University of Strathclyde
Department of Electronic and Electrical Engineering

Target Area Based Methods for Analytical Satellite Orbit Design and Tasking

by

Adah Majala Tole

Thesis Submitted to the Department of Electronic and Electrical Engineering in fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)

2025

Declaration of author’s rights

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.50.

Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Adah Majala Tole
15th May 2025

Dedication

This work is firstly dedicated to The Almighty God, Who gave me the wisdom and the resilience to keep going even when the PhD seemed far-fetched. It is also dedicated to all the people who look up to me. The sky is not the limit. Dreams can be achieved regardless of age, background and challenges that come with it, just start and make progress a day at a time; progress over perfection (Winston Churchill). May this work make a difference to someone, somewhere, and leave my mark on the world for future generations.

“Greatness can only be measured by what you do for others, at the end of the day, all of us, we’re going to leave, and nothing remains but the love we shared and the good we did”

- Chatri Trisiripisal

“Space isn’t just for astronauts; it is for everybody” - Christa McAuliffe

“Passion is important, but it only gets you so far”

Acknowledgements

First and most importantly, my gratitude is to God, who throughout this PhD, even during COVID times, has been my greatest companion, sustaining me, comforting me, encouraging me and giving me wisdom. Throughout this PhD there have been many challenges, but He has always been faithful.

I am forever in-debited to my research group members, Dr. Astrid Werkmeister and Dr. Ciara McGrath for the support and advice they gave me through the PhD. Most of all, I express my deep appreciation, to my supervisors; Prof. Malcolm Macdonald, for his mentorship and patience throughout the PhD, serving as a significant inspiration for my aspirations, Dr. Christopher Lowe, whose unwavering support and patience was an invaluable tool to realising my PhD goal, and Dr. Ruaridh Clark, who challenged me to become a better and greater researcher both in thinking and presenting the research work. This work would not have been possible without their unwavering support, patient guidance and insightful advice.

I sincerely and wholeheartedly thank my family for their essential support, and particularly Mrs. Purity Wanja, my mother, and Prof. Mwakio Tole, father, who always encourage me and anytime I felt like giving up they cheered me on, I will always be grateful for their emotional support and comfort. My sincere thanks also to my brothers, Davies Tole, Mwawuganga Tole and Samuel Mwaniki, as well as Stacy Mrunde and Dr. Nkirote Kalaine, thank you for always checking up on me to find out how I was fairing on. To my best friend Elizabeth Wachira, and her sisters, whose patience and encouragement during the writing up of this thesis was invaluable, I am forever grateful. To my other best friend, Grace Wangari, my friends and colleagues, Bonface Osoro, Fiona Jepleting, Charlynne Jepkosgei, Prof. Faustin Ondore, and many others from TUK and beyond as well as my relatives, I am grateful for the support, and counselling throughout my PhD. I also thank my therapist, for the help and guidance that helped stabilise my mental health. I am grateful to many more who include everyone who crossed my path during this PhD journey and those I knew even before starting the PhD like folk from the Sandyford Henderson Memorial church family and the Wellington church International Welcome team who have always been my home away from home at Glasgow.

Lastly but in no means least, words cannot express my gratitude to the UK Space Agency who sponsored me and enabled me to embark on this PhD journey. It would not have started, nor been realised without their financial support.

Abstract

Space has become increasingly congested, commercialised, and inter-connected over the past few decades. The "*New Space*" era and the use of federated space systems are intended to make space more affordable and accessible to more users. The use of agile space has also increased in the past few decades. To sustain this increase, new, fast, efficient, and robust methodologies are required for satellite orbit designs, constellation determination and satellite tasking paradigms. Additionally, methods aiding the increase in the value of assets already in space are also a valuable contribution to the space industry especially in reducing the rising space congestion and *"space pollution"*. The asset value increase may be found in filling coverage gaps or supplementing the functionality of existing space population for diversified service providers. A rapid, robust, and efficient approach for orbit design/determination based on desired target coverage is herein proposed. The proposed method is fully analytical and of the so-called *embarrassingly parallel* nature. The fully analytical property eliminates the necessity for iterative computations resulting in increased efficiency. The *embarrassingly parallel* nature makes the method analyse each orbital element and time combination within a given search space independently allowing for efficient parallel execution. This provides a basis for robust multi-objective optimisation of the determined orbits. The development of the method and some application scenarios while considering both single and multi-objective missions are presented. The research presented also develops a novel graph-theory based method for multi-satellite tasking which is aimed at value addition to satellites already in space. The interaction between satellites in different orbits and ground targets is modelled using bipartite networks. Determination of the satellite(s) to be tasked is based on the optimisation of different requirements and this can be modelled as a combinatorial network problem. The developed method uses an analysis of static bipartite graphs to determine the optimum satellite and ground target interactions based on the mission objectives, referred to as tasks. Optimum satellites for various tasks are determined using a combination edge weight and graph structure analysis. The network developed give insights for scalable analysis of options involving multiple ground targets and satellite options. To illustrate the application of the method, different networks are studied; simple networks and complex networks considering complexity both in terms of number of satellites and ground targets. This method proves to be fast, simple, efficient, and robust in determining optimal satellites for multiple objective function tasking.

Contents

List of Figures

List of Tables

List of Symbols and Acronyms

Roman Symbols

$a -$ $semi\ major\ axis, m$

$D -$ $number\ of\ days$

$e -$ $eccentricity$

$i -$ $orbital\ inclination, deg$

$J_2 -$ $coefficient\ of\ the\ Earth's gravitational\ zonal\ harmonic\ of\ the\ second\ degree$

$J_t -$ $Mean\ Duration\ of\ view\ objective\ function$

$J_{ts} -$ $Mean\ Number\ of\ times\ seen\ objective\ function$

$J_{RT} -$ $Time\ to\ next\ visit\ (Revisit\ time)\ objective\ function$

$H -$ $Altitude, m$

$M -$ $mean\ anomaly, deg$

$n -$ $Mean\ Angular\ Velocity\ or\ Mean\ Motion$

$R -$ $revolutions$

$R_E -$ $Radius\ of\ Earth, km$

$t_0 -$ $Initial\ time\ or\ Epoch\ Time, s$

$t -$ $time, s$

$T -$ $orbital\ period, s$

$T_\Omega -$ $nodal\ period, s$

$u -$ $argument\ of\ latitude, deg$

$d_a -$ $Earth\ Central\ Angle, deg$

Greek Symbols

$\varepsilon -$ *eccentricity*

$\delta -$ *latitude, deg*

$\Psi -$ *longitude, deg*

$\mu_E -$ *standard gravitational parameter of the Earth,* $\frac{m^3}{s^2}$

$$\mu_E = 3.986004418 \times 10^{14} \frac{m^3}{s^2}$$

$\omega_E -$ *angular velocity of the Earth, rad/s*

$\omega -$ *argument of perigee, deg*

$\theta -$ *true anomaly, deg*

$\Omega -$ *right ascension of the ascending node, deg*

$\Omega_s -$ *right ascension of the ascending node, deg*

$\Omega_0 -$ *right ascension of the ascending node, at Epoch deg*

$\Omega_{et0} -$ *greenwhich apparent sidereal time at epoch, deg*

$\gamma -$ *supplementary angle, deg*

$\beta -$ *spacecraft elevation angle, deg*

$\lambda -$ *effective Earth central angle, deg*

$\eta -$ *half effective angle, deg*

Acronyms/Abbreviations

ALT – Altitude, km

AoL – Argument of Latitude

AoP – Argument of perigee

DE – Differential Evolution

deg – *degrees*

Ecc – eccentricity

FoR – Field of Regard

FoV – Field of View

FSS – Federated Space Systems

GMAT – General Mission Analysis Tool

GAST – Greenwich Apparent Sidereal Time

INC – Inclination

LAT – Latitude

LON – Longitude

MOO – Multiple objective function optimisation

MTMP – Multi Tasking Proposed method

OF – Objective Function
PoI – Point of interest

RAAN – Right Ascension of Ascending Node

rads – radians

sec. – *seconds*

SA-DE – Self Adaptive Differential Evolution

SDG – Sustainable Development Goals

SGP4 – Simplified General Perturbations-4

SVD – Singular Value Decomposition

Sol – Solution

SMA – Semi – major axis

SSP – Subsatellite Point

STELA – Semi-analytical Tool for End-of-Life Analysis

$TA -$ *True Anomaly*

Subscripts

$E -$ *Earth*

$et0 -$ *time at epoch*

Chapter 1

Introduction

Satellite technology currently plays a big part in the modern world. Satellites are used for different purposes such as communication, Earth observation, weather monitoring, navigation, and research, amongst other uses [1, 2]. For decades, with the increase in the use and need of satellite technologies, many methods have been developed to design satellite orbits, and satellite constellations, based on the mission requirements such as desired coverage, revisit frequency, etc. Some current missions have used the Two-Line Elements (TLE), and different software such as the General Mission Analysis Tool, GMAT, and systems tool kit, STK to determine their orbits, then optimise these using different numerical methods [3-5]. The accuracy of tools like TLE are still however being developed [3]. More so, now with smaller satellites being used [6], and targeting smaller areas, the desire for increased precision in orbit design is continuously increasing [7, 8]. There is a need for satellite orbit designs that facilitate the overflight of specific targets with as much accuracy as possible [9]. Moreover, with the end of the decade of action fast approaching, satellites are used to monitor and evaluate the indicators of sustainable development goals (SDGs) and for this, the connectivity, the localisation, and the digitalisation of global areas is required [10, 11]. Constellations such as the first Sustainable Development Goals Satellite (SDGSAT-1), have been launched for monitoring the SDG indicators, and some areas have been found to still lack coverage and even very basic internet connectivity [12]. This means there is still a need to fill in coverage gaps. Filling these gaps require methods that get away from the traditional patterns for global and regional satellite orbit and constellation designs, and determination, as well as development of more robust methods for tasking satellites.

There are continuous efforts being made to develop the designs of optimum (or *"best"* depending on the requirements) responsive satellite orbits [9]. The developing designs are geared towards filling various coverage gaps, getting clearer images, trying to get as much spatial and temporal coverage as possible, maximising or minimising revisit times [13], and supporting the developments of agile space systems. The use of agile space has significantly increased in the past few decades [1]. Some agile space systems involve missions such as, tracking mobile ground targets, disaster imaging, and other emergency purposes [14-16].

These missions facilitate informed response in cases ranging from disaster assessment and urgent data collection in areas lacking connectivity.

The application spectrum of small satellites has significantly increased, and fairly broadened as they have become a powerful tool especially for global, regional, and local monitoring [17]. This can be attributed to their low cost of operation, and the recent emergence of “New space” [17, 18]. The low cost can be ascribed to the recent development of the Federated Space Systems (FSS), [19] which is aimed at improving the performance of core functions of the satellites or satellite constellations that have different missions and are operated by members of the federation [20].

1.1 Motivation

In some cases where orbit designs involve overflying a specific target area, the traditional approach is to firstly determine the orbital parameters that yield an orbit whose ground track almost facilitates an overflight. Subsequently, the parameters are optimised to obtain the optimal orbit for the desired ground track [15, 21]. When orbits that overfly specific target points or areas are required however, some previously used orbit design algorithms can eventually get the precise orbits but the compromise between accuracy and computational costs can be a disadvantage [22, 23]. Moreover, some numerical methods have tendencies to get stuck in local optimums hence missing the global optimums which might have better solutions [24].

For cases like response to on-going disasters or monitoring of complicated scenes, (e.g., natural disasters or meteorological phenomenon such as, hurricanes and tornadoes), fast and accurate, localised targeting is desired [15]. These are challenges that can be solved with satellite orbits designed with high degrees of accuracies to overfly the specific areas [15]. For example, if monitoring the eye of a hurricane to categorise it, a method that accurately determines the orbit that will overfly the next predicted position of the eye is needed. The gap of developing a fast and accurate orbit design methodology is still at hand. Despite the reduction of satellite operating costs due to the emergence of FSS and the “New Space” era in the recent decades, the pressure to achieve the mission objectives within the short lifespan of the satellites with operation cost efficiency has also increased. This enhances the need for more accurate and precise orbit designs that can achieve multiple objectives within a given period. Additionally, the search space of such federated missions may be large as mission requirements such as monitoring an on-going disaster may require determination of orbits that facilitate overflight of different localised parts of the globe. The urgency of each mission requirement, herein referred to as "*objective function*", and an understanding of the full search space is therefore crucial to getting to the desired orbits as fast as possible. For such cases, some numerical methods incur high computational costs whereas some fully analytical methods may compromise on the accuracy due to main assumptions related to the equations of motion used [25, 26]. The implementation of a general perturbation approach while minimising the assumptions is studied in the development of the proposed analytical orbit design method.

The need for multiple objective function optimisation methods that are robust have also been highlighted in the past few decades. Some orbit design methods require re-calculation of the orbits for each objective function to be optimised. Therefore, there is a gap for a method that would be more computationally efficient by determining the orbits once and use the results to optimise for multiple objective functions. Some real-life missions have different requirements hence require multiple objective function optimisation. Numerical methods such as differential evolution have been used for optimisation of one objective function [27], and the addition of adaptive grids enable them to be used for multiple objectives, but with a higher computational cost [27]. The analytical orbit design method proposed in this dissertation is aimed at enabling operators to explore multiple objective functions and still maintain computation cost and time efficiency. In some cases, an orbit that performs well over multiple objective functions could be preferable to an orbit that only maximises a single objective function. This leads to a selection of the "*robust*" orbit for multiple objective functions even if not the "*optimal*" for the individual objective functions. In a case where multiple objective function optimisation is needed, this work develops algorithms for robust orbit determination. For the same search space values and targets, the method developed in this work does not require re-simulation of the orbits, even when the objective function changes.

An addition of a multi-level adaptive grid to the developed analytical method is proposed. The adaptive grid addition aims to increase the efficiency of the developed method in terms of the computational time needed to determine the optimum orbits for large search spaces. Responsive satellites can be said to have dynamic tasks. A gap of methods aimed at tasking reconfigurable satellites is identified. Fast, accurate and efficient methods for satellite tasking need to be developed especially for agile space systems and maximisation of existing space assets. Using some of the orbital solutions obtained from the analytical method proposed in this dissertation, a graph theory method is proposed and developed with an aim to task satellites that are already in space. This is to maximise on value of the existing space population for any emerging tasks or new users. The graph theory method presented proves to be robust due to its applicability to multiple tasks hence, making it computationally efficient.

1.2 Orbit Design Methods

Orbit propagation techniques are essential tools in orbit design methodologies. Orbit propagation is an area that has been under research for decades and is used to determine the position of an object in space [28]. The methods of orbit propagation are divided into special perturbation methods and general perturbation methods, and they can either be numerical, analytical, or semi-analytical [29]. Special perturbation methods mostly use numerical integrators that provide accurate prediction of the positions and velocities of space objects. The disadvantages of such methods lie in the processing, where stepwise calculations are used from epoch to epoch [29-31]. Despite currently having more methods that have variable steps such as the Runge-Kutta-Fehlberg method, RKF45, see ref [32], such methods have a significant increase in computational costs. For more accuracy, both the computational costs and the computational time increases significantly for some numerical methods.

General perturbation methods make use of analytical equations of motion. Eliminating the stepwise processing technique of numerical methods enables them to compute solutions more generally, and the use of analytical equations have higher computational efficiencies. Their accuracy is however reduced due to the use of some restrictive assumptions [29-31] [33]. Semi-analytical methods combine both the general and the special perturbation methods depending on the period being considered [29]. Some third-party software such as the French National Centre of Space Research’s, Semi-analytical Tool for End-of-Life Analysis (STELA), use a semi-analytical approach [34].

For purposes of mission design as well as orbit determination, orbit propagation methods are required to be fast and efficient [29]. An example of this is the general perturbation models which are used in the Simplified General Perturbations-4 (SGP4) orbit model [35] and provide the basis for propagating the ‘‘2-line” orbital elements. R. Flores et. al.,[29] propose a method to improve the computational performance of some numerical methods while still maintaining the level of accuracy, by adjusting the number of geopotential spherical harmonics during the integration, and determining the allowable acceleration error. With the use of a Molniya orbit, R. Flores et. al., maintain the accuracy of their solution while reducing the costs of computation by a factor of 3 or more. R. Flores et. al., however, find some computational inefficiencies for some cases such as when using dynamic expansion, where an increase in accuracy of the baseline solutions by two orders of magnitude requires a 90% rise in computational costs [29].

Orbit design methodologies have been developing for decades with some past designs having focused on improving the efficiency of existing numerical methods, and some on developing semi-analytical methods [13, 23, 25, 36-38]. Orbit design methods are based on numerous criteria mainly related to satellite coverage [39], revisit strategies and repeat ground tracks amongst many others [21, 39-44].

1.2.1 Orbit Designs Based on Coverage

Traditionally, the main criteria of satellite orbital design methods focus on the type of coverage desired which was initially divided into global and regional coverage. As observed by G. Dai et. al., [39], Earth observation by satellites and the data that needs to be transmitted between the satellites and the users are both related to satellite coverage. The main criterion for orbital designs is the type of coverage desired.

At first, the desire to have global coverage dominated most design methodologies. Methods such as the Walker and the Flower methods are developed for constellation designs aiming at continuous global coverage[45, 46]. An example of a design for single satellite global coverage is presented by scholars like M. Xu et. al.,[47]. M. Xu et. al., formulate an analytical algorithm for global coverage using a revisit orbit and apply it to a mission revisiting all points on the Earth within long periods of time. As an extension to this but using constellations, S. Y. Ulybyshev et. al., [48] present an analytical technique based on basic solutions obtained by previous authors. S. Y. Ulybyshev et. al., use those solutions for a single spacecraft to calculate the optimum argument of latitude and the right ascension of ascending node between satellites in their constellations. With this, S. Y. Ulybyshev et. al., minimise the revisit for any point on the Earth's surface ensuring that the constellation has an operational global monitoring with a one-day repeat ground-track. The technique S. Y. Ulybyshev et. al. develop can only be used for orbits with direct and inverse inclinations for fields of view that cover polar points. S. Y. Ulybyshev et. al. suggest future work to be done using similar methods developed for orbits that have higher multiplicities than theirs. On the other hand, P. Zong et. al, [49] present a constellation global coverage design in one revisit time. In their work, P. Zong et. al, use a model of constellations which are fully connected by crosslinks and maintain continuous communication. P. Zong et. al, find that their designs are closer to optimal coverage of the globe, compared to results of some previously used methods.

The design of regional coverage satellites later on emerge and some of the algorithms used are such as the grid point algorithms and latitude strip algorithms amongst many others[50]. Regional coverage has become more attractive to research than global coverage. Z. Song et. al., [51] study the option of regional coverage and develop a novel grid point approach for efficiently solving the constellation to ground-regional coverage problem by giving a strategy that addresses some challenges of the previously used grid point approach. Z. Song et. al., in Ref. [52], give an approach to further solve the constellation to ground coverage problems using three main judgement theorems. Z. Song et. al., develop the judgment theorems to evaluate the coverage of satellite constellations and they achieve higher computational efficiency compared to more traditional methods such as grid point approaches. However, the method by Z. Song et. al., is only applicable to constellations that have complete ground region coverage.

Methods based on overflying sets of target areas and target points have more recently been developed especially due to Earth observation using responsive Low Earth Orbit (LEO) satellites [25]. Many numerical methods have been studied on how to design satellite orbits based on specific desired targets. T. Li, J. Xiang et. al., [22], design a method to address some problems of previous coverage area based numerical methods such as the computational cost, as well as, failure to obtain desired orbits by various methods. T. Li, J. Xiang et. al., use their method to determine the existing conditions of the solutions to revisit orbits by use of a special kind of repeating ground track orbit, which T. Li et. al., refer to as a circular revisit orbit (CRO).

Some methods for calculating optimal orbits focus on the duration of view of target areas, whereas some focus on the amount of time that it will take to revisit the target as in [6]. Prior to this however, Y. Chen et. al., in [53] consider a satellite orbit design using a set of target sites which need to be visited within an assigned period and the optimisation of the orbit design considers target priority and duration of observation time. Y. Chen et. al use a self-adaptive differential evolution (SA-DE) algorithm. This algorithm is an improvement to previously used differential evolution algorithms, and this is demonstrated in their regional and global case studies, as Y. Chen et. al show that the SA-DE is a more suitable optimizer for orbit design. Later, however, M. Pu et. al., [6] present an orbit design approach by combining a robust optimization model with a multi-objective optimization algorithm. M. Pu et. al., use Monte Carlo simulation results to demonstrate that the robust orbit solutions are more reliable compared to nominal orbit solution designed from the traditional single objective stochastic optimization algorithms. In Ref. [38], G. Zhang et. al., use approximate analytical solutions for single and dual coplanar impulsive maneuverers to observe given Earth sites for exact overflight and conical sensor cases. G. Zhang et. al., modify a known orbit and use impulse approaches to be able to get to the desired ground track. G. Zhang et. al., extend their work to a three-impulse method associated with bi-elliptical transfer and four impulse methods. G. Zhang et. al., later on formulate a semi analytical method to obtain the longitude differences, and then use approximations to get to the desired orbit as presented in Ref. [54].

