cost of an increase in computation time. Analytical methods operate oppositely, slightly worsening results with assumptions to achieve a better computation time. The strengths and weaknesses of the various methods discussed in this section have been summarized in Table 2.1. This table also shows the expected suitability of the method within the scope of this thesis, namely constellation optimization.

Method	Analytical or numerical	Strengths & weaknesses	Suitability
Point coverage simulation [63]	Numerical	+ Accurate + Easy to determine revisit time - Slow - Requires many parameters - Can lead to misleading results and conclusions	Too slow to be used in optimization
STK 12	Numerical	+ Accurate - Slow	Too slow to be used in optimization
2D visibility maps [59] [60]	Analytical	+ Fast - Restricted use case - Must be repeated for every point - Only estimates revisit time	Too restricted use case to be relevant
Graziano's SoC [14]	Analytical	+ Fast + Not limited to one point + Asymmetrical constellation - Restricted to unoptimal SoC constellations	Too restricted use case to be relevant
Sarno's method [51]	Analytical	+ Fast + Not limited to one point + Asymmetrical constellation - Not optimal constellation	No mention of revisit time computation
Li's revisit orbits [33]	Semi-analytical	+ Somewhat fast + Easy revisit time computation - Restricted to RGT orbits	Not used on constellations
Route theory [46]	Analytical	+ Fast + Accurate revisit time computation - Restricted to RGT orbits - Must be repeated for every point - Not validated	Promising method, but too constraint to RGT orbits
Crisp's method [6]	Semi-analytical	+ Fast + Applicable to all circular orbits + Easy revisit time computation - Must be repeated for each latitude bound - Only validated for symmetrical constellations	Can be applied if it can be modified to include eccentricity and if global analysis can be made fast enough
He's field mapping [18][19]	Analytical	+ Fast Accurate revisit time computation + Validated - Restricted to RGT orbits - Must be repeated for every point	Promising method, but too constraint to RGT orbits

Table 2.1: Summary of discontinuous coverage analysis methods

2.3. Constellation optimization

With the coverage analysis established as the way to evaluate constellations based on the figures of merit, the framework of the complete optimizer can now be envisioned. The biggest factor of the optimizer is the type of optimization algorithm that is to be used. Many different algorithms can be used for the optimization of satellite constellations, and various studies have been performed implementing numerous algorithms. In this section, first different popular algorithms are explored. This is then followed by the analysis of various studies on constellation optimization.

2.3.1. Optimization algorithms

Considering the figures of merit stated before in section 2.1, the optimization problem of constellation design can be considered a multi-objective problem. One could consider the minimization of the maximum revisit time, the maximization of the coverage, or even the minimization of constellation parameters such as the total number of satellites. Many different algorithms exist, each with their own strengths and weaknesses. It is therefore critical that the right choice of algorithm is made.

Gradient descent algorithms

Gradient descent optimization is one of the oldest types of optimization algorithms. It was first suggested by Cauchy [4] in 1847, to compute the orbit of heavenly bodies such as stars. The algorithm was first applied to non-linear optimization by Curry [7], and gained immense popularity among deep learning applications during the rest of the 20th century. The stochastic gradient descent method [49] in particular has been applied extensively to deep networks.

Stochastic gradient descent works as follows. The cost function of the optimization is in the form:

$$Q_w = \frac{1}{n}\sum_{i=1}^{n} \nabla Q_i(w) \tag{2.23}$$

To start the optimization, an initial guess is made for the parameter w. Then, a random point from the training set is selected, at which the cost is computed using the cost function with the estimated parameter w. The difference between the computed cost value and the actual value can then be used to determine the gradient of the cost function at this point. Using this gradient function, parameter w is updated via:

$$w := w - \eta \nabla Q_i(w) \tag{2.24}$$

Here, η is called the learning rate of the algorithm. By iterating these steps many times, the parameter w is optimized to eventually find the correct function to predict solutions.

A number of issues can be identified when trying to use gradient descent methods on satellite constellation optimization. First and foremost, the cost function of the constellation optimization is often not differentiable, especially when considering the revisit time. This means that gradient descent methods can often not be used as they require the cost function to be differentiable. Furthermore, as stochastic gradient descent is trained on only one training set point at a time, it is considerably slow, while other gradient descent methods that train on larger sets are more sensitive to bias and can get more easily stuck in local optima.

Genetic Algorithm (GA)

The genetic algorithm was extensively described by Holland [20]. The algorithm is based on the evolutionary process of natural selection found in biology. In genetics, the set of observable traits of an individual is called the phenotype. This phenotype can be split into traits related to the individual's genetic code (genotype) and the impact of the environment on the individual. More specifically, the environment can influence how well an individual with a certain genotype can thrive. This is based on the opportunities and threats that the environment imposes onto the individual. Mathematical studies of genetics allow for the summary of the performance of an individual's genotype within an environment by a single performance measure, the fitness. The fitness roughly translates to the amount of offspring of the individual that survives the environment to generate offspring of their own. More generally, the fitness relates to the influence of the individual upon the next generation. When taking a

sample of different genotypes, the fitness can be used to find the best one. In order to create even better genotypes, individuals are subject to various genetic operators. Most importantly are crossover and mutation. Crossover is the practice of combining two genotypes into a new one. In biology this occurs via the mating of two individuals. The resulting offspring has a combined genotype of its parents. The second operator mutation, allows for additional variety within the sample (or population) of individuals. Randomly, parts of the genotype of the offspring can be changed, i.e. mutated.

This process of evolution can be applied to optimization. Individuals can be evaluated using their fitness values, which can then be optimized using the processes of crossover and mutation. One other operator should be introduced, namely selection. Before the crossover, the parents that are to generate the offspring need to be selected. A number of variants exist for the different genetic operators. These differ mainly between preserving the most fit individuals, but narrowing the diversity of individuals, or preserving diversity, but possibly discarding optimal solutions. Figure 2.17 gives an overview of the framework of a genetic algorithm.

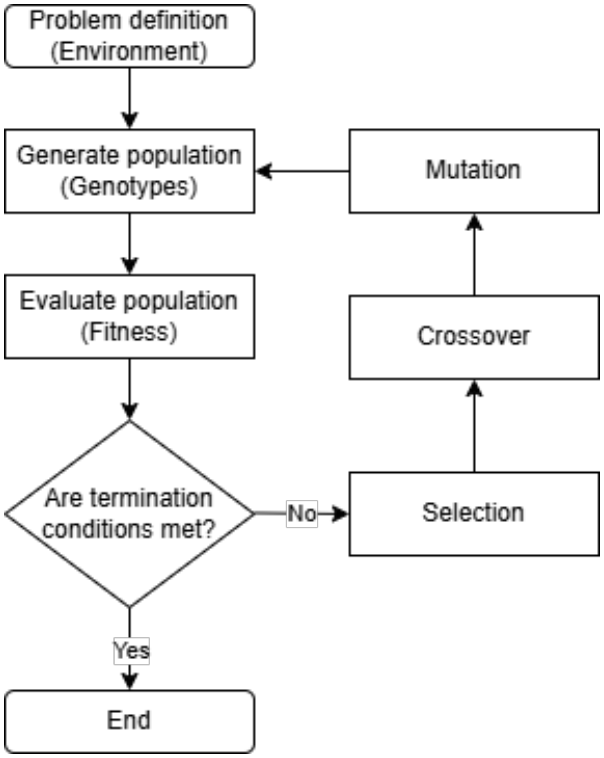


Figure 2.17: Overview of the genetic algorithm framework.

Particle Swarm Optimization (PSO)

Particle swarm optimization was introduced by Kennedy and Eberhart [22]. The algorithm is related to artificial life (A-life), bird flocking, fish schooling and swarming theory. Further, it has ties to genetic algorithms and evolutionary programming. Particle swarm optimization aims to simulate the social behavior of a swarm.

Every iteration, a particle considers two parameters, namely its personal best position and the global best position of the swarm. In optimization, this position can be viewed as the fitness of a combination of decision variables. Based on the current position of the particle, its personal best, and the global best, it is assigned a velocity (Equation 2.25).

$$vx = vx + 2rand()(pbestx - presentx) + 2(pbest[gbest] - presentx) \tag{2.25}$$

In this equation the cognitive and social parameters can clearly be seen. Both parameters include a bias, which is set to two in Equation 2.25. These biases influence how much a particle considers its

personal best versus the global best of the swarm. Further an inertia parameter can be introduced to control the weight of the particle's previous velocity when computing its new velocity. Using the velocity, the position of the particle is updated each iteration. If the position is better than its personal best, it gets updated. Further, if the swarm improves its global best, it is also updated. Changing the weights in the velocity equation can change both the convergence speed, as well as the exploration of the algorithm. Figure 2.18 provides an overview of particle swarm optimization.

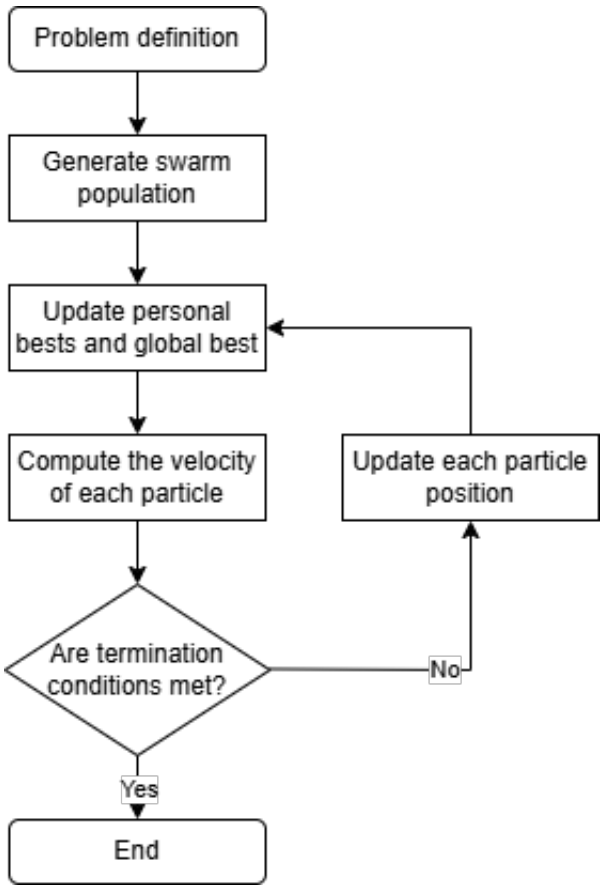


Figure 2.18: Overview of the particle swarm optimization framework.

Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

Covariance matrix adaptation evolution strategy was proposed by Hansen [16]. It was created to specifically handle non-linear optimization problems with non-convex or rugged search landscapes better. It further overcomes a number of problems typical to evolutionary algorithms. Firstly, it can perform on badly scaled and non-separable problems. Secondly, it has no inherent need for large population sizes. In fact, smaller population sizes lead to faster convergence. Thirdly, the CMA-ES algorithm is able to prevent premature convergence.

The working of the CMA-ES algorithm is described in detail by Hansen [15]. First, the parameters of the algorithm are set and the problem is initialized. Next, similarly to other evolutionary algorithms, selection and recombination is performed. During selection, the parents are greedily sampled from the population. Notably, the parents are used to compute the weighted average of the sample. This average is used to update the covariance matrix of the sample. The update of the covariance matrix consists of two parts, namely the rank-μ update and the rank-one update. In the rank-μ update, the information from the entire population is used. Additionally, the rank-one update uses the information of correlations between generations. By modifying the learning rates (i.e. weights), the update can be adapted to different populations. For large populations the rank-μ update is more important, while for small populations the rank-one update is more important.

Another aspect of the CMA-ES algorithm is its adaptable step-size. To control the step-size, the algorithm uses the evolution path, which is the sum of (previous) successive steps. Three cases can be identified based on the evolution path and the expected length of the step. Whenever the evolution path is short, i.e. shorter than the expected length, the single steps in the evolution path cancel each other out. This indicates that the step-size should be decreased. When the evolution path is longer than the expected length, the single steps in the path point in similar directions. This means that a longer step-size can be used. Finally, in the ideal situation the length of the evolution path is equal to the expected length (Figure 2.19).

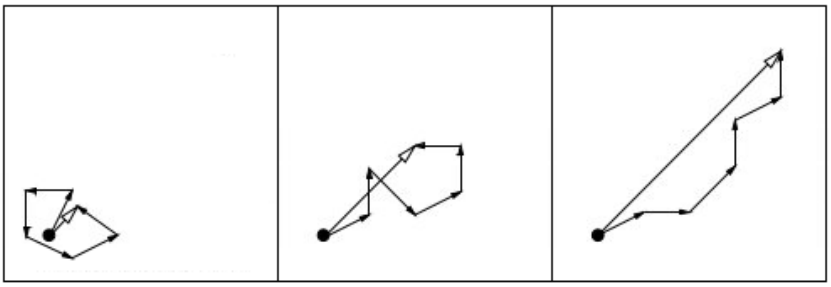


Figure 2.19: Three different evolution paths: left a short path, right a long path and middle the ideal situation.

The CMA-ES algorithm optimizes the average of the population, allowing it to converge to better solutions. The framework is described in Figure 2.20.

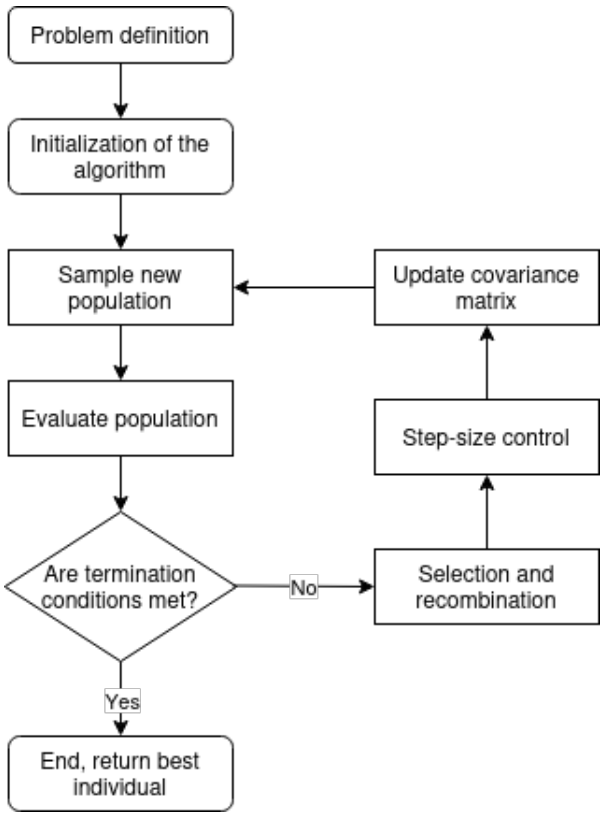


Figure 2.20: Overview of the covariance matrix adaptation evolution strategy framework.

Differential Evolution (DE)

Storn [56] introduced differential evolution. According to Storn, users of optimization algorithms generally demand that minimization techniques fulfill four requirements. Firstly, the algorithms should be

able to handle non-differentiable, nonlinear and multimodal cost functions. Secondly, the algorithm should allow for parallelization to handle computation intensive cost functions. Thirdly, there should be few control variables to control the minimization. These variables should also be robust and easy to choose. Finally, the algorithm should be able to sho good convergence, and be able to find the global minimum consistently. Differential evolution was designed to comply to all four of these requirements.

The differential evolution algorithm is initialized by the creation of the initial population of vectors. For each vector, the decision variables are chosen at random from a uniform distribution. The first step of the algorithm is called mutation. First, two random population vectors are chosen. The weighted difference between these vectors creates a difference vector. Adding this difference vector to another third population vector creates the mutated vector. The next step in the algorithm is crossover. Here, the mutated vector's parameters are mixed with those of another predetermined population vector, called the target vector, which results in the trial vector. The final step is selection, which is performed greedily. This means that if the fitness value of the trial vector is better than that of the target vector, the target vector is replaced by the trial vector. If the fitness of the trial vector is not better, it is discarded and the target vector remains in the population. This framework is shown in Figure 2.21.

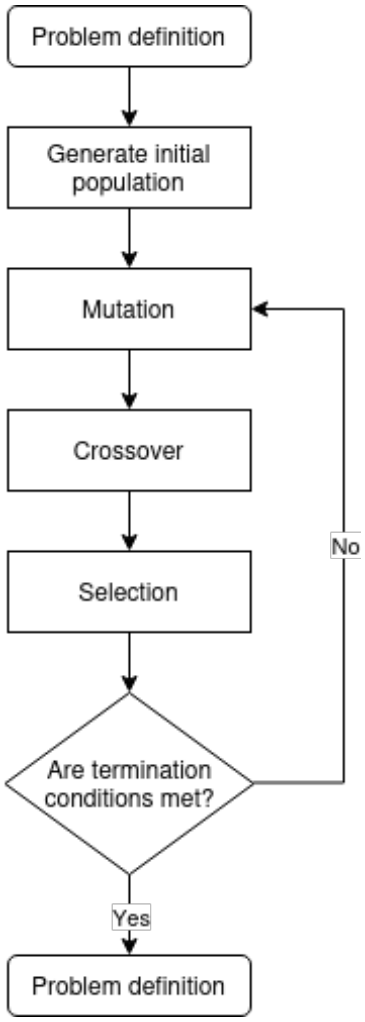


Figure 2.21: Overview of the differential evolution framework.

Simulated Annealing (SA)

Simulated annealing was described by Kirkpatrick [24]. The algorithm is based on annealing in metallurgy. In science, annealing is used in experiments to find the low-temperature state of a material. To create a crystal without defects, the material first needs to be heated up and then slowly cooled down,

spending a long time close to the freezing point. This perfect crystal can be seen as the global optimum. Finding the low-temperature state of a material can be viewed as an optimization problem. However, the changing temperature did not have an obvious equivalent in other optimization systems.

In 1953, Metropolis et al [39] developed a method to simulate atoms in equilibrium at a given temperature. In each step, an atom is given a small random displacement. This changes the energy of the system. If the change in energy is negative, the displacement is accepted and the changed atom is used as the starting point in the next iteration. However, when the change in energy was positive, the changed atom was not just discarded. Instead, the new atom was accepted probabilistically following:

$$P(\Delta E) = e^{\frac{-\Delta E}{k_B T}} \tag{2.26}$$

Thus, the lower the temperature, the less likely it is that the new atom would be accepted.

Kirkpatrick recognized that this simulation could be used in optimization problems by replacing the energy of the atoms by the cost function of the optimization. Then, to apply the annealing process, the algorithm is initialized with a high temperature. This means that the probability of accepting worse solutions is high, which allows for better exploration of the solution space, and reduces the risk of getting stuck in local optima. As the temperature is lowered with each iteration, the chance of accepting worse solutions lessens and the algorithm moves towards a single optimal solution.

There are two main operators within simulated annealing. First is the initial temperature. Setting it very high leads to great exploration of the solution space, but might either increase the computation time or is not able to fully find the most optimal solution. On the contrary, a lower initial temperature may find a solution quickly, but has a higher chance of getting stuck in local optima. The second operator is the annealing schedule. This is the formula the temperature follows for every iteration. The schedule also influences the amount of exploration and convergence depending on how fast the temperature changes. Figure 2.22 gives on overview of the framework of simulated annealing.

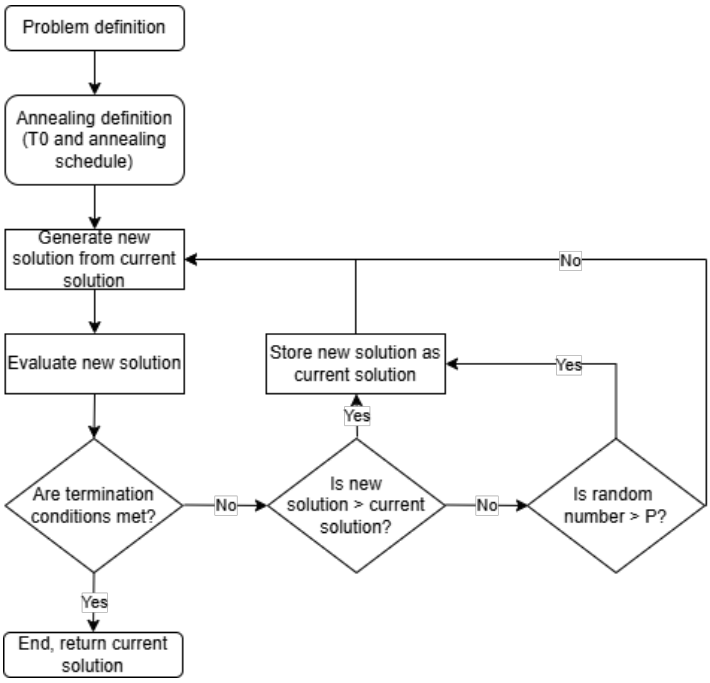


Figure 2.22: Overview of the simulated annealing framework.

Non-dominated Sorting Genetic Algorithm II (NSGA-II)

Dep [11] proposed the improved non-dominated sorting genetic algorithm. This algorithm is very similar to the generic genetic algorithm, but is used to optimize multi-objective problems. It is primarily used for

two-objective optimization, but can also be applied to optimization problems with three objectives. As a multi-objective optimizer, NSGA-II does not provide a single optimal solution. Instead, it generates a Pareto front, which shows the solutions with the best combination of objective values. With NSGA-II, Dep sought to improve the NSGA algorithm proposed by Srinivas and Deb [55]. Three main criticisms of the NSGA algorithm were to be addressed by NSGA-II. Firstly, the high computational complexity of non-dominated sorting. Secondly, the need for the sharing parameter σ_{share}. Thirdly, the lack of elitism.

In non-dominated sorting, the individuals of the populations are sorted into fronts (Figure 2.23). The first front contains all individuals who are not dominated by other individuals of the population. Domination occurs when an individual is worse in all objectives than another individual. When the first front is established it can be temporarily removed. Individuals in the new population that are then no longer dominated are sorted into the second front. This continues until all individuals are sorted. In a naive procedure, the whole population has to be evaluated to create the first front. Say the population is of size N and there are M objectives, then the amount of evaluations for creating the first front is $O(MN^2)$. In the worst case, only one individual is part of each front, meaning N fronts have to be created. This leads to a total of $O(MN^3)$ evaluations. To reduce the computational complexity of the non-dominated sorting, NSGA-II makes use of an improved procedure. First, two parameters are computed for each individual p: the domination count n_p, which is the number of individuals that dominate individual p, and the set S_p of the individuals that individual p dominates. All individuals with a domination count equal to zero are sorted into the first front. For each individual p in the first front, the members in the set S_p is visited, and their domination count is reduced by one. Any member whose domination count then becomes zero is sorted into the second front. This process is continued until all individuals are sorted into fronts. The complexity of this procedure is $O(MN^2)$.

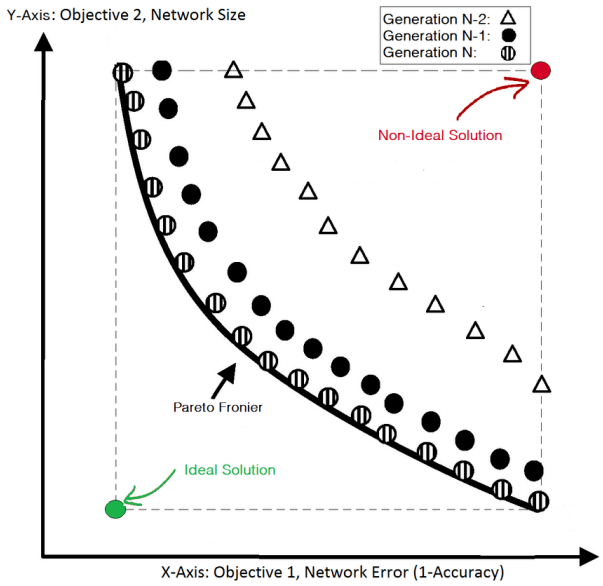


Figure 2.23: A visualization of an example Pareto front. [34]

The original NSGA algorithm made use of the sharing parameter σ_{share} to preserve diversity throughout the optimization. In NSGA-II, a new procedure is proposed that both aims to remove this additional parameter and reduce the complexity of the procedure. The new procedure is called the crowded-comparison approach. In this approach, the crowd density around an individual is estimated. For each individual, the normalized distance between its neighbouring individuals within the same front is computed. Thus, each individual now has two attributes. Firstly, the non-domination rank, i.e. the rank of the front which contains the individual, and secondly the crowding distance. These attributes can be used to compare two individuals. An individual with a lower rank is always preferred to an individual with a higher rank. If two individuals have the same rank, and thus are within the same front, the individual with the higher crowding distance, i.e. in the lesser crowded region, is preferred.

The main algorithm is as follows. First, a population is initialized and an offspring is generated according to a normal genetic algorithm. These two are combined and sorted according to the non-dominated sorting procedure. Starting from the first front, the fronts are selected for the next population, until the new population is the same size as the original. If the last front contains more individuals than there is space in the population, the individuals are selected based on crowding distance. The process then repeats by performing the classic genetic algorithm operations of selection, mutation and crossover to generate a new set of offspring. The framework of the algorithm is shown in Figure 2.24.

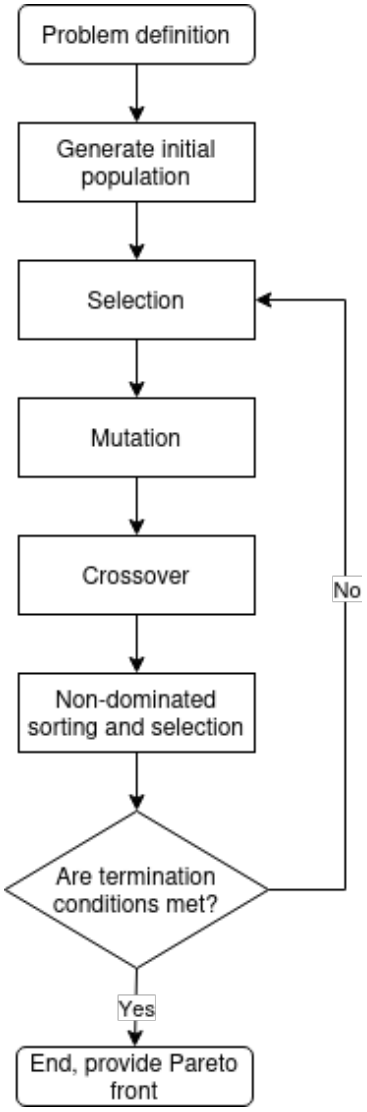


Figure 2.24: Overview of the non-dominated sorting algorithm II framework.