1.2.2 Agile Space System Orbit Designs

To enable better temporal coverage, there has been a recent rise in the desire to have more localised coverage, as global coverage focuses more on spatial performance[1, 25, 55]. The desire to overfly more specific target points and target areas gave rise to responsive satellites [25]. These are becoming more and more attractive to the space industry as the world, and its needs keep changing. Monitoring and predicting changes and disasters is an area of special interest to research at large. Mission requirements for responsive satellites change overtime. These changes include coverage area changes, reconnaissance changes, revisit schedule changes amongst others. Most of the sustainable development goals that are satellite technology based depend on such monitoring hence, making it very valuable. Monitoring the change of an area requires accuracy of overflight. Different numerical methods have been studied on how to design a satellite orbit based on specific desired targets. Indeed, Earth observation using LEO satellites that are more dynamic has become a common main objective. The envelope curve method by [50] design a coverage boundary about which a specific area target can be obtained by the intersection of the target area and envelope curves. To minimize the number of satellites in a constellation as well as to achieve a set maximum revisit time, C. Zhang et. al., [56] formulate a method to design a LEO constellation to eliminate long revisit time, and large number of satellites which is a problem faced by traditional design methods.

Z. Song et. al., [1] develop an agile satellite orbit design that considers spatial resolution and temporal resolution simultaneously and use a repeat ground track orbit in an accurate geopotential model of the Earth. Z. Song et. al., formulate a multi-objective optimization technique for obtaining the optimal feasible orbits. These objective constraints include the maximum repeat cycle for the satellite, the maximum tilt angle and full coverage at the equator. Other numerical methods such as the self-adaptive method, self-adaptive differential method and many more have been developed further for purposes of target overflight. W. Yao et. al., [57] for example, present an improved differential algorithm that is applicable in orbit designs. W. Yao et. al., use a double self-adaptive differential evolution (SA-DE) algorithm with a random mutant which is an improvement of the work by Y. Chen et. al., [53]. W. Yao et. al., [57] apply a random mutant as well as introduce a double self-adaptive scaling factor to the traditional differential evolution algorithm. The scaling factors of the proposed algorithm are found to be able to adjust with the optimization procedure, and this makes the algorithm able to jump out of the local optimal. Different from the previous research, the self-adaptive scaling factors in the work by W. Yao et. al., can be affected by not only the number of generations but also the fitness function of the last generation. When the algorithms are applied to several test function studies including low dimension and high dimension and compared with the other algorithms, the simulations demonstrate that the advanced algorithm give a better performance in solution accuracy, convergence, and the results' standard deviation. The case studies presented by W. Yao et. al., prove that the novel self-adaptive algorithm with random mutant can provide an improved performance on multiple targets and manoeuvre optimal problems than others.

1.3 Objective Function Optimisation in Orbit Designs

Just like in many real-life applications, orbit designs are optimised using several mission requirements which are equivalent to multiple objective functions [27]. For single objective function optimizations, methods such as differential evolution have proven to be efficient [27]. W. Yao et. al., [57], use a single objective function relating to the observation duration of targets, and optimise the orbits using an improved differential evolution method.

In the past couple of decades, there has been emergence and development of Federated Space Systems. This means that space systems have now geared towards agile space with different capabilities and missions[19]. The study of multi-objective optimisation in orbit designs is therefore essential. S. Ghorbanpour et. al., in [58] modify a differential evolution algorithm by using an adaptive grid for the multi objective optimisation. S. Ghorbanpour et. al., extend the mutation strategy that is used in single objective function optimisation to multi-objective optimization using the modified method. The main contribution in the work of S. Ghorbanpour et. al., is adding the adaptive grid to the commonly used traditional differential evolution algorithm. Other studies such as the one by H.-D. Kim et. al., [59] also aim to achieve multi-objective optimisation based on the average revisit time and average transmitted power. H.-D. Kim et. al., use multi-objective heuristic algorithms which include genetic evolution, particle swamp optimisation, and differential evolution algorithms, then compare the results from them. H.-D. Kim et. al., find that the developed and modified differential evolution algorithms obtain more efficient results than generic algorithms, particle swamp optimisation and analytical approaches.

In this work, multiple objective function optimisation is studied and the advantages of using the developed analytical method in analysing different objective functions is highlighted. In chapter 4 of this dissertation, an adaptive grid is introduced in the optimisation phase for enhanced computational efficiency. This is applied to both single and multiple objective functions and examples of the simulation results are given.

1.4 Design Optimisation Using Adaptive Grids

Adaptive grids have been incorporated into methods such as differential evolution and genetic algorithms to improve their performance. J. Cheng et. al., in ref. [27] for example, use a grid based adaptive algorithm to improve the dynamical adjustment of convergence and diversity of a differential evolution algorithm. The method by J. Cheng et. al, exploit the feedback during the evolutionary process and base their grid on three metrics i.e., the grid fitness, the grid density, and the grid objective-wise standard deviation. J. Cheng et. al, compare their method to nine other algorithms based on differential evolution and genetic algorithms, and find their method to be better at optimising multi-objective function problems. Traditional particle swarm optimisations are said to have inefficient convergence as well as a tendency to get stuck in local optimums [60, 61]. To solve the problem of particle swarm optimization (PSO), getting stuck in local optimums and having low convergence rates, K. H. Rubinder Mannan et. al., [61] propose an adaptive particle swarm optimisation algorithm based on directed and weighted complex networks. The method by K. H. Rubinder Mannan et. al., improves the convergence efficiency by having a higher convergence rate than some other particle swarm optimisation methods. In Ref. [60], J. E. Harries et. al., also propose an adaptive response strategy to activate stagnated particles so that they would not get stuck in local optimums.

The application of adaptive grids can also be implemented while using analytical methods to increase the computational efficiency of the method by refining the grids in the areas of interest hence, reducing the search space. This approach is explored herein as a multi-level adaptive grid which is implemented into the developed orbit design method.

1.5 Satellite Tasking Methods

In most cases, satellite tasking also involves scheduling. This has been done previously using different heuristic algorithms aimed at selecting the most urgent tasks. Satellite tasking and scheduling is an NP hard problem that has been recently achieved using different numerical algorithms to task and schedule the satellites [62]. S. Liu et. al., [63] for example, study the tasking of intelligent satellites for earth observation using a linear programming algorithm and a heuristic search algorithm, which when compared with some previously used search algorithms have better results in terms of profitability.

Graph theory is an area that has been under research for centuries even dating back to the seventeenth century [64]. It has been commonly used to find solutions to networking problems used in cases such as traffic systems, internet connectivity and the medical field amongst many others. Connecting a task to a satellite is a combinatorial problem, that can use graph theoretical methods to be solved. The work presented in this dissertation proposes the use of graph theory in tasking satellites already in space. This aims to maximise the assets that are already in space and reduce the emerging space congestion. At the same time, this can be profitable to an operator if new users emerge. The proposed method proves to be robust and can be used to identify satellites that optimally perform multiple tasks.

1.6 Gaps Identified in the Literature

Literature review of existing satellite tasking, orbit design and propagation algorithms, reveal a need for the development of methodologies that facilitate rapid orbital solutions of acceptable accuracy levels [29]. An identified research gap exists in orbit design methodologies that give fast and accurate orbital solutions for overflights of both static and moving targets within a defined field of regard.

The literature review highlights a gap of orbit design methods that offer fast comprehensive search space analysis enabling informed decision making for engineers and operators regarding optimal orbits. This research aims to address this gap by development of an orbit design methodology that will enhance computational time efficiency for large search space analysis. The application of the method presented in this work for example includes an analysis of orbits with inclinations from 50 degrees to 130 degrees and right ascension of ascending node of 0 to 360. Some previous orbit design methods only consider prograde orbits, due to computational time and cost constraints. The developed orbit design method is analytical and uses an *embarrassingly parallel* approach. An *embarrassingly parallel* method is an approach where there is elimination of reliance between the processes and so the results are independent of each other. In the developed method, the results of a certain timestep is generated without knowledge of the previous timestep results because the time, the inclination, and the right ascension of the ascending node are all analysed in parallel. The approach of the algorithm and implementation are further explained in section 2.2.2. A fast general overview gives engineering and design insights into complex search spaces that may show, for example, that some optimal solutions might not necessarily be the most robust solutions for different objective functions. The insights can also guide where an orbit designer can concentrate more to suit various design needs.

Some previous numerical orbit design methods require a re-run of the orbit determination simulations when the objective functions change. The developed orbit design method in this dissertation is efficient in that different objective functions can be explored without the need to re-run the simulation to determine the orbits because, the orbital elements are analysed in parallel and so is time. The method is generalised and independent of specific optimisation techniques while calculating the orbits. This means that the proposed method is robust for Multi-Objective Function Optimisation (MOO).

With the emergence of “New Space”, there is an increased desire for robust multi mission, multi objective and multifunctional satellites [65, 66]. Some previous orbit design methods that use numerical simulations such as differential evolution have been excellent optimisers for single objective functions [27]. For such methods to be used for multi-objective optimisation (MOO), additional computations are needed see Ref. [27]. Adaptive grids are used in some methods such as the one by S. Ghorbanpour et. al. [58], who add an adaptive grid based mutation to differential evolution in order to get MOO. Considering the high computational cost of numerical methods, such additions mean adding to these costs.

The adaptive grid added to the proposed method enhances the computational efficiency attained by refining the grid on the main areas of interest of a large search space. The analytical method can also be implemented with other numerical methods to increase the accuracy.

The lack of robust reconfigurable/agile satellite tasking strategies has been established. Despite the recent development of using smaller satellites, hence reducing costs, it would be more cost efficient and space sustainable for new emerging users to use the pre-existing space population than launching their own satellites. Some of the previous satellite tasking methods are ad hoc and not applicable to diverse scenarios or cases. A graph-theory based method is herein proposed for tasking in-orbit satellites to achieve user specific mission requirements even in a case where a new user emerges, or if the mission requirements change. The method takes advantage of insights that can be drawn from a graph network of satellites in orbit after ground propagation within a given field of regard and the desired ground targets.

1.7 Aims and Objectives of the Presented Work

In relation to the gaps identified in literature, this dissertation addresses the following issues and research questions:

i. Develop a novel fast analytical embarrassingly parallel orbit design/determination methodology (a method that parallelises the orbital elements and time for robust results). This is developed throughout Chapter 2 of this dissertation and highlighted in section 2.2.2.

ii. Develop a robust orbit design method that gives fast general insights into complex search spaces (search spaces that include both prograde and retrograde orbits for clearer insights) hence improving on exhaustive search algorithms. This is illustrated in section 2.3 where the method is used for a search space containing both prograde and retrograde orbits.

iii. Perform the optimisation of multiple objective functions without a need to re-simulate valid overflight orbits hence, an added efficiency to computational time and cost when considering multiple objective functions. The simulations in section 3.3 have been performed without the need to recalculate the orbits.

iv. Introduce an adaptive method to the developed orbit design method to increase its computational efficiency and still obtaining the optimum orbits (this also shows that the method can be used with a numerical algorithm once the area of interest is identified to increase the results accuracy). This is presented in chapter 4.

v. Develop a novel graph theory approach that can be used for tasking satellites for different/multiple user requirements with applicability to both new space populations and pre-existing space population. This can be used to determine optimum satellites for multiple tasks as is shown in chapter 5 of this dissertation.

In summary, the work herein proposes an analytical, *embarrassingly parallel* method that aims to fill in some of the gaps identified in literature that are related to previous orbit design methods. The developed analytical method determines orbits that are aimed at overflying specific target areas. Kepler's equations and spherical geometry equations are used for the calculations that relate the satellite orbit and the ground target points as derived by [67]. These equations have been previously used to find orbital solutions and obtain subsatellite points for various solutions. [25] for example, used the equations in the designing of their analytical method for reconfiguration of satellite constellations using low-thrust manoeuvres, and the work developed in this dissertation shares many of their advantages. The developed method is novel and a contribution to knowledge as it determines orbits that have greater values of an objective function when compared to a previously used method (see section 3.1). The results from the proposed method can also be used to optimise multiple objective functions without the need to recalculate the orbits for a given search space (see section 3.3). The work herein also explores the use of graph theory for satellite tasking based on required objective functions (see Chapter 5).

Chapter 2

Development of an Analytical Orbit Design Methodology

This chapter aims to fill in some of the gaps that are identified and highlighted in section 1.6. This aim is achieved by developing a methodology that; determines orbits based on desired ground target overflight, gives insights to large orbital search spaces with minimum computational time and cost, and is applicable to diverse and complex scenarios.

The proposed methodology is developed from general perturbation techniques and so it is analytical. Generally, analytical algorithms give deeper insights to the physical mechanisms of a system [29]. The method in this chapter shares this advantage by giving full insights to both small and large search spaces, with minimised computational time compared to some previous orbit design methods. The method develops previous analytical orbit propagation algorithms and uses the general basic equations of motion derived by J.E Harries [67]. Compared to some analytical orbit design methods, which make assumptions of a circular Earth, the method developed in this chapter includes the Earth's secular perturbations of the first zonal harmonic, J_2, for increased accuracy. For the method in this chapter however, other secular perturbations such as atmospheric drag are not considered. Validation of the developed method's accuracy is done by analysing the error between ground-tracks of the orbits obtained using the proposed method against those from simulating the same orbital values on a third-party software which uses numerical methods. This validation is presented in section 2.3, and the National Aeronautics and Space Administration's, (NASA's), General Mission Analysis Tool GMAT is the third-party software used.

The method developed herein also has an embarrassingly parallel nature which includes parallelising time. The factor of making each time-step parallel means that each orbital element, i.e., each inclination, right ascension of the ascending node and semi-major axis combination is evaluated against each time-step value after epoch; this is further explained in section 2.2.2. The embarrassingly parallel nature gives the developed method an added advantage over some previously used algorithms as it eliminates the need of re-simulating the orbits when optimising them for different objective functions; this is analysed in chapter 3.

2.1 Main Orbit Design Equations

The method proposed and developed in this chapter is an orbit design methodology based on desired ground passes. The method determines the main orbital elements, i.e., the inclination (INC), the argument of perigee (AOP), the right ascension of the ascending node (RAAN), and the semi-major axis (SMA), by considering sets of known longitudes (LON) and latitude (LAT) points that are to be overflown. This study is based on a circular orbit and so the Argument of latitude, (AoL), is considered because the argument of perigee is not defined. Generally, values of classical orbital elements (COEs) are used to determine orbits that facilitate the overflight of specific targets. Unlike traditionally used propagation algorithms, and orbit determination methodologies, in this work the target points, i.e., the longitudes and latitudes, are used as inputs to determine the orbital elements that satisfy overflight.

Previous work by authors such as C. N. McGrath et. al., [25], give derivations and equations for calculating the longitudes and latitudes from orbital elements i.e., ground track propagation. A method to reciprocate this is initially studied in this chapter. The algorithm takes the equations by Refs. [25, 67] as the initial ground point calculation method, then the equations are developed to make the longitudes and latitudes the inputs, and the orbital elements satisfying the overflight of these ground points the outputs. Generally, an orbital search space considered to facilitate an overflight of a given target would be infinite, and so to make the method feasible, a search space constituting the orbital elements and time range is also defined.

2.1.1 Initial Orbit and Ground-Track Propagation Equations

The initial calculations of the method are based on the ground track propagation equations presented in Ref. [67], where the inputs are values of inclination, Argument of Latitude, Eccentricity, and the Period (calculated from the semi-major axis). Some of the main assumptions applied to the initial development of the method include,

i. The orbit is circular; Eccentricity (e) = 0,

ii. The orbit calculation is from epoch, therefore starts at time, t = 0 to a given/desired value of t,

iii. The J_2 perturbations are neglected for the initial calculations.

The calculation steps are as presented in equation (2.1) – (2.6) which include some basic orbital equations.

As derived from Kepler's laws by authors such as Ref. [68], the period of the orbit is given as,

$$T = 2\pi\sqrt{\frac{a^3}{\mu_E}} \tag{2.1}$$

Where a is the semi-major axis and μ_E is the standard gravitational parameter of the Earth.

From the mean angular velocity also known as mean motion constant [67],

$$n = \frac{2\pi}{T} \tag{2.2}$$

the Mean Anomaly is,

$$M = \frac{2\pi}{T}t \tag{2.3}$$

The argument of latitude of a circular orbit is equal to the true anomaly, TA, which is the same as the Mean anomaly (for a circular orbit), and is given as,

$$u = 2\pi t \Big/ \sqrt{\frac{4\pi^2}{\mu_E}a^3} \tag{2.4}$$

And the spherical geometry orbit subsatellite points as derived by [67], is given as,

$$\delta = sin^{-1}(sin(i)\,sin(u)) \tag{2.5}$$

$$\Psi = tan^{-1}\left(\frac{cos(i)\,sin(u)}{cos(u)}\right) + \Omega_0 - \Omega_{et0} - \left(\frac{d\Omega_e}{dt} - \frac{d\Omega}{dt}\right)(t - t_0) \tag{2.6}$$

where δ is the Latitude, Ψ is the longitude of the subsatellite points, t_0 is the time at epoch (t==0), i is the inclination, Ω is the right ascension of the ascending node of the satellite, Ω_e is the right ascension of Greenwich, and $\left(\frac{\mathrm{d}\Omega_\mathrm{e}}{\mathrm{dt}}\right)$ is the relative rotation of the Earth, ω_e, relative to the orbital plane [67].

For the initial calculations, the following additional assumptions are made.

i. the rate of change of RAAN, $\frac{d\Omega}{dt}$, is not considered,

ii. the Greenwich apparent sidereal time (GAST) at epoch, Ω_{et0} , of the orbit is at 0 degrees, and the longitude of ascending node at epoch time, given by Ω_0 is also 0 degrees.

With these assumptions, the subsatellite points are then calculated as,

$$\delta = sin^{-1}(sin(i)\, sin(u)) \tag{2.7}$$

$$\Psi = tan^{-1}\left(\frac{cos(i)\, sin(u)}{cos(u)}\right) - \omega_e t \tag{2.8}$$

from equation (2.7) and (2.8) the ground track is propagated and can be plot from the subsatellite points. In this work, the simulation is done on MATLAB R2020, and using the values presented on Table 2.1, the ground track Figure 2.1 is obtained. The calculation does not consider any perturbations and so the ground track presented in Figure 2.1 may be accurate for short periods but for long periods, the error accumulates making the inaccuracy increase, this is further analysed in section 2.2.

Table 2.1: Orbital element values used to simulate the initial ground-track.

Input, units	Value
Inclination, deg	70
Argument of Perigee, deg	0
Semi-major axis, meters	7039000
Earth’s rotation rate, rads/sec	7.292115e-5
Start time - End time, sec	0 – 10000
Eccentricity	0

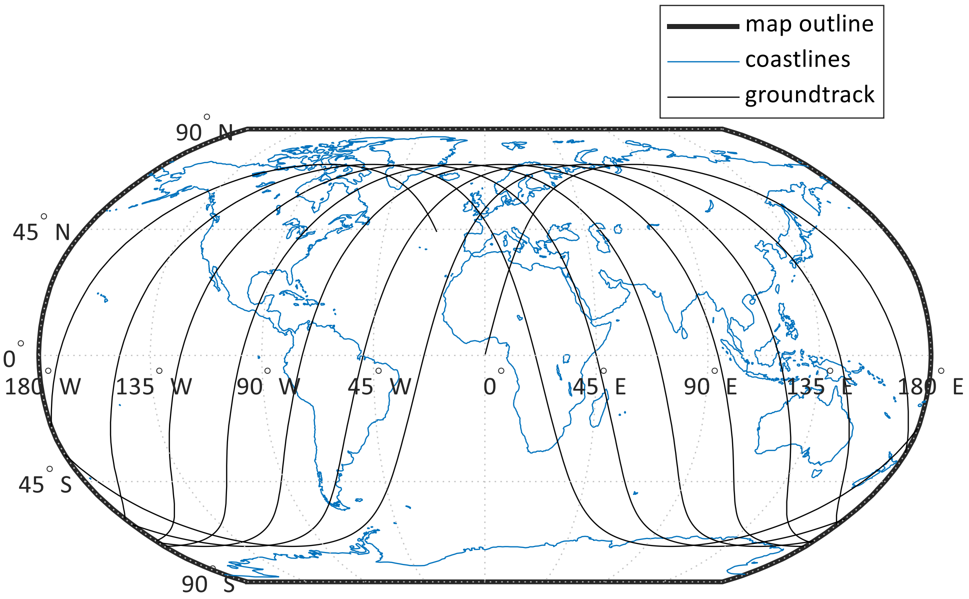


Figure 2.1: Initial Ground track from equation 2.8

From equation (2.1) – (2.8), the method is further developed to have the latitudes and the longitudes as inputs, and the values of the valid orbits in terms of their classical orbital elements, as the outputs. The results from both the initial orbit propagation equations and the developed method when compared and analysed verify that the developed method results are as expected, and this is done in section 2.1.2.