Summary of the optimization algorithms

In this section a variety of optimization algorithms have been studied and explained. First, the algorithms can be divided into two groups; the iterative gradient descent method and heuristic methods. It was concluded that the gradient descent method is not a good fit for satellite constellation optimization primarily due to the lack of a differentiable cost function, making the implementation of this algorithm impossible.

Within the heuristic algorithms further distinctions can be made, depending on which objectives are to be optimized. When considering just the revisit time and coverage figures of merit, a simple weighted sum cost function could be used. All of the studied single objective optimization algorithms can be

applied to this optimization. When the minimization of the number of satellites is also considered an objective of the optimization, two possibilities arise. First, the single objective weighted sum method can still be applied by adding the cost of the number of satellites to the sum. The other option is to use a multi-objective optimization algorithm, such as NSGA-II. Here, a Pareto front can be created with the revisit time and coverage weighted sum on one axis, and the number of satellites on the other. As these heuristic methods are all viable solutions to the problem, a comparison of their applications can be very useful.

2.3.2. The history of satellite constellation optimization

Throughout the last decades, many researchers have studied the optimization of satellite constellations. In this section, an exploration of past studies on constellation optimization is made.

The earliest optimizations of satellite constellations often considered symmetrical constellations such as Walker, Streets-of-Coverage or Flower constellations. One of the earliest research on the optimization of constellations was performed by Lang and Hanson [29]. They used an algorithm on symmetric constellations to minimize the revisit time for global or partial coverage. The user specifies an input consisting of the total number of satellites, the satellite altitude and their sensor characteristics. The algorithm solves for the optimal inclination and arrangement of the satellites to minimize the revisit time. Lang continued researching the optimization of discontinuous coverage of Walker constellations [28], Street-of-Coverage constellations [27] and even the use of genetic algorithms [26].

Dufour [12] used an optimization algorithm to optimize the Dilution of Precision (DOP) for a navigation satellite constellation. Five types of DOP were identified to optimize: geometric, position, horizontal, vertical and time. Dufour used Walker constellations as the starting point of the algorithm. Dufour introduced an alternative version of a Walker constellation which were named multi-Walker constellations. These are a combination of multiple basic Walker constellations, which differ in initial RAAN and initial argument of latitude. These multi-Walker constellations add more flexibility to the coverage while keeping the nice features of basic Walker constellations.

Dufour's optimization algorithm is as follows. First, all Walker constellations within a bounded range of satellites are evaluated and a selection is made of the most promising configurations. Next, the algorithm optimizes the semimajor axis and inclination of the constellation. Finally, the algorithm aims to improve the performance by splitting the best Walker constellation into multi-Walker configurations, and optimizes the initial RAAN and argument of latitude of the different basic Walker constellations.

Dufour applies two methods to evaluate the performance criterion; a discrete and a continuous evaluation. In the discrete evaluation, usage is made of a weighted summation of all DOP criteria. This has the benefit of allowing the user to choose (using weights) which DOPs to include and which they are not interested in studying. To evaluate the precision and the computation time, the continuous method is used. As the problem is nonlinear, Dufour makes use of the Nelder-Mead optimization method. This is a heuristic and thus does not guarantee that it will find the most optimal solution. However, it can be used to indicate the precision of the discrete solution.

In Zhang [68], optimization is used to minimize the mission cost. This is represented by the objective function which includes the number of satellites in the constellation, the number of spacecraft carry bus, and a penalty related to the geometric dilution of precision (GDOP). For simplicity, Zhang only considers homogeneous satellite constellations, specifically Flower and Walker constellations. The search space is even more constricted by only considering circular, repeating ground track orbits. To perform the optimization, Zhang makes use of a differential evolution (DE) algorithm, stating that DE exhibits better performance over particle swarm optimization and genetic algorithms. The previous explored methods optimized symmetrical constellations. However, according to Razoumny [46], for discontinuous coverage the optimal solution often is not a symmetrical constellation. In fact, Razoumny states that the solution should instead by searched for in weakly symmetrical or even asymmetrical constellations.

Like Dufour, Shtark and Garfill [54] also use an algorithm to optimize the (geometric) dilution of precision of navigation satellites. Their constellation optimization algorithm determines the orbital elements at an epoch t_0, of a constellation with n_s satellites, in circular repeating ground track orbits.

The algorithm consists of two stages. The first stage is like the coverage analysis. It consists of a nonlinear equation solver to produce a virtual reference satellite. It passes over a ground station and shares its mean semimajor axis, mean eccentricity and mean inclination with all other satellites. In the second stage, a cost function is defined consisting of the GDOP criteria and the elevation angle constraints. A genetic algorithm is used to optimize the cost function, after which a gradient-based optimization method is applied to refine the results. In this stage, the mean longitude of ascending node and mean argument of latitude are determined for each satellite.

Zhang [67] uses an optimization algorithm to minimize the number of satellites in a repeating ground track constellation, while ensuring that the revisit time does not exceed the predefined one. The revisit time is evaluated for multiple targets.

The algorithm consists of three parts. First, a grid search and numerical method are performed to construct a database of repeating ground track orbits. Second, the least number of satellites is selected. A visiting matrix is defined, where for each orbit in the database it is stated whether or not a target point is visited. The problem then becomes how to visit all the target points with the least number of orbits from the visiting matrix. This problem can be abstracted as a search tree, and a pruning method can be applied to solve it. Finally, the constellation is constructed to meet the revisit time constraint. This is achieved by placing more satellites in the repeating ground track orbits defined in the second step.

Savitri [52] applies a genetic algorithm to maximize percent coverage and minimize the revisit time for a small satellite constellation. First, a semi-analytical initial guess is made by evaluating the coverage of the target area by repeating ground track satellites. It must be noted that the considered orbits are constraint to circular, RGT, low Earth orbits. The initial guess helps lower the overall computational load of the computation.

As Savitri optimizes for two objective functions, usage is made of a multi-objective optimization algorithm, namely the nondominated sorting genetic algorithms (NSGA-II). Four of the six orbital parameters are optimized, only excluding eccentricity and argument of perigee due to the circular orbits constraint. Four optimization objectives were considered: 1) maximum area percent coverage; 2) maximum average area time coverage; 3) minimization of maximum coverage gap time; 4) average coverage gap time. These were combined into the two objective functions using weighted sums.

Lee [30] aims to optimize constellations for optimal regional coverage. The constellation design can be split into two parts: defining the orbital elements of the reference satellite and defining the constellation pattern. No simplifying assumption is made for symmetric constellations. The design is constrained to only consider circular repeating ground track orbits.

Lee introduces a novel method, binary integer linear programming (BILP). This method takes the reference satellite and the coverage requirement to find the minimum number of satellites required. Further, the method is able to provide solutions using sub-constellations.

Mencarelli [36] introduces the mixed integer (non)linear programming (MI(N)LP) methods. These methods minimize the total number of satellites while considering a maximum revisit time constraint. Like in the other papers, only circular repeating ground track orbits are considered. The method optimizes the other four orbital parameters (semi-major axis, inclination, RAAN and mean anomaly) as well as the placement of satellites within the constellation.

Summary on established optimizations of satellite constellations

The optimization of satellite constellations can take many forms, as could be seen in the highlighted studies. Researchers optimized for revisit times, for dilutions of precision, for mission cost or coverage.

Further, algorithms ranging from integer programming to NSGA-II have been applied. The main goal of these studies was to optimize as many parameters of the constellation as possible, without drastically impacting the computational complexity. However in the studies, the optimal constellations were often constrained, either to symmetrical constellations, such as Lang and Hanson [29], Dufour [12] and Zhang [68], or to repeating ground track orbits such as Shtark and Garfill [54], Zhang [67], Savitri [52], Lee [30] and Mencarelli [36]. This leaves room for the exploration of asymmetrical, non-repeating ground track constellations. Further, the only study that involved elliptical orbits was that of Zhang [68], but this was only in the context of symmetrical flower constellations.

3

Research gap and research questions

In light of the studies analyzed in the previous chapter (chapter 2), a number of research gaps can be identified for both the coverage analysis and the constellation optimization. Combining this research gap with the objectives of the thesis stated in the introduction, the research questions can be established.

3.1. Research gap

Various research gaps can be identified from the literature analyzed in chapter 2, both in the coverage analysis methods as well as the constellation optimization studies. Starting with the coverage analysis, the different studies were summarized and evaluated in Table 2.1. In this table, only one method was deemed suitable for the purpose of this research, namely the semi-analytical coverage analysis method developed by Crisp [6]. Compared to the other coverage analysis methods, Crisp's method was not constrained to repeating ground track orbits or symmetrical constellations. However, the method is not fully analytical, having to be repeated for every latitude bound, impacting the computational speed. Further, the constellations analyzed by Crisp were still limited to circular orbits only. Thus the research gaps for the coverage analysis are as follows:

- The coverage analysis method must be able to analyze all orbit types, even elliptical orbits.
- The coverage analysis method must be able to analyze all constellation configurations, including non-symmetrical constellations.
- The coverage analysis method must be as fast as possible, i.e. as analytical as possible.

In the literature for the constellation optimization similar research gaps were identified, as the evaluated studies all constrained the constellations to either symmetrical constellations, or to repeating ground track orbits. Additionally, the chosen optimization algorithm should be able to handle the non-differentiable cost function as well as the non-convex solution space. To summarize the identified research gaps for constellation optimization are:

- The optimizer must be able to optimize all orbit types, not just repeating ground track orbits.
- The optimizer must be able to optimize all constellation configurations, including non-symmetrical constellations.
- The optimization algorithm must be able to handle a non-differentiable cost function.
- The optimization algorithm must be able to find a good solution in a non-convex solution space.

3.2. Research questions

By combining the objective as stated in chapter 1 with the research gap, the research questions can be created. As the thesis can be divided into two parts, the coverage analysis method and the constellation optimization, the research questions are split accordingly. For the coverage analysis method, the

research questions are as follows:

1. Which figures of merit should the new coverage analysis method analyze?
2. How to make the new coverage analysis method as fast as possible?
3. How can the new coverage analysis method be used for elliptical orbits?
4. How can the new coverage analysis method be used for asymmetrical constellations?

Some hypotheses can be created to give an initial direction to the development of the coverage analysis method. Firstly, from the figures of merit evaluated during the literature study in chapter 2, two stand out, namely the revisit time and coverage. Secondly, the fastest way to perform a coverage analysis is by making it as analytical as possible. Thus, efforts should be taken such that this is the case. Thirdly, by combining, modifying and improving the existing coverage analysis methods, one could find a way of analyzing elliptical orbits. Finally, if the coverage analysis can be performed for a singular satellite, a combination of coverage analyses of other satellites could lead to the analysis of any constellation configuration imaginable.

Similarly, research questions can be created for the optimization of the satellite constellations.

1. For what objectives should the satellite constellations be optimized?
2. How can the amount of decision variables be maximized?
3. What optimization algorithm performs the best?

Again, hypotheses can be created to answer the research questions. Firstly, the objectives of the optimization should at the very least contain the figures of merit analyzed by the coverage analysis. Furthermore, constellation parameters such as the number of satellites could also be considered as objectives to be optimized. Secondly, the maximum amount of decision variables would be the inclusion of all orbital parameters of all satellites in the constellation. To ensure maximum flexibility, this should be aimed for. Finally, a comparison should be made of a variety of optimization algorithms to see which performs the best. It can be expected that algorithms that perform well in non-convex, non-linear solution spaces will outperform algorithms that are more suited to linear, predictable solution spaces.

Part II

Methodology

4

Computationally-efficient Earth coverage analysis

Before being able to optimize the coverage and revisit time of a satellite, the method to compute the figures of merit has to be established. This coverage analysis method has to be especially fast, as many calls to this function would be made during the constellation optimization. Thus, great effort has to be made to ensure that the analysis is as fast as possible while retaining accuracy. In this chapter the development of the coverage analysis method is stated, starting with the initial idea and evolving into the final established methodology.

4.1. Initial concept

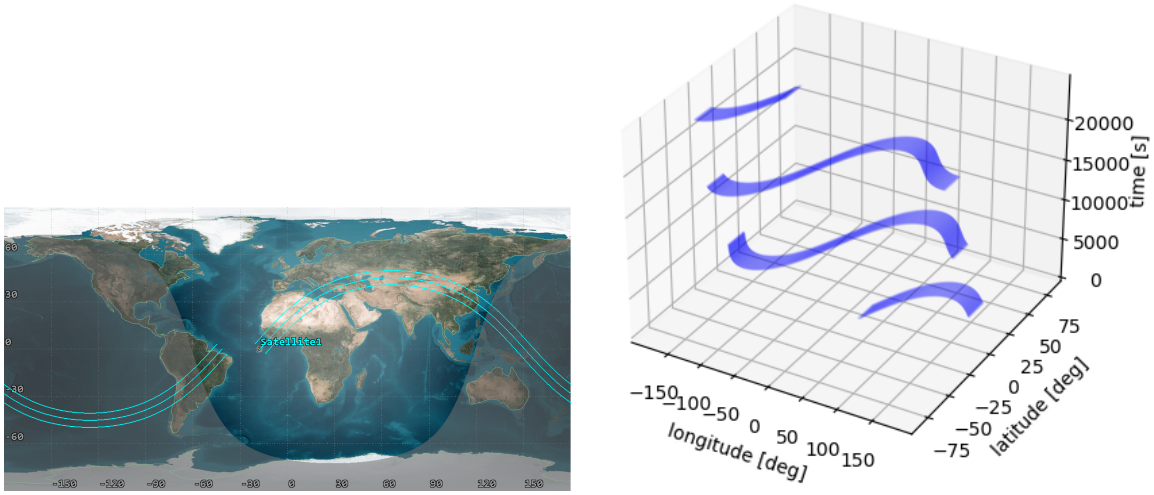
In light of the Earth observation figures of merit, a satellite can be best be evaluated by performing a coverage analysis. Various methods exist, including analytical, semi-analytical, and numerical variations, as shown in Table 2.1. The aim of the initial concept was to create a coverage analysis method that was both not computationally expensive, while allowing great flexibility in the types of satellite orbits it could evaluate. Therefore, efforts were made to keep the initial concept as analytical as possible. Further, to ensure flexibility, the concept should allow for the analysis of all orbits, including elliptical orbits.

A satellite's path around Earth can be visualized by its ground track. This ground track is a 2D representation of the orbit of the satellite. When modeling an Earth observation satellite, its instrument can also be accounted for, which creates an area over Earth that shows which parts will be observed during the orbit of the satellite (Figure 4.1a). By computing the percentage of Earth's surface that the ground track covers, the coverage figure of merit can be analyzed. To also determine the revisit time, a third dimension, time, can be added to the model. Now the ground track is modeled as a surface within latitude-longitude-time space (Figure 4.1b). The revisit time is then the difference in time between two overlapping parts of the ground track surface. By limiting the 3D space according to the user objectives, specific parts of Earth could be analyzed within a specific time frame for coverage and revisit time.

The feasibility of the initial concept depended on two questions. The first of these questions was whether there exists an analytical way of representing the 3D ground track surface. To answer this question, it is important to first formulate the ground track line of the satellite as an equation. In Crisp [6] the equations are given which transform Kepler elements to spherical coordinates (Equation 4.1).

$$\begin{cases} \phi = sin^{-1}\left(sin\left(\nu_\phi + \omega\right) sin(i)\right) \\ \lambda = tan^{-1}\left(\frac{cos(\omega+\nu_\phi)sin(\Omega)+sin(\omega+\nu_\phi)cos(\Omega)cos(i)}{cos(\omega+\nu_\phi)cos(\Omega)-sin(\omega+\nu_\phi)sin(\Omega)cos(i)}\right) + \frac{\nu_\phi}{2\pi}\Delta\lambda \end{cases} \tag{4.1}$$

Further, the time coordinate can be found by relating the true anomaly to the mean anomaly via the eccentric anomaly. This completes the set of equations. Next, the swath should be included to transform



(a) Example 2D ground track made with STK 12.

(b) Example 3D ground track surface.

Figure 4.1: Different representations of an example satellite ground track.

the parametric ground track line to a parametric surface. A number of assumptions are made to model the swath. Firstly, the swath is modeled as a slit. Furthermore, the swath is assumed to always be a straight line. Finally, the swath is assumed to be perpendicular to the ground track of the satellite. This swath line is bounded on both sides of the ground track by the field of view constraint of the instrument of the satellite. These assumptions allow the surface to be classed as a ruled surface. A surface is ruled if for every point a straight line lies on the surface. As a ruled surface, the ground track surface can also be defined by either the two outer bound lines, or by one bound line and the equation of the swath lines. [3]

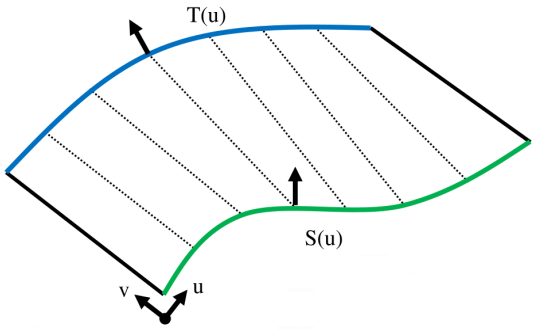


Figure 4.2: An example of a ruled surface. [17]

The second question that is to be answered is whether there is an analytical way of computing the relevant figures of merit from the ground track surface representation. Unfortunately, due to the complex nature of the surface definition, only iterative methods would be able to compute the coverage and revisit time of the satellite. Considering the latitude-longitude-time space and the system of equations defining the surface, there are two ways of iterating. The figures of merit can be evaluated by either propagating the satellite through time, or by iterating through the latitude-longitude grid. As the regular spacing between observation points in the latitude-longitude grid is desirable for both the coverage and revisit time purposes, it was decided to develop a method using this type of iteration.

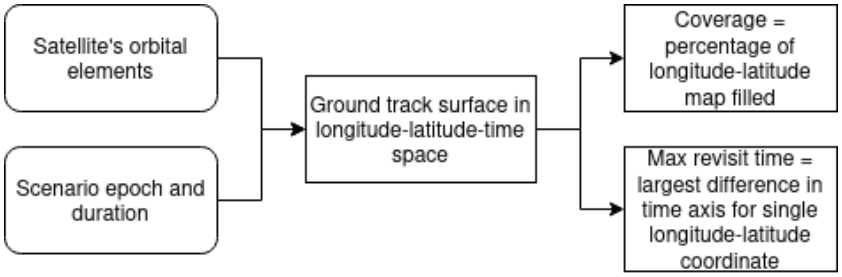


Figure 4.3: Simplified framework of the initial idea.

4.2. Development of grid-based analysis

The grid-based coverage analysis method consists of two parts. Firstly, the grid itself has to be defined in a smart way. This means making sure that the resolution of the grid is not too coarse, which would lack detail, but also not too fine, as this could make the computation much more expensive. Ideally, the pixel size of the grid would be defined by the user. Furthermore, the creation of the grid should be made as efficient as possible to keep the computation time low. The second part of the coverage analysis is the computation of the visit times per grid point. The method should be as accurate as possible. Efforts should also be made to keep the computation as fast as possible.

4.2.1. First grid-based method: the derivative method

During the development of the first grid-based method, the grid creation was kept rather simple. The latitude-longitude grid points were evenly-split, with one degree distance in both latitude and longitude. Every grid point was evaluated during the visit time computation. More focus was laid on establishing the method for the computation of the visit times. Usage was made of the initial assumptions of the swath, mainly that the swath line is straight and perpendicular to the ground track. This meant that the slope of the swath line could be found by finding the slope of the ground track line and rotating it by 90 degrees. Using the slope of the swath line, the ground track of the satellite and the coordinates of the grid point, the position of the satellite, the true anomaly, could be found. This position can then be related to the time at which the grid point is observed. The slope of the ground track could be found by differentiating Equation 4.1 (see Equation 4.2). The full derivation can be found in the Appendix B.

$$\left(\frac{d\phi}{d\lambda}\left(\nu_\phi\right)\right)_{swath} = -\left(\frac{d\phi}{d\lambda}\left(\nu_\phi\right)\right)_{GT}^{-1} = -\left(\frac{d\phi\left(\nu_\phi\right)}{d\nu_\phi}\frac{d\nu_\phi}{d\lambda\left(\nu_\phi\right)}\right)^{-1} = -\frac{d\lambda\left(\nu_\phi\right)}{d\nu_\phi}\left(\frac{d\phi\left(\nu_\phi\right)}{d\nu_\phi}\right)^{-1} \tag{4.2}$$

As can be seen, the slope of the swath line is a function of the true anomaly. This causes a problem in the fact that the position of the satellite is to be computed by a function that takes the position as an argument, meaning that it can not be solved analytically. Instead a numerical solver is used that can find the solution using an initial guess. Once the position of the satellite is found, a check has to be performed whether the grid point is within the field of view of the satellite. First the distance between the sub-satellite point, the projection of the satellite on Earth, and the grid point is computed. This is done by using the Haversine formula [38], which computes the distance between two points on a sphere. This distance is then compared to the length of the swath. If a grid point is observed, the time at which this occurs is saved in a dictionary. This dictionary can then be evaluated to easily find the revisit time at every (visited) grid point, the maximum revisit time of the satellite, and the total coverage of Earth. Finally, to decrease the computation time, it was decided that the visit time computation only needed to be performed for the first orbit. The coverage for additional orbits could be derived using the information of the (full) first orbit, and shifting the points according to the rotation of Earth within one orbital period. The framework of the method is visualized in Figure 4.6.

A number of issues were identified when developing this method. Firstly, the simple grid creation was very inefficient as most of the evaluated grid points would not be able to be observed during the first orbit. At these points, the numerical solver would run for the maximum amount of iterations trying to find a solution, which impacted the computation time greatly. Efforts could be made to reduce the size of the grid in areas that are not close to the ground track.

Secondly, the numerical solver caused various issues. It often got stuck in local optima resulting in bad solutions and longer computation times. Further, it was unable to find solutions for grid points close to the maxima of the ground track (Figure 4.4). This was partially solved by interpolating the visit times for these points, but this made the model much less accurate (Figure 4.5).

Thirdly, the usage of the Haversine formula required the Earth to be assumed spherical instead of oblate, making the model slightly more inaccurate. Finally, it was also found that Equation 4.1 did not work for elliptical orbits. All in all, the method was too slow, taking in the order of tens of minutes to analyze the first orbit, as well as too inaccurate. However, the usage of the first orbit to compute the additional orbits proved to be very promising, as over a thousand orbits could be analyzed in a matter of seconds.

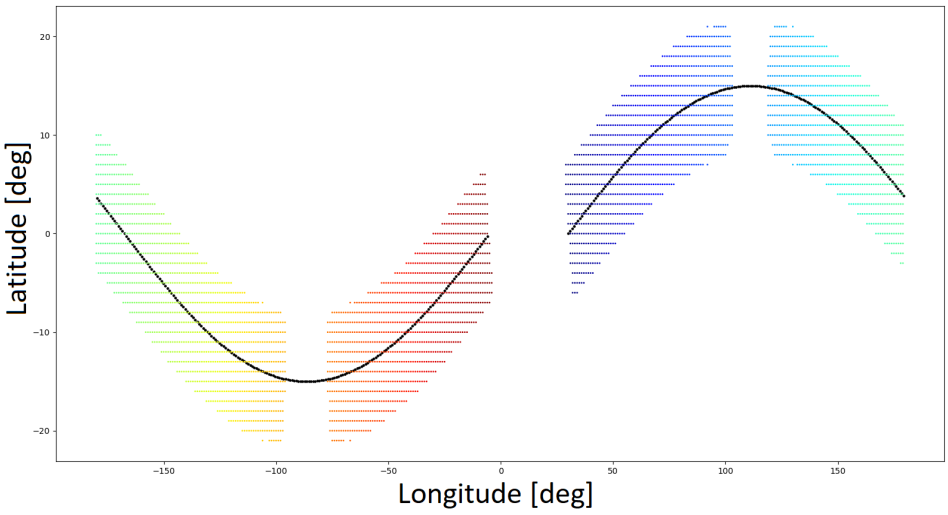


Figure 4.4: Grid coverage for first method without interpolation.

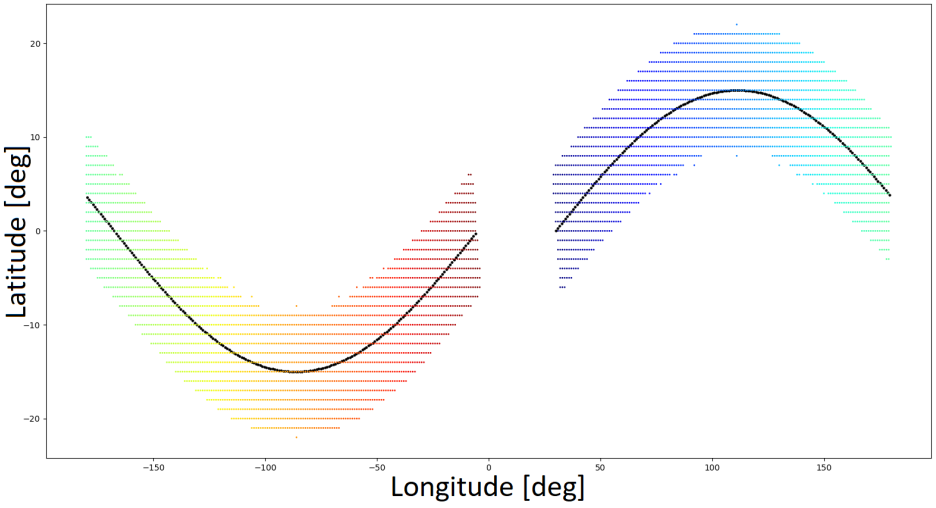


Figure 4.5: Grid coverage for first method with interpolation.

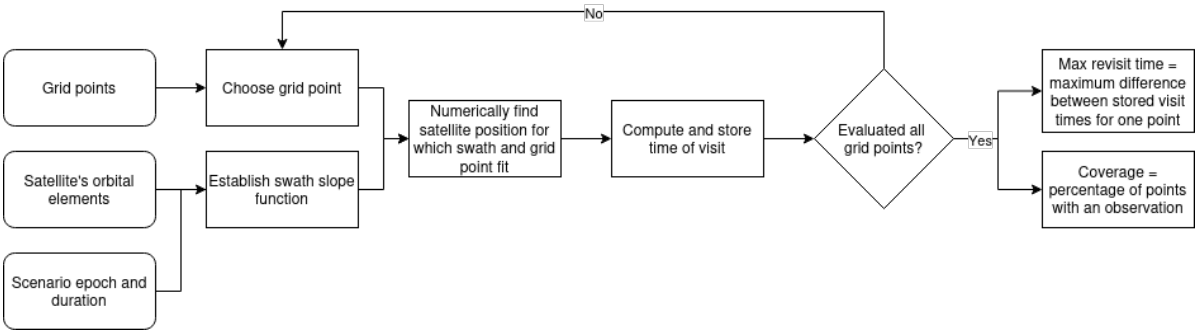


Figure 4.6: Framework for the first grid-based coverage analysis method.