The development of the inverse method equations is as follows; considering two known ground points (δ_1, Ψ_1), and (δ_2, Ψ_2), equation (2.7) and (2.8) are,

$$\delta_1 = sin^{-1}(sin(i)\, sin(u_1)) \tag{2.9}$$

$$\delta_2 = sin^{-1}(sin(i)\, sin(u_2)) \tag{2.10}$$

$$\Psi_1 = tan^{-1}\left(\frac{cos(i)\, sin(u_1)}{cos(u_1)}\right) - \omega_e t_1 \tag{2.11}$$

$$\Psi_2 = tan^{-1}\left(\frac{cos(i)\, sin(u_2)}{cos(u_2)}\right) - \omega_e t_2 \tag{2.12}$$

Equation (2.11) – (2.12) are simplified to,

$$\Psi_1 = tan^{-1}\left(\frac{cos(i)\, tan(u_1)}{1}\right) - \omega_e t_1 \tag{2.13}$$

$$\Psi_2 = tan^{-1}\left(\frac{cos(i)\, tan(u_2)}{1}\right) - \omega_e t_2 \tag{2.14}$$

For the time, t, the latitude, δ, and longitude, Ψ , to be the inputs, equation (2.9) – (2.14) are rearranged as follows,

$$sin\, \delta_1 \frac{1}{sin(i)} = sin(u_1) \tag{2.15}$$

$$sin\, \delta_2 \frac{1}{sin(i)} = sin(u_2) \tag{2.16}$$

$$tan(\Psi_1 + \omega_e t_1) \times \frac{1}{cos\,(i)} = tan(u_1) \tag{2.17}$$

$$tan(\Psi_2 + \omega_e t_2) \times \frac{1}{cos\,(i)} = tan(u_2) \tag{2.18}$$

Because time, t, is one of the inputs used to determine the orbital element values, the argument of latitude, u, can be written in terms of t as presented in equation (2.4), resulting in having the semi-major axis, a, as an unknown. Equation (2.4), is substituted into the latitude and longitude equations, (2.15) – (2.18) as follows,

$$\sin\delta_1 \frac{1}{\sin(i)} = \sin\left(2\pi t_1 \Bigg/ \sqrt{\frac{4\pi^2}{\mu_E} a^3} \right) \tag{2.19}$$

$$\sin\delta_2 \frac{1}{\sin(i)} = \sin\left(2\pi t_2 \Bigg/ \sqrt{\frac{4\pi^2}{\mu_E} a^3} \right) \tag{2.20}$$

$$\tan(\Psi_1 + \omega_e t_1) \times \frac{1}{\cos{(i)}} = \tan\left(2\pi t_1 \Bigg/ \sqrt{\frac{4\pi^2}{\mu_E} a^3} \right) \tag{2.21}$$

$$\tan(\Psi_2 + \omega_e t_2) \times \frac{1}{\cos{(i)}} = \tan\left(2\pi t_2 \Bigg/ \sqrt{\frac{4\pi^2}{\mu_E} a^3} \right) \tag{2.22}$$

The latitude equations can be rearranged to calculate the inclination , i, as follows,

$$\sin^{-1}\left(\frac{\sin(\delta_n)}{\sin\left(t_n \Big/ \sqrt{\frac{a^3}{\mu_E}} \right)} \right) = i \tag{2.23}$$

For a single orbit to overfly multiple targets, the inclination does not change. This means that for a single orbit,

$$\frac{\sin(\delta_1)}{\sin\left(t_1 \Big/ \sqrt{\frac{a^3}{\mu_E}} \right)} = \frac{\sin(\delta_n)}{\sin\left(t_n \Big/ \sqrt{\frac{a^3}{\mu_E}} \right)} = \sin(i) \tag{2.24}$$

This can be re-arranged to be,

$$\sin(\delta_1) \sin\left(2\pi t_n \Bigg/ \sqrt{\frac{4\pi^2}{\mu_E} a^3} \right) = \sin(\delta_n)\ \sin\left(2\pi t_1 \Bigg/ \sqrt{\frac{4\pi^2}{\mu_E} a^3} \right) \tag{2.25}$$

The equation can be solved for a then for i, if the three inputs, time, Latitude, and Longitude are given.

Using the inclination equation (2.23), the longitude equation can be rewritten in terms of the semi-major axis,

$$\frac{tan(\Psi_N+\omega_e t_n)}{cos\left(sin^{-1}\left(\frac{sin(\delta_n)}{sin\left(t_n \Big/ \sqrt{\frac{a^3}{\mu_E}}\right)}\right)\right)} = tan\left(t_n \Big/ \sqrt{\frac{a^3}{\mu_E}}\right) \tag{2.26}$$

Which can be simplified and rewritten as,

$$\frac{tan(\Psi_n+\omega_e t_n)}{\sqrt{1-Cosec\left(t_n \Big/ \sqrt{\frac{a^3}{\mu_E}}\right)^2 Sin\,(\delta_n)^2}} = tan\left(t_n \Big/ \sqrt{\frac{a^3}{\mu_E}}\right) \tag{2.27}$$

Equation (2.27) requires inputs of time, latitude, and longitude to find the unknown value, a. This cannot be solved directly as the time of target overflight is unknown and therefore there are two unknowns in these equations, i.e., the semi-major axis and the time. For this, a simple solver can be used to find one unknown. Alternatively, an appropriate time range can be used to solve equation (2.27) for the possible semi-major axis solutions. These semi-major axis solutions can then be used, with the chosen time range to solve for the required inclination solutions using equation (2.23). From these, three vectors can be obtained, related to time, semi-major axis, and inclination. The three vectors can be plotted in 3D space. The projection of this line in 3D onto the semi-major axis/inclination plane is a variable time contour of the span of possible solutions. To demonstrate that the three vectors obtained can be used to give the desired results, the following steps are taken:

1. A simulation of the initial equations, is done on MATLAB using the values presented in *Table 2.2*,
2. The values used for step 1 above yield the SSPs, (longitude, latitude), and time results presented in *Table 2.3*,

3. These results are then used to check if equation (2.23) – (2.27) generate the expected orbit values. One of the SSP Latitude and Longitude values and expected time of overflight are selected,
4. When these are simulated, on MATLAB, the results are calculated and are found to be approximate to the expected semi-major axis value of a, ≅ 7700km and inclination angle value of ≅70 degrees. This shows that the inverse equation method works within a given accuracy level.
5. The method is further developed to finding the best single orbit without the knowledge of time of passage over each point. The three SSPs previously calculated are used to simulate this. To show that given a time range, the vectors of time, inclination and semi-major axis solutions can be used to determine an orbit that will overfly all the SSPs, the simulation results are presented in section 2.1.2.

Table 2.2: Orbital values used for inputs of the initial method simulation.

a (km)	i(deg)	$\dot{\Omega}$(rad/s)	ω(deg)	Ω_{et0}(deg)	e
7700	70	0	0	0	0

Table 2.3: Three subsatellite points and times obtained from the initial calculations.

	Latitude (deg)	Longitude (deg)	Time (sec)
1	21.06	6.30	420
2	46.32	18.48	940
3	47.71	19.53	970

2.1.2 Analysis of The Developed Analytical Method Equations

To verify the validity of the proposed method, Equation (2.27) can be used with subsatellite point sets that are known to be valid, and this should resolve an orbit that would overfly the known ground points at defined times within a given time range. Additionally, for times such as, t == 0, there will not be a valid semi-major axis that solves this equation. However, there will be a range of times at which a range of semi-major axis will overfly the target.

Using any one of the combinations of subsatellite points on *Table 2.3*, equation (2.27) can be used to find the number of solutions within a given time span. To verify this, an analysis is done using the last set of the SSPs on *Table 2.3*, (latitude 3 and longitude 3). This analysis of (2.27) is done using a solver and by plotting the left-hand side against the right-hand side for a range of times, an equality check, of the equation. The plots resulting from this analysis are in Figure 2.2 (a) and (b) where the intersections show equation (2.27) time solutions. Figure

2.2 (a) shows the equality check for 7700km semi-major axis, for multiple orbits over a period of just under a day and it graphically presents the number of solutions that can be obtained from equation (2.27). Figure 2.2 (b) gives a graphical view of the equality check for 7700km semi-major axis, for just over one orbit showing the number of valid solutions expected for 1 orbit.

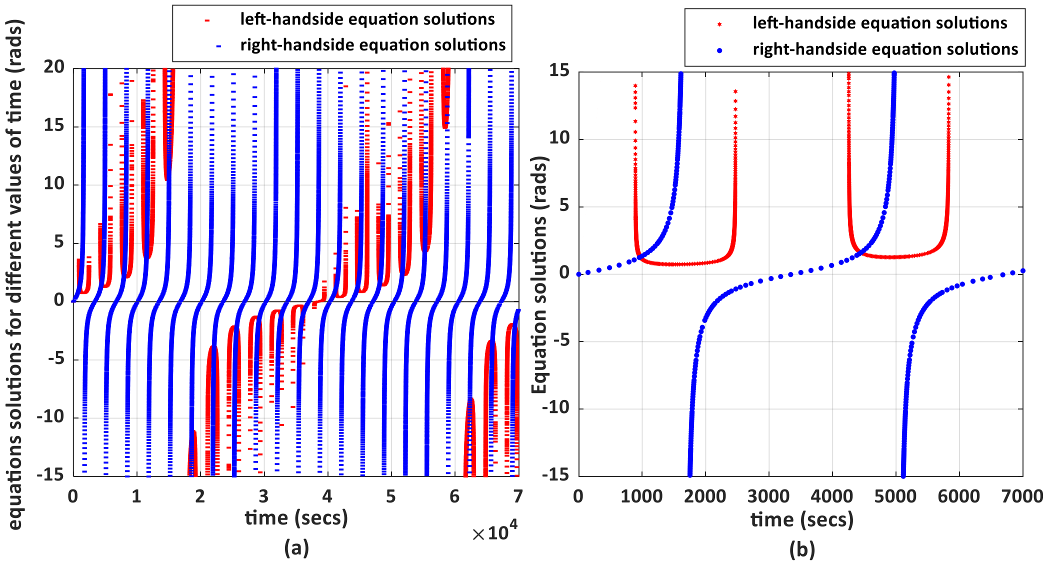


Figure 2.2: Equality check for 7700km semi-major axis, latitude 47.71 deg., longitude 19.53 deg.; (a) for multiple orbits over a long period (70000 seconds) and (b) for just over one orbit

Figure 2.2 (a) and (b) show that Equation (2.27) has up to two solutions per orbit for a given semi-major axis and, a real solution only exists when,

$$1-\csc\left(\frac{t}{\sqrt{a^3/\mu_E}}\right)^2\sin(\delta)^2 \geq 0 \qquad (2.28)$$

which can be re-arranged as

$$csc\left(\frac{t}{\sqrt{a^3/\mu_E}}\right)^2 sin(\delta)^2 \leq 1 \qquad (2.29)$$

Finding the limits allows the solver to be constrained to the correct time span, so

$$\csc\left(\frac{t}{\sqrt{a^3/\mu_E}}\right)^2\sin(\delta)^2 = 1 \qquad (2.30)$$

which can be rearranged to find time as,

$$t = \sqrt{a^3/\mu_E}\,\csc^{-1}\sqrt{\frac{1}{\sin(\delta)^2}} \qquad (2.31)$$

For further analysis, the first of the previously defined sub-satellite points, from *Table 2.3*, (Latitude = 21.06 deg. & Longitude = 6.30 deg.), is used to solve equation (2.27) for time at a defined semi-major axis range. The corresponding inclination for each (semi-major axis, time) solution can be calculated using equation (2.23).

At this point three vectors can be acquired: inclination, time, and altitude (H). The altitude is a function of the semi-major axis and the Earth's radius where,

$$H = a - R_E \tag{2.32}$$

Where R_E is the radius of the Earth (6378km).

The three vectors can be plot in 3D space or 2D space of each. The 2D solutions are presented in Figure 2.3. The time – Inclination graph shows that just as in the equations, the time and inclination relate in a sinusoidal manner. Figure 2.3 shows the solutions for latitude 21.06deg., longitude 6.30deg. Whereas Figure 2.4 shows the solutions for latitude 46.32deg., longitude 18.48deg. Figure 2.5 then shows the combination of these two solutions and from these the solutions of satellites at the orbit Inclinations and altitudes that overfly these two ground points can be determined. These are presented and explained in Figure 2.6 - Figure 2.9 and an explanation given for each.

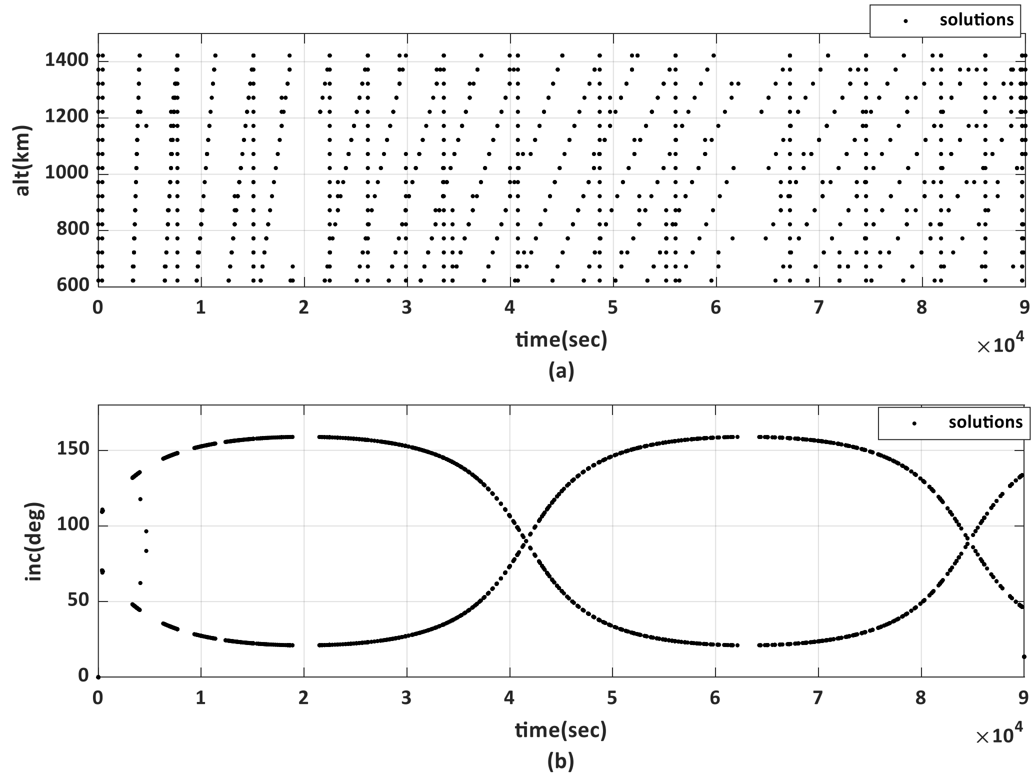
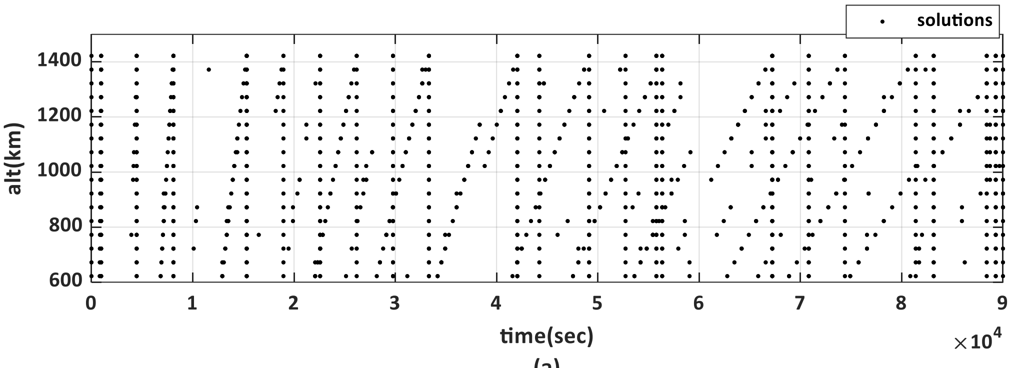
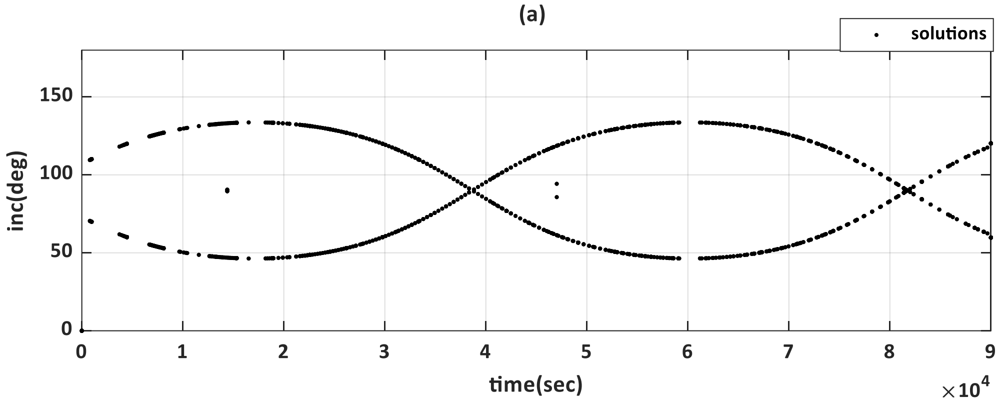


Figure 2.3: Orbital element solutions for overflights of latitude 21.06deg., longitude 6.30deg.: (a) time vs altitude, and (b) time vs inclination.

Figure 2.3 (a) shows the possible orbital altitude(s) that can be used to facilitate overflight of the given ground point at the inclination(s) given in Figure 2.3 (b). At times of approximately 20000 seconds after epoch for example, no orbit facilitates overflight of the target. This can be seen as there is no altitude nor inclination points corresponding to this time for the altitude range and the inclination range selected. This is further clarified when a second ground point is considered and the plots from both are compared as in Figure 2.5. To get to these figures however, the process to obtain Figure 2.3 is repeated for the set of sub-satellite points (Latitude = 46.32 deg. & Longitude = 18.48 deg.).



(a)



(b)

Figure 2.4: Orbital element solutions for overflights of latitude 46.32deg., longitude 18.48deg. (a) time vs altitude, and (b) time vs inclination.

From both Figure 2.3 and Figure 2.4, there are two graphs, (one per ground point), that can be plotted against each other and where there is a crossing on the inclination/altitude plane, a satellite on that orbit may be said to overfly both points if the time of the two solutions is not equal. If an orbit is valid to overfly both ground points, a satellite in that orbit will not overfly the two points at the same time after epoch. Figure 2.5 presents the result of plotting the two graphs, Figure 2.3 and Figure 2.4, against each other.

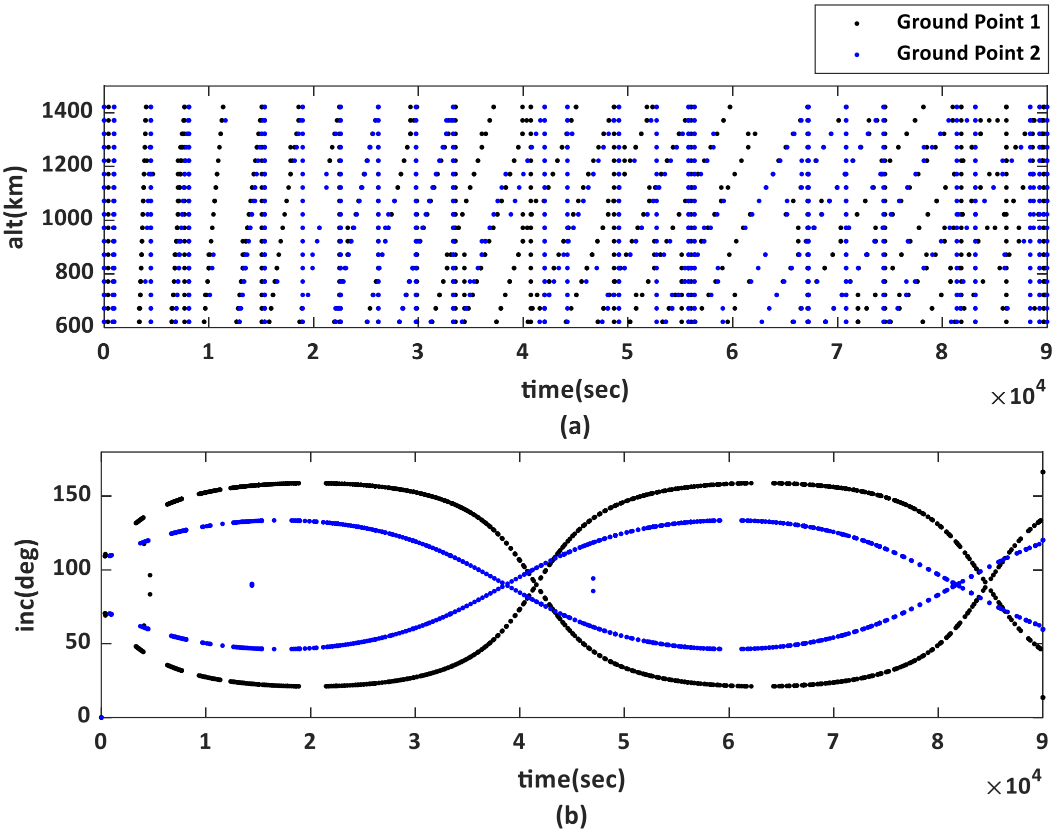


Figure 2.5: Orbital element solutions for overflights of Ground Point 1 (latitude 21.06 deg., longitude 6.30 deg.), Ground Point 2 (latitude 46.32 deg., longitude 18.48 deg.)

From Figure 2.5, a few solutions, other than the original orbit, are found to be possible. It is however worth noting that the inclination is not an exact match, rather it is very close and so whilst an exact overflight may not happen, both ground points would likely be visible within a certain spacecraft swath width. This will depend on the field of view (FoV) of the on-board instrument. The Field of View is influenced by the semi-major axis as presented in section 2.2.1.2.

From Figure 2.5, it can be observed that determining the points with the same orbits directly from the graph presents a challenge. An algorithm is therefore implemented to be able to print out and to also mark the determined orbits for the different ground positions on the graphs. These are indicated by the green squares in Figure 2.6 – Figure 2.9 and the valid orbits vary depending on the inclination increments used.

The formulation is based on analysis of ground point 1 and ground point 2 in *Table 2.3*. The altitude at the valid inclinations must be equal for both ground points and the times must be different for a solution to be valid.

Different scenarios are simulated where the inclination tolerance is varied for each case. As the exact values are known for this formulation, it is observed that for an exact overflight solution, a tolerance of Inclination +/- 0.001 degrees is the most accurate and it yields the graph presented in Figure 2.6.

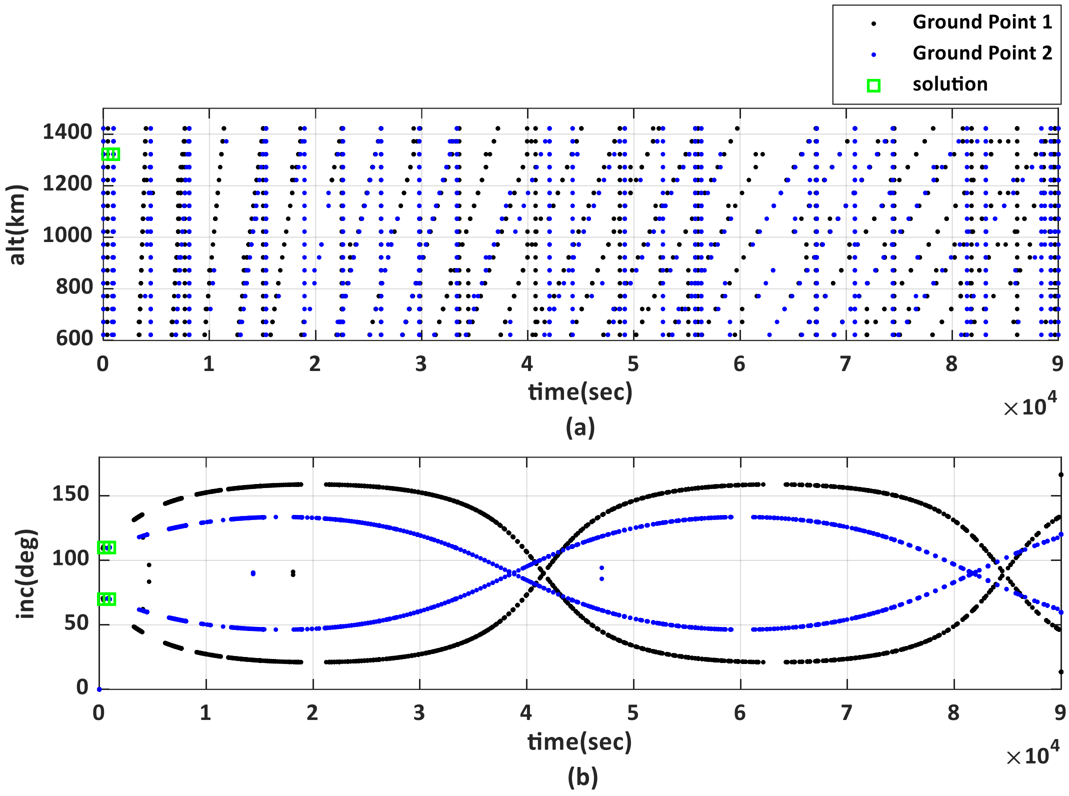
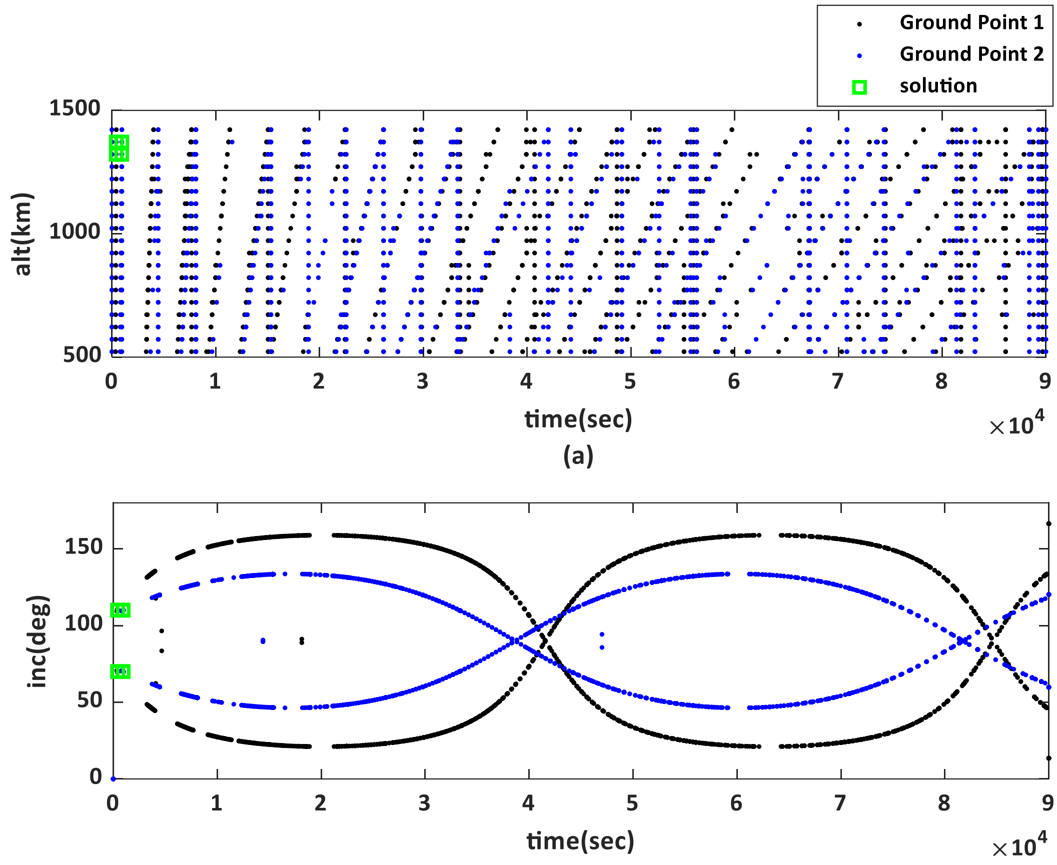


Figure 2.6: Overflights for Ground Point 1 (latitude 21.06 deg., longitude 6.30 deg.), Ground Point 2 (latitude 46.32 deg., longitude 18.48 deg.) (inc. tolerance +/- 0.001 degrees). The green squares are the orbits inclinations and altitudes valid for crossing both target points.

From Figure 2.6, it can be observed that two orbits are found to facilitate overflight over both targets. The orbital altitude found is the same for both SSPs and the inclination obtained is one a prograde and one a retrograde inclination. This is the altitude and inclinations expected hence showing that the method works as expected. For a desired satellite swath width allowance, more solutions may be valid, but this depends on factors such as the instrument on-board. When the tolerance is set to and Inclination +/- 0.01, more near solutions are obtained as shown on Figure 2.7. The near solutions show that for an allowable larger swath width, the target points will be viewed by some additional orbits. In the case presented in Figure 2.7, there are four orbit solutions.

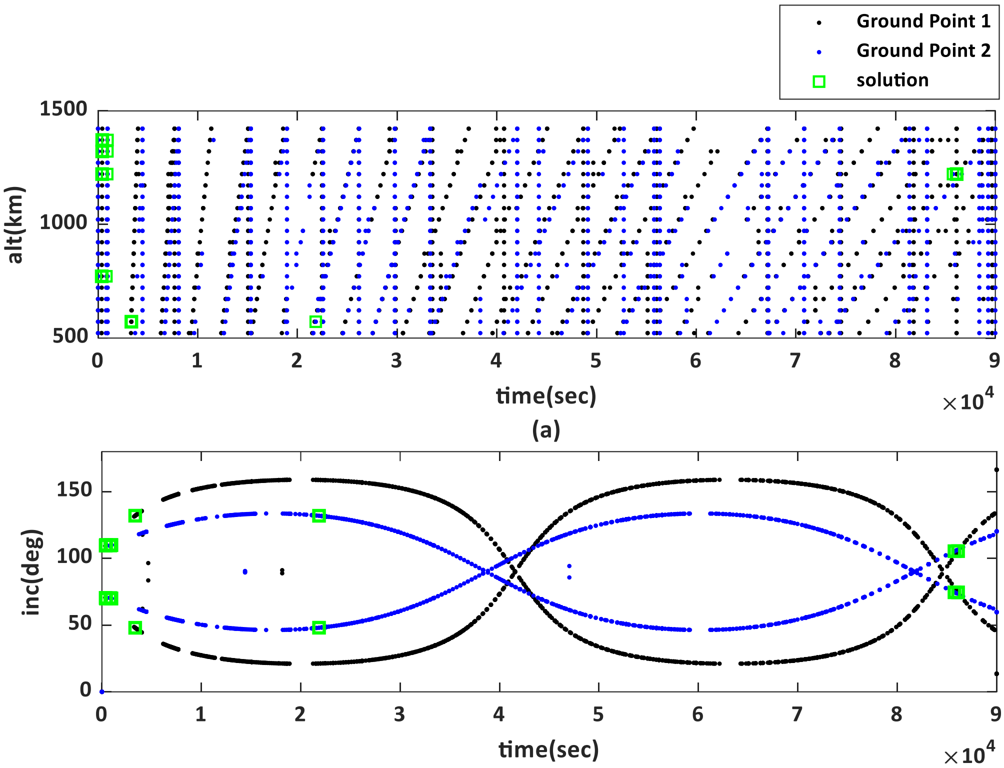


(a)

(b)

Figure 2.7: Overflights for Ground Point 1 (latitude 21.06 deg., longitude 6.30 deg.), Ground Point 2 (latitude 46.32 deg., longitude 18.48 deg.) inclination tolerance 0.01degrees. The valid orbits (green squares) are more than when the tolerance was lower.

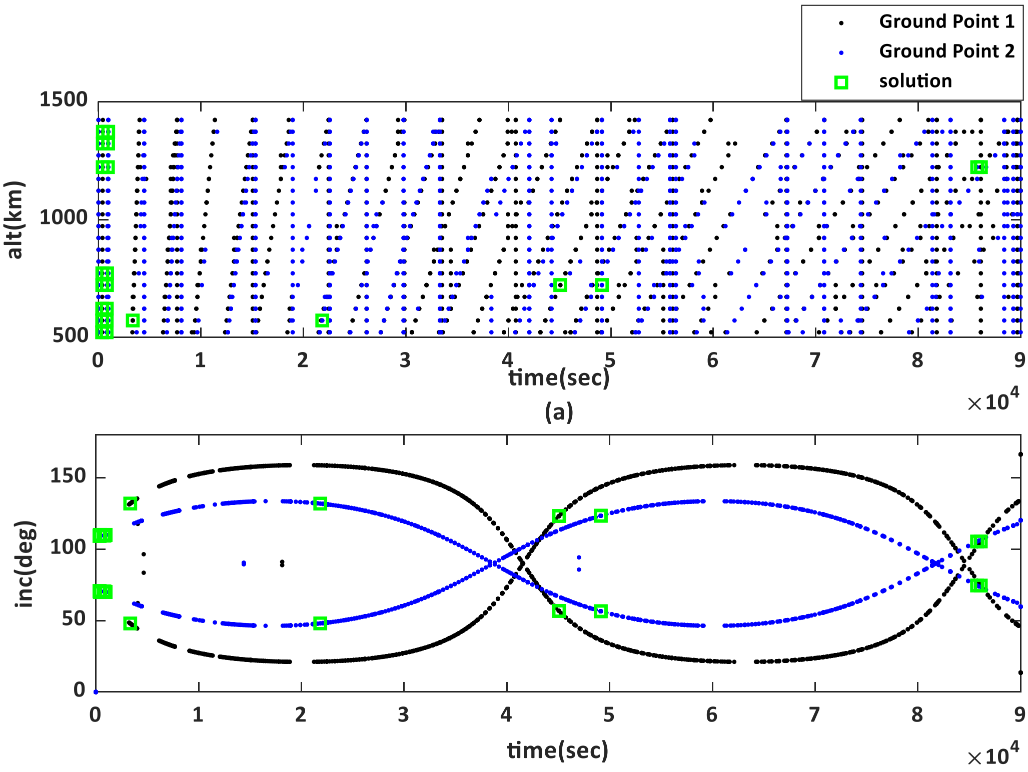
Four scenarios are analysed, i.e., when the inclination tolerance is 0.001 degrees, 0.01 degrees, 0.1 degrees and 0.2 degrees respectively. The solutions for inclination increments of 0.1 degrees and 0.2 degrees are presented in Figure 2.8 and Figure 2.9. These figures show that the near solutions for 0.2 deg. increments are more. These results for the graphically presented solutions are numerally presented in *Table 2.4 – Table 2.8*. For an inclination tolerance of 0.001 degrees, only the known to be valid orbit where the altitude is 1322km, the inclination is approximately 70 degrees, and the times are 420 seconds and 940 seconds respectively are found as the solutions.



(a)

(b)

Figure 2.8: Overflights for Ground Point 1 (latitude 21.06 deg., longitude 6.30 deg.), Ground Point 2 (latitude 46.32 deg., longitude 18.48 deg.) inclination tolerance 0.1 degrees.



(a)

(b)

Figure 2.9: Overflights for Ground Point 1 (latitude 21.06 deg., longitude 6.30 deg.), Ground Point 2 (latitude 46.32 deg., longitude 18.48 deg.) inclination tolerance 0.2 degrees.

Multiple near solutions are obtained at a tolerance of 0.2 degrees compared to the lower increments as expected. The tolerance in this case is related to the precision of the orbital inclination needed to overfly the target.

Table 2.4: Key for Table 2.5 – Table 2.8

Time 1	Time 2	Inclination 1 (deg)	Inclination 2 (deg)	Semi-Major Axis
Time for ground point 1 overflight (sec)	Time for ground point 2 overflight (sec)	Retrograde inclination (deg)	Prograde inclination (deg)	Altitude for targets overflight (km)

Table 2.5:Near solutions at inclination increments of +/-0.2 degrees.

Time 1	Time 2	Inclination 1	Inclination 2	Semi-Major Axis
86120	85756	105	75	1222
355	794	109	71	522
359	803	109	71	572
363	812	109	71	622
371	830	110	70	722
375	839	110	70	772
412	921	110	70	1222
420	940	110	70	1322
424	949	110	70	1372
45075	49134	123	57	722
45075	49134	123	57	722
3346	21839	132	48	572

Table 2.6: Near solutions at inclination increments of +/-0.1 degrees.

Time 1	Time 2	Inclination 1	Inclination 2	Semi-Major Axis
86120	85756	105	75	1222
375	839	110	70	772
412	921	110	70	1222
420	940	110	70	1322
424	949	110	70	1372
3346	21839	132	48	572

Table 2.7: Near solutions at inclinations of +/-0.01 degrees

Time 1	Time 2	Inclination 1	Inclination 2	Semi-Major Axis
420	940	110	70	1322
424.2	949.4	110	70	1372

Table 2.8: Near solutions at inclination increments of +/-0.001 degrees.

Time 1	Time 2	Inclination 1	Inclination 2	Semi-Major Axis
420	940	110	70	1322

From the results presented in the *Table 2.4 – Table 2.8*, the method of analysing a search space to determine orbits based on ground target overflight proves to be valid. The exactness of the solutions depends on the number of steps taken/inclination tolerance used. Nevertheless, instead of considering an exact point of contact, this method is further developed to include a field of view, FoV and considers the swath width. This takes into consideration the FoV of an on-board instrument and the method is further developed in section 2.2.1.2 to include this. Additionally, J_2 perturbations are included in further developments of the method. In section 2.2 also, the equations relating to the rate of change of RAAN, and the Greenwich Apparent Sidereal Time (GAST), are included in the development of the method.

2.2 Development of Ground-Target Based Orbit Determination Method Equations Eliminating Previous Assumptions

To track a satellite from the earth and to position it in space, [67] use two methods, the vector rotation method and the spherical geometry method. The method presented in this chapter is focused on the second method, i.e., the spherical geometry method. This method is used by [25] in the analytical description of the ground track motion. It has also been used by other researchers to enable them to propagate a ground track.

2.2.1 Main Analytical Equations to Determine the Valid Orbit(s)

From spherical geometry, the subsatellite latitudes and longitudes of a satellite can be calculated, from a known orbit by using the following equations.

$$\delta = sin^{-1}(sin(i)\, sin(u)) \tag{2.33}$$

$$\Psi = tan^{-1}\left(\frac{cos(i)\, sin(u)}{cos(u)}\right) - \omega_e t + \Omega_s + \frac{d\Omega}{dt}t - \Omega_{et0} \tag{2.34}$$

Where δ and Ψ are the subsatellite latitudes and longitudes respectively. i and u are the orbital inclination and the argument of latitude respectively. ω_e, Ω_s and Ω_{et0} are the relative rotation rate of the Earth, the right ascension of the satellite and the right ascension of Greenwich at epoch time, respectively. ($\Omega_s - \Omega_{et0}$) is the equivalent to the longitude of the ascending node at epoch time[67].

The argument of latitude is a sum of the true anomaly and the argument of perigee.

$$u = \theta + \omega \tag{2.35}$$

Where θ is the true anomaly and ω is the argument of perigee.

The orbit being considered is however a circular orbit. For this reason, the argument of perigee is zero and so the argument of latitude, AoL, is equal to the true anomaly. This therefore means that the argument of latitude can be calculated as,

$$u = \frac{2\pi t}{T} = \frac{2\pi t}{2\pi\sqrt{a^3/\mu_E}} = \frac{t}{\sqrt{a^3/\mu_E}} \tag{2.36}$$

Where T is the period, a is the semi major axis, and μ_E is the gravitational constant of the Earth. In the initial methods presented in this chapter, there are assumptions such as not considering J_2 perturbations as well as assuming the last two terms of the longitude calculation to be negligible i.e., (==0). This simplifies the work, but the results incur some errors compared to the results from some more accurate methods especially for longer periods. This is one disadvantage of using simplified analytical equations. To solve this problem and reduce the errors incurred, the assumptions are eliminated only maintaining one of the conditions that the orbit is circular, i.e., e=0.

To compute the errors, the great circle distance between the subsatellite points is calculated. This distance is given known as the haversine distance. An example of the errors incurred when using a non-perturbed method as compared to a perturbed method is graphically represented in Figure 2.10 which gives a haversine distance between the ground tracks obtained when including J_2 and an unperturbed simulation using different periods. The accumulation of errors leads to inaccuracy of the results especially over a long period of time.

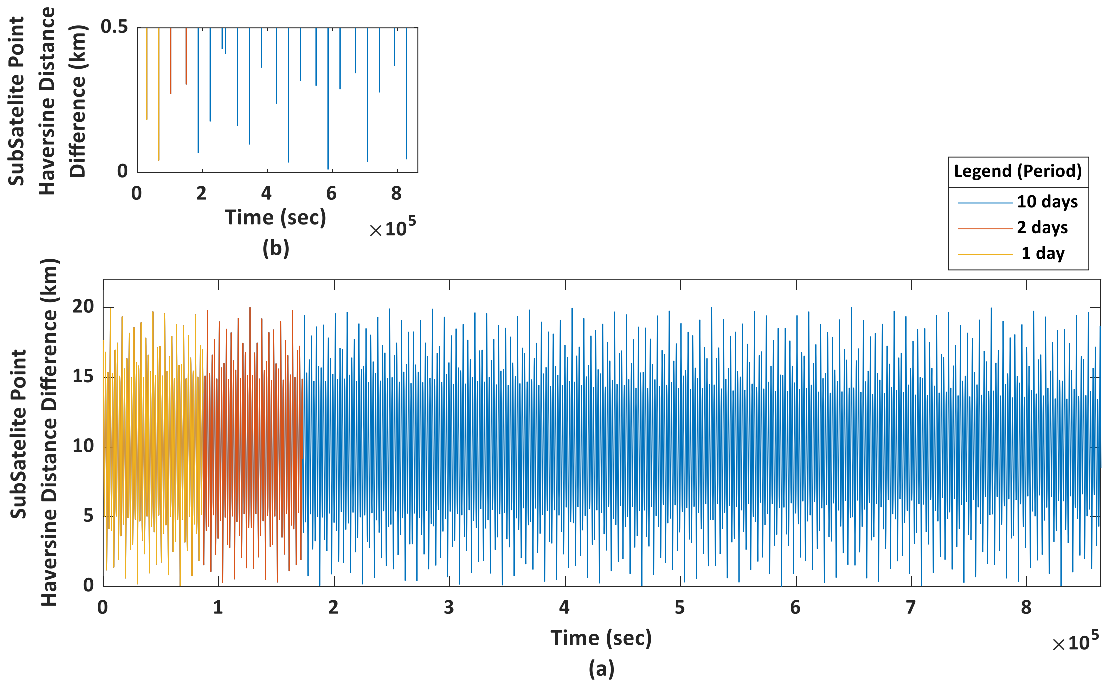


Figure 2.10: (a) Haversine distance between ground-track of orbit when J_2 is included and when J_2 is not included for short period (1-Day), Medium Period (2-Days) and Long period (10-Days) and (b) zoomed in graph showing that the distance between the ground-tracks is never zero.