4.2.2. Second grid-based method: the minimum distance method

In the second grid-based method efforts were made to improve and eliminate the limitations of the first method. The size of the grid was reduced by using the position of the ground track. By iterating over the ground track and evaluating the swath, relevant grid points could be identified. Grid points that were far from the satellite's orbit were not to be evaluated. This reduced the computation time substantially, as the numerical solver no longer had to be called at every single grid point.

Next, the ground track equation 4.1 has to be analyzed. The issue with the elliptical orbits was identified to be in the equation for the longitude, and more specifically in the second part of this equation. In this part the drift in longitude due to Earth's rotation and J2 perturbations are computed. The true anomaly is used to identify the passage of time since the start of the orbit. This works for circular orbits, however, for elliptical orbits the evolution of the true anomaly is no longer linear. This means that the true anomaly should not be used to indicate the passage of time. The mean anomaly could be used instead, as this behaves linearly for all orbits. This leads to the revised equation for longitude (Equation 4.3).

$$\lambda = tan^{-1}\left(\frac{cos(\omega+\nu_\phi)sin(\Omega)+sin(\omega+\nu_\phi)cos(\Omega)cos(i)}{cos(\omega+\nu_\phi)cos(\Omega)-sin(\omega+\nu_\phi)sin(\Omega)cos(i)}\right)+\left(\frac{MA_\phi}{nT}+j\right)\Delta\lambda \tag{4.3}$$

The orbit number j is also added to allow the evaluation of orbits other than the first orbit. This modification was checked by the implementation of the unique Molniya orbit (Figure 4.7).

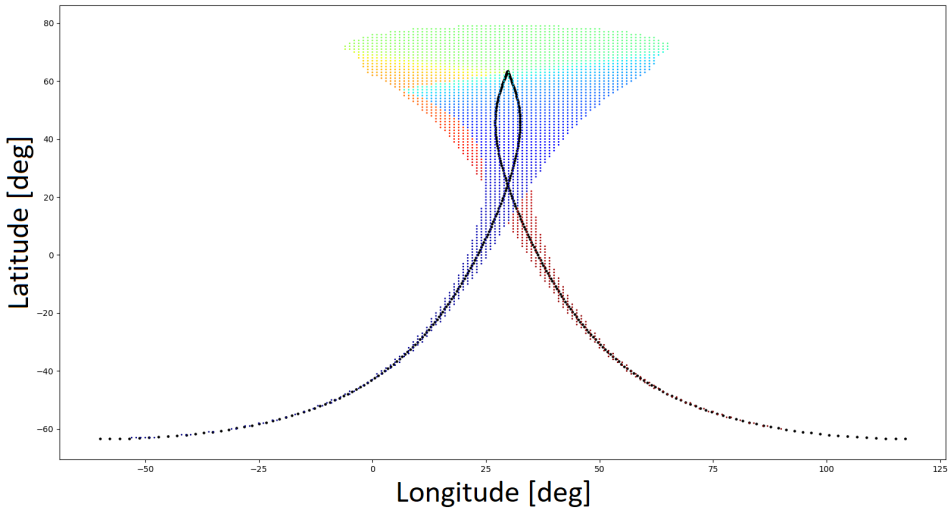


Figure 4.7: Grid coverage of the Molniya orbit using the seconds method.

Some new issues were identified by analyzing the Molniya orbit. Due to the loop in the Molniya orbit, some grid points would be visited twice in one orbit. Thus, a distinction had to be made between

ascending and descending visits, meaning that the grid had to be evaluated twice.

Next, the method to compute the visit times was reevaluated. If a numerical solver was to be used, and the assumptions of the swath line were kept, one could also try to find the satellite position for which the distance to the grid point is minimized. Usage was still made of the Haversine formula to compute the distance between the two points. The implementation of this method also meant that the field of view check could be made instantly. However, the need arose for check on the perpendicularity of the swath line. At the start and end of the orbits, points were incorrectly identified as observed due to the lack of restraints on the slope of the swath line. This caused a bloom of identified points to occur. By ensuring that the line connecting the grid point and the sub-satellite point was (close to) perpendicular to the ground track, these wrong points could be removed. The distance minimization method also was able to correctly evaluate points at the maxima of the ground track, removing the need for the less accurate interpolation. The framework of the method is visualized in Figure 4.8.

The minimum distance method seemed very promising. The improvements to the grid creation and the visit time computation reduced the computation time to the order of tens of seconds. However, there still exist inaccuracies due to the usage of the Haversine formula. Furthermore, the grid is still very rigid and the resolution can not be modified by the user.

To verify the results of the method, a comparison was made with the numerical coverage analysis in STK 12. However, the results of STK 12 were based on geographic coordinates. Thus, a correction had to be made to account for the orientation of Earth at the start of the scenario. First the Kepler elements of the satellite at the start of the orbit were transformed to ECEF coordinates. Then, Equation 4.3 was used to compute the not corrected longitude of the satellite at the start of its orbit. By comparing these two coordinates, the longitudinal shift could be determined. This could then be applied to the rest of the computations.

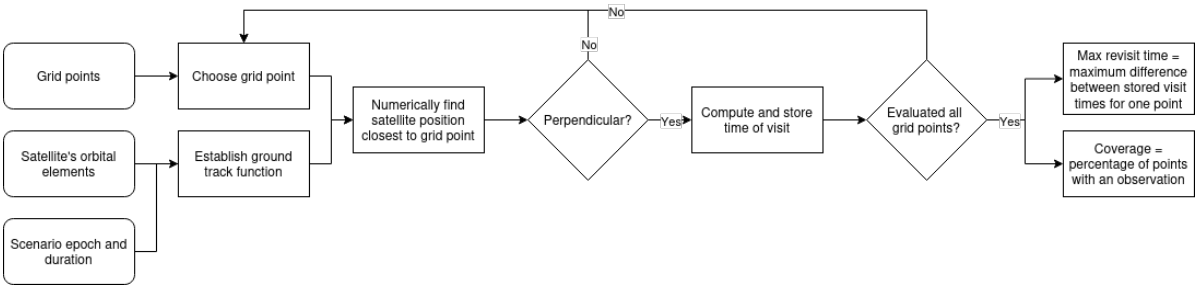


Figure 4.8: Framework of the second grid-based coverage analysis method.

4.3. Finalized concept

While verifying the results of the second grid-based method, a crucial error was discovered. The assumption that the swath line is perpendicular to the ground track was incorrect. The swath line is perpendicular to the 3D representation of the orbit, but this does not translate to the 2D ground track. This discovery meant that the computation of the visit times had to be revised again, as the minimum distance method no longer holds.

4.3.1. Wrong assumption

A substantial wrong assumption was made during the development of the previous coverage analysis methods. The swath was modeled as a slit, meaning that in the 3D visualization of the problem, the swath would be a straight line perpendicular to the orbital track of the satellite. This is a correct representation of a slit swath. Where it went wrong was in the transfer from the 3D case to 2D. In 2D, a map representation is made of Earth. This is, however, not a true representation of Earth, as latitude-longitude points are shifted due to the lack of representation of the curvature of Earth.

During the previous coverage analysis methods, this change in representation of Earth was not taken into account. Instead, it was assumed that if the swath is a straight line perpendicular to the orbital track in 3D, then it can be represented as a straight line perpendicular to the ground track in 2D. This is wrong as it disregards both the curvature of Earth as well as the rotation of Earth. This in turn completely invalidated the computation of the position of the satellite. In the first grid method the swath line could only be fitted between the grid point and the ground track due to the straight line formula used. Further, in the second method, the minimization of the distance between a grid point and the ground track only holds if the swath connecting the two can be represented by a straight line. Thus, the coverage analysis method had to be reenvisioned.

4.3.2. Observation time computation

The shift from 2D to 3D meant that Equation 4.3 could no longer be used to compute the time of observation. Instead, a combination of the first and second methods could be made, while applying them in 3D. The swath vector could be fitted between the satellite and the grid point by envisioning a plane. This plane lies on the swath line and encompasses the satellite position as well as the center of Earth. Then, the distance between the plane and the grid point could be minimized by optimizing the position of the satellite. This is how the satellite position can be found for which the swath crosses a grid point in 3D space. To start, the Kepler elements of the satellite are transformed to inertial Cartesian coordinates, using Equation 4.4.

$$\begin{cases} p = a(1-e^2) \\ r = \frac{p}{(1+ecos(\nu))} \\ h = \sqrt{\mu p} \\ x = r(cos(\Omega)cos(\omega+\nu) - sin(\Omega)sin(\omega+\nu)cos(i)) \\ y = r(sin(\Omega)cos(\omega+\nu) + cos(\Omega)sin(\omega+\nu)cos(i)) \\ z = r(sin(i)sin(\omega+\nu)) \\ \dot{x} = \frac{xhe}{rp}sin(\nu) - \frac{h}{r}(cos(\Omega)sin(\omega+\nu) - sin(\Omega)cos(\omega+\nu)cos(i)) \\ \dot{y} = \frac{yhe}{rp}sin(\nu) - \frac{h}{r}(sin(\Omega)sin(\omega+\nu) + cos(\Omega)cos(\omega+\nu)cos(i)) \\ \dot{z} = \frac{zhe}{rp}sin(\nu) + \frac{h}{r}sin(i)cos(\omega+\nu) \end{cases} \tag{4.4}$$

As the latitude of the satellite is still the initial guess, the true anomaly can be found using the latitude equation in Equation 4.1. A position and velocity vector can be formed from the inertial Cartesian coordinates. The direction of the swath in 3D space can be found by taking the cross product of the position and velocity vectors of the satellite. Next, the plane is to be formed using the swath direction and the position of the satellite. The minimum distance between a point and a plane will be along a line parallel to the normal vector of the plane. Thus, by taking the cross product of the swath vector and the satellite position vector, the direction of the normal vector is found. Finally, the distance can be computed using Equation 4.5 [57].

$$d = \frac{|\vec{n} \cdot \vec{P}|}{|\vec{n}|} \tag{4.5}$$

By minimizing this distance, the satellite position is determined for which the grid point is observed. The satellite position, i.e. the true anomaly of the satellite, can then be used to find the observation time. This is done using Equation 4.6. The framework of the coverage analysis method is visualized in Figure 4.9

$$\begin{aligned} EA &= tan^{-1}\left(\frac{\sqrt{1-e^2}sin(\nu)}{1+ecos(\nu)}\right) \\ MA &= EA - esin(EA) \\ t &= \frac{MA}{n} \end{aligned} \tag{4.6}$$

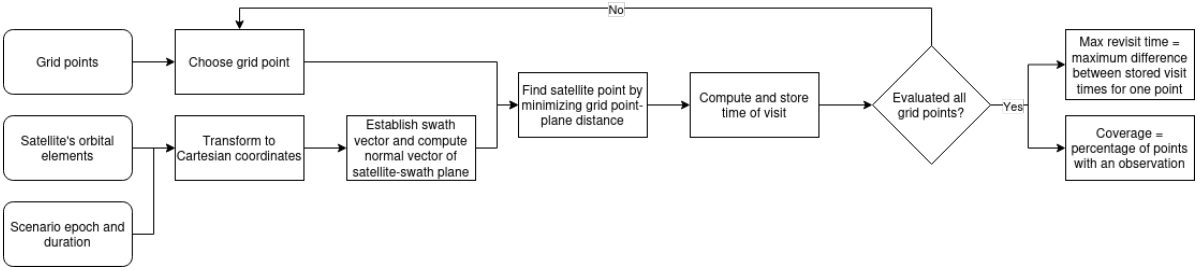
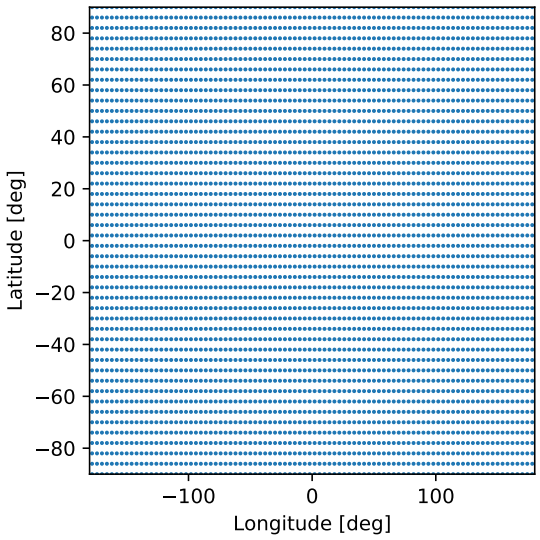


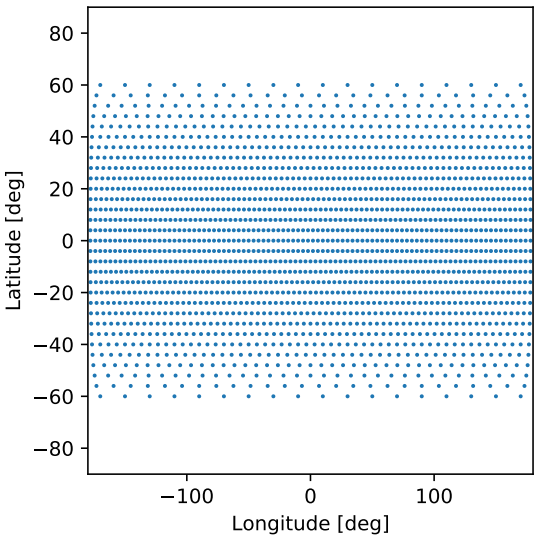
Figure 4.9: Framework of the finalized coverage analysis method.

4.3.3. Grid creation

The overall grid creation method is similar to the one developed during method two. To summarize, only grid points close to the ground track are to be included in the grid. A flag was added to indicate whether a point is to be analyzed during the ascending phase, the descending phase or both, which reduced the amount of iterations necessary. Further, a number of features were added to make the grid more flexible and more customizable. Firstly, it was made possible to analyze specific latitude bands. If the user is only interested in the coverage of the latitudes around the equator, this can now be indicated and used. Secondly, the grid was made less rigid. The size of the pixels can be input in degrees. Finally, the option is added to reduce the amount of grid points at higher latitudes. The user can input the minimum and maximum pixel width in degrees. The grid will then be created with equally sized pixels per latitude as close to the pixel width desired by the user. Figure 4.10 shows what such a customized grid could look like.



(a) Rectangular grid with equal pixel width and height of four degrees.



(b) Variable grid with pixel height of four degrees and width of four to twenty degrees. The grid is bounded in latitude between -60 and 60 degrees.

Figure 4.10: Comparison of a standard grid and a customized grid.

4.3.4. Assumptions and limitations

A number of assumptions and limitations occur in the new coverage analysis method. In this section, a summary is made of these particularities.

- The swath line is no longer assumed to be perpendicular to the ground track. Instead, it is assumed to be perpendicular to the 3D orbital path of the satellite. This also means that the swath line is no longer assumed to be a straight line in the 2D representation. Instead it is a curved line due to the curvature of Earth.

- The Earth is assumed to be spherical.
- The coverage analysis can not evaluate more than two visits of a grid point in one orbit. This is due to the fact that during the grid creation visited grid points are assigned to either the ascending phase, the descending phase or both. Per phase only one visit is analyzed.
- The computation time of the analysis is sensitive to certain scenarios, specifically scenarios for which the constellation observes large swaths of Earth, such as high altitudes and/or wide swaths.
- The coverage of grid points involves rounding, as partially observed points are not considered. When considering the small inaccuracies due to the assumptions made, this can result in some points incorrectly being labeled as observed or not. Furthermore, as the partial observations are not stored, the inaccuracy is amplified when considering additional orbits of the same satellite, due to the copying of the observations in the first orbit.

5

Constellation optimization

With the coverage analysis method established for a single satellite, it can be applied in the constellation optimization. First, the coverage analysis should be expanded to be able to analyze the figures of merit for the full constellation. Then, the overall framework of the optimizer should be defined. Finally, numerous optimization algorithms are tuned and compared.

5.1. From satellite to constellation

The coverage analysis is established on a single satellite basis. To optimize a satellite constellation, the analysis needs to be expanded. The coverage analysis for a single satellite yields all the observed grid points with their respective times of observation. This can be repeated for all satellites in the constellation, and merged into a result for the full constellation, which describes all grid points that were observed by at least one of the satellites in the constellation.

5.2. Architecture of the optimization

A number of parameters and functions need to be defined clearly before starting the optimization. First and foremost, the objectives should be clearly defined. There are three objectives that the optimization should aim for, namely minimizing the revisit time, maximizing the coverage and minimizing the number of satellites in the constellation. There are multiple ways in which these objectives can be represented within the optimization. For single-objective optimization, the objectives can be combined into a single cost function, such as Equation 5.1. Weights are added into the cost function to tune the importance of the objectives with respect to each other.

$$C = w_{rt} t_{revisit} + w_{cvg}(1 - cvg) + w_{sat} n_{sat} \tag{5.1}$$

For multi-objective optimization, different strategies can be used, as explained in subsection 2.3.1. The most straightforward would be to optimize the three objectives separately, creating a 3D Pareto front to show the most optimal solutions. If the algorithm is limited to two objectives, the choice can be made to optimize for a cost function consisting of the revisit time and coverage, and for the number of satellites. This creates a 2D Pareto front, showing the cost function versus number of satellites.

Second are the decision variables. As the optimization is be kept as flexible and unconstrained as possible, the decision variables consist of all six orbital elements per satellite in the constellation. This means that the length of an input sequence is variable, as it is dependent on the number of satellites in the constellation. Not all optimization algorithms are able to handle variable-length input sequences and measures should be taken such that this is adequately taken into account.

Thirdly are the constraints on the decision variables. These constraints are either set by the user, or to ensure that the constellation is practically feasible. Each orbital element should be within the desired

range specified by the user. The user can also specify the range in the amount of satellites that are to be used. More practically, it should be ensured that the combination of the semi-major axis and the eccentricity does not lead to a perigee that is within, or very close to, Earth. Finally, if the user does not constrain the angular orbital elements, they are practically constraint by the definition of angles and can not exceed 360 degrees (or 180 degrees for inclination).

5.3. Optimization algorithm

Many different optimization algorithms exist. In subsection 2.3.1, a variety of popular optimization algorithms that could be used in satellite constellation optimization were presented. This section provides an overview of the implementation of these different optimization algorithms. Two main Python packages were used to implement the optimization, namely Distributed Evolutionary Algorithms in Python (DEAP) and SciPy. These packages allow for the creation and customization of the algorithms. Various settings can be tuned and changed for each different algorithm. In this section an overview is given on how the algorithm is implemented and what settings were used during the optimization.

5.3.1. Genetic Algorithm (GA)

The genetic algorithm was implemented using DEAP. DEAP allows for great customization of GA's and provides various methods for selection, crossover and mutation. It also allows for the creation of custom methods to be used by the GA. As stated in section 5.2, not all algorithms are able to evaluate variable-length input individuals. For these algorithms, the optimization was performed multiple times, resulting in a solution for each constellation size. These solutions could then be compared by the user and the best solution based on the user's preferences is then chosen. If the optimization algorithm is able to handle variable-length individuals, only the most optimal solution is provided. The user is able to influence the desired solution by modifying the weights in the cost function.

Selection

Various studies have been performed on the comparison of different selection methods in genetic algorithms. Pandey [43] compared rank based, roulette and tournament selection on a network security use case. It was found that rank based selection performed the best, followed by roulette selection. Tournament selection performed the worst. Another study was performed by Goldberg [13] on deterministic differential equations. Proportionate reproduction, ranking selection, tournament selection, and Genitor (or "steady state") selection were compared. It was concluded that tournament selection with large tournament sizes worked well, in constrast to the conclusion of Pandey. Thus, the choice in selection method can be quite dependent on the use case of the algorithm.

Various different selection procedures can be used for a genetic algorithm in DEAP. Two of these can be used in the single objective minimization optimization, namely tournament selection and roulette selection. For tournament selection, twenty percent of the individuals in the population is randomly chosen. The best individual in the tournament is then chosen to be part of the new population. This is repeated until the new population is the same size as the old population. In random selection, a random individual from the old population is selected to be part of the new population. This repeats until the new population is the same size as the old population.

During the selection phase, elitism can also be applied to the optimization. In elitism, the top individuals of the old population are directly copied to the new population. The rest of the new population is then chosen by either tournament or random selection.

Crossover

Like selection procedures, crossover methods have also been extensively studied and compared. Jaradat [21] compared various crossover operators in a WSN lifetime use case. It was concluded that one-point crossover performed the best. Magalhães-Mendes [37] concluded the same, one-point performed the best followed by uniform crossover.

In the satellite constellation optimization use case a distinction has to be made for the crossover method of the GA. Some methods are able to handle variable-length individuals while others can not.

Thus, two different genetic algorithms were created to compare the influence of variable-length versus fixed-length individuals. For the variable-length individuals two different crossover methods were compared, namely one-point and two-point crossover. These crossover procedures are very similar. Either one or two random points within the individual length are chosen. Two individuals are then split in these points and their parts are interchanged. This creates two new children individuals that are added to the new population. As the individuals are of variable length, it must be made sure that the point(s) at which the individuals are split lie within the shortest individual.

For fixed-length individuals three different crossover methods were compared, namely one-point, two-point and blend crossover. One- and two-point crossover work the same as for variable-length individuals. In blend crossover, the children are created by selecting the decision variables between the two parents using a uniform distribution. The interval in which the decision variable lie is determined by a parameter α and the distance between the parents for the decision variable d_i.

$$x_{child}^{(i)} \sim U[min_i - \alpha d_i, max_i + \alpha d_i] \tag{5.2}$$

This crossover method introduces an additional level of mutation and thus exploration into the optimization.

Mutation

Finally, the mutation methods have also been compared in previous studies. Cazacu [5] compared uniform, polynomial and Gaussian mutation on an OOGA framework. it was concluded that uniform mutation performed better than Gaussian. Further, polynomial mutation also performed well.

In the use case of constellation optimization, mutation methods that switched around decision variables could not be applied, due to the structured nature of the input individuals. Further, due to the difference in order of magnitude between the decision variables a standard mutation method would not work correctly. Instead, two custom methods were created based on existing mutation methods, Gaussian and uniform mutation. Instead of specifying one distribution for the whole individual, a separate distribution is made for each decision variable, depending on the range of the variable. For the Gaussian mutation, the standard deviation of the distribution is set to the difference between the maximum and minimum value of the decision variable, divided by three. Due to the empirical rule, this means that 99.7 percent of values will be within the range. This value is then added to the original decision variable of the individual. For uniform mutation, the mutation value is found by randomly picking a value within the possible range of the decision variable.

An additional mutation has to be added to the GA handling variable-length individuals. This is due to the fact that the number of satellites is technically also a parameter that should be mutated. Thus, an additional mutation check is made to see if a random satellite in the constellation gets deleted, or if a random satellite gets added to the constellation.

5.3.2. Particle Swarm Optimization (PSO)

The PSO algorithm was implemented using DEAP. In particle swarm optimization, the main settings that influence the optimization are the social and cognitive bias, the inertia of the particle and the velocity bound. For this optimization the social and cognitive biases were kept equal. Further, the velocity bound is set at twenty percent to ensure that the particle does not stray too fast. There exist multiple methods for implementing the inertia of the particle. Based on Bansal [1], two of these methods were chosen for comparison, namely constant inertia and random inertia. For constant inertia, the inertia value was to 0.5. For random inertia, a value is selected from the standard uniform distribution. This value is halved and added to 0.5.

$$w = 0.5 + \frac{Rand()}{2} \tag{5.3}$$

5.3.3. Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

CMA-ES was implemented using DEAP. There are no parameters to change within the algorithm. Instead, attention has to be paid to the restrictions on the input individuals. The algorithm requires a

singular value for the standard deviation of the initial distribution. As the original decision variables have differing orders of magnitude, they have to be normalized to conform to the same standard deviation. The normalization is performed by subtracting the minimum possible value of the decision variable and then dividing it by the difference between the minimum and maximum value. This transforms each variable to lie between zero and one. The initial centroid then becomes 0.5 for each decision variable. The standard deviation of the initial distribution is set to $1/3$.

5.3.4. Differential Evolution (DE)

The SciPy package was used to implement the differential evolution algorithm. The main setting that can be changed is the strategy. These strategies influence the tuning parameters of DE, the scale factor and the crossover ratio. Three strategies were chosen to be compared, the SciPy default of ”best1bin”, and two standard strategies [66], ”rand1bin” and ”rand1exp”. In best1bin two members of the population are randomly chosen. The difference between these individuals is used to mutate the best member so far. This allows for the construction of a trial vector. Starting from a randomly chosen parameter, the trial vector is filled with parameters from either the mutated or the original member. This choice is made with a binomial distribution. The final parameter is always chosen from the mutated member. If the trial vector performs better than the original, it replaces the original member. In rand1bin, a random member is chosen for mutation instead of the best member. In rand1exp, an exponential distribution is used instead of a binomial one. [44]

5.3.5. Simulated Annealing (SA)

Due to its simplicity, the simulated annealing optimization was implemented without using any package. Instead, a simple Python script was written. The main settings to change in simulated annealing are the initial temperature and the annealing schedule. Two annealing schedules were compared, exponential and linear, which are both widely used [42]. In linear annealing the temperature of the simulation is gradually and constantly lowered by a given parameter. This allows for wide exploration of potential solutions, but often does converge slower. In exponential annealing the temperature decreases exponentially. This allows for a faster converges, but might have a higher risk of premature convergence or getting stuck in local optima.

The initial temperature is dependent on the order of magnitude of the objective function. For the satellite constellation optimization, the objective function values are in the order 10^4 to 10^6. Therefore, three initial temperatures were compared, 10^4, 10^5 and 10^6.

5.3.6. Non-dominated Sorting Genetic Algorithm II (NSGA-II)

The NSGA-II algorithm was implemented using DEAP. It is the only multi-objective optimization algorithm that was used, as it allows for the usage of variable-length input individuals. Like the single objective genetic algorithm, a variety of options are provided by the DEAP package.

Selection

In DEAP, the NSGA-II algorithm is characterized by its selection procedure. To perform an NSGA-II optimization, the selection procedure has to be set to a specific selection method in DEAP. Thus no comparison could be made between different selection methods.

Crossover

Similarly to the variable-length GA, two crossover methods were considered. These methods were one-point crossover and two-point crossover.

Mutation

Two mutation methods were chosen to be compared, Gaussian mutation and uniform mutation, just like the GA. As the NSGA-II algorithm handles variable-length individuals, the number of satellites should also be mutated.

Summary of the optimization algorithms settings

In Table 5.1 the previous sections are summarized. It gives an overview of the various settings that are to be compared for each different optimization algorithm.

Algorithm	Tuning parameters
GA	• Selection: Tournament, Roulette, Elitist • Crossover: One-point, Two-point, Blend • Mutation: Gaussian, Uniform
PSO	Inertia function: Constant, Random
CMA-ES	-
DE	Strategy: best1bin, rand1bin, rand1exp
SA	• Initial temperature: 10^4, 10^5, 10^6 • Annealing schedule: Exponential, Linear
NSGA-II	• Crossover: One-point, Two-point • Gaussian, Uniform

Table 5.1: Summary of the tuning parameters of each algorithm

5.4. Multiprocessing

To speed up the optimization tool as much as possible, multiprocessing could be applied. There are two main ways or levels at which parallel computing could be applied. Firstly, it could be applied at satellite level during the coverage analysis. Here it can be used to speed up both the grid creation as well as the observation time computation. During the grid creation, the script iterates over each true anomaly step of the satellite orbit. To apply multiprocessing, the steps have to be divided among the different processors. However, as more points are expected at higher latitudes, the split should be made alternating. This will more evenly split the required work over the processors. During the computation of the observation time, the script iterates over the established grid. Thus, to apply multiprocessing the grid is split evenly among the processors. This split does not have to be alternating, as the work required for each grid point is the same for every grid point.

The second way that multiprocessing could be implemented is at constellation level, during the optimization. The two packages that are used to implement most of the optimization algorithms, DEAP and SciPy, both allow for the parallel evaluation of individuals. This means that parallel calls can be made to the evaluation function to evaluate different constellations at the same time. On the contrary, the coverage analysis multiprocessing is used with the evaluation function and requires many more calls.

It can be expected that the constellation-level multiprocessing would outperform the satellite-level due to the reduced overhead caused by starting the multiprocessing process. Nevertheless, a comparison should be made to verify this hypothesis.

Part III

Results and discussions

6

Coverage analysis results

In this chapter the results of the coverage analysis are provided. There are three main objectives for which the coverage analysis method is to be tested, accurate coverage of the grid, accurate revisit time computation and an as low as possible computation time. These results are verified with the numerical software of STK 12.

6.1. Coverage

It is of utmost importance that the coverage analysis is able to correctly evaluate all visited grid points for all types of orbits. Thus a variety of orbital element combinations are to be tried to see if the script is able to handle them. Figure 6.1 and Figure 6.2 show the coverage plots for different orbit types. The colors of the grid points correspond to the first time of observation, where blue is the beginning of the scenario and red the end.

The tested scenarios include specific edge case scenarios that often are used in the real world. The first edge case is the equatorial orbit, shown in Figure 6.1a. When the satellite flies over the same latitude for a large fraction of its orbit, a problem arises in the observation time computation. In the algorithm, the latitude of the satellite is used as an initial guess to initialize the problem. This latitude ϕ is then used to compute the true anomaly ν_ϕ according to:

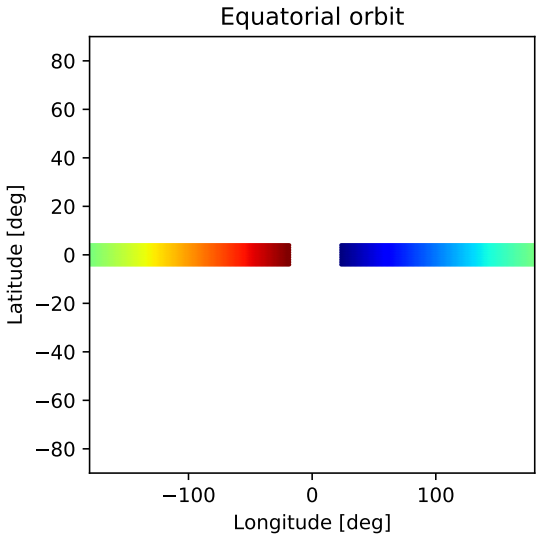
$$\nu_\phi = sin^{-1}\left(\frac{sin(\phi)}{sin(i)}\right) - \omega \tag{6.1}$$

For small inclinations i, Equation 6.1 can not be solved. Thus, to allow for the analysis of (close to) equatorial orbits, the algorithm has to be modified. Instead of using the latitude of the satellite as an initial guess, from which the true anomaly is computed, an initial guess is immediately made of the true anomaly. Although the argument of perigee is arbitrary for equatorial (and circular) orbits due to the lack of the ascending node, it can still be used in the algorithm. By defining an argument of perigee, the ascending node is also defined, dividing the orbit in an ascending and descending phase. This helps in providing a more accurate initial guess, depending on the phase of the satellites:

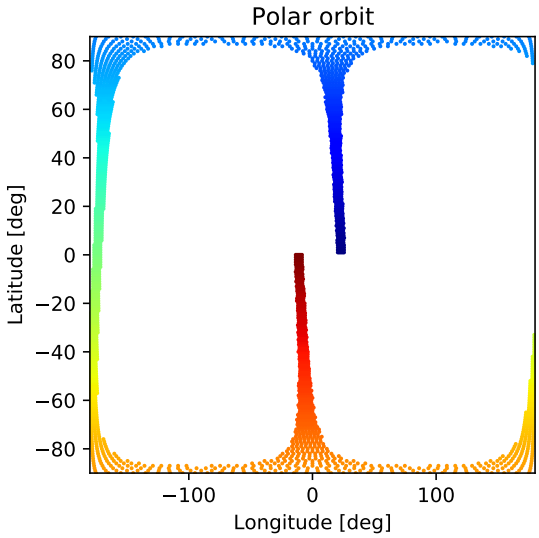
$$\begin{cases} \phi_0 = 90^\circ - \omega, & \text{if ascending} \\ \phi_0 = 270^\circ - \omega, & \text{if descending} \end{cases} \tag{6.2}$$

The initial true anomaly guess is thus set in the middle of the respective phase of the satellite. This allows the minimization algorithm to work and provide the correct results.

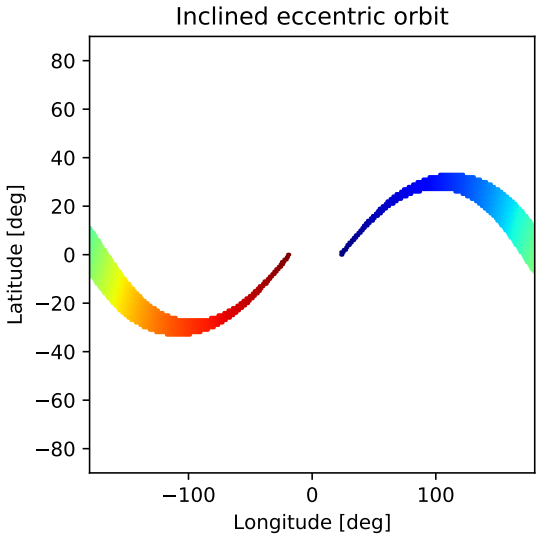
The second edge case is the polar orbit (Figure 6.1b). For (close to) polar orbits the swath of the satellite is able to reach over the poles and observe grid points with a difference of 180 degrees. Due to the conversion to Cartesian coordinates during the algorithm, this proved no issue. Furthermore, the polar orbit case also shows the correct implementation of the rotation of Earth, as the coverage plots



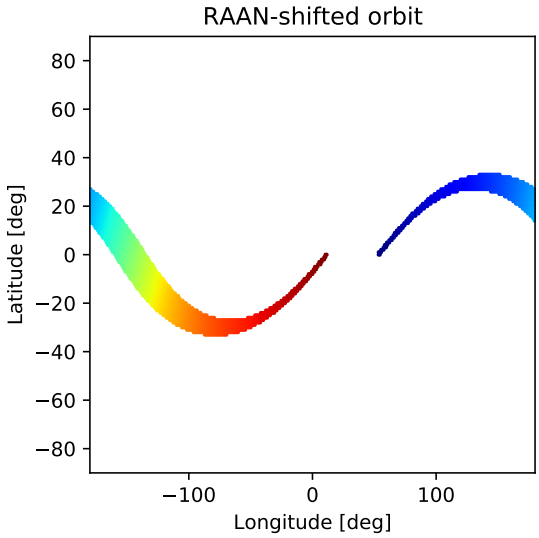
(a) Coverage for equatorial orbit. $a = 10000$ km, $e = 0$, $i = 0°$, $fov = 15°$



(b) Coverage for polar orbit. $a = 10000$ km, $e = 0$, $i = 90°$, $fov = 15°$



(c) Coverage for inclined, eccentric orbit. $a = 10000$ km, $e = 0.3$, $i = 30°$, $fov = 15°$



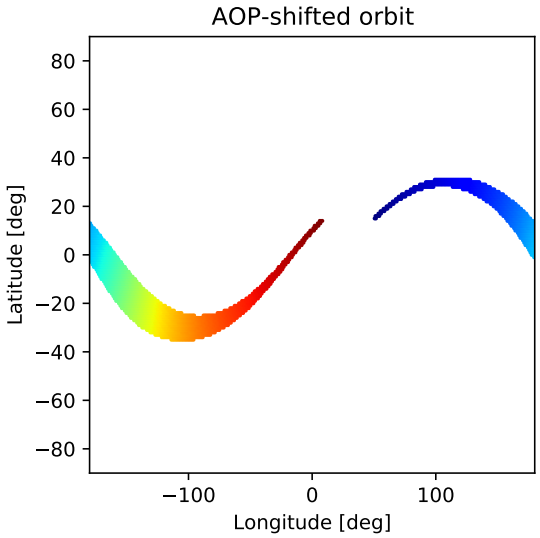
(d) Coverage for RAAN-shifted, inclined, eccentric orbit. $a = 10000$ km, $e = 0.3$, $i = 30°$, $\Omega = 30°$, $fov = 15°$

Figure 6.1: Some coverage plots for a variety of different orbits.

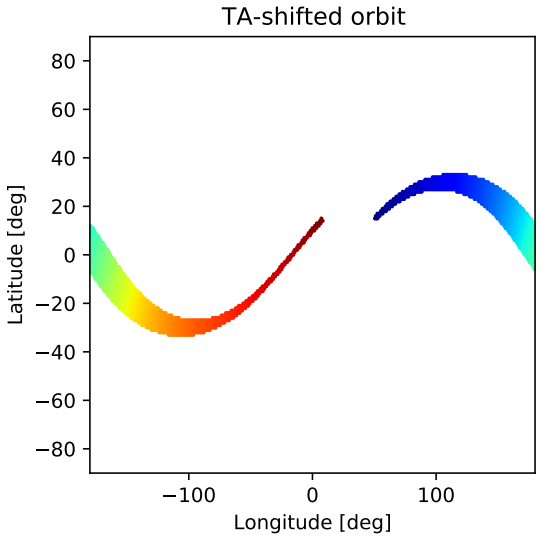
shows the satellite moving slowly eastward. Additionally, the flexibility of the grid creation method is also shown, as grid points closer to the poles are more spaced apart in longitude than the grid points at the equator.

With the edge cases of zero and 90 degrees inclination correctly working, a medium inclination should also be checked. In the coverage plots the orbit inclined at 30 degrees is combined with the eccentric orbit case, as seen in Figure 6.1c. It can be seen that the algorithm is able to correctly follow the satellite's inclined path, forming a wave pattern. Furthermore, the change in width of the coverage swath shows the effect of the eccentricity.

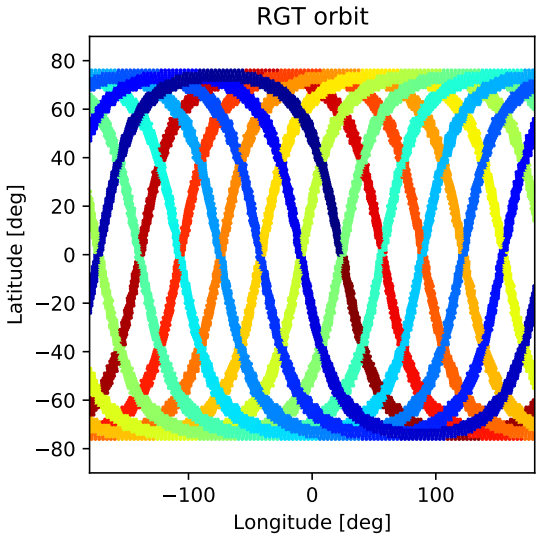
Next to the inclination and eccentricity, the coverage analysis should also be able to handle differences in the other orbital elements. Figure 6.1d, Figure 6.2a and Figure 6.2b show the case for different values of the right ascension of ascending node, argument of perigee and initial true anomaly respec-



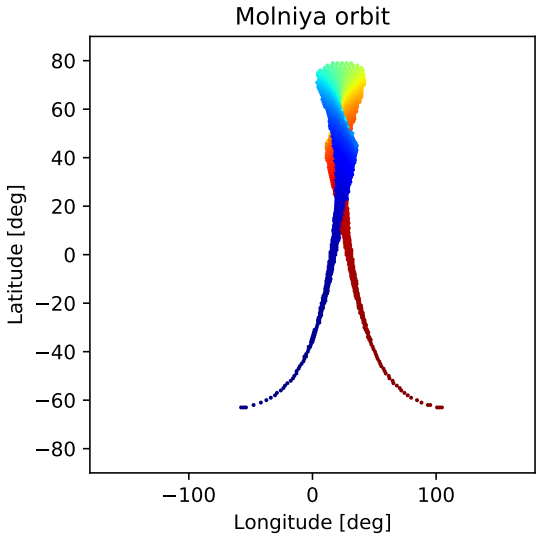
(a) Coverage for argument of perigee shifted, inclined, eccentric orbit. $a = 10000$ km, $e = 0.3$, $i = 30°$, $\omega = 30°$, $fov = 15°$



(b) Coverage for true anomaly shifted, inclined, eccentric orbit. $a = 10000$ km, $e = 0.3$, $i = 30°$, $\nu_0 = 30°$, $fov = 15°$



(c) Coverage of one repeat cycle for repeating ground track orbit, with repeat cycle of eleven orbits. $a = 8524.7$ km, $e = 0$, $i = 105.85°$, $fov = 15°$



(d) Coverage for Molniya orbit. $a = 26600$ km, $e = 0.74$, $i = 63.4°$, $\Omega = 0°$, $\omega = 270°$ $fov = 5°$

Figure 6.2: More coverage plots for a variety of different orbits.

tively. The rest of the parameters were kept the same as the inclined, eccentric orbit (Figure 6.1c). Comparing the RAAN-shifted orbit with the unshifted one, it can be seen that the starting position is correctly shifted eastward according to the change of thirty degrees. For the orbit with a shift in the argument of perigee, it can be seen that the satellite is at a higher latitude than for the unshifted orbit. This is due to the fact that the coverage analysis always assumes that at the satellite is present at its perigee at the scenario start. Thus a shift in the location of the perigee equates to a different initial position of the satellite. It can also be seen that the width of the swath is different between the two orbits. This is again due to the shift in perigee position. Finally, the orbit shifted in true anomaly is very similar to the AOP-shifted orbit. The initial position of the satellite is very similar. The main difference lies in the swath width throughout the orbit. For the TA-shifted orbit this is the same as the unshifted orbit as its perigee location is the same. Thus, the coverage analysis method correctly places the satellite at the start of the simulation, not at the perigee, but thirty degrees true anomaly further.

The next coverage plot, Figure 6.2c, shows a number of important factors of the coverage analysis. In Earth observation, the repeating ground track or sun synchronous orbit is very often used. Thus, it was of great importance that the coverage analysis was able to correctly evaluated this orbit type. Further, the coverage also shows more than one orbit of the satellite. Instead of performing the algorithm for each orbit, which is very time consuming, grid points are copied from the first orbit and shifted according to the rotation of Earth. This coverage plot shows that this time saving method works correctly and is viable as a time saving method. By plotting multiple orbits, the coverage plot also shows the repeating ground track nature of the satellite, as the end of the last orbit connects back to the start of the first orbit. Additionally, this coverage plot also confirms the coverage analysis to work for inclinations exceeding 90 degrees, as this particular orbit is sun synchronous.

The last coverage plot is that of the Molniya orbit (Figure 6.2d). The Molniya orbit is a very specific orbit case where the ground track of a satellite makes a loop over its specific target. Correct implementation of this orbit proves that the coverage analysis can handle different arguments of perigee, as this Molniya orbit has an argument of perigee of 270 degrees, as well as extreme eccentricity. What makes the Molniya orbit special is that grid points can be observed more than once during one orbit due to the loop in its ground track. To handle this, the algorithm categorizes the grid points in ascending and descending points. Points that are observed in both will thus be evaluated twice. A limitation is that the algorithm does not evaluate points more than three times, even though points within the loop might be observed three or more times in a real scenario. Additionally, the Molniya orbit showed another limitation of the coverage analysis method. In the case that many points are observed by the satellite, such as a high altitude orbit with a wide field of view, the algorithm becomes very slow. Although these scenarios are not often analyzed, caution should be taken.

6.2. Revisit time

When using the coverage analysis on a scenario with multiple orbits, either by extending the scenario length or adding additional satellites, the revisit time can be computed. The analysis can provide the maximum and average revisit time for both singular grid points as well as global values. A multitude of scenarios were explored to examine the global revisit time results of the coverage analysis. The results are shown in Table 6.2. It must be noted that in order to compute the revisit time for a grid point, it needs to be visited at least twice. Thus, points that are only visited once are not accounted for in either the maximum or average revisit time computation. Further, a nonlinear grid was used, were points closer to the poles were spaced further apart in longitude than points closer to the equator.

Scenario	Orbital elements	Maximum revisit time [s]	Average revisit time [s]
One satellite, ten orbits	$a = 10000$km $e = 0$ $i = 30°$ $fov = 15°$	90015.1	34380.64
One satellite, ten orbits	$a = 15000$km $e = 0$ $i = 30°$ $fov = 15°$	163054.3	61122.23
Four equally TA-spaced satellites, five orbits	$a = 10000$km $e = 0$ $i = 30°$ $fov = 15°$	45204.03	5936.435
One satellite, one Molniya orbit	Molniya $fov = 5°$	38521.46	32034.79

Table 6.1: Maximum and average global revisit times for different scenarios.

The first two scenarios in the table show the results for the same satellite with a change in semi-major axis. As a higher orbit has a longer orbital period, it can be expected that the revisit time would be longer. This is indeed the case, as the higher semi-major axis scenario has almost double the revisit

time of the lower scenario. The third scenario analyses a constellation of equally spaced satellites. here it can be expected that the average revisit time is low, as the satellites fly in a train-like constellation. Table 6.2 shows that this is exactly what the coverage analysis found. The average revisit time of the constellation is very low. Finally, the fourth scenario analyses a Molniya orbit. Although only one orbit was analyzed, a revisit time was found. This is due to the loop in the Molniya orbit, which causes the satellite to visit certain grid points multiple times within a singular orbit.

6.3. Verification and validation

To verify the results of the coverage analysis, a similar analysis was made using STK 12. Both the coverage results and the revisit time results can be compared. Figure 6.3 shows a visual comparison between the two coverage analyses. It can be seen that the results are very similar. By performing a comparison test in Python, the exact difference between the results can be computed. Three main factors were analyzed: the (normalized) difference in the visit times between the coverage analyses, the percentage of points that were present in of the analyses but not in the other, and the number of visited points for which the amount of visits differed between the two analyses. These results are summarized in Table 6.2. For the difference in visit times to values are given. In general, the normalized difference is checked according to:

$$diff_n = \frac{t_{cvg} - t_{stk}}{t_{stk}} \tag{6.3}$$

However, when the value of the visit time is small, this normalized difference is not a good indicator anymore. In this case the absolute difference is evaluated. Thus, the first value in the table represents the normalized difference limit, while the second value represents the absolute limit for values not adhering to the normalized difference limit.

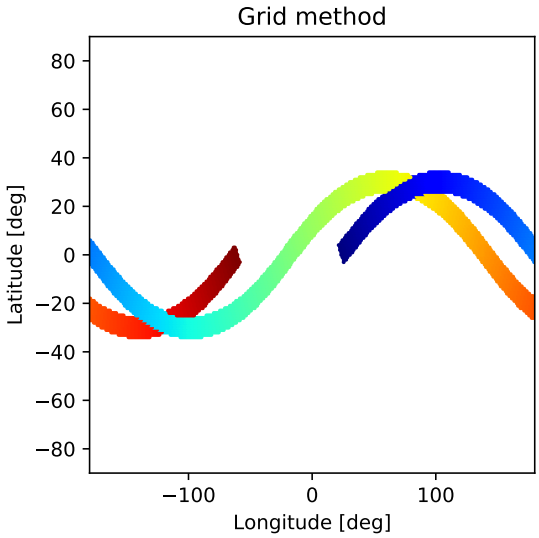
Scenario	Orbital elements	Difference in visit time [%, s]	Not in both [%]	Unequal visits [%]
One satellite, two orbits	$a = 10000$km $e = 0$ $i = 30°$ $fov = 15°$	<0.5, <17	5.38	0.570

Table 6.2: Verification test results.

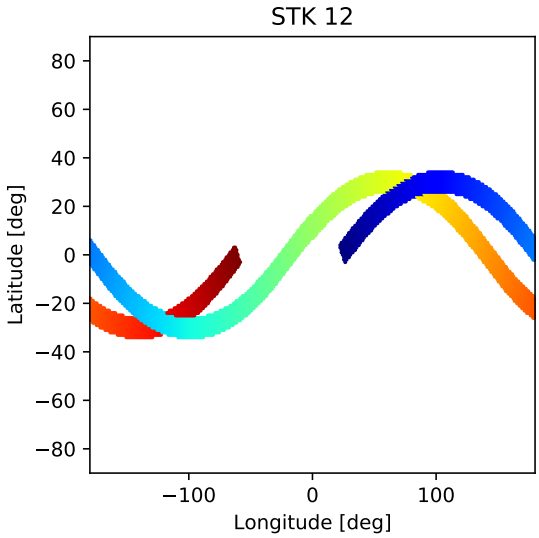
6.4. Computation time

One of the most important aspects of the novel coverage analysis method is that it should be as fast as possible. The computation speed was tested for a multitude of scenarios. These results are shown in Table 6.3. These results show the influence of various levels of multiprocessing. It can also be seen that the specific scenarios have an impact on the computation speed of the analysis. In general, the more points the satellite visits, the longer the computation time. Thus, when the scenario includes either a high altitude, a wide field of view, or both, the computation time can increase drastically. The results of the STK 12 coverage analysis are also shown in the table. It should be noted that multiprocessing was not applied to the STK 12 analysis. Further, the comparison is not completely fair due to the large amount of different, non-related computations that STK 12 makes during its analysis. Still, it helps put the coverage analysis computation into perspective.

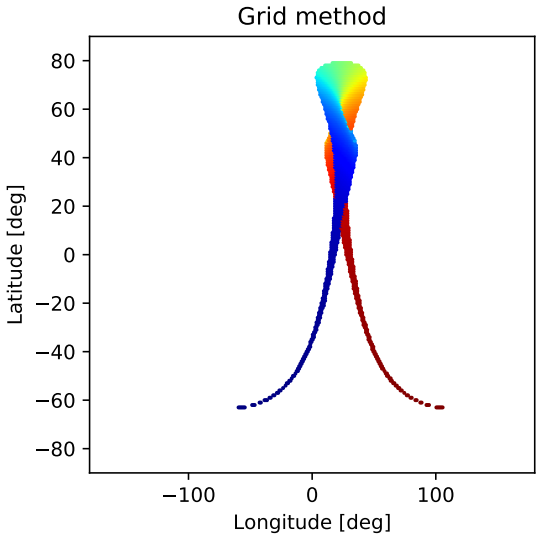
The results clearly show the impact of multiprocessing. Using eleven processors can improve the single processor computation time by two to five times. Furthermore, comparing the results to those from STK 12, the impact of the orbit copying is noticed. The coverage analysis method is able to scale much better with additional orbits than STK 12. Finally, the impact of a higher altitude is also seen. By increasing the semi-major axis from 10000 km to 15000 km the computation time more than doubled for the case with eleven processors and more than tripled for the single processor computation.



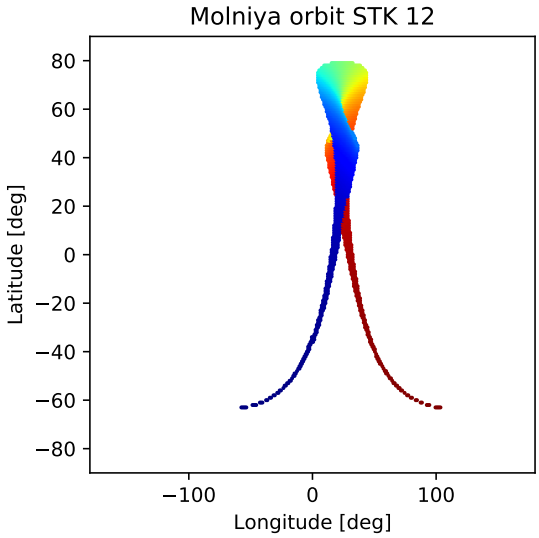
(a) Coverage for inclined orbit. $a = 10000$ km, $e = 0$, $i = 30°$, $fov = 15°$



(b) STK 12 coverage for inclined orbit. $a = 10000$ km, $e = 0$, $i = 30°$, $fov = 15°$



(c) Coverage for Molniya orbit. $a = 26600$ km, $e = 0.74$, $i = 63.4°$, $\Omega = 0°$, $\omega = 270°$ $fov = 5°$



(d) STK 12 coverage for Molniya orbit. $a = 26600$ km, $e = 0.74$, $i = 63.4°$, $\Omega = 0°$, $\omega = 270°$ $fov = 5°$

Figure 6.3: Coverage plots for the grid method and STK 12.

Scenario	Orbital elements	1P [s]	5P [s]	11P [s]	STK 12 [s]
One satellite, one orbit	$a = 10000$km $e = 0$ $i = 30°$ $fov = 15°$	35.548	12.711	9.740	64.152
One satellite, ten orbits	$a = 10000$km $e = 0$ $i = 30°$ $fov = 15°$	36.424	14.483	11.246	147.29
One satellite, ten orbits	$a = 15000$km $e = 0$ $i = 30°$ $fov = 15°$	122.553	39.371	27.738	220.47
One satellite, 100 orbits	$a = 10000$km $e = 0$ $i = 30°$ $fov = 15°$	55.580	32.373	27.919	734.48
One satellite, one Molniya orbit	Molniya $fov = 5°$	87.384	27.757	19.377	165.308

Table 6.3: Computation time results.