From the haversine distance on Figure 2.10, it is observed that the assumption of no perturbations attracts errors regardless of the period being considered. Moreso, Figure 2.10 (b) shows that using the same propagation values and software (MATLAB), the SSPs of the non-perturbed orbit and the perturbed orbit are never equal. In a case of monitoring an area such errors can lead to coverage gaps, hence data gaps and even in a case of a target revisit schedule.

Third party software and methods such as SGP4 orbit propagator account for secular and periodic orbital perturbations caused by Earth's geometry and atmospheric drag and is applicable to near-Earth satellites whose orbital period is short [69]. One such software that includes J_2 perturbations is used for comparison of the J_2 included method developed. The software used is NASA’s GMAT, version GMAT R2022a.

2.2.1.1 Inclusion of Secular Perturbations to Orbital Calculations

Secular perturbations due to the oblate nature of the Earth must be considered for accuracy of orbital results. In this work, the second order perturbations, J_2, are considered. This mainly affects the period (in the case of a repeating ground track orbit), mean anomaly, hence the argument of latitude and the right ascension of the ascending node. These affect the equations of motion as follows, [30, 68, 70, 71]

The Mean Anomaly,

$$M = n = \frac{2\pi}{T} = \sqrt{\frac{\mu_E}{a^3}} \tag{2.37}$$

With the inclusion of J_2 perturbations, the mean anomaly changes with time as follows,

$$\bar{n} = \frac{dM}{dt} = n\left[1 + \frac{3}{2}\, J_2 (\frac{R_E}{a})^2\, (1-\varepsilon^2)^{-\left(\frac{3}{2}\right)} (1 - \frac{3}{2} sin^2(i)\right] \tag{2.38}$$

The Right Ascension of the ascending node,

$$\Omega = \ \Omega_0 + \left(\frac{d\Omega}{dt}\Delta t\right) \tag{2.39}$$

For which,

$$\frac{d\Omega}{dt} = -\,\bar{n}\,\left[\frac{3}{2}\, J_2 (\frac{R_E}{a})^2\, (1-\varepsilon^2)^{-2} cos\,(i)\right] \tag{2.40}$$

The argument of latitude,

$$u = \ u_0 + \left(\frac{du}{dt}\Delta t\right) \tag{2.41}$$

For which,

$$\frac{du}{dt} = \frac{dM}{dt} + \frac{d\omega}{dt} \tag{2.42}$$

And,

$$\frac{d\omega}{dt} = \ \bar{n}\left[\frac{3}{2}\, J_2 (\frac{R_E}{a})^2\, (1-\varepsilon^2)^{-(2)} (2 - \frac{5}{2} sin^2(i))\right] \tag{2.43}$$

$$\frac{du}{dt} = \bar{n} + \ \bar{n}\left[\frac{3}{2}\, J_2 (\frac{R_E}{a})^2\, (1-\varepsilon^2)^{-(2)} (2 - \frac{5}{2} sin^2(i))\right] \tag{2.44}$$

The secular perturbation due to the Earth's oblateness has effects on the semi-major axis of a repeat-ground track orbit. It mainly affects the nodal period, defined as the period from ascending node to ascending node, and this in turn affects the semi-major axis. The Keplerian period, defined from perigee to perigee without considering the perturbations is given as,

$$T = \frac{2\pi}{n} \; and \; n = \sqrt{\frac{\mu_E}{a^3}} \tag{2.45}$$

This period is however affected by the oblateness of the Earth as it will change depending on the inclination of the orbit as well as the argument of perigee. The period is as defined by [33], the anomalistic period or the osculating Keplerian period. When perturbations are not considered, this value is the same as the Keplerian period. To obtain the actual period and in turn the semi-major axis, J_2 should be considered because the nodal period changes according to the Earth's rotation, the Earth's oblateness, and the argument of perigee. This will have effects on the equator crossings as well as the Greenwich meridian cross times. To calculate the nodal period, the formula for a circular orbit is given by [33] as;

$$T_\Omega = T\left[1 - \frac{3J_2}{2}\left(\frac{R_E}{a_o}\right)^2 (3 - 4sin^2(i))\right] \tag{2.46}$$

But as derived and explained by ref. [31], the mean values of these osculating semi-major axis can be calculated using the formula;

$$a = a_0 \left\{1 - \frac{J_2}{{a_0}^2}\left(1 - \frac{3}{2}sin^2(i)\right)\right\} \tag{2.47}$$

For a repeat ground track, the effect of this change is mainly the distance to the successive equator crossing points. This is influenced by the nodal period and is given by [33] as;

$$\Delta\lambda_{rev} = (\omega - \dot{\Omega})T_\Omega \tag{2.48}$$

In some previous work however, to maintain a fully analytical solution it is shown that the effect of J_2 on the semi-major axis, when the period being considered is short and the inclinations are high, the effects are minimal and can be neglected. This is mentioned in the work by C. N. McGrath et. al., [25], who do not consider the changes due to J_2 but only focus on the propulsive acceleration effects on the semi-major axis during manoeuvres. The effect of J_2 on the semi-major axis is however considered for the method proposed herein.

In all cases, the subscript ($_0$) denotes the initial value (values at epoch time 0).

Due to the secular perturbations, equation (2.33) and (2.34) can be rewritten as,

$$\delta = sin^{-1}\left(sin(i)\, sin\left(u_0 + \left(\frac{du}{dt}\Delta t\right)\right)\right) \tag{2.49}$$

$$\Psi = tan^{-1}\left(\frac{cos(i)\,sin\left(u_0+\frac{du}{dt}\Delta t\right)}{cos\left(u_0+\frac{du}{dt}\Delta t\right)}\right) - \omega_e t + \left(\Omega_s + \frac{d\Omega}{dt}\Delta t\right) - \Omega_{et0} \qquad (2.50)$$

And for the semi-major axis, when considering a repeat ground track, the equation is,

$$a_{x+1} = \frac{\mu^{1/3}}{(n\omega_E)^{2/3}}\left[1-\frac{3}{2}(-J_2)\left(\frac{R_E}{a_x(1-\varepsilon^2)}\right)^2\left(1-\frac{3}{2}sin^2 i\right)\right]^{3/2}\left[1-J_2\left(\frac{R_E}{a_x(1-\varepsilon^2)}\right)^2\frac{3}{2}\left(n\cos i - \frac{3}{4}(5\cos^2 i-1)\right)\right]^{3/2} \qquad (2.51)$$

Where it is iterated to a tolerance where $a_{x+1} - a_0 < 0.000001$ metres and,

$$a_0 = \frac{\mu^{1/3}}{(n\omega_E)^{2/3}} \quad and\; n = \frac{R}{D} \qquad (2.52)$$

Where R = number of revolutions and D = number of days – these give the expected repeat schedule. From the repeat schedule, a_0 is the desired semi-major axis to obtain the repeat. a_{x+1} is therefore iterated until the difference between it and a_0 is almost zero. This is done to get the semi major axis of a repeat ground track orbit while considering the secular perturbation, J_2.

Apart from the J_2 inclusions, to accurately propagate the ground track and determine whether a satellite in an orbit overflies a certain target or not, the field of view or field of regard, FoR, must be considered. Especially for data collection, the field of view determines the amount of time the overflight takes and so this should be included in the equations for added accuracy of the method. Section 2.2.1.2 highlights the addition of the FoV to the method.

2.2.1.2 Extension of Ground-Track Equations to Overflight Based on Swath Width.

The instrument on board has an impact on the capability of a satellite in some ways such as the quality of data received and most of all the collection of the correct data from a target area. The collection of data from a target area is dependent on whether the satellite swath width is large enough to get views of the target. The FoV also influences the duration that a target will be viewed for. This overflight is generally dependent on factors such as the angles of the ground, the instrument, the distance from the instrument to the ground, etc. In this study, the field of view angle is important because if the earth central angle between the sub satellite point and a target are not within the field of view angle of the instrument, the orbit will not be valid for an overflight.

From the work by [53] and further developed by [57], the effective Earth central angle of a given field of view are calculated using the following equations:

$$a = R_E + H \qquad (2.53)$$

$$\gamma = sin^{-1}\left(\frac{a\, sin\, \eta}{R_E}\right) \qquad (2.54)$$

$$\rho = R_E\, cos\, \gamma + a\, cos\, \eta \qquad (2.55)$$

$$\lambda = sin^{-1}\left(\frac{\rho\, sin\, \eta}{R_E}\right) \qquad (2.56)$$

Where γ is the supplementary angle, i.e., the spacecraft elevation angle, β, plus 90 degrees, a is the semimajor axis, η is the half effective angle (the Field of view angle) which is specified by the mission requirements depending on the instrument on board, R_E is the radius of the Earth, ρ is the distance between the satellite and the bounds of the sensor projected onto the Earth's surface and λ is the effective Earth central angle. These angles and the spherical geometry are presented in Figure 2.11.

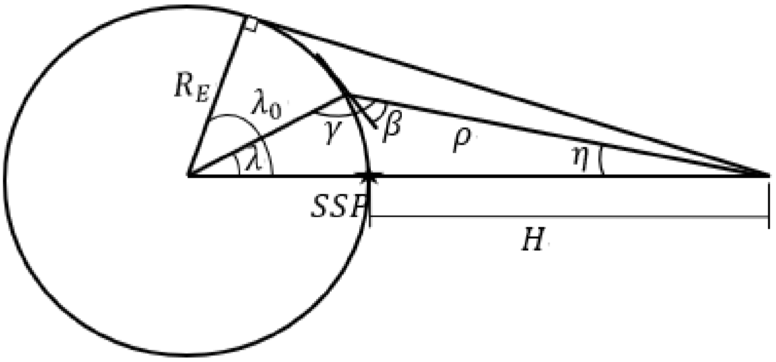


Figure 2.11: Spherical geometry of the field of view of a satellite

Using spherical triangles, the angle between the ground point and the subsatellite point can then be calculated. If it is assumed that the projection of the view from the satellite to the Earth is a circle, an Earth central angle, d_a, can be calculated using spherical triangles and using the sub-satellite points and the target points as shown in equation (2.57) [57].

$$d_a = cos^{-1}\left(cos\left(\frac{\pi}{2} - \varphi_1\right)cos\left(\frac{\pi}{2} - \varphi_2\right) + sin\left(\frac{\pi}{2} + \varphi_1\right)sin\left(\frac{\pi}{2} + \varphi_2\right)cos(\tau_1 - \tau_2)\right) \tag{2.57}$$

Where subscript 1 indicates longitudes and latitudes of the sub-satellite point and subscript 2 is for the target point.

The angles calculated from the target and sub-satellite points are then compared to those of the desired half-effective angle. If the half-effective angle is greater than the angle from the calculation of target points and sub-satellite points, then the satellite overflies the target i.e., if $\lambda > d_a$, then the orbit facilitates an overflight of the target.

2.2.2 Development of the Proposed Method's Algorithm Using an Embarrassingly Parallel Approach

Search and optimisation problems can be classified into sequence or parallel, amongst other classifications [72]. For some cases, running processors in parallel tend to be more efficient than in sequence. This has been found to be true in cases that getting stuck in local optimums is a possibility. Parallel methods are also used to save on computational time [72]. Parallelisation techniques can be used with algorithms such as, evolutionary algorithms and genetic algorithms amongst many other numerical and analytical methods [72, 73]. For some methods however, there is still a level of communication between the processes. In some cases, the processes are not independent on time. The elimination of this communication completely is what in this case is known as an embarrassingly parallel method. In other works, it is referred to as a naturally parallel method or pleasingly parallel methods. According to Amdahl's Law, the advantage of using parallel algorithms is limited by the part of the computation that ends up being sequential [74]. For generally parallel or almost embarrassingly parallel algorithms, the communication between the processes cause a significant increase in computational costs [75]. In their work, J.-C. Régin et. al., [76] use an embarrassingly parallel algorithm to conduct a search in constraint programming. J.-C. Régin et. al., present an Embarrassingly parallel search where they divide their problem into subproblems and find it more efficient with average factor gains of over 10. J.-C. Régin et. al., extend this work in A. Malapert et. al., [75] where the method then proves to be efficient, easy to implement, and has almost no communication between workers.

Parallel algorithms can be divided into data division, where the processes are run on each batch of data or task division where each worker performs different tasks. In the case of the work herein presented, data division is used, where each worker performs the same process on different batches of data.

The simulation of the developed method is done in MATLAB R2022B. The flow diagram of this simulation is presented in

Figure ***2.12*** which gives the embarrassingly parallel architecture used for propagation.

The summary of the algorithm is as follows:

i. The search space is defined with the INC, RAAN and Time, range and increments respectively, the target Longitudes and Latitudes and physical constant parameters are also given.

ii. The search space parameters are combined to form a 4-dimensional matrix. From the matrix, each of the values are combined to determine if an orbit at a certain time facilitates the overflight of a target. For example, if Inclination = 50:1:60 degrees, Right Ascension of the Ascending Node = 0:1:360 degrees, and Time = 0:10:86400 seconds, (i.e. the search space is of 50 degrees inclination to 60 degrees inclination with a step size of one degree, Right Ascension of the Ascending node of 0 degrees to 360 degrees in 1 degree step sizes, and time is from 0 seconds to 86400 seconds in 10 second increments), and there is a target with a latitude and a longitude of 56 degrees North and 65 degrees West respectively, each combination such as an Inclination of 60 degrees, a Right Ascension of Ascending Node of 34 degrees, and a time of 400 seconds will be analysed against that target.

iii. Once this is done and the subsatellite point to target Earth central angle is obtained it is compared to the effective Earth central angle of the onboard instruments' field of view which can be calculated. If the Earth central angle of the SSP to the target point is less than the effective Earth central angle of the instrument, then there is an overflight hence that orbit is valid. If it is greater however, there is no overflight, and this orbit is discarded.

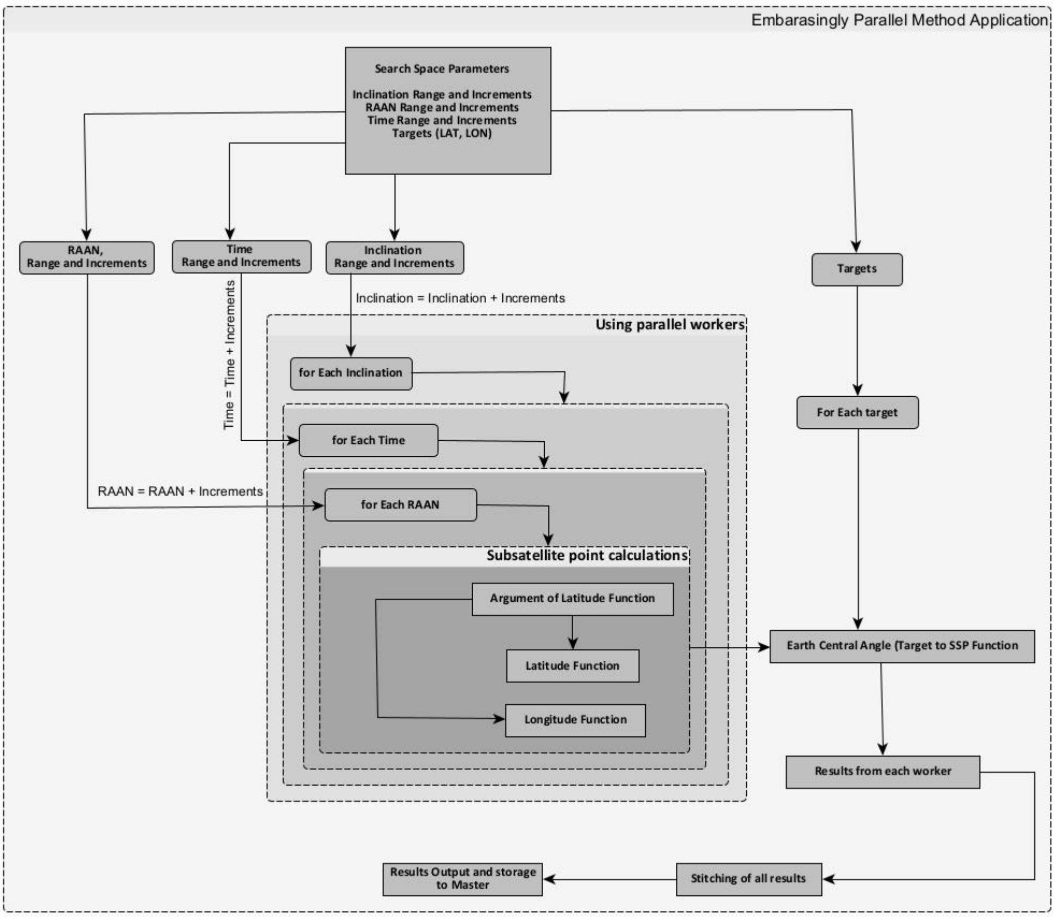

Embarasingly Parallel Method Application
Search Space Parameters
Inclination Range and Increments
RAAN Range and Increments
Time Range and Increments
Targets (LAT, LON)
RAAN, Range and Increments
Time Range and Increments
Inclination Range and Increments
Targets
Inclination = Inclination + Increments
Time = Time + Increments
Using parallel workers
for Each Inclination
For Each target
for Each Time
RAAN = RAAN + Increments
for Each RAAN
Subsatellite point calculations
Argument of Latitude Function
Latitude Function
Longitude Function
Earth Central Angle (Target to SSP Function
Results from each worker
Results Output and storage to Master
Stitching of all results

Figure 2.12: Embarrassingly Parallel Method Architecture

For each value of the inclination, a RAAN value is simulated with a time value. What this means is that for example, if the INC range is 50 – 60 degrees in increments of 1 degree, RAAN range of 0 – 360 degrees in increments of 1 degree and time range of 0 – 86400 seconds in increments of 10 seconds, the simulations will be done $10 \times 360 \times 8640 = 31{,}104{,}000$ times. This is done in parallel then stitched together to give the results. In this work it is done using MATLAB R2023b and uses the inbuilt MATLAB Parfor, on an intel core i7 64-bit laptop. It can be done using other programming languages, where different processors can be used. The algorithm developed based on the *embarrassingly parallel* approach is as presented in Algorithm 2.1.

Algorithm 2.1: Developed method algorithm

```
Inputs: Physical constants, Desired targets (LAT, LON), INC (max, min, increments),
RAAN (max, min, increments), Time (max, min, increments)
Outputs: Orbits – Targets relationships and data
for each RAAN
```

```
        for each INC
            for each time
                subsatellite LON and LAT do// use parallel loop to calculate for each
                RAAN, INC, time combination
                for each target
                    Earth central angle of the field of view of satellite do// calculate from
                    each SSP and target combination
                    storage do// store in a 4D matrix of target, INC, RAAN and Time
                End
            End
        End
    End
    for each target
        Test if within field of view of instrument do// filter the matrix of field of view for orbits
        that see all targets
        Stitch the search space together
        generate and store the orbits and results
    End
    orbits = generated results
```

The developed method has an advantage where each of the simulated orbits, (Inclination – RAAN) combinations, are simulated for each time. Giving an advantage of having a full view of the search space with less computational costs. Also, for different objective functions, there is no need to simulate the results again as each orbit result is already independent of the time. Nevertheless, the method efficiency depends on the increments used for the search space given. Generally, the smaller the increments the more the computational cost, but the better the chances of locating global maximums when optimising the results. For this reason, an adaptive grid that only refines the most optimum orbits is studied in chapter 4 of this dissertation.

The orbit determination method proposed applies the inverse of Kepler's equations that have been developed. The inputs to this method are the target points and physical constants. The search space which includes time, inclination and RAAN ranges are also given for the method to simulate in finite time. Depending on the search space size, the method yields thousands of possible orbits that will facilitate overflight of the given targets. A condition that the orbits determined must facilitate overflight of all the targets or not, can be applied to the method as a filtration to the results obtained. The next step in this work is therefore optimising the orbits based on the specific requirements. In Chapter 3, different objective functions are used to optimise the orbital results. Depending on the operator requirements, the objective functions can be selected and optimisation done without the need for a re-run of the simulations. For this work the studies include the number of times that a satellite in an orbit views the targets and the duration of view of the targets within a given period of two days. One of these objective functions is compared to some previous work as a validation of the developed method; see section 3.1. Section 2.3 gives a presentation of the simulation of the method. Throughout the work, unless otherwise stated, the physical constants used are as presented in table 2.9.

Table 2.9: Simulation Physical Search Space Parameters used throughout the thesis (unless otherwise stated).

Parameter	Symbol	Value	Units
Revolutions	R	29	Cycles
Number of days	D	2	Days
Radius of the Earth	R_E	6,378,000	Meters
Half Effective Angle	H	20	Degrees
Start Time	t_0	00:00:00 1 January 2017	Julian Date
Greenwich Hour Angle	Ω_{et0}	100.84	Degrees
Earth's rotation	Ω	7.292106590880652e-05	Rads/sec
Standard gravitational parameter of the Earth	μ_E	3.986004418e14	M^3/s^2
Earth's gravitational zonal harmonic	J_2	1.0827e-3	-

2.3 Developed Method Simulation, Results and Analysis

This section presents the results, analysis, and discussions obtained from simulating the method described in section 2.1 – 2.2. The simulations, results and analysis given in this section are in the following sequence:

i. The main simulation of the method is done and the orbit solutions obtained are given.

ii. The orbits are simulated from two different angles; the first is when a selected orbit must facilitate overflight of all targets at least once, and the second, is that an orbit is selected if it facilitates overflight of any one of the targets at least once.

iii. Insights obtained from the results are presented and some advantages of using the proposed methods as seen in the results are highlighted.

iv. Justification of the added accuracy when including J₂ perturbations to the orbit determination methodology is given.

v. Method error analysis is done using a third-party software.

2.3.1 Simulation of the Proposed Method and the Results Obtained

Using the developed method algorithm described in section 2.2, the simulation is done using MATLAB R2022b. The inputs to the simulations as required by the method are, orbital search space parameters, physical parameters, and desired ground targets. The inputs used for the simulations in this section are presented in Table 2.10 – Table 2.11 for the orbital search space and the targets respectively. The physical parameters used are presented in Table 2.9.

Table 2.10: Search Space parameters giving the range and increments of the inputs.

Parameter	Range	Increments	Units
Inclination	50-130	0.05	Deg
Right Ascension of Ascending Node	0-360	0.5	Deg
Time (From Epoch)	0 – 172800	10	Sec

Table 2.11: List of targets required to be overflown.

Target	City	Longitude, deg	Latitude, deg
1	Moscow	37.4	55.5
2	London	0.1	51.3
3	Peking	116.2	39.6
4	Washington, D.C	-77.0	38.5
5	Los Angeles	-118.2	34
6	Miami	-80.1	25.5
7	Hong Kong	115.1	21.2
8	Rio	-43.2	-22.5
9	Sydney	151.1	-33.5
10	Buenos Aries	-58.3	-34.4

These 10 geographical target locations are presented in Figure 2.13.

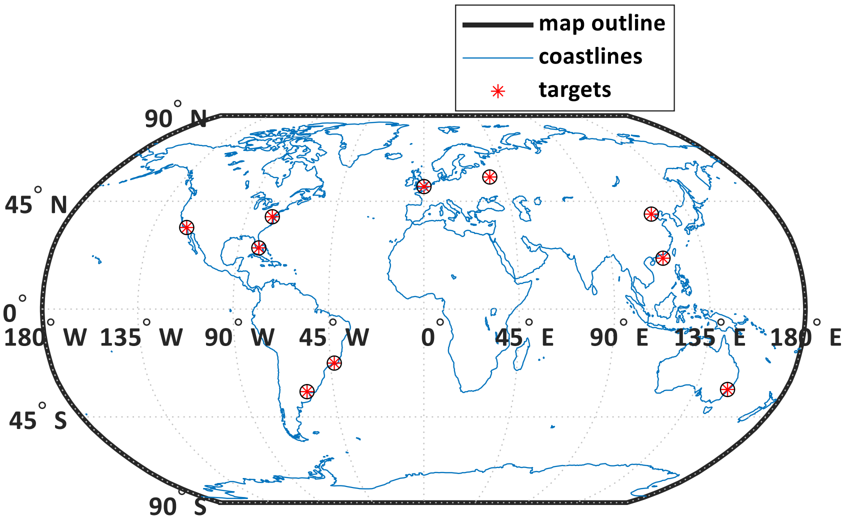


Figure 2.13: Global view of the 10 Targets Used for the simulation of the developed analytical method in this section.

The developed method algorithm is simulated using the given search space values, physical constants values and targets. First, the orbits that facilitate overflight of any of the given targets are determined. The orbital results from these are graphically presented in Figure 2.14 in terms of the inclination and Right Ascension of the Ascending Node values. The black and shaded parts show the inclination and right ascension of the ascending node values of orbits that have been determined to facilitate overflight of the targets.

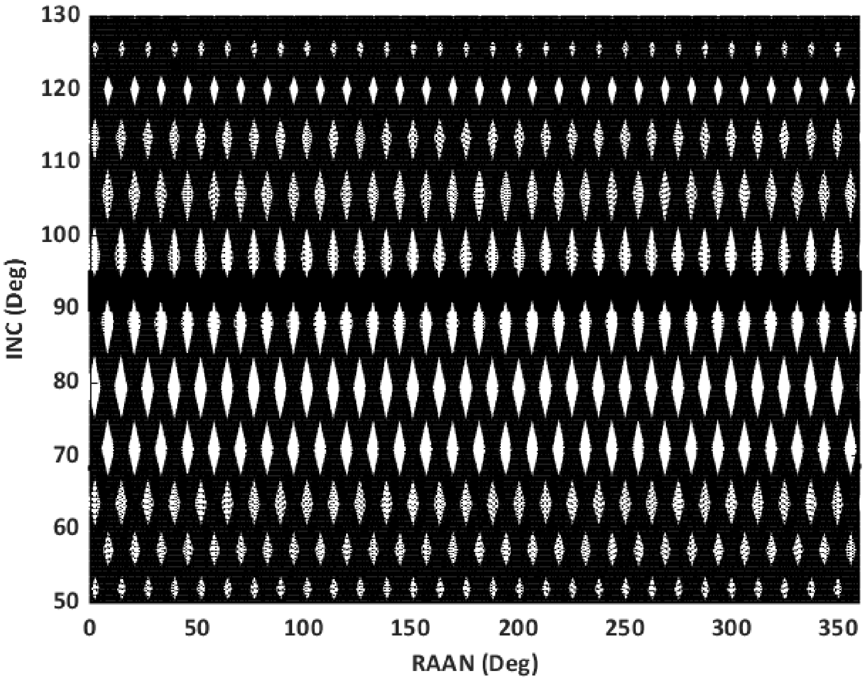


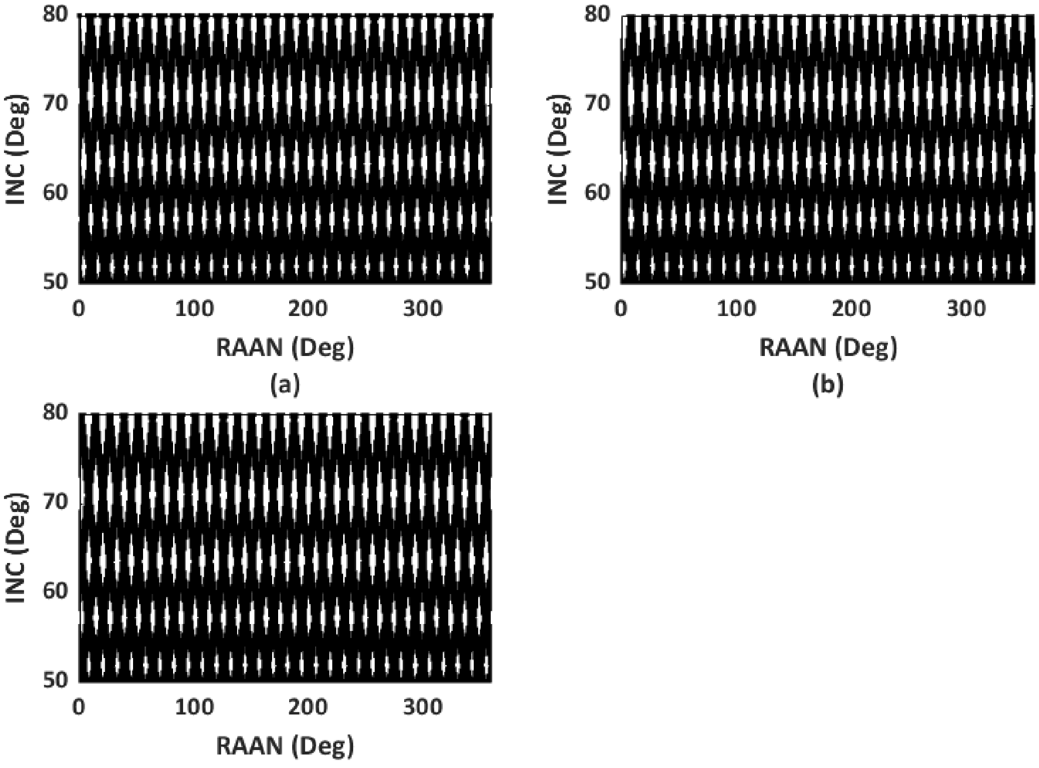
Figure 2.14: Solutions of the orbits, (Inclination and RAAN), which can facilitate the overflight of any of the targets at least once in 2 days.

From Figure 2.14, it is evident that within the given search space, most orbits can ensure overflight of at least one of the desired targets at a certain time after epoch. There are, however, some orbits that at no given time will they facilitate a view of any the targets within the duration of 2 days. The results need extra analysis to determine the optimum orbits depending on the mission requirements. Insights such as which targets will be overflown by satellites on which orbits, which satellites overfly the targets for longer durations of time, etc need further analysis and querying of the results. This analysis is presented in the Case Studies section of this dissertation, chapter 3.

An advantage of this method being analytical, is a simplified analysis of a ground-track shift using the longitude values. A RAAN-shift while maintaining the shape of the ground-track and the other orbital values can be achieved by changing the epoch time. For example, when the start date is set to 1st January 2017 00:00:00, UTC Gregorian time, the longitude of the ascending node at epoch time, in this work referred to as the Greenwich Apparent Side Real Time (GAST) is calculated as 100.8 degrees. The GAST value indicates an addition of a factor of 100.8 degrees to the longitude calculation and if this value is set to zero, the ground-tracks will have a similar shape but there will be a shift in the longitudes hence RAAN. To verify this, a simulation is performed using a GAST of 100.8 degrees, 70 degrees and 0 degrees. The analysis is done for search space parameters presented in Table 2.12, with physical parameters and targets previously presented in Table 2.9 and Table 2.11, respectively. The results for these simulations are presented in Figure 2.15 which shows the differences between orbits obtained when using the different GAST values.

Table 2.12: Search Space parameters used for the different GAST values

Parameter	Range	Increments	Units
Inclination	50-80	0.2	Deg
Right Ascension of Ascending Node	0-270	1	Deg
Time (From Epoch)	0 – 172800	10	Sec



(a)

(b)

(c)

Figure 2.15: Inclinations and RAANs of orbits that were found after simulating GASTs of (a) 0 degrees, (b) 100.84 degrees, and (c) 70 degrees on the bottom left.

From Figure 2.15, the impact of the GAST value on the orbit solutions can be observed. At 80 degrees inclination for example, when RAAN is 0 degrees, there is an overflight of at least one target by a satellite on those orbits, when the GAST is 0 and 70 degrees, but when the GAST is 100.84 degrees, there is no overflight.

To further analyse the effect of the GAST and the advantage in terms of using the presented method, a ground-track comparing the different GAST values is presented in Figure 2.16. This figure shows that the main effect is a shift of the initial ground track longitude. To produce the ground track, one of the calculated orbits obtained when GAST = 100.84 deg. is used. The orbital values and the time span are as presented in Table 2.13. From these results, it can be observed that the orbital solutions are dependent on the reference epoch time and to get to a target, a ground-track shift can be achieved using the GAST. This is shown by the shift in ground tracks as the GAST value changes.

Table 2.13: Orbital element values used to simulate the difference in ground-tracks for different GAST values.

Parameter	Inclination	RAAN	Time Span
Value	56.60 deg	213.01 deg	7200 sec

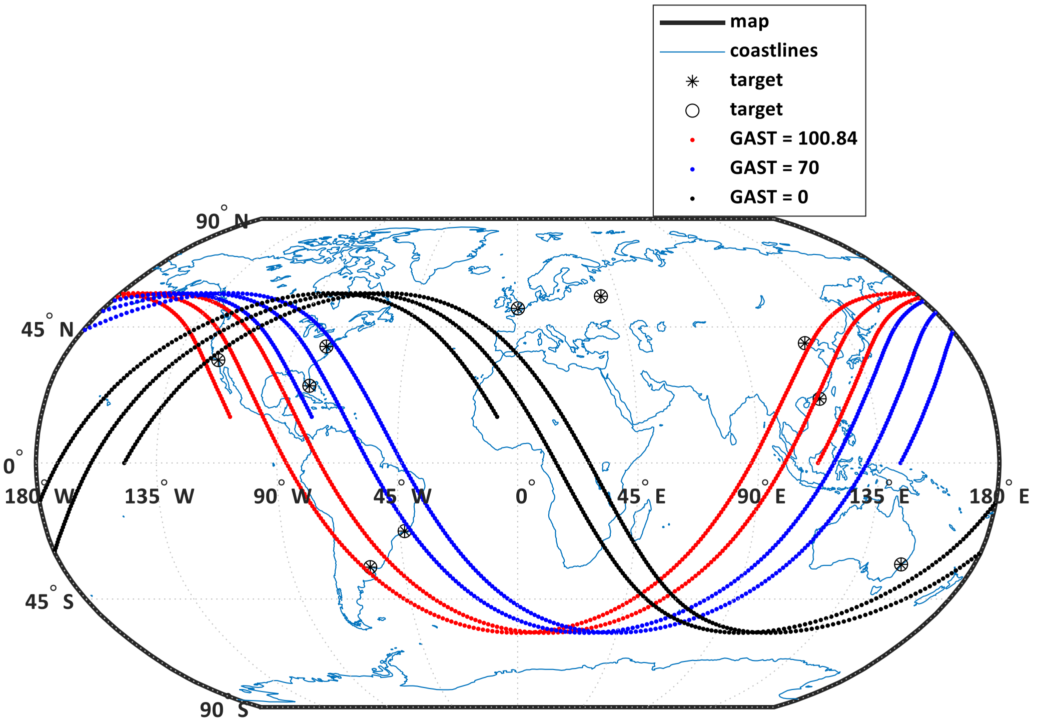


Figure 2.16: Ground tracks of different GAST values when all other orbital elements are the same

From this analysis, it is observed that a ground track shift using the starting time can ensure a pass over a desired target. The difference in longitude to get to a target in case an orbit does not facilitate its overflight, can be calculated. This may determine how much the ground track's starting longitude needs to be shifted to ensure an overflight. The case presented in Figure 2.16 uses an orbit that facilitates the overflight of the targets with a GAST of 100.84 degrees; when the orbit is shifted however, i.e., using a different GAST, some of the targets viewed by the GAST 100.84 degrees orbit are not viewed. This concept of shifting ground tracks to get the desired orbits has been studied by scholars such as C. Circi et. al., [21] who use a sliding ground track method to get to their desired orbit. C. Circi et. al., do this by implementing manoeuvres in inclination and altitude. The method developed in this chapter can therefore provide guidance on an orbit that would require shifting the RAAN instead and this can be done as future work. Generally, this analysis shows that the derived method can be an ideal guide for, orbit designers, satellite orbit determination, and the fact that it is fast and gives general insights, it saves on computational time and ideally computational costs.

Some valuable initial insights can be drawn from the results as it shows the full search space solutions. The errors and validation are however analysed in section 2.3.2 and 3.1 of this dissertation. To show an application of the results obtained from the method, a condition that all determined orbits must facilitate overflight of all ten targets at least once is imposed. The full simulation does not need to be re-run but an algorithm that includes this condition is added. From this, the full search space solutions of the determined orbits are graphically given on Figure 2.17 in terms of the orbit Inclinations and the RAANs. The values of the search space, the targets and the physical values are as in, Table 2.10, Table 2.11, page 43, and Table 2.9, page 42, respectively.

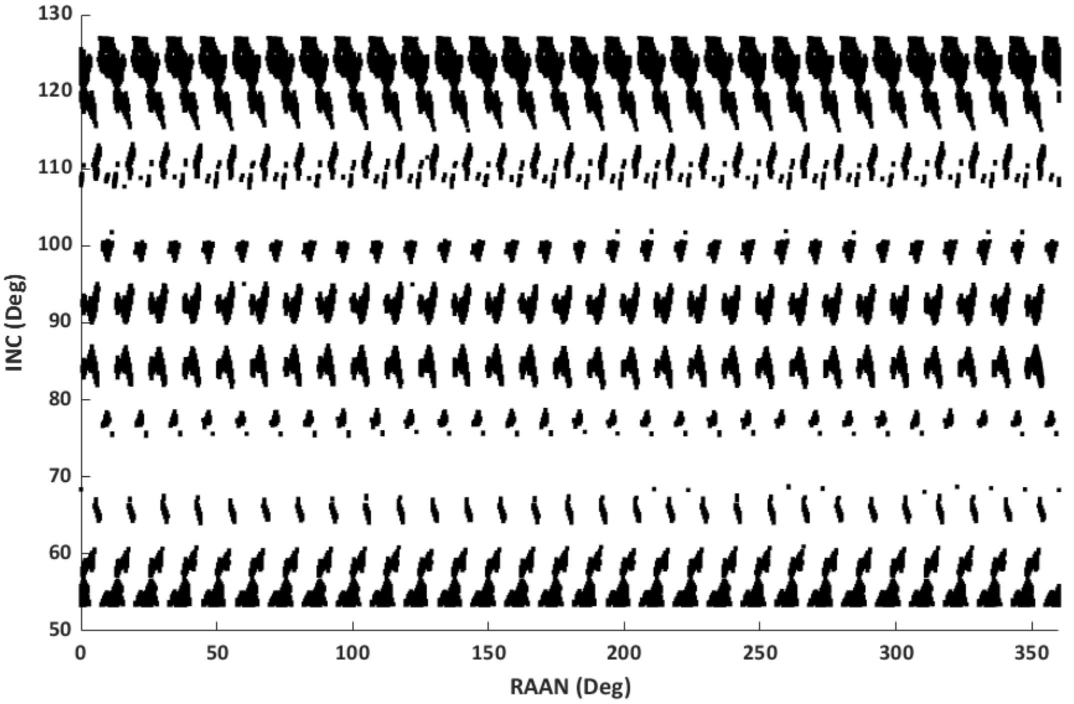


Figure 2.17: Solutions of orbits, (Inclination and right-ascension combinations), which facilitate a view of all the targets at least once in 2 days.

Figure 2.17 illustrates the orbital solutions that provide at least one overflight of all ten targets within 48 hours of epoch time. Due to the condition that the determined orbits must facilitate overflight of all targets, the solutions are fewer than those that overfly at least one of the targets. This shows that the method gives results that are logical. From the results presented in Figure 2.17, it can also be observed that satellites on some orbits below 55.5 degrees inclination are found to view all the targets but that no orbits below approximately 53.4 degrees inclination enable the view of all the targets. This can be attributed to the fact that the highest latitude target point is target 1, Moscow, which is at 55.5 degrees latitude. Due to the field of view angle of the instrument selected, i.e., 20 degrees, at an inclination of 53.4 degrees, target 1 is viewed. This again shows that the method logically works as expected. Some insights can be drawn from these results. One is that even above an inclination of 55 degrees, there are some inclinations that do not view all the targets, and those that do, have some RAAN values that do not view all the targets. This can be quite a useful insight to an operator or for certain satellite tasking purposes as well as revisit orbit designs.

Another insight is that satellites on more orbits that are between inclination 50-60 degrees and approximately 120-130 degrees, view all the targets at least once than inclinations above 60 degrees and below 110 degrees. Generally, changes in inclination require more fuel and change in velocity, Δ-V, increase for such manoeuvres. It is therefore more viable to change the RAAN than changing the inclination. At some Inclinations also, some RAANs do not facilitate the viewing of all the targets but when the RAAN changes, they eventually do. This shows that if the inclination, semi-major axis and other classical orbit elements are held constant, the drift in RAAN will cause it to eventually view the targets again without using much fuel. This however depends on the type of mission as some missions require either a specific revisit or in cases of emergencies, it requires exact overflight at specific times. The change of RAAN with time can be computed using equation (2.40). Taking for example an Inclination = 56.9 degrees, Semi-major axis of 7042.1 km, and physical constants same as the ones on Table 2.9 the rate of change of RAAN and other values can be calculated. In this example, the rate of change of RAAN per second is calculated to be -4.45e-05 degrees per second. The change of RAAN per day is therefore 3.84 degrees.

From this, analysis, if RAAN starts at 0 degrees, it will take approximately 93.57 days for it to be at 0 degrees again in an ideal scenario. Further analysis and simulation of this using third party software gave 91 days for the RAAN to be equivalent to the epoch RAAN value. The proposed method can therefore give further insights into what orbital elements to change to achieve overflight of the desired targets.

2.3.2 Justification of Including J2 to the Proposed Method Calculations

In section 2.3.1, the full developed method while including the J_2 earth's oblateness perturbation is simulated. In section 2.2 on the other hand, the haversine distance between a perturbed orbit and an orbit using the same orbital elements but without considering J_2 is given. In this section, further analysis is done on the results obtained from the method developed in this dissertation when J_2 is included in the calculations and when J_2 is not included. A comparison of the two is used to show the impact of not including the perturbation in the developed method. For this, three target points are considered. The input orbital parameters, the targets, and the physical constants are presented in Table 2.14, Table 2.15 and Table 2.9, respectively.

Table 2.14: Search Space parameters, the increments are now 0.2 degrees for inclination and 0.5 degrees for right ascension of the ascending node.

Parameter	Range	Increments	Units
Inclination (deg.)	50-70	0.2	Deg
RAAN (deg.)	0-360	0.5	Deg
Time (From Epoch) (sec.)	0 – 172800	10	Sec

Table 2.15: List of targets used for simulation of inclusion and exclusion of J_2, Earth secular perturbations.

Target	Description	Longitude, deg	Latitude, deg
1	Celtic Park, Centre spot	-4.2055	55.8497
2	Fenway Park, Home Plate	-71.0977	42.3462
3	Eden Park, Centre Spot	174.744	36.8749

From the simulations, Figure 2.18 and Figure 2.19 which show the orbit solutions when perturbations are considered and when they are not is obtained.