7

Constellation optimization

In this chapter the results of the constellation optimization are provided. Following the establishment of the coverage analysis method, effort can now be made to optimize the constellation of satellites. To create the best optimization tool, a number of comparisons have to be made. First, the tuning parameters for each optimization algorithm are compared. This allows for the selection of the best settings for each algorithm. Then a comparison is made between the different optimization algorithms to find the best performing algorithm for constellation optimization. To allow fair comparison, the seed is kept the same for each optimization. Further, the scenario variables are the same and described in Table 7.1. It should be noted that the number of satellites restraint is not used for variable-length GA and NSGA-II optimization, as these algorithms also optimize for the number of satellites.

Parameter	Value	Unit
Scenario length	200000	s
# of satellites	4	-
Field of view	15	deg
Semi-major axis	6800-8000	km
Eccentricity	0-0.2	-
Inclination	0-180	deg
RAAN	0-360	deg
Argument of perigee	0-360	deg
True anomaly	0-360	deg

Table 7.1: Scenario variables for algorithm comparisons.

7.1. Comparison of tuning parameters

In this section a comparison is made between different tuning parameters for each optimization algorithm. The parameters are compared by their convergence during the optimization. The convergence is visualized as the best objective value found per iteration. The comparison is evaluated based on the speed of convergence and the final result. To make sure the comparison is fair, all other settings should be the same between the optimizations. For each algorithm, the standard settings are shown. Furthermore, it was decided that each algorithm should perform around 1000 function evaluations, based on the combination of population size and iterations. For the single objective optimizations the weight on the revisit time was set to one, while the weight on the coverage was set to two.

7.1.1. Fixed-length genetic algorithm

Various settings were compared for the genetic algorithm using fixed-length individuals, as described in subsection 5.3.1. The standard settings are stated in Table 7.2. Further, for each optimization the chance of crossover was set to 0.5 and the chance of mutation to 0.2. For the elitist tournament selection, the top two individuals were kept as elites.

Selection	Crossover	Mutation	Population	Generations
Tournament	One-point	Gaussian	40	25

Table 7.2: Standard settings for GA with fixed-length individuals.

First, the different selection procedures are compared. The three methods that were compared were tournament selection, roulette selection and tournament selection with elitism.Figure 7.1 shows the convergence of the objective value for the various selection methods.

All three selection methods show very similar convergence behaviour. During the early generations, the algorithms using elitist and non-elitist tournament selection outperform roulette selection. This can be explained by the fact that in tournament selection, especially elitist tournament selection, better individuals are often selected. On the contrary, although there is a higher chance for more fit individuals to be selected in roulette selection, there is still a chance that bad performing individuals are also included. Interestingly, after eleven generations the roulette selection optimization catches up to the two tournament selection algorithms. Further, after sixteen generations the roulette selection is able to find a better solution than the elitist tournament selection. After twenty-five generations, non-elitist tournament selection has found the most optimal individual, closely followed by the roulette selection. Elitist tournament selection performs the worst by converging to the least fit individual out of the three optimizations. These results show that a more exploration focused selection method is able to find better solutions, meaning that the solution space is quite non-linear with many local optima. More elitist selection methods get stuck in these local optima and end up performing worse than other methods.

Next, three different crossover methods are compared. These are one-point crossover, two-point crossover and blend crossover. In Figure 7.2 the convergence of the objective value is shown for the different crossover methods. Using the blend crossover method, the optimizer was able to immediately find a substantially better solution after one generation than the other two optimizations. After ten generations, all three optimizations stagnate, with blend crossover still providing the best solution. However, after eighteen generations one-point crossover overtakes blend crossover, and is able to steadily find better solutions. A couple generations later, two-point crossover also finds a more optimal solution than blend crossover. Seemingly, blend crossover converged prematurely and was not able to find a better solution after the initial ten generations. In contrast, the other two are able to keep optimizing the solution throughout the optimization.

Finally, two mutation methods are also compared, namely Gaussian mutation and uniform mutation. The convergence of these optimizations is shown in Figure 7.3. It can be seen that Gaussian mutation performs better than uniform mutation for every generation of the optimization. This could be explained by the fact that Gaussian mutation stays more true to the unmutated individual compared to uniform mutation. This allows good individuals to not be mutated to drastically, retaining the good genes within the population.

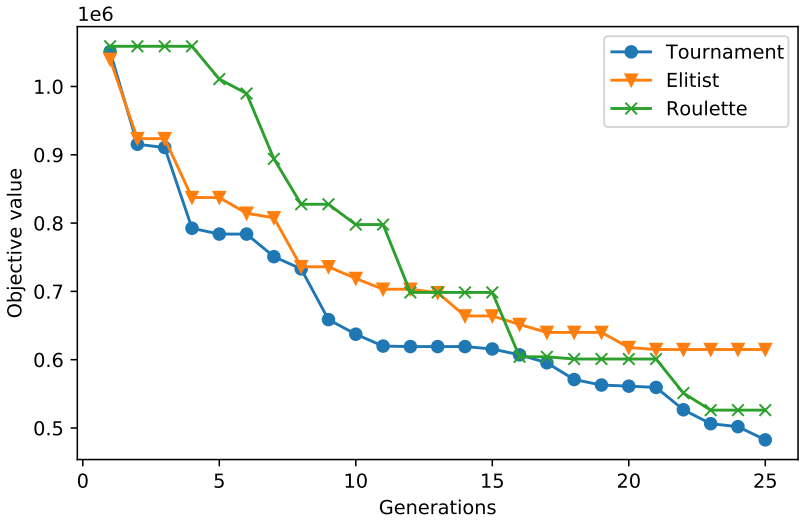


Figure 7.1: Comparison of selection methods for GA with fixed-length individuals.

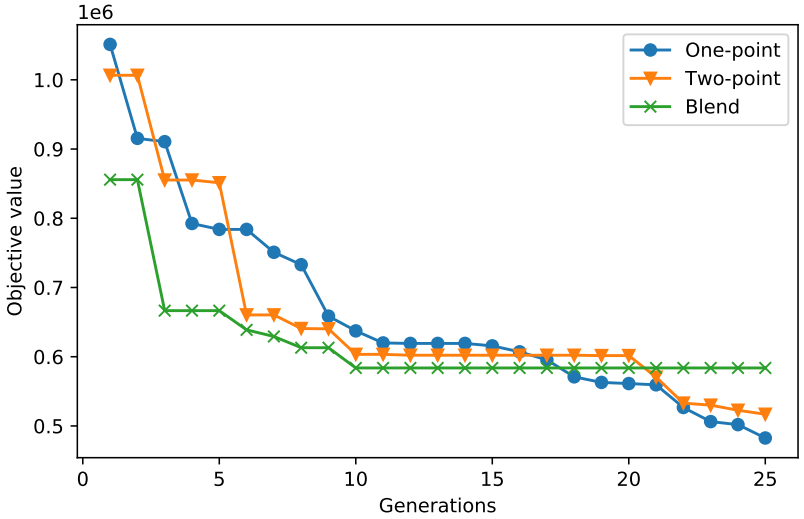


Figure 7.2: Comparison of crossover methods for GA with fixed-length individuals.

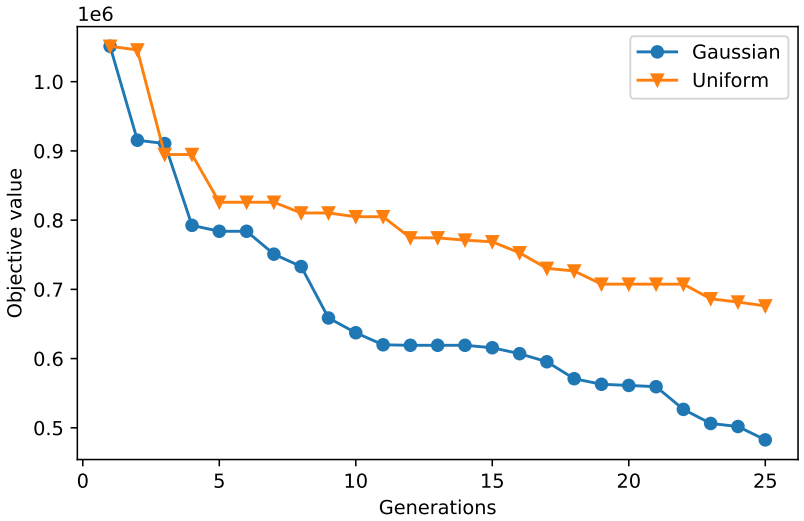


Figure 7.3: Comparison of mutation methods for GA with fixed-length individuals.

From these comparisons, the final settings for the fixed-length GA can be chosen. The configuration includes elitist tournament selection, blend crossover and Gaussian mutation. These are the settings that will be used for the fixed-length GA during the comparison with the different algorithms (Table 7.3).

Selection	Crossover	Mutation
Tournament	One-point	Gaussian

Table 7.3: Final settings for GA with fixed-length individuals.

These final settings are used to further analyze the impact of the population versus generations ratio. The number of function evaluations was kept the same at one thousand. Three different ratios were compared, a population of 25 for 40 generations, a population of 40 for 25 generations, and a population of 100 for ten generations. The convergence for these optimizations is shown in Figure 7.4. Comparing the convergence of the three optimizations, it can be seen that the higher the population, the better the initial best solution and the faster the convergence. This is expected as a higher population allows for a better exploration of the solution space during every generation. It can further be noted that the 100/10 optimization is still in the process of converging when the optimization was ended. On the contrary, the other two optimizations seemed to have stagnated at the end of the optimization. The final result and computation time for each optimization is shown in Table 7.4. An interesting pattern can be seen in the computation time, the higher the population, the lower the computation time. This is explained by the crossover and mutation probability. These cause a little over half of the population to generate new offspring which needs to be evaluated. The rest of the population has already been evaluated, reducing the amount of function calls performed. This causes a slight difference in the amount of function calls between the different population sizes, where larger populations (with fewer generations) perform less function evaluations than smaller populations (with more generations). Comparing the final results of the optimizations, it is interesting to see how much worse the 40/25 optimization performed. Seemingly, it got stuck within a local optima and was not able to converge as well as the other two optimizations. The 100/10 optimization was chosen to be the best out of the three compared, due to its good final result, fast computation time, and the lack of stagnation.

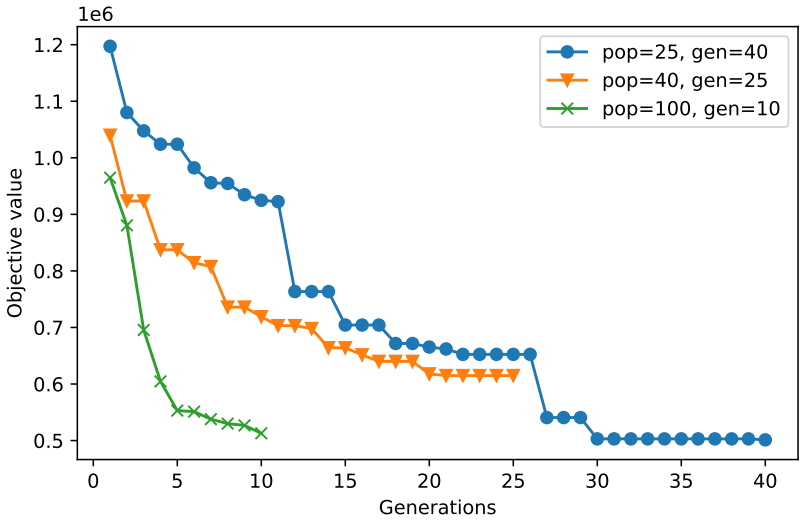


Figure 7.4: Comparison of different population versus generation ratios for GA with fixed-length individuals, non-elitist tournament selection, one-point crossover and Gaussian mutation.

Pop/gen ratio	Optimal cost function value	Computation time [s]
25/40	501234.2	9044.1
40/25	614883.1	8899.7
100/10	512810.6	8628.5

Table 7.4: Final result and computation time for different population versus generation ratios of GA with fixed-length individuals, non-elitist tournament selection, one-point crossover and Gaussian mutation.

7.1.2. Simulated annealing (SA)

Two settings were tuned for the simulated annealing optimization, the initial temperature and the annealing scheme. The standard settings are shown in Table 7.5.

Initial temperature	Annealing scheme	Iterations
10E6	Exponential	1000

Table 7.5: Standard settings for SA optimization.

The magnitude of the initial temperature is related to the magnitude of the objective values that are commonly obtained by the fitness function. As the order of magnitude of the objective is typically 10E5, the initial temperatures that were compared were chosen to be 10E4, 10E5 and 10E6. The convergence of the three initial temperatures can be compared in Figure 7.5. In the early generations, all three optimizations perform similarly and are able to converge rapidly to better solutions. After around 100 generations, the optimization with an initial temperature of 10E4 plateaus. Due to its lower initial temperature, it does not allow to accept worse solutions, limiting the exploration of the algorithm. This causes it to get stuck in a local optima early on in the optimization. The algorithms with an initial temperature of 10E5 and 10E6 are able to converge further, albeit slower than in the first 100 generations. The optimization with an initial temperature of 10E6 is able to consistently outperform the other two optimizations. The fact that the highest tested initial temperature performs best indicates that the solution space of the problem is quite nonlinear, with many local optima in which the optimization can get stuck. Next, the annealing scheme was compared. Two schemes were chosen for the comparison, namely an exponential annealing scheme and a linear scheme. In Figure 7.6 it can be seen that both converge rapidly during the early generations.

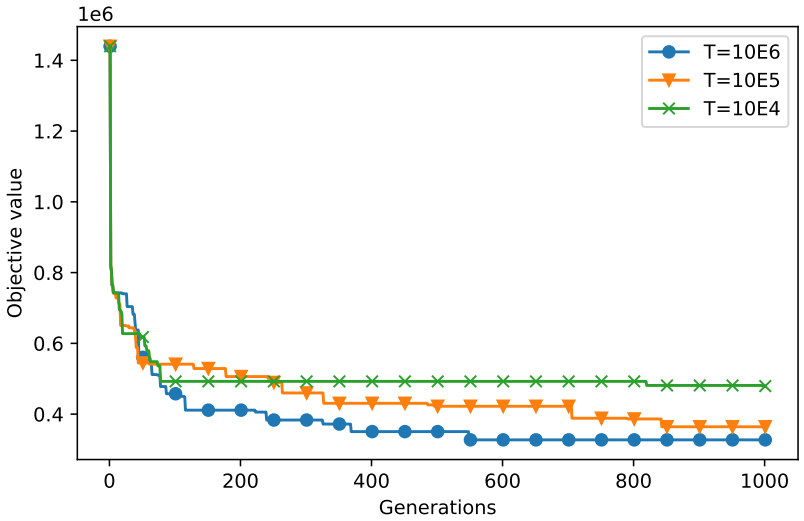


Figure 7.5: Comparison of initial temperatures for SA optimization.

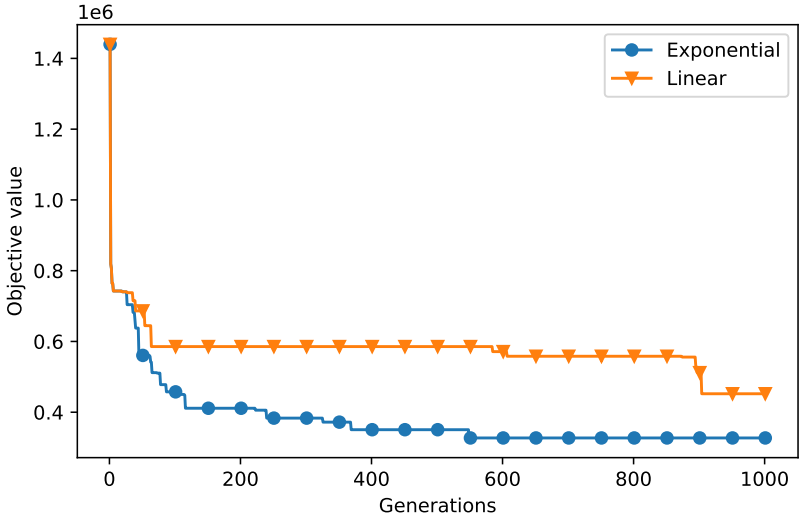


Figure 7.6: Comparison of annealing schemes for SA optimization.

7.1.3. Particle swarm optimization (PSO)

For particle swarm optimization, only one setting was evaluated. This was the implementation of the inertia. Two methods were evaluated namely a constant inertia and a random inertia. Other settings of the PSO algorithm were kept the same for both optimizations. These settings are shown in Table 7.6.

Cognitive bias	Social bias	Velocity bound	Population	Generations
2.0	2.0	0.2	40	25

Table 7.6: Standard settings for particle swarm optimization.

For the constant inertia, the inertia was kept at 0.5 for every generation. For random inertia, the inertia is a random fraction between 0.5 and one. The convergence of both optimizations is shown in Figure 7.7. It can be seen that both methods behave rather similar, following similar rates of convergence. With little variance between the two optimizations, the constant inertia method was chosen due to its

simplicity. This optimization was then used to analyze the impact of the population size and number of generations. Three variants were compared, a population of 25 for 40 generations, a population of 40 for 25 generations, and a population of 100 for ten generations. The convergence of these three optimization can be seen in Figure 7.8. All three optimizations behave quite similarly. The optimizations with a larger population size converge slightly faster than the smaller population optimization. However, all optimizations seem to converge to similar results. Although the 25/40 optimization converges the slowest, Table 7.7 shows that it is able to find the best solution in the least amount of time. Therefore, the 25/40 optimization is selected as the best setting for PSO.

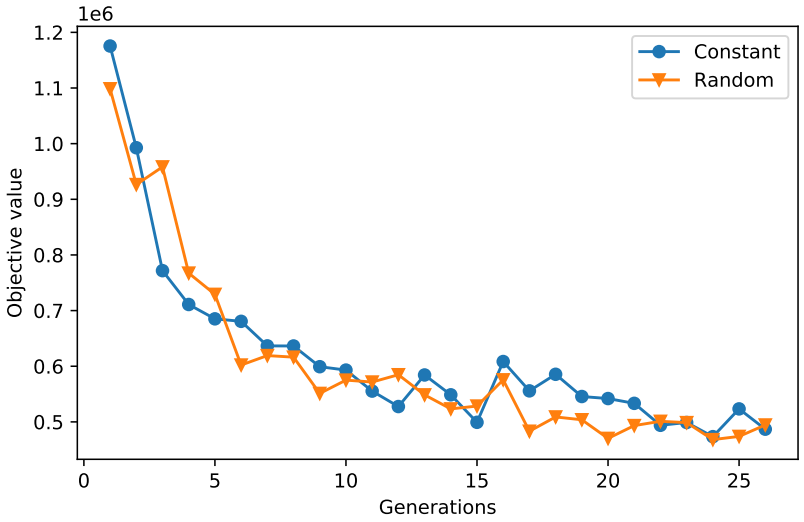


Figure 7.7: Comparison of inertia methods for PSO.

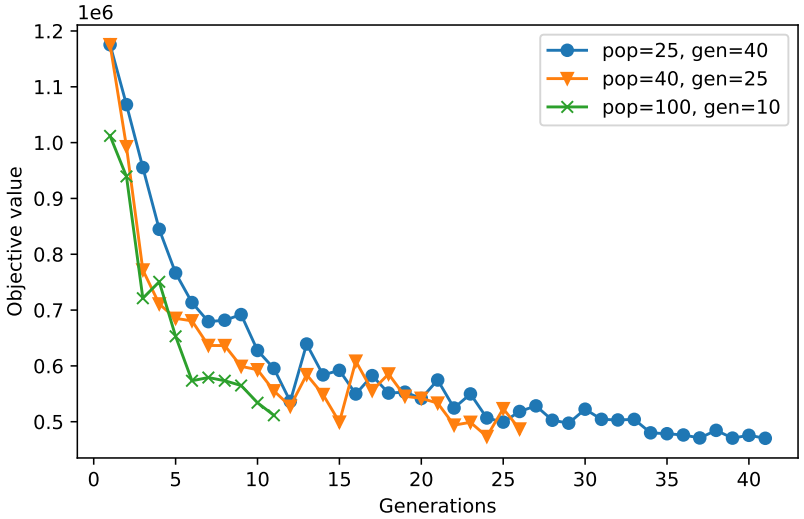


Figure 7.8: Comparison of different population versus generation ratios for PSO with constant inertia.

Pop/gen ratio	Optimal cost function value	Computation time [s]
25/40	470316.3	14859
40/25	473477.5	18577
100/10	511445.3	15655

Table 7.7: Final result and computation time for different population versus generation ratios of PSO using constant inertia.

7.1.4. Differential evolution (DE)

For the differential evolution algorithm, different strategies were compared. These were best1bin, rand1bin and rand1exp. The popsize and number of generations were kept the same between the different optimizations, and are shown in Table 7.8. Popsize is a parameter used to determine the size of the population in differential evolution, along with the amount of decision variables in an individual. Thus, when using four satellites, meaning the individual has a length of 24, the total population is popsize times 24.

Popsize	Generations
2	20

Table 7.8: Standard settings for DE optimization.

The convergence of the three strategies can be seen in Figure 7.9. Here, it is observed that the best1bin strategy outperforms the other two strategies throughout the whole optimization. Rand1bin and rand1exp perform very similar, alternating which performed better throughout the optimization. The main difference between best1bin and the other two strategies is in the mutation step. In best1bin, the best individual of the population is mutated, compared to a random individual in rand1bin and rand1exp. Thus, best1bin introduces a form of elitism into the algorithm. This elitism seemingly enables the optimization to find better solutions than the non-elitist rand1bin and rand1exp. This gives some insight in the solution space of the problem. It hints towards some form of linearity in the solutions, as a mutated individual that lies closer to the best individual is more often found to be a better solution than when a random individual is mutated.

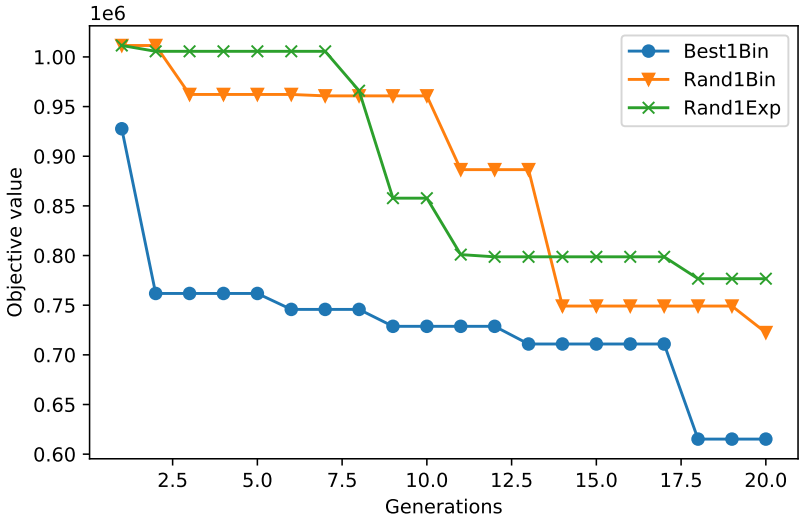


Figure 7.9: Comparison of strategies for DE optimization.

The best1bin optimization was used to analyze the impact of varying the population size and number of generations. Three different optimizations were compared, a popsize of one for 41 generations, a popsize of two for twenty generations, and a popsize of three for thirteen generations. The amount of generations were chosen to roughly equate to 1000 function evaluations. Figure 7.10 shows the convergence of the three optimizations. One would expect a larger population to find a better initial solution in the first generation, and to converge faster than an optimization with a smaller population. Interestingly, the optimization with a popsize of three does not find an initial solution better than the 2/20 optimization. It does, however, converge faster and overtakes the 2/20 optimization after seven generations. Comparing the 1/41 and 2/20 optimizations, it can be seen that the larger population optimization does find a better initial solution. However, the rate of convergence seems to be very similar between the two optimizations. Due to the much better initial solution, the 2/20 optimization is able

to keep finding better solutions than the 1/41 optimization. It should be noted that all three optimizations do not show significant signs of stagnation during their convergence, indicating that with more generations they might be able to converge to even better solutions. Table 7.9 shows the final result and computation time of the different optimizations. It can be seen that the 2/20 optimization is able to find the most optimal solution. Further, it can be shown that the higher the popsize/generation ratio, the lower the computation time. This can be explained due to the greedy selection in the differential evolution algorithm. If an offspring individual performs worse than the parent individual, the parent is selected for the next generation. As the parent has already been evaluated, it does not need to be evaluated again later on, reducing the amount of function evaluations. This means that in larger populations, the fraction of function evaluations per generation is less than that of smaller populations, lowering the overall computation time. Between the 2/20 optimization and the 3/13 optimization, 2/20 is chosen as the better setting. This is due to the optimization finding the best solution, while not being much slower than the 3/13 optimization,

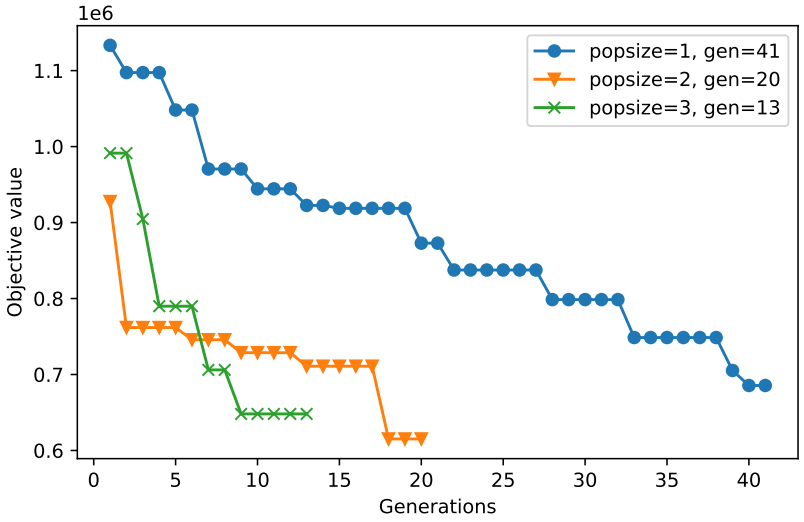


Figure 7.10: Comparison of different popsize versus generation ratios for DE using the best1bin strategy.

Popsize/gen ratio	Optimal cost function value	Computation time [s]
1/41	685541.0	15524
2/20	615165.7	13987
3/13	648109.3	13517

Table 7.9: Final result and computation time for different popsize versus generation ratios of DE optimization using the best1bin strategy.

7.1.5. Covariance matrix adaptation evolution strategy (CMA-ES)

In covariance matrix adaptation evolution strategy the choice is parameters has to be made very carefully as many combinations can lead to a bad optimization. Therefore it was decided to keep all tuning parameters of the algorithm to the default settings as described in DEAP. Instead, the impact of the population size and number of generations was investigated. Four different population/generation ratios were compared. These ratios were chosen to keep the number of function evaluations as close to 1000 as possible. The default ratio with population of thirteen for 77 generations, a population of 25 for 40 generations, a population of 40 for 25 generations, and a population of 100 for ten generations. The convergence for these four optimizations is shown in Figure 7.11. It can be seen that all four optimizations follow a similar convergence rate. The main difference between the optimizations lies in the initial found solution. Seemingly, the higher the population size, the better solution is found. In Table 7.10 the final result and computation time for the optimizations is shown. A clear trend is visible, the higher the λ/generation ratio, the lower the computation time, but the worse the solution. An outlier is the 25/40

optimization, which was able to find a better solution than the 13/77 optimization. This motivated the choice to select the 25/40 ratio as the best.

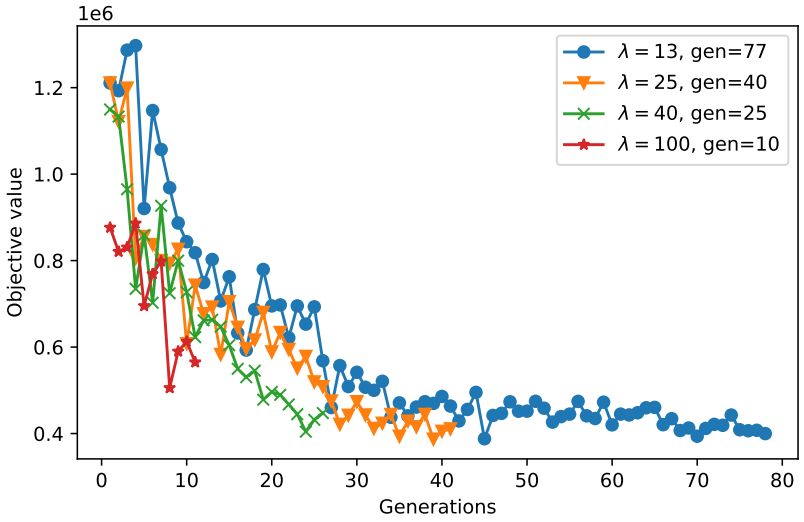


Figure 7.11: Comparison of different population versus generation ratios for CMA-ES optimization.

λ/gen ratio	Optimal cost function value	Computation time [s]
13/77	388015.8	18681
25/40	386894.8	14494
40/25	404369.5	13656
100/10	505464.3	12877

Table 7.10: Final result and computation time for different population size λ versus generation ratios of CMA-ES optimization.

7.1.6. Non-dominated sorting genetic algorithm (NSGA-II)

The settings that can be tuned in the non-dominated sorting genetic algorithm are very similar to those of the standard genetic algorithm. However, due to limitations of the DEAP package, only one option exists for the selection method, aptly called deap.tools.selNSGA2. The crossover and mutation were able to be compared. All four combinations of these settings were tested. The standard settings of this algorithm include the population size and amount of generations. Furthermore, as the amount of satellites is also compared, this is also part of the standard settings. These settings are shown in Table 7.11. Further, like the standard GA, the crossover probability is set to 0.5 and the mutation probability is set to 0.2.

Amount of satellites	**Population**	**Generations**
1-8	40	25

Table 7.11: Standard settings for NSGA-II algorithm.

Two crossover methods and two mutation methods were compared. For crossover, one-point and two-point crossover were compared, while for mutation Gaussian and uniform were compared. The Pareto fronts of the four combinations of settings are shown in Figure 7.12. It can be seen that all four optimizations result in similar Pareto fronts. An interesting observation is that none of the four optimizations found first front solutions for every amount of satellites possible. Two-point Gaussian optimization found a first front solution for five satellites but not for four, the other three vice versa. Further, none of the optimizations found a first front solution for eight satellites. This means that for certain numbers of satellites the algorithm is not able to find non-dominated solutions within the amount

of generations. As the Pareto fronts are very similar, the decision on the settings was made on the computation time, shown in Table 7.12. It can be seen that the algorithm using two-point crossover and uniform mutation was much quicker than the other three, thus it was selected as the final settings.

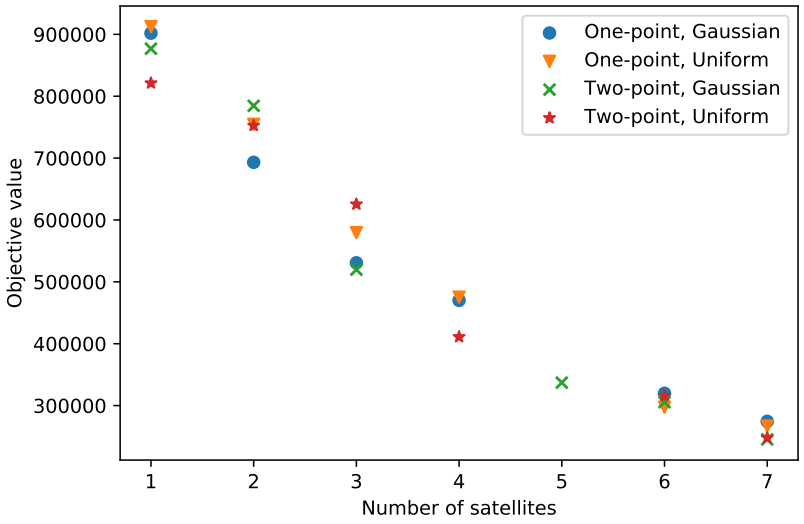


Figure 7.12: Comparison of different settings for NSGA-II optimization.

Settings	Computation time [s]
One-point, Gaussian	10376
One-point, Uniform	12343
Two-point, Gaussian	9727.4
Two-point, Uniform	7759.8

Table 7.12: Computation times for different settings for NSGA-II optimization

Next, the impact of the population size and number of generations was also analyzed using the finals settings. Three different ratios were compared, each translating to around 1000 function evaluations. The Pareto fronts of these optimizations are shown in Figure 7.13. The first thing to notice is the fact that the 25/40 optimization was able to find Pareto optimal solutions for every number of satellites. On the contrary, 100/10 optimization is missing one (at eight satellites) and 40/25 optimization is missing two (at five and eight satellites). Further, the 100/10 optimization is not able to find comparably good solutions at three to six satellites compared to the other two optimizations. The computation times are also compared and shown in Table 7.13. It can be seen that the higher the population versus generation ratio, the lower the computation time. Although 25/40 has the longest computation time, it was still selected as the best setting due to its ability to find Pareto optimal solutions for every number of satellites, and due to its good quality of found solutions compared to the other optimizations.

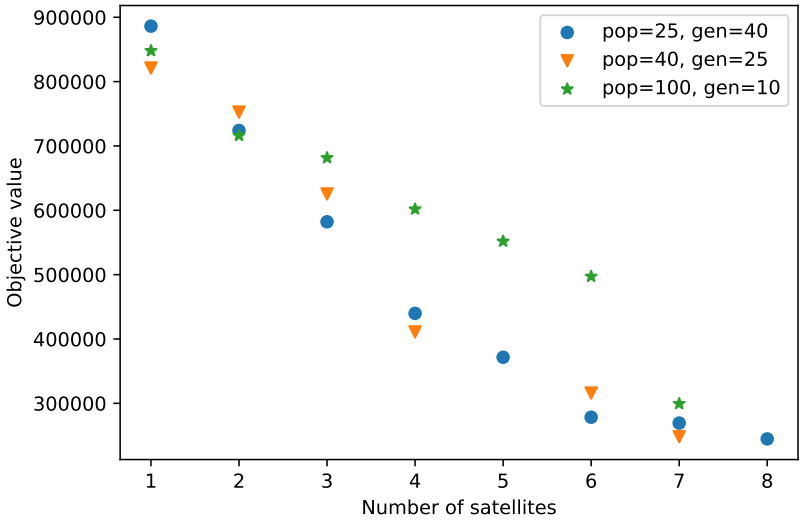


Figure 7.13: Comparison of different population versus generations ratios for NSGA2 optimization using two-point crossover and uniform mutation.

Pop/gen ratio	Computation time [s]
25/40	9536.3
40/25	7759.8
100/10	7576.0

Table 7.13: Computation times for different population versus generation ratios for NSGA-II optimization using two-point crossover and uniform mutation.

7.1.7. Variable-length genetic algorithm

Similarly to the fixed-length genetic algorithm, the settings of the variable-length genetic algorithm for selection, crossover and mutation can be changed. The standard settings of the optimization can be found in Table 7.14. For each optimization the crossover and mutation probability are set to 0.5 and 0.2 respectively. As the number of satellites is also a variable, the standard settings also includes the possible number of satellites. The cost function of the optimization includes a weight for the number of satellites, which was set to one.

Selection	Crossover	Mutation	Amount of satellites	Population	Generations
Tournament	One-point	Gaussian	1-8	40	25

Table 7.14: Standard settings for GA with variable-length individuals.

Three different selection methods were compared, namely tournament selection, roulette selection and tournament selection with two elites. Figure 7.14 shows the convergence for each of the methods. It can be seen that roulette selection performs much worse than either tournament selection method. In the early generations, elitist tournament selection slightly outperformed the non-elitist selection. However, the elitist selection plateaued already around seven generations. Non-elitist tournament selection was able to continue converging and outperforms elitist selection after eight generations. Seemingly, the increased exploration of the non-elitist tournament selection helps the algorithm to climb out of local optima to find more globally better solutions. However, increasing the exploration too much, such as with the roulette selection, leads to even worse results than the elitist tournament selection.

For the crossover phase, two methods were compared, one-point crossover and two-point crossover. In Figure 7.15 it can be seen that one-point crossover outperforms two-point crossover in the early

generations. However, two-point crossover is able to find a better solution after five generations and outperforms one-point crossover throughout the optimization.

Finally, the performance of Gaussian and uniform mutation were compared. Figure 7.16 shows the convergence of both optimizations. In early generations, Gaussian mutation greatly outperforms uniform mutation, converging to a better solution quickly. However, after thirteen generations, the Gaussian mutation optimization stalls, and is overtaken by the uniform mutation after eighteen generations.

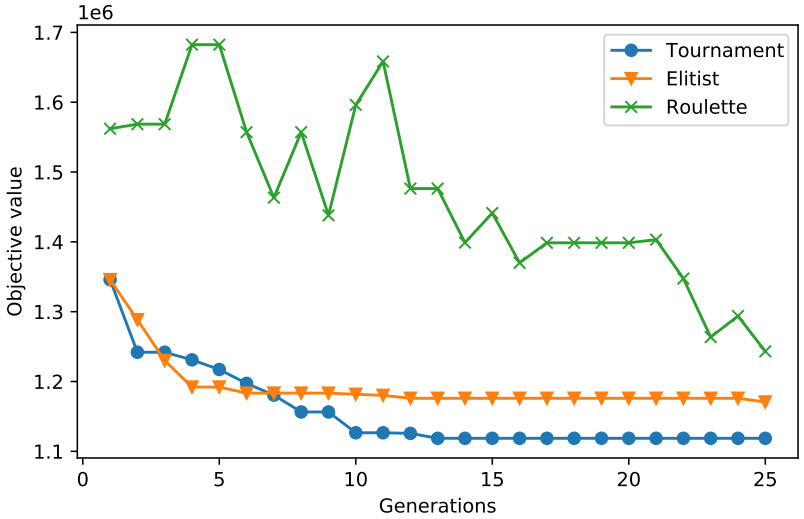


Figure 7.14: Comparison of selection methods for GA with variable-length individuals.

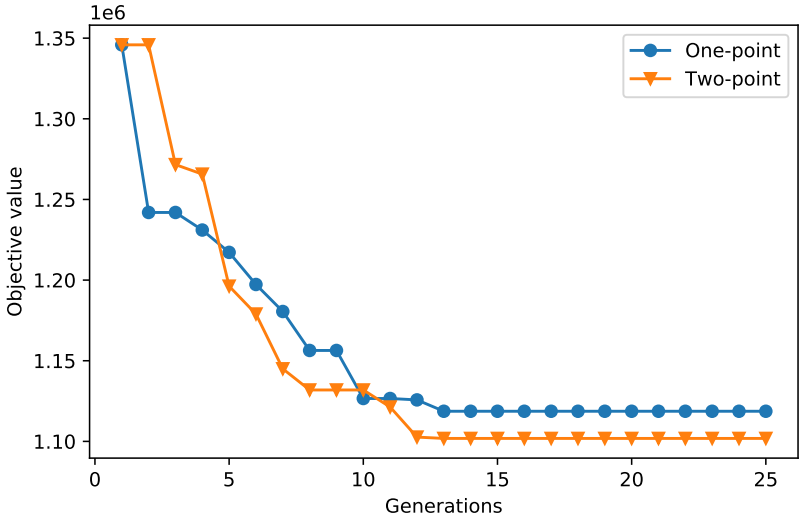


Figure 7.15: Comparison of crossover methods for GA with variable-length individuals.

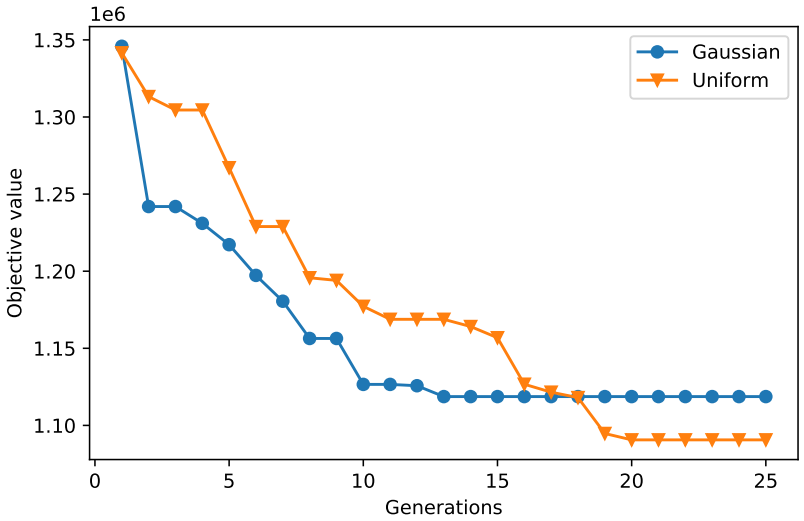


Figure 7.16: Comparison of mutation methods for GA with variable-length individuals.

The final tuning parameters are chosen to be non-elitist tournament selection, two-point crossover and uniform mutation (TourTwoUni). These settings were compared with the best-performing optimization from during the comparison, namely tournament, one-point, uniform optimization (TourOneUni). Their convergence is shown in Figure 7.17. It can be seen that the combination of best performing settings in fact does not perform better than the existing TourOneUni optimization. Thus, the final settings are instead non-elitist tournament selection, one-point crossover and uniform mutation.

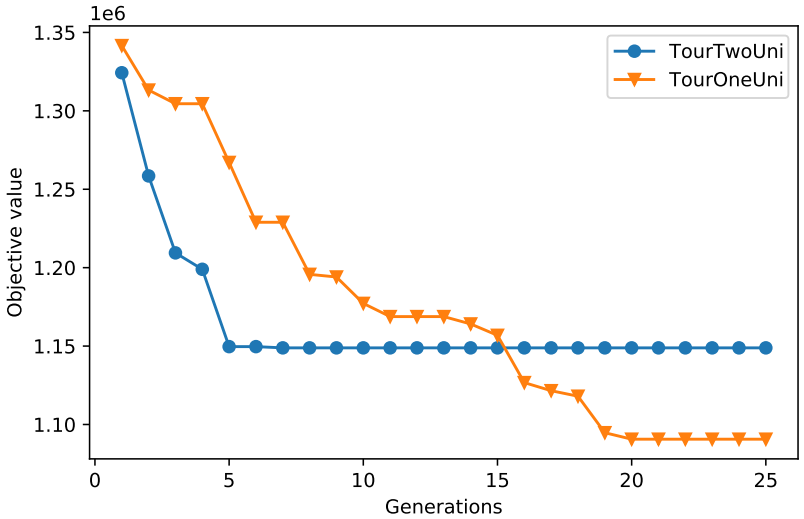


Figure 7.17: Comparison of tournament, one-point, uniform optimization and tournament, two-point, uniform optimization.

The final settings were used to analyze the impact of the population size and number of generations. Three different population to generation ratios were compared, making sure to keep the number of function evaluations similar. The convergence can be seen in Figure 7.18. It can be seen that the lower population of the 25/40 optimization results in a slower rate of convergence compared to the other two optimizations. The 40/25 and 100/10 optimizations have a similar rate of convergence. Due to the higher amount of generations, the 40/25 optimization is able to outperform the other two optimizations and find the most optimal result. In Table 7.15 it can be seen that the computation time for the three optimizations is very similar. Interestingly, where one would expect to see the lower the population, the

higher the computation time, this is not true for this particular case. Instead, the 40/25 optimization took the longest to compute. As the difference is relatively small, it could be attributed to hardware particularities. As the computation times are so similar, and 40/25 is able to find the best solution, 40/25 is chosen as the best setting for the variable-length individual GA.

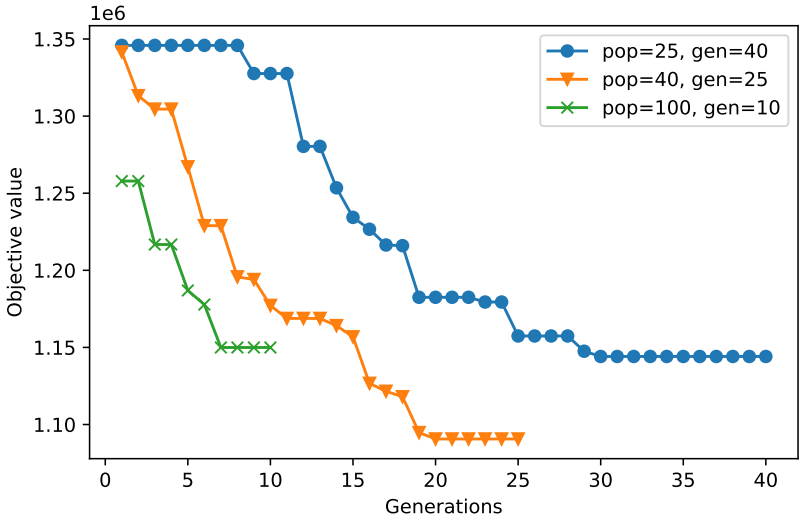


Figure 7.18: Comparison of different population versus generations ratios for GA with variable-length individuals, tournament selection, one-point crossover and uniform mutation.

Pop/gen ratio	Optimal cost function value	Computation time [s]
25/40	1144120	13332
40/25	1090620	13727
100/10	1149951	12951

Table 7.15: Final result and computation time for different population versus generations ratios of GA with variable-length individuals.

7.2. Comparison between optimization algorithms

With the best settings identified for each optimization algorithm, a comparison can be made to find the overall best performing algorithm. This comparison is first made between the algorithms using fixed-length individuals. Next, the results of the variable length GA are compared to the fixed-length algorithms. Finally, multi-objective optimization is compared with single-objective optimization.

7.2.1. Comparison of algorithms using fixed-length individuals

The convergence and results of the different algorithms are shown in Figure 7.19 and Table 7.16. Note that simulated annealing has been excluded from Figure 7.19 due to the much larger amount of generations used. Looking at the computation time, a clear winner emerges, namely the genetic algorithm. It is almost twice as fast as the next algorithm. Interestingly, PSO, DE and CMA-ES all perform very similar in terms of computation time. Finally, simulated annealing performs by far the worst in terms of speed. The fact that SA does not make use of a population, and thus does not save part of its solution to the next generation makes it very slow.

The algorithms can also be compared in terms of their best found solution. The objective function for each algorithm was as follows:

$$C = t_{rev} + 2(1 - f_c)10^6 \tag{7.1}$$

The coverage is scaled to have the same order of magnitude as the revisit time. Further, the coverage is weighted twice as highly as the revisit time, as it is more desired to have full coverage and worse

revisit time than a fast revisit time but bad coverage. Comparing the algorithms in terms of the best solution found results in a different ranking. Now CMA-ES clearly outperforms the other algorithms. This is due to the great coverage value it was able to find. In second place lies PSO, followed by GA and SA who found very similar results. In last place is DE, although it found the best revisit time, its solution had the worst coverage of all algorithms.

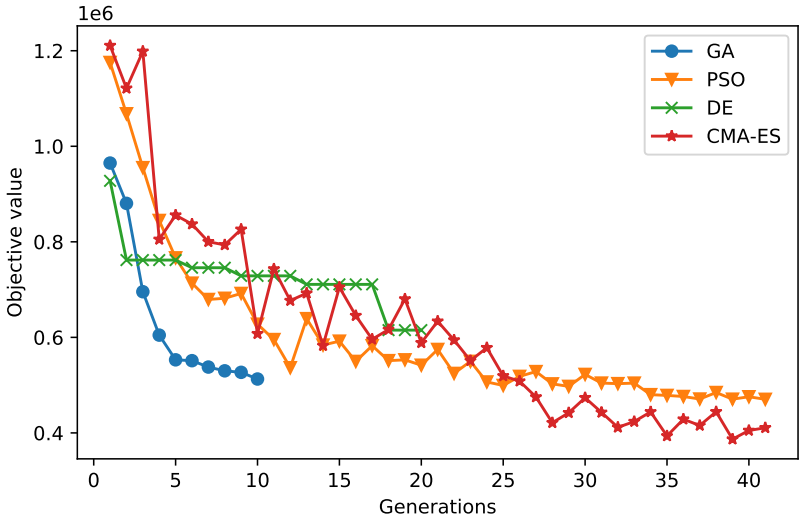


Figure 7.19: Comparison of convergence of algorithms using fixed-length individuals excluding simulated annealing.

Algorithm	Optimal cost function value	Max revisit time [s]	Coverage [%]	Computation time [s]
GA	512810.6	196674.5	84.19	8628.5
SA	513494.2	199736.5	84.31	39801
PSO	470316.3	198728.3	86.42	14859
DE	615165.7	192553.2	78.87	13987
CMA-ES	386894.8	196399.0	90.48	14494

Table 7.16: Final result and computation time for difference fixed-length algorithms.

In Figure 7.19, it can be seen that the genetic algorithm did not yet plateau at the end of its simulation. By increasing the amount of generations to match the computation time of the CMA-ES algorithm, it might be able to outperform it. The computation time for the genetic algorithm becomes similar to that of the CMA-ES algorithm when increasing the number of generations to 18. Figure 7.20 shows the convergence of the two optimizations and Table 7.17 shows the detailed results. It can be seen that the CMA-ES optimization still outperforms the GA optimization. Thus, it can be concluded that the CMA-ES optimization is the overall best algorithm for the fixed-length individual use case.

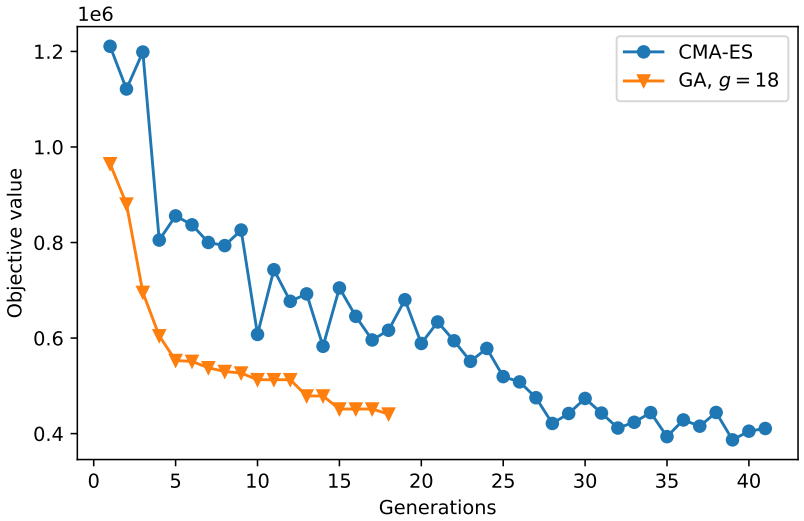


Figure 7.20: Comparison of convergence of the CMA-ES optimization and the genetic algorithm with 18 generations.

Algorithm	Optimal cost function value	Max revisit time [s]	Coverage [%]	Computation time [s]
GA, 18 generations	440924.2	196674.5	84.19	15087
CMA-ES	386894.8	196399.0	90.48	14494

Table 7.17: Final result and computation time for difference fixed-length algorithms.

7.2.2. Comparison of different optimization types

Two additional algorithms were used in the optimization, namely the variable-length individual genetic algorithm and the NSGA-II algorithm, i.e. the multi objective genetic algorithm. The NSGA-II optimization can also be seen as a variable-length individual optimization as it also optimizes the number of satellites. First the variable-length GA optimization is compared with the fixed-length GA, using the settings as described in section 7.1. The result can be seen in Table 7.18. As the result of the variable-length GA optimization is a constellation of seven satellites, it is to be expected that the revisit time and especially the coverage results are better than that of the fixed-length GA. Comparing the computation time, the fixed-length optimization is faster than the variable-length optimization. As the variable-length optimization converged to a larger constellation size, the evaluation of the individuals take longer, explaining this difference. However, this comparison is not quite fair, as the variable-length GA performs an extra optimization in the number of satellites. A better comparison on computation time would be when repeating the fixed-length GA for every possible number of satellites, which would take much longer than the variable-length GA. Thus, when the optimization of the number of satellites is required, the variable-length genetic algorithm is a great choice.

Algorithm	Number of satellites	Max revisit time [s]	Coverage [%]	Computation time [s]
Fixed-length GA	4	196674.5	84.19	8628.5
Variable-length GA	7	185871.1	99.41	13727

Table 7.18: Final result and computation time for fixed-length and variable-length GA optimization.

The results of the Pareto front found by the NSGA-II optimization are stated in Table 7.19. The higher weight on the coverage objective can clearly be seen in the results. The coverage objective steadily increases the more satellites are included, compared to the revisit time which remains largely constant. The results of the NSGA-II optimization can also be compared to both the variable-length GA (Table 7.18) and the best fixed-length optimization CMA-ES (Table 7.17). Firstly comparing with the

variable-length GA, the solution of seven satellites in the Pareto front is used. It can be seen that the variable-length GA does outperform the solution of the NSGA-II optimization. However, looking at the computation time of NSGA-II (Table 7.13), it is much faster. Next, the NSGA-II optimization can be compared with the CMA-ES optimization (Figure 7.20). Using the solution for four satellites, it can be seen that the NSGA-II optimization is again outperformed. However, its computation time is quite a bit lower than that of the CMA-ES optimization. All in all, the NSGA-II optimization can be useful when it is desired to check adequate solutions for all possible number of satellites.