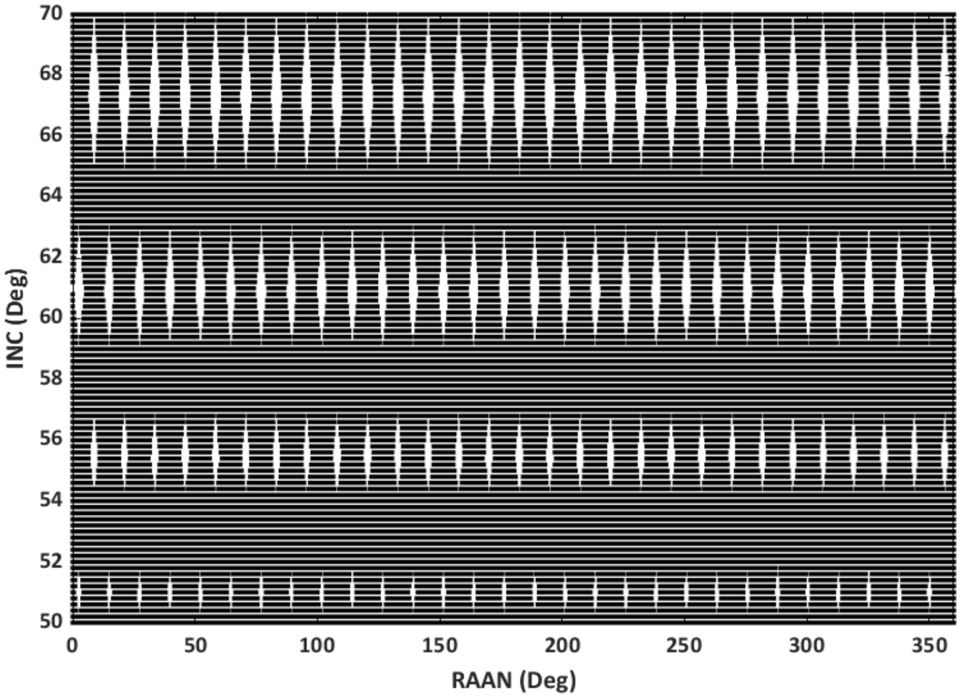


Figure 2.18: Inclinations and Right-ascensions of orbits that enable the view of all targets at least once when no perturbations are included.

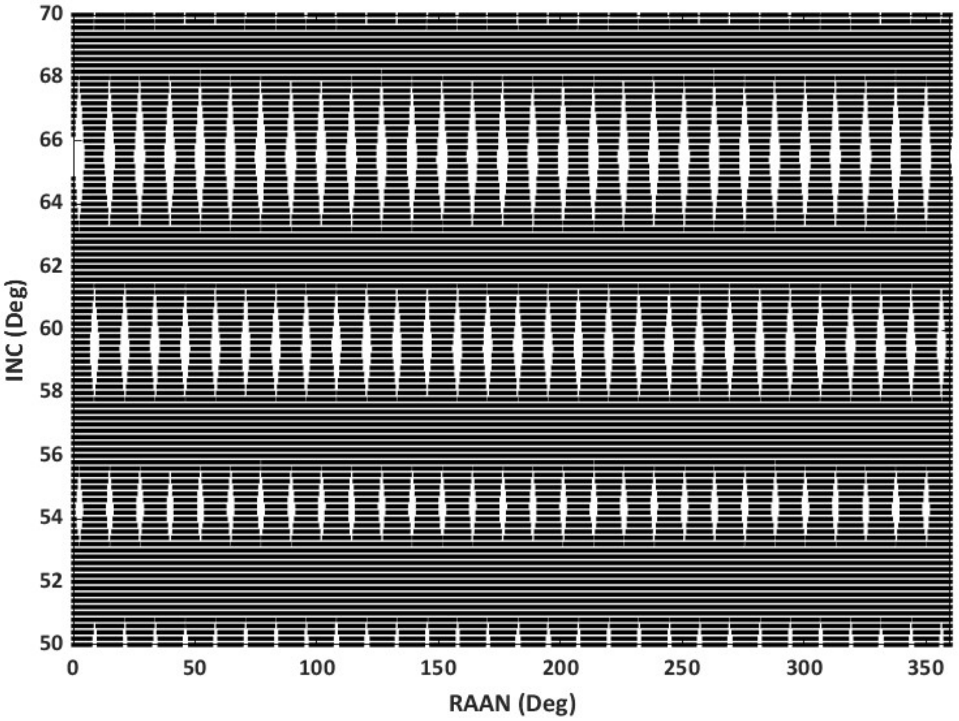


Figure 2.19: Inclinations and Right-ascensions of orbits that enable the view of all targets at least once when J_2 perturbations are included.

From Figure 2.18 and Figure 2.19 presented, it is evident that when J_2 is not included, there are not only RAAN errors but inclination errors are evident as well. The impacts of the errors are illustrated when considering mission objectives such as maximising the mean number of target overflights. Some of these differences are highlighted in the 3D plot presented in Figure 2.20 showing the inclination, RAAN, and the mean maximum number of overflights of target point 1. Table 2.16 highlights some of these major differences in orbits for the results when J_2 is included and when not for the different target locations. The considered results are those within an inclination range of 50-60 degrees. However, as the period of simulation is short, set to two days, atmospheric drag is not considered despite the orbits being Low Earth Orbits, LEO. For greater periods however, this would be considered for increased accuracy.

Table 2.16: Differences observed for orbits when J_2 is included and when J_2 is not included.

Parameter	Value with J_2	Value without J_2
Number of orbits determined	46381	46391
Maximum number of target 1 overflights	7	6
Maximum number of target 2 overflights	3	4
Maximum number of target 3 overflights	2	3

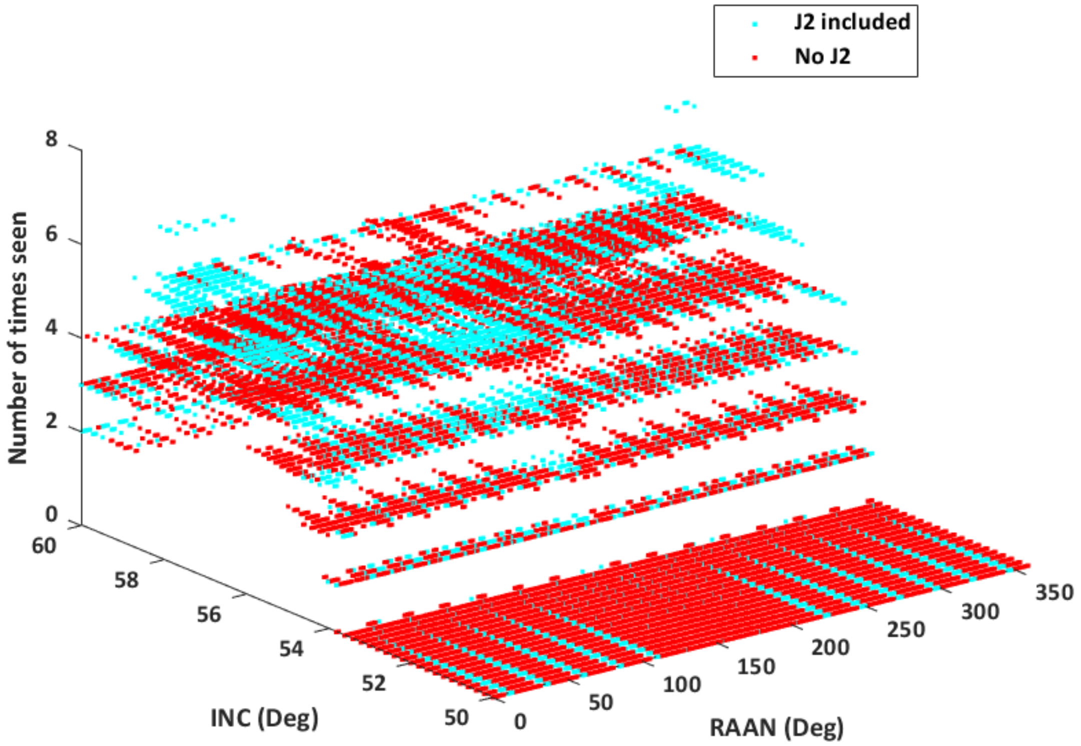


Figure 2.20: 3D graph of Inclination, RAAN and number of times viewed showing the differences when J_2 is present and J_2 not present for target point 1.

For the other targets, as well, there are expected errors in the calculations when J_2 is not included compared to the simulation when it is. Other errors are in the duration of view, which is expected due to different orbits being found, but these are further presented in the appendix section of this dissertation. This shows that the inclusion of J_2 is necessary for the accurate usage of the developed method and more so when target specific missions are being considered.

2.3.3 Error Analysis Using Third Party Software

As research for more accurate methods for both orbit and ground-track propagation are still on going, numerical methods are said to have a higher accuracy but also higher computational cost than analytical methods. This section gives an analysis of the accuracy errors incurred from the presented method by comparing the calculated results to ground-track propagation done using a third-party software. The analysis in this section uses NASA’s GMAT, version GMAT R2022a. The numerical propagator of GMAT is set to use RungeKutta89 and the JGM-3 gravity model of degree 70 and order 70. The targets used are the 10 targets presented in Table 2.11 and the simulation is done using the orbital search space values on Table 2.17.

Table 2.17: Orbit values used for simulation, in this case the RAAN increments are 1 degree.

Parameter	Value(increments)
Inclination Range (Increments) (deg.)	50-60 (0.2)
RAAN Range (Increments) (deg.)	0-360 (1)
Time Range (Increments) (sec.)	0-172800 (10)

Some of the orbits that the method identifies to enable a view of all the targets are also found to overfly all the targets on GMAT. To enable the simulation of an exact scenario used in the method to be used on GMAT, the target minimum elevation angle which is related to the spacecraft elevation angle is calculated to be 67.82 degrees. The calculation used to obtain this angle is presented in Appendix C. To illustrate the validity of the method, two orbits that the method yielded are selected. The total duration of view of the targets by the orbits in 48 hours is calculated. The selected orbits and their total duration of view values both from the method and GMAT are presented in Table 2.18. The two orbits are found to have almost the same total duration of view values when calculated from the developed method and when simulated on GMAT.

Table 2.18: Solutions found for total duration of view by the highlighted orbits when ground-track is propagated by the developed analytical method and on GMAT (2 orbits)

Parameter	GMAT	Developed method	GMAT	Developed method
Semi-Major Axis (km)	7040.5	7040.5	7040.9	7040.9
Inclination (deg)	55.2	55.2	55.6	55.6
RAAN (deg)	150.0074	150.074	225.112	225.0112
Duration of view of targets (sec)	830.407	830	809.1896	810

These results of the ground track from GMAT and the ground track from the method for the 55.2-degree inclination orbit are presented in Figure 2.21 and Figure 2.22 respectively.

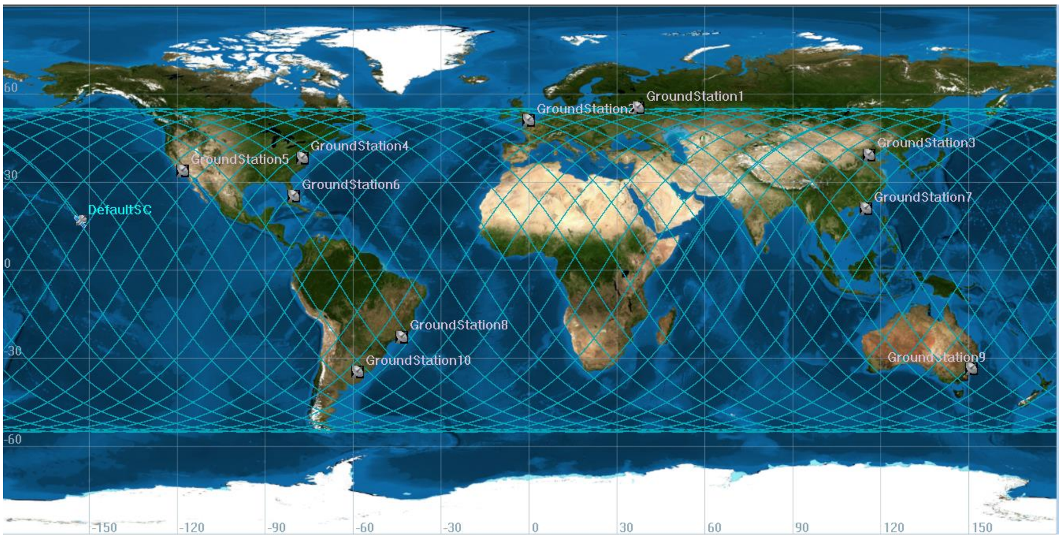


Figure 2.21: Ground track simulated from GMAT propagation for orbit with 55.2 deg. Inclination and RAAN 150.074 deg.

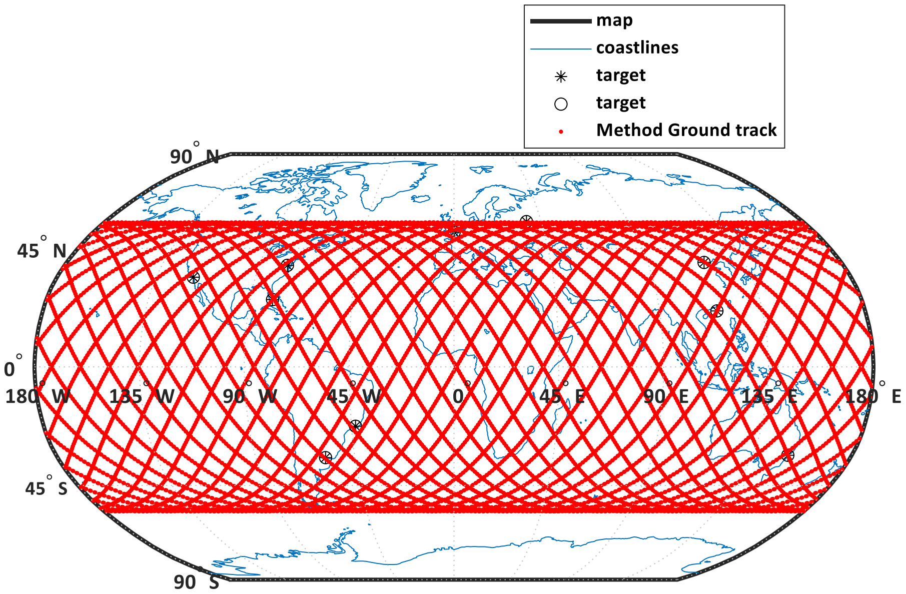


Figure 2.22: Ground track plot from MATLAB simulation for orbit with RAAN==150.0074 deg. and Inclination==55.2 deg.

The differences in longitude and latitude are not obvious from the ground track plots. The results from both simulations are therefore analysed further by considering the longitudes and latitudes individually. The start point from the two methods are similar in epoch longitude as the GAST used is the same for both. The latitudes on the other hand are found to be very similar and have a positive covariance diagonal and a correlation factor of approximately 1 whereas the longitudes have a negative covariance diagonal and a correlation coefficient of -0.0284 which is approaching zero.

Table 2.19: Covariance and correlation comparison of longitude and latitudes from method ground-track and GMAT.

Method	Longitude result	Latitude result
Covariance	-306.00	1376.057
Correlation	-0.0284	0.9865

When ground-track results from the proposed method and the GMAT longitudes and latitudes are analysed and compared for errors, it is observed that the method and GMAT have some differences in both latitude and more so in longitude. To further analyse the differences, and hence validate the developed method, the haversine distance between both simulations are also calculated using the equation for haversine distance, given as,

$$H_d = 2R_E \, sin^{-1}\left[\sqrt{sin^2\left(\frac{\delta_1 - \delta_2}{2}\right) + cos(\delta_1)\, cos(\delta_2)\, sin^2\left(\frac{\psi_1 - \psi_2}{2}\right)}\right] \tag{2.58}$$

From this analysis, Figure 2.23 shows the difference in the distance of SSPs between the method results and the GMAT simulation is obtained. It is observed from Figure 2.23 that the differences in ground distance between the simulations of both methods are no more than 20km. The results of the errors in distance are due to the longitude differences. This can be due to the ground-track simulation using an analytical method and a numerical method (GMAT) hence the differences in methods used, not limited to but including the error models used.

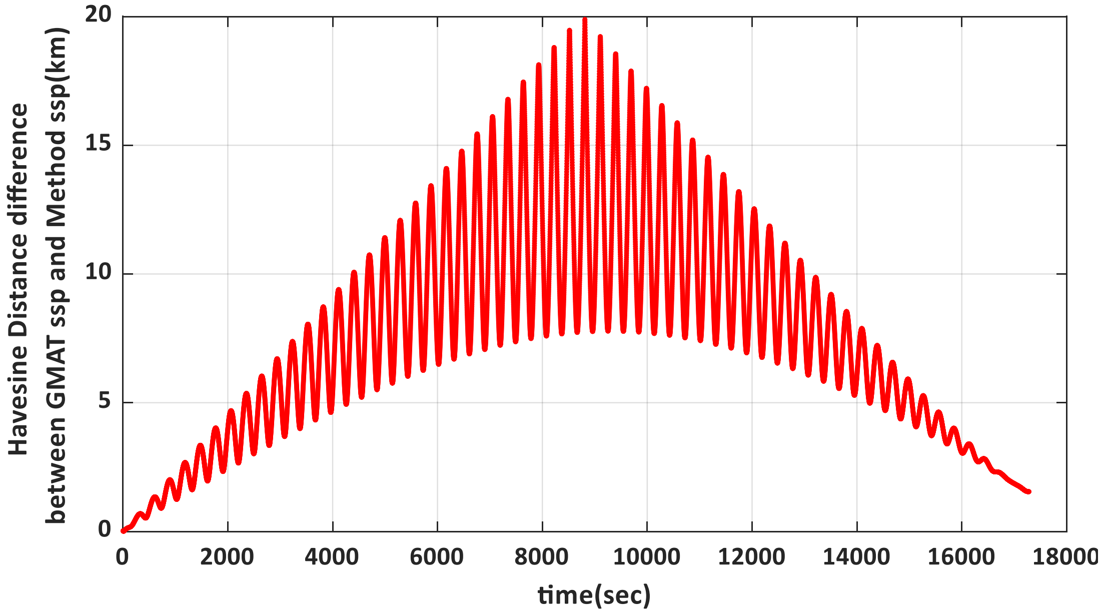


Figure 2.23: Haversine distance differences between the orbit ground track from the developed analytical method and GMAT simulation.

Other orbits are also compared, and, in some cases, the method found some orbits enabling the viewing of all the targets but when simulated on GMAT do not view some targets and have a larger value of the duration of view. This can be attributed to the differences in the methods of simulation, the increments used for the presented method, as well as some errors that may occur when using an analytical method compared to using a numerical method. Based on this analysis however, the ground-track errors of the method can be identified and are not too extreme. The difference in propagation methods and possibly equations used for the propagation and more so, the methods used for ground-track plotting may be the reasons for these errors. This is further analysed in appendix D.

2.4 Chapter Summary

A method that determines orbits that can overfly a given number of desired ground passes has been developed in this chapter. The method is analytical as well as embarrassingly parallel in nature. It is robust and applicable to large search spaces. In section 2.3, the method has demonstrated capabilities of determining orbits given both prograde and retrograde search spaces. The key novelties obtained from the development of the method are contributing a novel method that can determine orbits for a given set of ground points using analytical equations and minimising assumptions by including the J2 secular perturbation. The accuracy of the method has also been studied by doing an error analysis against a third-party software, GMAT, that is used by orbit designers to propagate their orbits. A case study that would include the atmospheric perturbations may be included to further develop the method's accuracy. The method has filled the gap in literature of an analytical ground-track propagator that determines orbits based on desired overflights.

Chapter 3

Validation and Case Study Applications of the Proposed Method

Some previous methods for orbit determination and design use single objective function optimisation [77]. More recently the need for multi-objective function optimisations (MOOs), has been realised [78]. Some MOOs are high-dimensional and dynamic hence challenging to the performance of some algorithms [78]. In general, most realistic scenarios call for MOO. For the study of multi-objective functions, scholars use methods such as, Genetic Algorithms (GA), improved GA's, Differential Evolution, and other Evolutionary Algorithms, amongst others [58, 77, 79-81]. As presented in studies done by authors such as Refs. [78, 81], all these algorithms have their advantages and disadvantages depending on the optimisation. This chapter presents the optimisation of the orbits determined by the developed analytical orbit design method. Optimisation is done with single-objective function then with multiple objective functions. The embarrassingly parallel nature of the presented method makes it more robust in that the determined orbits are not objective function based and so even when the objective function changes, there is no need to re-simulate the valid orbits. This makes the method robust and ideal even for multiple objective function optimisation. A validation of the developed method is first presented using a single objective optimisation comparison.

3.1 Validation of the Results of the Proposed Method by Comparison with Results from Previous Methods

To validate the presented method, a comparison is done with the results from a previous method, which uses an improved self-adaptive differential equation, (SA-DE), to determine the orbit that maximises the total duration of view over multiple targets [57]. W. Yao et. al., [57], use a double SA-DE algorithm with a random mutant to determine the orbit that facilitates an optimum total duration of view of 10 targets. The targets W. Yao et. al., use are the 10 targets presented in section 2.3, Table 2.11 but they assigned a priority value for each target. The method by W. Yao et. al., improves on previously used numerical methods by enabling them to jump out of local optimums due to the introduction of a random mutant.