Number of satellites	Max revisit time [s]	Coverage [%]
1	176617.6	29.48
2	181852.4	45.90
3	199383.2	75.59
4	201396.8	84.18
5	197873.8	92.50
6	197697.3	95.51
7	197385.4	96.84
8	186932.9	97.53

Table 7.19: Final results of the Pareto front of the NSGA-II optimization.

7.3. Verification and validation

The verification and validation of the various optimization algorithms is done in multiple ways. Firstly, the optimization scripts need to be verified to be correctly implemented. This can be done by using the script to work on a verified problem with a known solution. The Rastrigin problem was used to verify the optimizers. Secondly, the verification of the results is done by checking their sensibility and if they are in line with what can be expected. Thirdly, a sensitivity analysis was made to evaluate the behavior of the algorithms when presented with a slightly different scenario. Finally, the optimization can be validated by optimizing for an existing constellation and comparing the results.

7.3.1. Verification using the Rastrigin function

The Rastrigin function is a performance test function that can be used to verify implemented single objective optimization algorithms. It is a very non-convex function and is thus best solved by global optimization algorithms, as local search algorithms would get stuck in one of the many local minima. For n dimensions the function is as follows:

$$f(x) = An + \sum_{i=1}^{n} [x_i^2 - Acos(2\pi x_i)] \tag{7.2}$$

Typically, A is ten and x_i is constrained between -5.12 and 5.12. For two dimensions, the global minima of the function is zero at [$x_1 = 0$, $x_2 = 0$] and the global maxima is 80.707 for $x_i = \pm 4.523$. Knowing these solutions, the various optimization algorithms can be checked if they have been implemented properly by running this particular optimization and comparing the found results.

Algorithm	Minimum	Maximum
GA	$1.551e^{-5}$ at $x_1 = 3.176e^{-5}$, $x_2 = -2.778e^{-4}$	80.706 at $x_1 = -4.523$, $x_2 = 4.522$
SA	$1.042e^{-3}$ at $x_1 = -1.989e^{-3}$, $x_2 = -1.139e^{-3}$	75.375 at $x_1 = -4.697$, $x_2 = -4.482$
PSO	0.0 at $x_1 = 3.095e^{-9}$, $x_2 = -7.286e^{-10}$	80.707 at $x_1 = -4.523$, $x_2 = 4.523$
DE	2.024 at $x_1 = -0.9818$, $x_2 = 0.9950$	80.331 at $x_1 = -4.479$, $x_2 = 4.524$
CMA-ES	0.0 at $x_1 = -3.282e^{-9}$, $x_2 = 7.985e^{-10}$	80.707 at $x_1 = -4.523$, $x_2 = -4.523$

Table 7.20: Results of the Rastrigin optimization for various single objective optimization algorithms

Table 7.20 shows the results of the optimizations. It can be seen that almost all optimization algorithms are able to find the correct values for both the minimum and maximum of the problem. Only the differential evolution algorithms for minimization finds a different value. Comparing this with the solution values

of the Rastrigin function, it is found that the DE algorithm got stuck in the local minimum at $x_i = \pm 1$, for which the solution value is two.

7.3.2. Validation of the results

For the case of optimizing the revisit time and coverage, hypotheses can be made on what type of orbits or constellations could be expected. In the particular scenario used in section 7.1 and section 7.2, due to the use of a fixed field of view and the emphasis on the coverage objective, it can be expected that results with a high semi-major axis perform better. Furthermore, as coverage of all latitudes is analyzed, it can be expected that the orbits should have high enough inclination to observe the poles. Finally, a combination of different perigee positions with a high eccentricity could also lead to overall better coverage.

Table 7.21 shows the solution of the best performing fixed-length single objective algorithm, CMA-ES. This solution can be compared with the scenario settings (Table 7.1) to check if the hypotheses are met. Looking at the semi-major axes of the four satellites, it can be seen that they indeed are near the upper limit set in the scenario. Looking at the inclinations of the four orbits, all are high, with three of the four being close to polar. This also meets the expectation. Finally, the eccentricity of the four orbits is high. Looking at the arguments of perigee, it can be seen that they are all quite similar, lying close to pi radians. Combining this with the spaced out right ascensions of ascending node, it indicates that the apogees are quite well spread out, allowing for maximum coverage and adhering to the hypothesis.

Satellite	a **[km]**	e	i **[rad]**	Ω **[rad]**	ω **[rad]**	ν_0 **[rad]**
1	7859.4	0.17666	1.6264	5.8227	2.8810	5.7271
2	7838.1	0.17441	2.4022	2.3391	3.0595	1.7356
3	7847.4	0.17539	1.6501	3.7113	3.1276	4.5403
4	7960.0	0.18706	1.8684	0.063036	2.9505	1.0174

Table 7.21: Solution found by the CMA-ES algorithm.

7.3.3. Sensitivity analysis

For each optimization algorithm a sensitivity analysis was performed using a slightly different scenario. This scenario differed in the instrument type of the satellites. Instead of a fixed field of view, the instruments now have a fixed swath of 400 kilometers. The rest of the scenario settings and constraints were kept the same (Table 7.1). Table 7.22 shows the results of the various fixed individual, single-objective optimizations.

Algorithm	**Optimal cost function value**	**Max revisit time [s]**	**Coverage [%]**	**Computation time [s]**
GA	272678.9	197807.0	96.26	14230
SA	281268.0	203880.5	96.13	40604
PSO	295669.4	199804.0	95.21	19427
DE	286774.3	201794.5	95.75	19955
CMA-ES	280746.4	197047.2	95.82	17504

Table 7.22: Results of the sensitivity analysis.

It can be seen that the results of the different algorithms lie much closer together then those in Table 7.16. This is primarily due to the coverage metric being much more equal between the algorithms. This could be attributed to the larger overall swath of the scenario, allowing the constellations to cover more of Earth of easily. It can be seen that for this scenario the genetic algorithm finds the best solution, closely followed by CMA-ES and SA.

To further identify if the algorithms behave as expected, a closer look can be taken at the results of the optimizations. In particular, a comparison can be made between the final results of the CMA-ES algorithm for the two scenarios. Due to the fixed swath size of the new scenario, the semi-major axis

and eccentricity have a reduced impact on the coverage. In fact, one could expect that a lower semi-major axis could provide better results, as more orbits could be completed within the scenario time. This is exactly what can be seen in Table 7.23, the semi-major axes and eccentricity are lower than those in Table 7.21. Further, it can be seen that the fourth satellite has an inclination close to polar to allow for the coverage of the highest latitudes.

Satellite	a **[km]**	e	i **[rad]**	Ω **[rad]**	ω **[rad]**	ν_0 **[rad]**
1	6800.0	0.048382	2.1996	1.4591	3.4210	3.2686
2	6813.1	0.050211	2.6724	2.6714	0.0	2.8438
3	7432.9	0.12941	1.9722	2.4787	1.9923	2.4964
4	7316.6	0.11557	1.5585	2.1699	5.6973	4.0461

Table 7.23: Solution found by the CMA-ES algorithm for the new scenario.

7.3.4. Validation with sentinel-2 constellation

To further validate the optimizer, a scenario can be created that closely resembles an existing constellation. Here, the existing constellation was chosen to be the sentinel-2 constellation of ESA [1]. The sentinel-2 constellation consists of two sun synchronous satellites flying in the same orbit, phased apart by 180 degrees true anomaly. To validate the optimizer (using the CMA-ES algorithm), a scenario was created to closely match the sentinel-2 constellation, namely restricting the two satellites to be in the same plane, to constrain the satellites to only sun synchronous orbits, and to constrain the semi-major axis to values close to those of the sentinel-2 constellation. This means that the only parameters that can be optimized is the true anomaly. The results of the optimization are shown in Table 7.24. It can be seen that the optimizer indeed finds the most optimal phasing of the satellites to be pi radians, or 180 degrees, in true anomaly.

Satellite	a **[km]**	e	i **[rad]**	Ω **[rad]**	ω **[rad]**	ν_0 **[rad]**
1	7158.8	0.0	1.7195	2.3509	3.0797	1.7539
2	7158.8	0.0	1.7195	2.3509	3.0797	4.8398

Table 7.24: Results of the CMA-ES optimization for the sentinel-2 scenario.

[1] `https://sentiwiki.copernicus.eu/web/s2-mission#S2Mission-OrbitS2-Mission-Orbittrue`

Part IV

Conclusion and recommendations

8

Conclusion

Conclusions can be made by answering the research questions from chapter 3. The research questions were split into two parts relating to the coverage analysis method and the constellation optimization. First the research questions related to the new coverage analysis method are answered.

1. Which figures of merit should the new coverage analysis method analyze?

During the literature study, various different Earth observation figures of merit were found. In particular, for discontinuous coverage, the most used metrics were the revisit time and coverage. However, further specifications exist of these figures of merit, as established by Wertz [63]. For the new coverage analysis method, the specific figures of merit were established as:

- The global maximum revisit time. This is the maximum revisit time found in any of the grid points observed at least twice by any satellite in the constellation.
- The coverage fraction of Earth. This is defined as the fraction or percentage of grid points that are observed by the satellite during the analysis period out of all grid points.

2. How to make the new coverage analysis method as fast as possible?

In the hypothesis, it was stated that the coverage analysis method should be made as analytical as possible to ensure a low computation time. After researching various approaches, it was concluded that a fully analytical coverage analysis method is not feasible. Instead a semi-analytical grid-based method was developed. To make the computation time as low as possible various actions were taken:

- By evaluating the ground track of the satellite before creating the grid, only grid points with a high likelihood to be observed, i.e. close to the ground track, are added to the grid. This reduced the amount of calls to the coverage analysis function, reducing the overall computation time.
- Only the first orbit of a satellite is analyzed using the coverage analysis function. Additional orbits can be evaluated by copying the results from the initial orbit, shifting the results according to the rotation of Earth, and finally modifying the observation time using the orbital period of the satellite. This massively reduces the number of coverage analysis function calls, especially for long analysis periods.
- The grid can be made non-rectangular, where fewer points are created at higher latitudes. Here the pixels still represent similar area sizes due to the curvature of Earth. Reducing the amount of grid points available reduces the amount of coverage analysis function calls, which in turn reduces the computation time.
- Multiprocessing can be applied to analyze grid points in parallel, reducing the overall computation time.

3. How can the new coverage analysis method be used for elliptical orbits?

The eccentricity is used during various steps of the coverage analysis. It is first used before the grid creation to establish the ground track of the satellite. To represent the ground track of an elliptical orbit, the equation from Crisp [6] (Equation 4.1) is modified into Equation 4.3.

Secondly, the eccentricity is used during the evaluation of the observation. More specifically during the transformation of the Kepler elements to inertial Cartesian coordinates (Equation 4.4).

Finally, the eccentricity is also taken into account during the computation of the observation time. This is done in the transformation from the true anomaly of the satellite to the time of observation (Equation 4.6).

4. How can the new coverage analysis method be used for asymmetrical constellations?

Expanding the coverage analysis to constellations is easily done by evaluating the individual satellites and combining the observation times together. This has the additional benefit of not being restricted to any form of symmetric or asymmetric constellation configuration, as the coverage analysis already works for any type of satellite orbit.

Next, the research questions for the constellation optimization part are answered.

1. For what objectives should the satellite constellations be optimized?

In the hypothesis for this research question it was stated that at the very least the figures of merit of the coverage analysis should be included as optimization objectives. Furthermore, constellation parameters such as the number of satellites could also be included. This is exactly what has been done during the optimization of the satellite constellations. The optimization objectives are as follows:

- Minimize the maximum revisit time of the constellation.
- Maximize the coverage of the constellation.
- Minimize the number of satellites in the constellation.

2. How can the amount of decision variables be maximized?

The hypothesis stated that the maximum amount of decision variables possible would be all orbital elements of all satellites within the constellation. This has been implemented into the optimization framework. All orbital elements are subject to the optimization, allowing for maximum flexibility of the optimizer. The drawback is the fact that the number of decision variables is dependent on the number of satellites in the individual constellation. This means that when optimizing the number of satellites, the number of decision variables also varies per individual. This causes issues with most single-objective optimization algorithms. Only the genetic algorithm is able to truly handle variable-length individuals.

3. What optimization algorithm performs the best?

Before comparing the different optimization algorithms with each other, the settings of each algorithm had to be determined. For each optimization algorithm, a small optimization was performed with different combinations of settings. The convergence of the solution was compared, as well as the final optimal solution that was found by the optimization. The optimization with the best configuration of settings was chosen to be compared with the other algorithms.

For the small test scenario, the genetic algorithm and the CMA-ES algorithm were found to perform the best for single objective optimization using fixed-length individuals. CMA-ES found the best performing solution, while the genetic algorithm found an acceptable solution in a considerably lower amount of time. An additional comparison was performed between the CMA-ES optimization and a GA optimization using more generations to make the time of computation more comparable. Even with additional generations, GA was still outperformed by CMA-ES, making CMA-ES the best performing algorithm for this scenario.

A sensitivity analysis was performed by analyzing a slightly different scenario. Again, all algorithms were compared based on the convergence of the optimization and the final solution. Again, the genetic algorithm and CMA-ES performed the best for the single objective optimization using fixed-length individuals. However, for this scenario, the GA was able to find a better solution than the CMA-ES algorithm.

Only two algorithms were able to be used to optimize variable-length individuals. This was the adjusted genetic algorithm for single objective optimization, and NSGA-II for multi-objective optimization. These two optimizations were compared with the results of the fixed-length GA optimization and the CMA-ES optimization. It was found that both the adjusted GA and the NSGA-II found acceptable results, but not better results than the fixed-length optimizations.

The primary conclusion that can be formed from the results is that the choice of best optimization algorithm depends on the use case. If it is desired to find the most optimal constellation of a specific number of satellites, either the genetic algorithm or the CMA-ES algorithm perform the best. If it is desired to find one solution for which the number of satellites is also optimized, the adjusted GA is to be used. Finally, if an overview is desired of the best constellation for each possible number of satellites, the Pareto front of the NSGA-II optimization provides the most relevant information.

9

Recommendations

The limitations of the thesis open up new paths for further research. Both for the coverage analysis method as well as the optimization of satellite constellations, some recommendations for further improvements can be given.

9.1. Recommendations for the coverage analysis method

In this thesis an overview was given of the development of a new coverage analysis method, starting with an initial idea for a fully analytical design. While it was concluded that the analytical approach for an analytical design was not feasible, further research could be made on different ways to design a fully analytical coverage analysis method. Achieving this feat could allow for even better and faster analysis and optimization of satellite constellations.

Some improvements can also be made on the developed coverage analysis method. Firstly, efforts could be taken to remove the spherical Earth assumption. This would allow for more accurate results of the coverage analysis. Secondly, the sensitivity of the computation time to changes in the scenario parameters could be improved. This would allow the coverage analysis to have a more consistent computation time when analyzing different constellations, making the optimization more predictable in terms of computation length. Thirdly, the use of a grid should be evaluated as the method involves rounding, causing slight inaccuracies throughout the analysis. In particular, efforts could be taken in allowing the storage of partial observations, making the coverage analysis method more accurate.

9.2. Recommendations for the constellation optimization

A number of recommendations can be made on the selection of the optimization algorithm. Much more exploration can be performed within the constellation optimization topic. Firstly, for the various algorithms that have been evaluated in this thesis, numerous other settings can still be compared and analyzed. This should also be done for different scenarios, to evaluate the impact of different parameters on the settings of the algorithms. Further, many more optimization algorithms exist apart from the algorithms that were analyzed in this thesis.

References

[1] J. C. Bansal et al. “Inertia Weight strategies in Particle Swarm Optimization”. In: *2011 Third World Congress on Nature and Biologically Inspired Computing*. 2011, pp. 633–640. DOI: 10.1109/NaBIC.2011.6089659.

[2] Pau Garcia Buzzi et al. “Assessment of constellation designs for earth observation: Application to the TROPICS mission”. In: *Acta Astronautica* 161 (2019), pp. 166–182. ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2019.05.007.

[3] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces*. 1st. Prentice-Hall, 1976, pp. 188–197. ISBN: 978-0132125895.

[4] Augustin Cauchy. “Méthode générale pour la résolution des systèmes d’équations simultanées”. In: *Comp. Rend. Sci. Paris* 25 (1847), pp. 536–538.

[5] Razvan Cazacu. “Comparative Study between the Improved Implementation of 3 Classic Mutation Operators for Genetic Algorithms”. In: *Procedia Engineering* 181 (2017), pp. 634–640. DOI: https://doi.org/10.1016/j.proeng.2017.02.444.

[6] Nicholas Crisp, Sabrina Livadiotti, and Peter Roberts. *A Semi-Analytical Method for Calculating Revisit Time for Satellite Constellations with Discontinuous Coverage*. July 2018. DOI: 10.48550/arXiv.1807.02021.

[7] Haskell B. Curry. “The Method of Steepest Descent for Non-Linear Minimization Problems”. In: *Quarterly of Applied Mathematics* 2.3 (1944), pp. 258–261. ISSN: 0033569X, 15524485. URL: http://www.jstor.org/stable/43633461.

[8] Giacomo Curzi, Dario Modenini, and Paolo Tortora. “Large Constellations of Small Satellites: A Survey of Near Future Challenges and Missions”. In: *Aerospace* 7.9 (2020). ISSN: 2226-4310. DOI: 10.3390/aerospace7090133.

[9] Alex Da Silva Curiel et al. “First results from the disaster monitoring constellation (DMC)”. In: *Acta Astronautica* 56.1-2 (2005), pp. 261–271. DOI: 10.1016/j.actaastro.2004.09.026.

[10] Jeremy J. Davis, Martín E. Avendaño, and Daniele Mortari. “The 3-D lattice theory of Flower Constellations”. In: *Celestial Mechanics and Dynamical Astronomy* 116.4 (Aug. 2013), pp. 339–356. DOI: 10.1007/s10569-013-9494-7.

[11] Kalyan Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *Evolutionary Computation, IEEE Transactions on* 6 (May 2002), pp. 182–197. DOI: 10.1109/4235.996017.

[12] F Dufour et al. “Constellation design optimization with a DOP based criterion”. In: *14th International symposium on space fligt dynamics*. 1995.

[13] David E. Goldberg and Kalyanmoy Deb. “A Comparative Analysis of Selection Schemes Used in Genetic Algorithms”. In: ed. by GREGORY J.E. RAWLINS. Vol. 1. Foundations of Genetic Algorithms. Elsevier, 1991, pp. 69–93. DOI: https://doi.org/10.1016/B978-0-08-050684-5.50008-2.

[14] M.D. Graziano, M. D’Errico, and E. Razzano. “Constellation analysis of an integrated AIS/remote sensing spaceborne system for ship detection”. English. In: *Advances in Space Research* 50.3 (2012). Publisher: Elsevier Ltd, pp. 351–362. DOI: 10.1016/j.asr.2012.04.020.

[15] Nikolaus Hansen. *The CMA Evolution Strategy: A Tutorial*. 2023. URL: https://arxiv.org/abs/1604.00772.

[16] Nikolaus Hansen and Andreas Ostermeier. “Completely Derandomized Self-Adaptation in Evolution Strategies”. In: *Evol. Comput.* 9.2 (June 2001), pp. 159–195. ISSN: 1063-6560. DOI: 10.1162/106365601750190398.

[17] Ramy F. Harik, Hu Gong, and Alain Bernard. “5-axis flank milling: A state-of-the-art review”. In: *Computer-Aided Design* 45.3 (2013), pp. 796–808. ISSN: 0010-4485. DOI: https://doi.org/10.1016/j.cad.2012.08.004.

[18] X. HE and H. LI. “Analytical solutions for Earth discontinuous coverage of satellite constellation with repeating ground tracks”. English. In: *Chinese Journal of Aeronautics* 35.10 (2022). Publisher: Elsevier B.V., pp. 275–291. DOI: 10.1016/j.cja.2021.11.012.

[19] X. He and H. Li. “General analysis method for global revisit characteristics of satellite constellation with repeating ground tracks”. English. In: *Acta Astronautica* 202 (2023). Publisher: Elsevier Ltd, pp. 319–332. DOI: 10.1016/j.actaastro.2022.10.051.

[20] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, Apr. 1992. ISBN: 9780262275552. DOI: 10.7551/mitpress/1090.001.0001.

[21] Yousef Jaradat et al. “Comparison of Genetic Algorithm Crossover Operators on WSN Lifetime”. In: May 2022, pp. 356–360. DOI: 10.1109/MI-STA54861.2022.9837587.

[22] J. Kennedy and R. Eberhart. “Particle swarm optimization”. In: *Proceedings of ICNN’95 - International Conference on Neural Networks*. Vol. 4. 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[23] Yunjoong Kim et al. “Optimum design of an SAR satellite constellation considering the revisit time using a genetic algorithm”. In: *International Journal of Aeronautical and Space Sciences* 18.2 (2017), pp. 334–343. DOI: 10.5139/IJASS.2017.18.2.334.

[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by Simulated Annealing”. In: *Science* 220.4598 (1983), pp. 671–680. DOI: 10.1126/science.220.4598.671.

[25] Jaeyoul Ko, Beomjin Gwon, and Jaemyung Ahn. “Proposal on Figure of Merits for Heterogeneous Earth Observation Satellite Constellation; [□□ □□ □□ □□□□ □□□ □□□□ □□ □□□ □□ □□]”. In: *Journal of the Korean Society for Aeronautical and Space Sciences* 53.4 (2025), pp. 411–419. DOI: 10.5139/JKSAS.2025.53.4.411.

[26] T.J. Lang. “A parametric examination of satellite constellations to minimize revisit time for low earth orbits using a genetic algorithm”. English. In: *Adv Astronaut Sci*. Vol. 109 I. Journal Abbreviation: Adv Astronaut Sci. 2002, pp. 625–640. ISBN: 00653438 (ISSN). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0036069188&partnerID=40&md5=9725fe9b9619ca9fc7b0b15ae8fe4fda.

[27] T.J. Lang. “Streets of coverage constellations to minimize revisit time in low Earth orbit”. English. In: *Adv Astronaut Sci*. Ed. by Vallado D.A., Gabor M.J., and Desai P.N. Vol. 120. Issue: I Journal Abbreviation: Adv Astronaut Sci. 2005, pp. 795–809. ISBN: 00653438 (ISSN). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-26944499357&partnerID=40&md5=888ec6320c312540b611b0f9bcc24455.

[28] T.J. Lang. “Walker constellations to minimize revisit time in low earth orbit”. English. In: *Adv Astronaut Sci*. Ed. by Sheeres D.J. et al. Vol. 114 II. Journal Abbreviation: Adv Astronaut Sci. 2003, pp. 1127–1141. ISBN: 00653438 (ISSN). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-1542359592&partnerID=40&md5=f8a5d37414a32ec7f8f1bebeb2a4e161.

[29] Thomas J. Lang and John M. Hanson. “ORBITAL CONSTELLATIONS WHICH MINIMIZE REVISIT TIME.” English. In: *Advances in the Astronautical Sciences*. Vol. 54. Issue: Pt 2 Journal Abbreviation: Advances in the Astronautical Sciences. r American Astronautical Soc by Univelt Inc, 1983, pp. 1071–1086. ISBN: 00653438 (ISSN); 0877031908 (ISBN). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0020872520&partnerID=40&md5=e332f33224ad3f3d01af7e1e7ae3ea60.

[30] H.W. Lee et al. “Satellite constellation pattern optimization for complex regional coverage”. English. In: *Journal of Spacecraft and Rockets* 57.6 (2020). Publisher: AIAA International, pp. 1309–1327. DOI: 10.2514/1.A34657.

[31] Israel Leyva-Mayorga et al. *NGSO Constellation Design for Global Connectivity*. Mar. 2022. DOI: 10.48550/arXiv.2203.16597.

[32] H. LI, D. LI, and Y. LI. “A multi-index assessment method for evaluating coverage effectiveness of remote sensing satellite”. English. In: *Chinese Journal of Aeronautics* 31.10 (2018). Publisher: Chinese Journal of Aeronautics, pp. 2023–2033. DOI: 10.1016/j.cja.2018.05.015.

[33] T. Li et al. “Circular revisit orbits design for responsive mission over a single target”. English. In: *Acta Astronautica* 127 (2016). Publisher: Elsevier Ltd, pp. 219–225. DOI: 10.1016/j.actaastro.2016.05.037.

[34] Mohammad Loni, Masoud Daneshtalab, and Mikael Sjödin. “ADONN: Adaptive Design of Optimized Deep Neural Networks for Embedded Systems”. In: Aug. 2018, pp. 397–404. DOI: 10.1109/DSD.2018.00074.

[35] R. DAVID LUDERS. “Satellite Networks for Continuous Zonal Coverage”. In: *ARS Journal* 31.2 (Feb. 1961), pp. 179–184. DOI: 10.2514/8.5422.

[36] L. Mencarelli et al. “Mixed integer (non)linear approaches for the satellite constellation design problem”. English. In: *Optimization and Engineering* 24.4 (2023). Publisher: Springer, pp. 2299–2320. DOI: 10.1007/s11081-022-09774-9.

[37] Jorge Mendes. “A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem”. In: *WSEAS Transactions on Computers* 12 (Apr. 2013), pp. 164–173.

[38] José de Mendoza y Rios. *Memoria sobre algunos métodos nuevos para calcular la longitud por las distancias lunares, y aplicación de su teoría a la solución de otros problemas de navegación*. Spanish. Madrid, España: Imprenta Real, 1795, p. 13. URL: https://uvadoc.uva.es/handle/10324/57278.

[39] Nicholas Metropolis et al. “Equation of State Calculations by Fast Computing Machines”. In: *The Journal of Chemical Physics* 21.6 (June 1953), pp. 1087–1092. ISSN: 0021-9606. DOI: 10.1063/1.1699114.

[40] John J. Morrison. “A system of sixteen synchronous satellites for worldwide navigation and surveillance”. English. In: (1973). Ed. by John A. Volpe National Transportation Systems Center (U.S.) URL: https://rosap.ntl.bts.gov/view/dot/9196.

[41] D. Mortari, M.P. Wilkins, and C. Bruccoleri. “The Flower Constellations”. English. In: *J Astronaut Sci*. Vol. 52. Issue: 1-2 Journal Abbreviation: J Astronaut Sci. 2004, pp. 107–127. ISBN: 00219142 (ISSN). URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-14744288142&partnerID=40&md5=bc0b189bea7908b8046b936e7b19c80b.

[42] Yaghout Nourani and Bjarne Andresen. “A comparison of simulated annealing cooling strategies”. In: *Journal of Physics A: Mathematical and General* 31.41 (Oct. 1998), p. 8373. DOI: 10.1088/0305-4470/31/41/011.

[43] Hari Mohan Pandey. “Performance Evaluation of Selection Methods of Genetic Algorithm and Network Security Concerns”. In: *Procedia Computer Science* 78 (2016), pp. 13–18. DOI: https://doi.org/10.1016/j.procs.2016.02.004.

[44] JI Qiang. “A unified differential evolution algorithm for global optimization”. In: (2014).

[45] Y.N. Razoumny. “Fundamentals of the route theory for satellite constellation design for Earth discontinuous coverage. Part 1: Analytic emulation of the Earth coverage”. English. In: *Acta Astronautica* 128 (2016), pp. 722–740. DOI: 10.1016/j.actaastro.2016.07.013.

[46] Y.N. Razoumny. “Route satellite constellations for earth discontinuous coverage and optimal solution peculiarities”. English. In: *Journal of Spacecraft and Rockets* 54.3 (2017). Publisher: American Institute of Aeronautics and Astronautics Inc., pp. 572–581. DOI: 10.2514/1.A33689.

[47] Arjuman R. Reshi, Subbarao Pichuka, and Akshar Tripathi. “Applications of Sentinel-5P TROPOMI Satellite Sensor: A Review”. In: *IEEE Sensors Journal* 24.13 (2024), pp. 20312–20321. DOI: 10.1109/JSEN.2024.3355714.

[48] A.B. Reut and T. Hara. “Remote monitoring of military assets using commercial LEO satellites”. In: *Proceedings of MILCOM ’95*. Vol. 2. 1995, 869–873 vol.2. DOI: 10.1109/MILCOM.1995.483651.

[49] Herbert Robbins and Sutton Monro. “A Stochastic Approximation Method”. In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. DOI: 10.1214/aoms/1177729586.

[50] Nasir Saeed et al. *CubeSat Communications: Recent Advances and Future Challenges*. Aug. 2019. DOI: 10.36227/techrxiv.12179397.

[51] S. Sarno, M.D. Graziano, and M. D'Errico. "Polar constellations design for discontinuous coverage". English. In: *Acta Astronautica* 127 (2016), pp. 367–374. DOI: 10.1016/j.actaastro.2016.06.001.

[52] T. Savitri et al. "Satellite Constellation Orbit Design Optimization with Combined Genetic Algorithm and Semianalytical Approach". English. In: *International Journal of Aerospace Engineering* 2017 (2017). Publisher: Hindawi Limited. DOI: 10.1155/2017/1235692.

[53] Joel Segarra. "Satellite Imagery in Precision Agriculture". In: *Digital Agriculture: A Solution for Sustainable Food and Nutritional Security*. Ed. by P. M. Priyadarshan et al. Cham: Springer International Publishing, 2024, pp. 325–340. DOI: 10.1007/978-3-031-43548-5_10.

[54] Tomer Shtark and Pini Gurfil. "Regional positioning using a low Earth orbit satellite constellation". In: *Celestial Mechanics and Dynamical Astronomy* 130.2 (Jan. 2018), p. 14. DOI: 10.1007/s10569-017-9811-7.

[55] N. Srinivas and Kalyanmoy Deb. "Multi-Objective Function Optimization Using Non-Dominated Sorting Genetic Algorithms". In: 1994. URL: https://api.semanticscholar.org/CorpusID:60032759.

[56] Rainer Storn and Kenneth Price. "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces". In: *Journal of Global Optimization* 11 (Jan. 1997), pp. 341–359. DOI: 10.1023/A:1008202821328.

[57] Gilbert Strang and Kai Borre. *Linear Algebra, Geodesy, and GPS*. English. Wellesley-Cambridge Press, 1997. ISBN: 0961408863.

[58] S.Y. Ulybyshev and A.A. Lysenko. "Design of Satellite Constellations for Operational Global Monitoring with a Daily Repeat of Flight Track". English. In: *Cosmic Research* 57.3 (2019). Publisher: Pleiades Publishing, pp. 204–212. DOI: 10.1134/S0010952519030080.

[59] Y. Ulybyshev. "Geometric analysis and design method for discontinuous coverage satellite constellations". English. In: *J Guid Control Dyn*. Vol. 37. Issue: 2 Journal Abbreviation: J Guid Control Dyn. American Institute of Aeronautics and Astronautics Inc., 2014, pp. 549–557. DOI: 10.2514/1.60756.

[60] Yuri Ulybyshev. "General analysis method for discontinuous coverage satellite constellations". English. In: *Journal of Guidance, Control, and Dynamics* 38.12 (2015), pp. 2475–2482. DOI: 10.2514/1.G001254.

[61] David A. Vallado and Wayne D. McClain. "Fundamentals of astrodynamics and applications". English. In: College customs series. Section: xviii, 922 pages : illustrations ; 24 cm. New York: McGraw-Hill Companies, Inc., 1997. Chap. 11. ISBN: 0-07-066834-5 978-0-07-066834-8.

[62] J.G. Walker. "Some circular orbit patterns providing continuous whole earth coverage". undefined. In: *J Brit Interplanet Soc* 24.7 (1971), pp. 369–384. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0015094458&partnerID=40&md5=fe88cd32a3ae31a96ceca4b1b2de938e.

[63] James Richard Wertz, David F. Everett, and Jeffery John Puschell. "Space mission engineering : the new SMAD". English. In: Space technology library; v. 28; volume 28. Section: xvi, 1033 pages : illustrations, maps ; 28 cm. Hawthorne, CA: Microcosm Press, 2011. Chap. 10. ISBN: 1-881883-16-7 978-1-881883-16-6 1-881883-15-9 978-1-881883-15-9.

[64] James Richard Wertz and Wiley J. Larson. "Space mission analysis and design". In: Journal Abbreviation: Space mission analysis and design Publication Title: Space mission analysis and design. Jan. 1999. Chap. 7.

[65] R. Wilkinson et al. "Environmental impacts of earth observation data in the constellation and cloud computing era". In: *Science of The Total Environment* 909 (Nov. 2023), p. 168584. DOI: 10.1016/j.scitotenv.2023.168584.

[66] Anil Yaman, Giovanni Iacca, and Fabio Caraffini. "A comparison of three differential evolution strategies in terms of early convergence with different population sizes". In: *AIP Conference Proceedings* 2070.1 (Feb. 2019), p. 020002. ISSN: 0094-243X. DOI: 10.1063/1.5089969.

[67] Chen Zhang et al. “LEO constellation design methodology for observing multi-targets”. In: *Astrodynamics* 2.2 (June 2018), pp. 121–131. DOI: 10.1007/s42064-017-0015-4.

[68] T.-J. Zhang et al. “Restricted constellation design for regional navigation augmentation”. English. In: *Acta Astronautica* 150 (2018). Publisher: Elsevier Ltd, pp. 231–239. DOI: 10.1016/j.actaastro.2018.04.044.

A

Additional results

In this appendix, some visualizations are provided to give some more context to the optimal constellations found in chapter 7. Further, an additional result is provided for a larger optimization.

A.1. Visualizations of the CMA-ES optimal constellation

The following visualizations are based on the optimal constellation found by the CMA-ES optimization. The specific orbital elements of the satellites are given in Table 7.21. The visualizations are shown in Figure A.1.

A.2. Results of a larger optimization

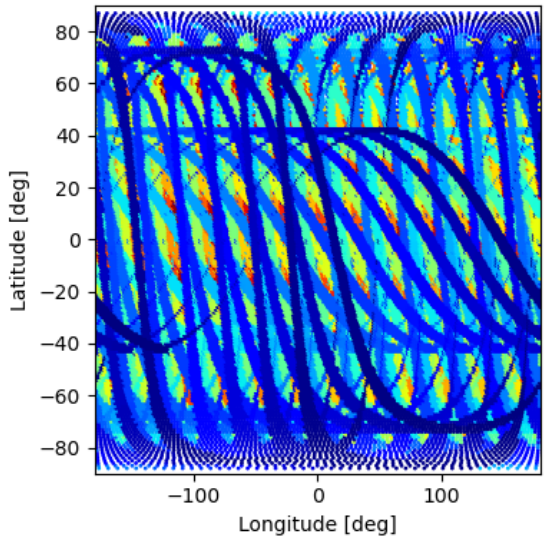
The CMA-ES algorithm was also applied to a larger optimization. The scenario constraints are stated in Table A.1. The main differences with the optimization performed in chapter 7 lie in the higher semi-major axis and bigger eccentricity constraints. Further, a population of 100 individuals as well as 100 generations were used.

Parameter	Value	Unit
Scenario length	200000	s
# of satellites	4	-
Field of view	15	deg
Semi-major axis	8000-10000	km
Eccentricity	0-0.4	-
Inclination	0-180	deg
RAAN	0-360	deg
Argument of perigee	0-360	deg
True anomaly	0-360	deg

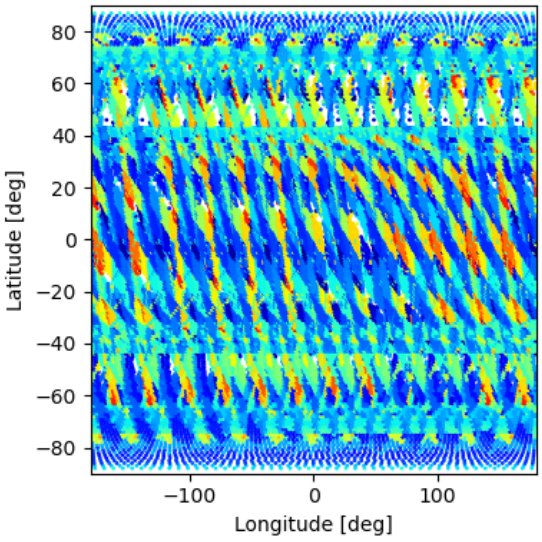
Table A.1: Scenario variables for algorithm comparisons.

Visualizations of the found constellation are shown in Figure A.2. Further, the numerical results are stated in Table A.2 and Table A.3. It can be seen that the longer optimization benefited the coverage metric greatly, as almost full coverage was achieved. Further, compared to other results in the thesis, the revisit time is quite low, although this could also be a result of the higher semi-major axis and eccentricity.

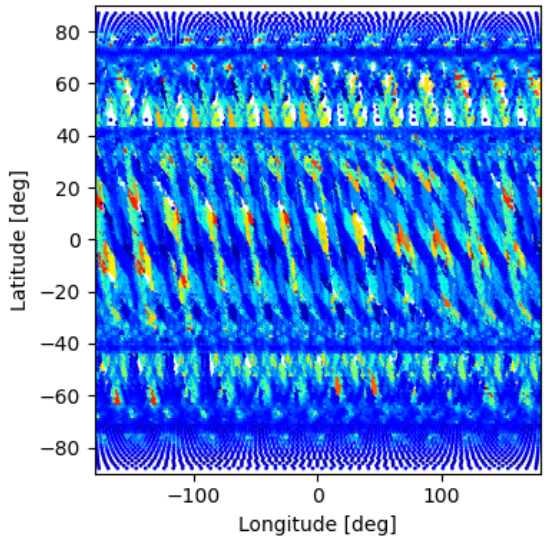
The orbital elements are in line with what was to be expected, namely a high semi-major axis and eccentricity. Interestingly, comparing this result with the optimal constellation of the smaller optimization, some parallels can be drawn. Namely, that three of the four satellites have a high inclination, to cover



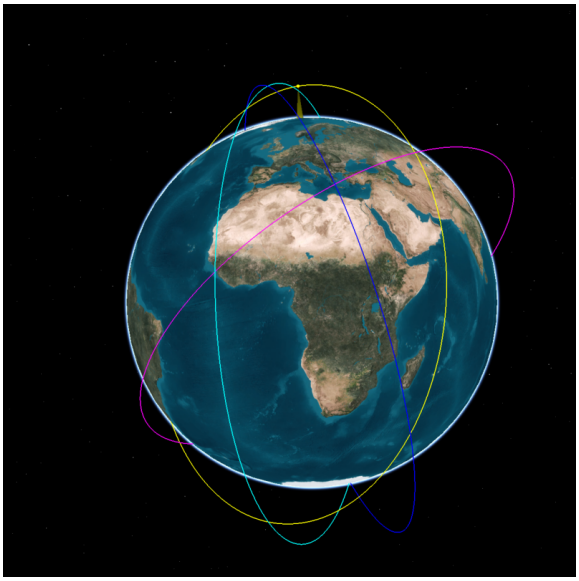
(a) The color indicates the time of first observation, with blue being earlier and red later.



(b) The color indicates the maximum revisit time, with blue being shorter and red being longer.



(c) The color indicates the average revisit time, with blue being shorter and red being longer.



(d) A 3D visualization of the constellation made using STK 12.

Figure A.1: Visualizations for the optimal constellation found by the CMA-ES optimization for the small test scenario.

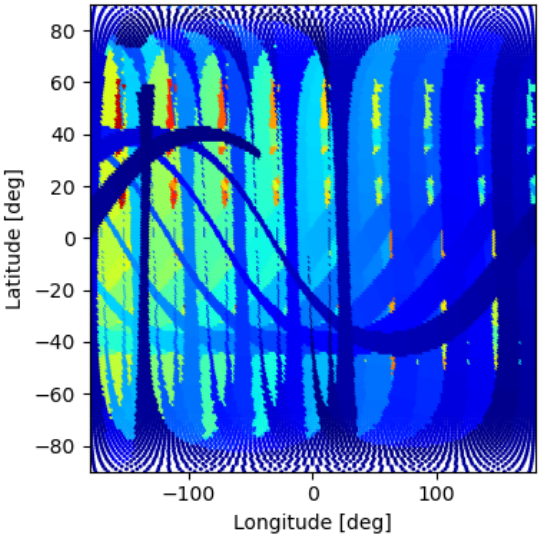
all latitudes, while one satellite has a lower inclination. This last satellite might be used to lower the revisit time for area of Earth closer to the equator, which have a higher revisit time than the poles.

Optimal cost function value	Max revisit time [s]	Coverage [%]	Computation time [s]
141018.2	138045.2	99.85	419391

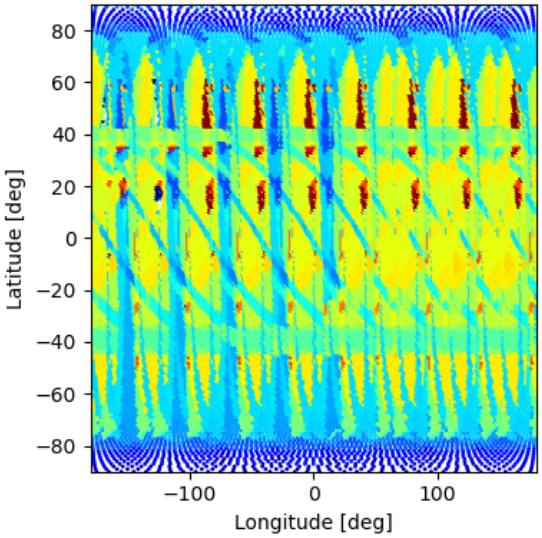
Table A.2: Final result and computation time for CMA-ES algorithm for the longer scenario.

Satellite	a [km]	e	i [rad]	Ω [rad]	ω [rad]	ν_0 [rad]
1	9984.9	0.35192	1.5162	3.1060	2.9241	4.6555
2	9967.7	0.060111	1.5072	0.49661	3.3939	4.9835
3	9962.2	0.24255	2.4341	5.9691	0.32424	0.62591
4	9981.7	0.34425	1.4573	2.6787	3.1110	0.13438

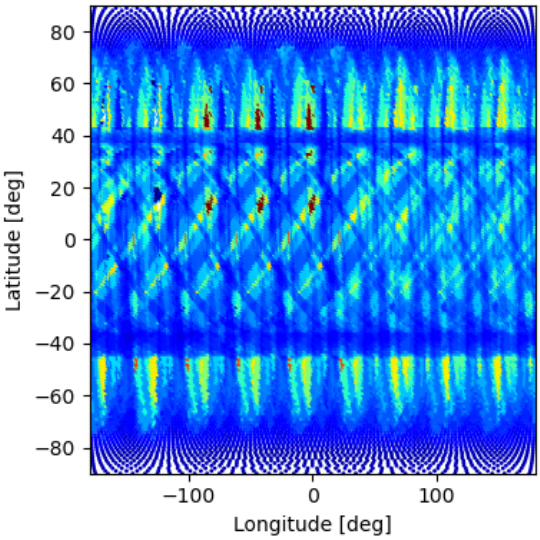
Table A.3: Solution found by the CMA-ES algorithm for the longer scenario.



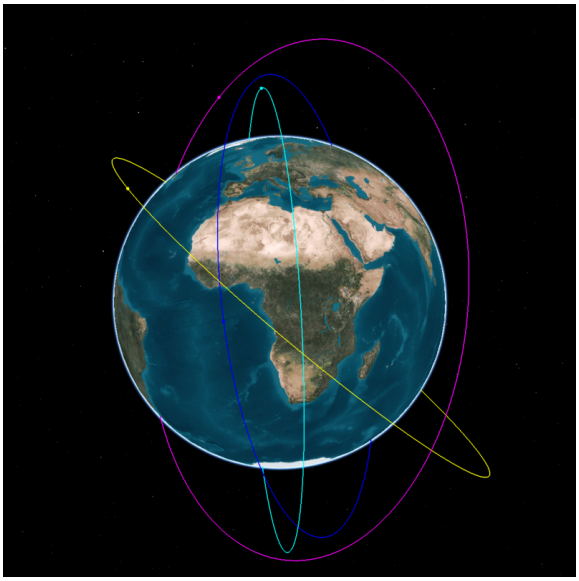
(a) The color indicates the time of first observation, with blue being earlier and red later.



(b) The color indicates the maximum revisit time, with blue being shorter and red being longer.



(c) The color indicates the average revisit time, with blue being shorter and red being longer.



(d) A 3D visualization of the constellation made using STK 12.

Figure A.2: Visualizations for the optimal constellation found by the CMA-ES optimization for the larger test scenario.

B

Swath slope derivation

This appendix provides the derivation of the swath slope formula of the first grid-based method. More information can be found in subsection 4.2.1. The derivation starts with the representation of the ground track as provided by Crisp [6].

$$\begin{cases} \phi = sin^{-1}\left(sin\left(\nu_\phi + \omega\right) sin(i)\right) \\ \lambda = tan^{-1}\left(\frac{cos(\omega+\nu_\phi)sin(\Omega)+sin(\omega+\nu_\phi)cos(\Omega)cos(i)}{cos(\omega+\nu_\phi)cos(\Omega)-sin(\omega+\nu_\phi)sin(\Omega)cos(i)}\right) + \frac{\nu_\phi}{2\pi}\Delta\lambda \end{cases} \tag{B.1}$$

According to subsection 4.2.1, the swath slope formula can be derived following:

$$\left(\frac{d\phi}{d\lambda}\left(\nu_\phi\right)\right)_{swath} = -\left(\frac{d\phi}{d\lambda}\left(\nu_\phi\right)\right)_{GT}^{-1} = -\left(\frac{d\phi\left(\nu_\phi\right)}{d\nu_\phi}\frac{d\nu_\phi}{d\lambda\left(\nu_\phi\right)}\right)^{-1} = -\frac{d\lambda\left(\nu_\phi\right)}{d\nu_\phi}\left(\frac{d\phi\left(\nu_\phi\right)}{d\nu_\phi}\right)^{-1} \tag{B.2}$$

Thus the derivatives $\frac{d\lambda(\nu_\phi)}{d\nu_\phi}$ and $\frac{d\phi(\nu_\phi)}{d\nu_\phi}$ should be found. Starting with $\frac{d\phi(\nu_\phi)}{d\nu_\phi}$:

$$\begin{aligned} \phi &= sin^{-1}\left(sin\left(\nu_\phi + \omega\right) sin(i)\right) \\ \frac{d\phi\left(\nu_\phi\right)}{d\nu_\phi} &= \frac{\frac{d}{d\nu_\phi}[sin(i)sin(\nu_\phi+\omega)]}{\sqrt{1-(sin(i)sin(\nu_\phi+\omega))^2}} = \frac{sin(i)\frac{d}{d\nu_\phi}[sin(\nu_\phi+\omega)]}{\sqrt{1-(sin(i)sin(\nu_\phi+\omega))^2}} \\ &= \frac{sin(i)cos(\nu_\phi+\omega)}{\sqrt{1-(sin(i)sin(\nu_\phi+\omega))^2}} \end{aligned} \tag{B.3}$$

Next is $\frac{d\lambda(\nu_\phi)}{d\nu_\phi}$. To simplify, the equation for $\lambda(\nu_\phi)$ is split into $tan^{-1}(\frac{N}{D}) + \frac{\nu_\phi}{2\pi}\Delta\lambda$, where N stands for numerator and D for denominator. Thus, the derivative can be written as:

$$\begin{aligned} \frac{d\lambda\left(\nu_\phi\right)}{d\nu_\phi} &= \frac{d}{d\nu_\phi}\left[tan^{-1}\left(\frac{N}{D}\right) + \frac{\nu_\phi}{2\pi}\Delta\lambda\right] = \frac{d}{d\nu_\phi}\left[tan^{-1}\left(\frac{N}{D}\right)\right] + \frac{\Delta\lambda}{2\pi} \\ &= \frac{\frac{d}{d\nu_\phi}\left[\frac{N}{D}\right]}{1+\left(\frac{N}{D}\right)^2} + \frac{\Delta\lambda}{2\pi} = \frac{\frac{\frac{dN}{d\nu_\phi}D - \frac{dD}{d\nu_\phi}N}{D^2}}{1+\left(\frac{N}{D}\right)^2} + \frac{\Delta\lambda}{2\pi} = \frac{\frac{dN}{d\nu_\phi}D - \frac{dD}{d\nu_\phi}N}{D^2+N^2} + \frac{\Delta\lambda}{2\pi} \end{aligned} \tag{B.4}$$

$D^2 + N^2$ is easily computed:

$$\begin{aligned} N^2 &= (cos(\nu_\phi+\omega)sin(\Omega) + sin(\nu_\phi+\omega)cos(\Omega)cos(i))^2 \\ &= cos^2(\nu_\phi+\omega)sin^2(\Omega) + 2sin(\nu_\phi+\omega)cos(\nu_\phi+\omega)sin(\Omega)cos(\Omega)cos(i) + sin^2(\nu_\phi+\omega)cos^2(\Omega)cos^2(i) \\ D^2 &= (cos(\nu_\phi+\omega)cos(\Omega) - sin(\nu_\phi+\omega)sin(\Omega)cos(i))^2 \\ &= cos^2(\nu_\phi+\omega)cos^2(\Omega) - 2sin(\nu_\phi+\omega)cos(\nu_\phi+\omega)sin(\Omega)cos(\Omega)cos(i) + sin^2(\nu_\phi+\omega)sin^2(\Omega)cos^2(i) \\ N^2 + D^2 &= cos^2(\nu_\phi+\omega) + sin^2(\nu_\phi+\omega)cos^2(i) \end{aligned} \tag{B.5}$$

Next, the derivatives of N and D should be computed:

$$\begin{aligned}\frac{dN}{d\nu_\phi} &= \frac{d}{d\nu}[cos(\nu_\phi+\omega)sin(\Omega)+sin(\nu_\phi+\omega)cos(\Omega)cos(i)]\\ &= -sin(\nu_\phi+\omega)sin(\Omega)+cos(\nu_\phi+\omega)cos(\Omega)cos(i)\\ \frac{dD}{d\nu_\phi} &= \frac{d}{d\nu}[cos(\nu_\phi+\omega)cos(\Omega)-sin(\nu_\phi+\omega)sin(\Omega)cos(i)]\\ &= -sin(\nu_\phi+\omega)cos(\Omega)-cos(\nu_\phi+\omega)sin(\Omega)cos(i)\end{aligned} \tag{B.6}$$

Which then becomes:

$$\frac{dN}{d\nu_\phi}D-\frac{dD}{d\nu_\phi}N=cos(i) \tag{B.7}$$

Therefore, the derivative of the longitude function becomes:

$$\begin{aligned}\frac{d\lambda\left(\nu_\phi\right)}{d\nu_\phi} &= \frac{\frac{dN}{d\nu_\phi}D-\frac{dD}{d\nu_\phi}N}{D^2+N^2}+\frac{\Delta\lambda}{2\pi}=\frac{cos(i)}{cos^2(\nu_\phi+\omega)+sin^2(\nu_\phi+\omega)cos^2(i)}+\frac{\Delta\lambda}{2\pi}\\ &= \frac{cos(i)}{1-sin^2(\nu_\phi+\omega)+sin^2(\nu_\phi+\omega)(1-sin^2(i))}+\frac{\Delta\lambda}{2\pi}\\ &= \frac{cos(i)}{1-sin^2(\nu_\phi+\omega)sin^2(i)}+\frac{\Delta\lambda}{2\pi}\end{aligned} \tag{B.8}$$

Combining the derivatives for the longitude and latitude:

$$\begin{aligned}-\frac{d\lambda\left(\nu_\phi\right)}{d\nu_\phi}\left(\frac{d\phi\left(\nu_\phi\right)}{d\nu_\phi}\right)^{-1} &= -\left(\frac{cos(i)}{1-sin^2(\nu_\phi+\omega)sin^2(i)}+\frac{\Delta\lambda}{2\pi}\right)\left(\frac{sin(i)cos(\nu_\phi+\omega)}{\sqrt{1-(sin(i)sin(\nu_\phi+\omega))^2}}\right)^{-1}\\ &= -\left(\frac{cos(i)}{1-sin^2(\nu_\phi+\omega)sin^2(i)}+\frac{\Delta\lambda}{2\pi}\right)\left(\frac{\sqrt{1-(sin(i)sin(\nu_\phi+\omega))^2}}{sin(i)cos(\nu_\phi+\omega)}\right)\\ &= -\frac{\sqrt{1-sin^2(\nu_\phi+\omega)sin^2(i)}\left(\frac{cos(i)}{1-sin^2(\nu_\phi+\omega)sin^2(i)}+\frac{\Delta\lambda}{2\pi}\right)}{cos(\nu_\phi+\omega)sin(i)}\end{aligned} \tag{B.9}$$

Thus, The formula for the slope of the swath line is:

$$\left(\frac{d\phi}{d\lambda}\left(\nu_\phi\right)\right)_{swath}=-\frac{\sqrt{1-sin^2(\nu_\phi+\omega)sin^2(i)}\left(\frac{cos(i)}{1-sin^2(\nu_\phi+\omega)sin^2(i)}+\frac{\Delta\lambda}{2\pi}\right)}{cos(\nu_\phi+\omega)sin(i)} \tag{B.10}$$