The methodology developed in Chapter 2 of this dissertation aims to improve this search for optimum solutions by using analytical equations to determine orbits that facilitate overflight of the targets, and to give an overview of solutions of the entire search space. The duration of view objective function is used for this validation. The objective function is formulated as follows just like in W. Yao et. al., [57]:

$$J_t = \sum_{i=1}^{N1} \frac{P_i(\sum_{j=1}^{N2} t_{i,j})}{10} \tag{3.1}$$

$$J_b = \sum_{i=1}^{N1} P^*. o_i \tag{3.2}$$

J_t represents the priority weighted part of the objective function and in this case the division by 10 is because there are 10 targets being overflown. J_b is a binary function of the objective function indicating whether a target has been overflown or not. Where o_i indicates whether a target has been visited or not in binary i.e., for a visit $o_i = 1$ and when not visited $o_i = 0$; P^* is then assigned 100 if the target needs to be visited. $t_{i,j}$ represent the duration of time each visit is, per target, P_i, is the priority value assigned to each target, and $N1, N2$ are the number of targets and the times of visit for each target respectively. The objective function in W. Yao et. al. is then optimised using,

$$max(J) = J_t + J_b \tag{3.3}$$

For the comparison however, as all the 10 targets need to be visited at least once, the objective function only considers $\max(J_t)$ as J_b (== 1000) for all orbits (i.e., all orbits visit all targets at least once).

The parameters for this simulation are, physical constants in Table 2.9, search space parameters presented in Table 3.1 and the targets with their priority values presented in Table 3.2.

Table 3.1: Search Space parameters for validation of the developed method

Parameter	Range	Increments	Units
Inclination	50-130	0.05	deg.
Right Ascension of Ascending Node	0-360	0.05	deg.
Time (From Epoch)	0 – 172800	10	sec.

The desired targets with their priority values as per the work of W. Yao et. al., [57] are as follows,

Table 3.2: List of targets and the priority values used for the objective function analysis and comparison with previously used methods' results.

Target	City	Longitude, deg.	Latitude, deg.	Priority values
1	Moscow	37.4	55.5	0.72
2	London	0.1	51.3	0.85
3	Peking	116.2	39.6	1.00
4	Washington, D.C	-77.0	38.5	1.00
5	Los Angeles	-118.2	34	0.85
6	Miami	-80.1	25.5	0.73
7	Hong Kong	115.1	21.2	0.68
8	Rio	-43.2	-22.5	0.62
9	Sydney	151.1	-33.5	0.90
10	Buenos Aries	-58.3	-34.4	0.65

The results obtained from the objective function simulation of the duration of view, J_t, are presented in Figure 3.1. From this analysis, the solution scope for the optimum J_t value orbit is large and the optimum and sub-optimum values are very close to each other. This can easily lead some numerical methods to getting stuck in local optimums [24].

Unlike previous methods used, the developed method includes a search space of both retrograde and prograde orbits. Retrograde orbits prove to have higher objective functions than prograde orbits. In the work by W. Yao et. al., [57], they only consider the prograde orbits and this may be due to the computational time needed. There is a gap in literature on the analysis of retrograde orbits for objective functions such as the duration of overflight over targets. This shows that the presented method is an improvement of some previously used orbit design methods. As a direct comparison to the method by Ref. [57] therefore, a comparison of the prograde orbit results is included for validation of the method herein developed.

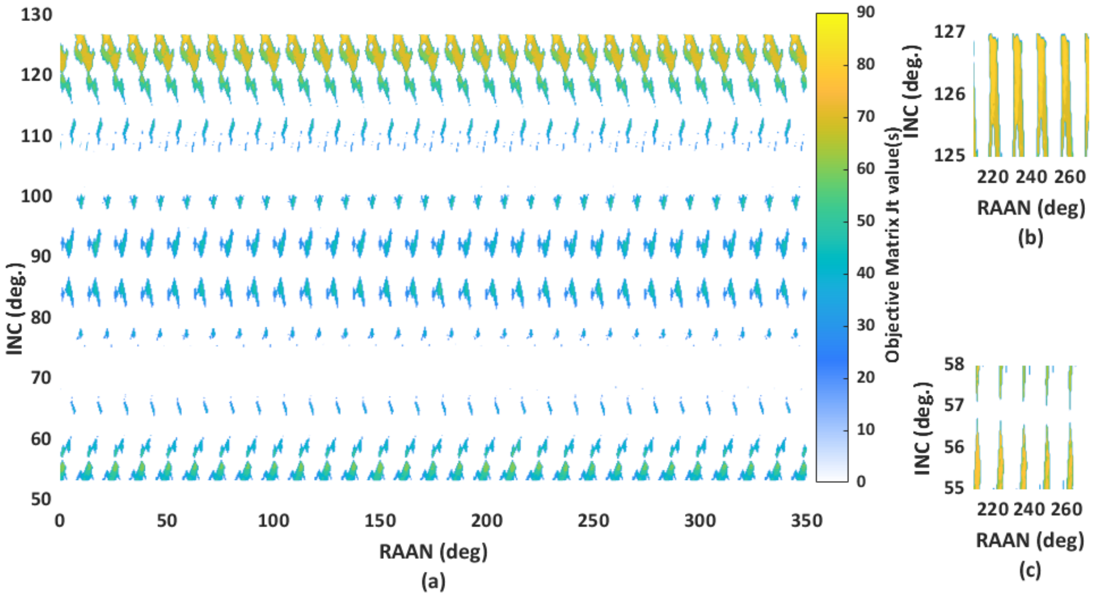


Figure 3.1: Inclination and RAAN of determined orbits with the heat plot representing the duration of view objective function values where optimum duration of view orbits are the yellow spots in graph (a) and zoomed into on subgraph (b) and (c).

Comparing the results from the developed method and that of Refs. [53] and [57], the methodology developed in this dissertation find a higher objective function value. For [53] and [57], the two optimum objective functions are J_t values of 79.43 and 80.11 respectively. The priority values used for each target and the physical constants are the same for all methods and the result of this analysis is presented in Table 3.3.

Table 3.3: A comparison of objective functions of duration of view calculated from propagation by developed analytical method and previously used numerical methods.

Methodology	Objective Function Value (J_t)	Inclination, deg.	RAAN, deg.
[57]	80.11	55.51	125.69
[53]	79.43	55.52	30.20
Developed Method Optimum orbit	94.16	126.20	111.21

For a more direct comparison and to show the developed methods capabilities, the inclination of 55.51 +/- 0.05 deg. is simulated using the method developed in this study. When an optimisation of the orbits obtained is done, a more optimum duration of view orbit is obtained. The optimum objective function value is 84.47 at an inclination of 55.55 deg. and a RAAN of 54.20 deg. This can be attributed to the embarrassingly parallel nature of the developed method which eliminates the possibility of getting stuck in local optimums. It is also impacted by the refinement used in the simulation. The developed method calculates an optimum viewing duration for prograde orbits that is more than 30 seconds longer than the results obtained by the previous numerical methods. Additionally, when simulating the orbits of the referenced methods on third-party GMAT, the optimum J_t value orbit presented by Y. Chen et. al., [53] does not enable overflight of target 8, and the optimum J_t value orbit presented by W. Yao et. al, [57], enables overflight of all the targets but almost misses target 4. This shows the propagation errors whilst using different methods. With the GMAT simulation, the orbit calculated by the method developed in this dissertation have a greater duration of view compared to the referenced methods. The results from the presented method prove to determine/calculate more optimum J_t orbits. This shows that the method is an improvement of some previous orbit design methods. The methods by Y. Chen et. al. and W. Yao et. al, also include J_2 perturbations and do not include the atmospheric drag; they also use the same epoch time as in the developed method presented.

3.2 Case Study 1: One Objective Function

From the simulation of the presented method as given in chapter 2, the results are analysed and optimised based on different objective functions without a need to re-run the orbital simulations unless the inputs, i.e., the search space values (range or increments), the target points, or the physical constants, change. For the case studies, the targets and their three assigned priority values are presented in Table 3.4. The search space physical constants are as in Table 2.9, with a GAST of 259.16 degrees, and the orbital search space parameters are presented in Table 3.5.

Table 3.4: List of targets with additional objective function priority values.

Target	City	Longitude, deg	Latitude, deg	Priority values 1	Priority values 2	Priority values 3
1	Moscow	37.4	55.5	0.72	0	2
2	London	0.1	51.3	0.85	1	1
3	Peking	116.2	39.6	1.00	1	1
4	Washington, D.C	-77.0	38.5	1.00	1	1
5	Los Angeles	-118.2	34	0.85	1	1
6	Miami	-80.1	25.5	0.73	1	1
7	Hong Kong	115.1	21.2	0.68	1	1
8	Rio	-43.2	-22.5	0.62	1	1
9	Sydney	151.1	-33.5	0.90	1	1
10	Buenos Aries	-58.3	-34.4	0.65	1	1

Table 3.5: Search Space parameters used to determine valid orbits.

Parameter	Range	Increments	Units
Inclination	50-130	0.2	Deg
Right Ascension of Ascending Node	0-360	0.2	Deg
Time (From Epoch)	0 – 172800	10	Sec

3.2.1 Mean Number of Times Seen Objective Function:

The first objective function in this study is based on the number of times that a target is seen. For this first analysis, no target is prioritised over the others and from the method, the determined orbits are those that facilitate overflight of at least one of the targets. The equation used for the optimisation is,

$$J_{ts} = \sum_{i=1}^{N1} \frac{(\sum_{j=1}^{N2} t_{i,j})}{10} \qquad (3.4)$$

Where J_{ts} is the calculated mean number of times seen objective function value of each orbit, and $t_{i,j}$ is a visit of each target is overflown by each orbit.

From the simulation and the calculations, the optimisation heat plot results obtained for the mean number of times of view facilitated by each determined orbit are presented in Figure 3.2. It can be observed from Figure 3.2 that prograde orbits give lower mean number of times in view values than retrograde orbits. The optimum J_{ts} value and the associated orbit Inclination and RAAN are given in Table 3.6.

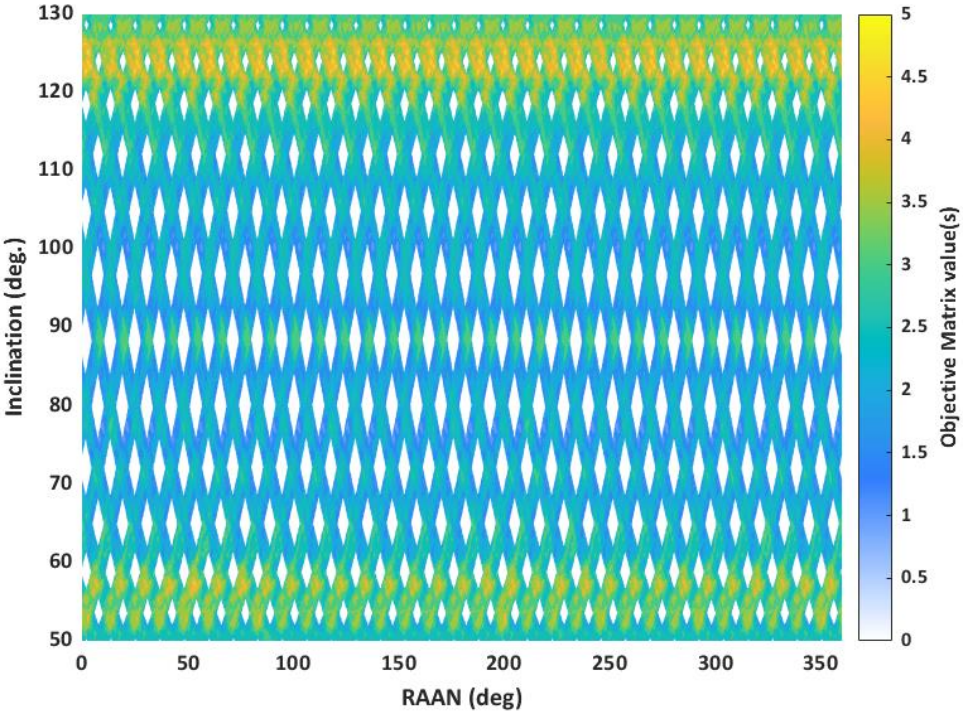


Figure 3.2: Inclination and right ascension of orbit heat plot showing the mean number of times of view (objective matrix value) of each orbit when the targets are viewed without assigning any priority values i.e., each target has equal priority.

Table 3.6: From *Figure 3.2*, these is the data of the Maximum Objective function value calculated.

Parameter	Maximum Jts	INC (Deg) (Maximum Jts)	RAAN (Deg) (Maximum Jts)
Value	5.00 times	124.20	153.42

In the first optimisation, for the results generated on Figure 3.2, all targets are considered to have equal priority. In some cases, however, the targets may have different levels of priority. The varying priority values of each target are given in Table 3.4, and the calculation for this is equation (3.5),

$$J_{ts} = \sum_{i=1}^{N1} \frac{P_i(\sum_{j=1}^{N2} t_{i,j})}{10} \qquad (3.5)$$

Where P_i is the priority value assigned to each target accordingly. On running an optimisation of the determined orbits, the results are presented in Figure 3.3 where it is observed that as the priority values of the targets change as per Table 3.4, the maximum objective function value of the mean number of times each target is seen changes. This shows that as the priorities of the targets change, the optimum J_{ts} orbit also changes. As this work requires no re-run of the orbital simulation, the advantage is that the changing optimum J_{ts} orbits can be determined with minimum computational time and cost. On Figure 3.3, the colour bar shows the objective function values. Table 3.7 shows some of the optimum J_{ts} values obtained from this simulation and the orbits that facilitate this maximisation.

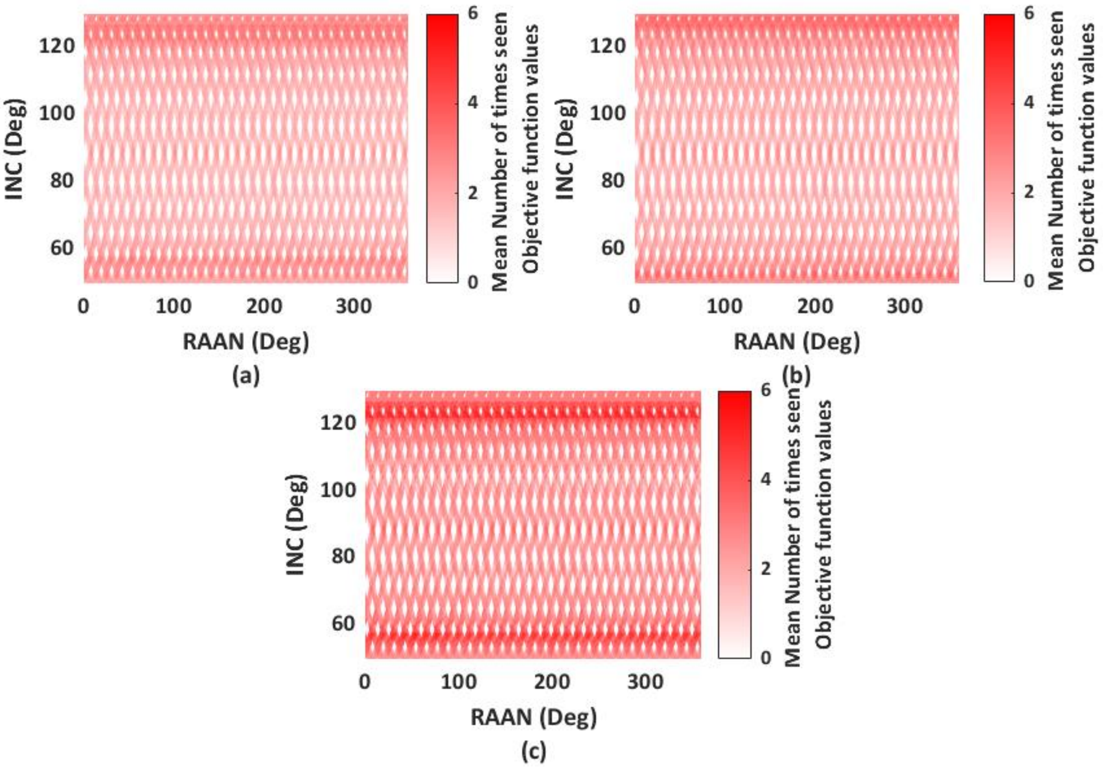


Figure 3.3: Objective function values of mean number of times seen for different target priority values as listed in *Table 3.4* respectively i.e., graph (a) target priority values are varying according to Priority values 1 on *Table 3.4*, graph (b) target priority values are varying according to Priority values 2 on *Table 3.4*, graph and graph (c) target priority values are varying according to Priority values 3 on *Table 3.4*, graph.

Table 3.7: The details of the Maximum and minimum mean number of times seen objective function when targets have varied priority values.

Parameter	Priority Values of target 1 to 10 respectively	Maximum J_{ts}	Inclination (deg.)	RAAN (deg.)
Value	0.72,0.85,1.00,1.00,0.85, 0.73,0.68,0.62,0.90,0.65	3.97	124.20	153.42
Value	0,1,1,1,1,1,1,1,1,1	4.30	52.60	84.20
Value	2,1,1,1,1,1,1,1,1,1	6.00	122.20	2.00

These values show that different orbits facilitate the “best” or “Optimum” J_{ts} values for different priority values of the targets even when considering just one objective function.

The problem becomes even more complicated when there are multiple objective functions. As previously mentioned however, in real life applications, it is rare that a satellite is needed for only one objective. Most of the real-life scenarios involve multi-objective optimisation. Such kind of a problem is considered highly combinatorial. Section 3.3, highlights cases of multiple objective functions and the solutions.

3.3 Case Study 2: Multiple Objective Function Applications

In the recent past, some space missions have been found to have more than one mission requirement. As in the case of the Korean SAR satellite, KOMPSAT-5, the focus is on monitoring a local target area. The main requirements for their mission are, minimising the average revisit time as well as maintaining a repeat ground track orbit [59]. Other requirements such as viewing the targets for as long as possible within a given period, viewing the targets a given minimum number of times per day, and many other different requirements call for multiple objective function optimisation. Multiple mission requirements may also be used for tasking satellites on different orbits. Determination of orbits that facilitate the performance of multiple objective functions optimally with computational efficiency is a gap identified. This optimisation for multiple satellite missions can be used to also determine satellite constellation designs instead of using the traditional approaches such as the Fowler and Walker constellation paradigms.

In this section, the possible orbits that can overfly 10 targets are determined using the embarrassingly parallel analytical method. This method as presented before, yields more than 10,000 orbits. For purposes of this work, one of the main mission requirements is that the orbit must be a repeat ground track orbit. The repeat period is set to be 2 days with a repeat revolution of 29. For all cases, J_2 coefficient for earth is considered.

Three objective functions are optimised and the optimum orbits identified. The three objective functions are as per the following sections:

3.3.1 Mean Duration of View Objective Function

This objective function is as in the work of both W. Yao et. al. [57] and Y. Chen et. al. [53] and is formulated as in equation, (3.1), page 58. That is,

$$max(J) = J_t \tag{3.6}$$

3.3.2 Mean Number of Times Seen Objective Function

The second objective function in this study is based on the number of times that a satellite in an orbit will overfly the targets. The mean number of times that a target is viewed by a satellite in each orbit is selected using equation (3.5), Page, 63.

3.3.3 Revisit Time Objective Function.

The third objective function considers the time it takes to revisit a target. H.-D. Kim et. al., [59], consider the average revisit time objective function. H.-D. Kim et. al., focus on the average revisit time between observations for the same ground point. The method by H.-D. Kim et. al., minimise this to be no more than 24 hours per revisit. In the study presented herein, three scenarios are simulated and presented. Scenarios such as determination of the orbits that enable views of each target at least once in a 6-hour period, then at least once in less than 6 hours can also be a mission requirement.

Different conditions have been studied and applied in the case studies herein presented. These conditions can also be applied to different targets, for example, if target 1 needs more than two visits with a spacing greater than 12 hours each within the 48 hours.

For a case where a satellite in orbit (o) first view of target T is x seconds after epoch time, if it views target T again after 12 hours, the orbit is selected. This can also be applied for 6 hours and for 24 hours and the approach for this is presented in Algorithm 3.1 .

Algorithm 3.1 Calculation of the revisit time objective function

```
input: data from propagation
output: target revisit limits
for each orbit and target, T(i,j)
    x(T(i,j)) == first view // first view of each target by each orbit
    if next view <(x + 21,600) seconds do// checks if the next view is within 6 hours
        $\beta(o,t) = {}^{1}_{0}\begin{cases} if\ true \\ if\ false \end{cases}$
        each orbit number of views within this period = sum $(\beta(i,j))$
        if 0 < number of views >3
```

```
            rt = 1
        Else
            rt = 0
        End
        if 1<rt<3
            Jrt = 1
        Else
            Jrt = 0
        End
    End
End
store Jrt
```

The detailed method formulae are equations (3.7) and (3.8),

$$J_1 = \sum_{\substack{i=1\\j=1}}^{N1} K(i,j) \tag{3.7}$$

Where J_1 is the total number of times the targets are overflown by an orbit within the given time x, K indicates an overflight of target j by orbit i;

$$J_{RT} = \begin{matrix}1\\0\end{matrix}\begin{cases} if\ 0 < J_1 < 3 \\ \quad if\ not \end{cases} \tag{3.8}$$

3.3.4 Multi-Objective Function Optimisation

The objective functions are first calculated individually. The objective functions are then optimised simultaneously without having to re-simulate the valid orbits. The individual objective functions are normalised where the maximum objective function of each == 100. Priority functions are assigned to each normalised objective function as X, Y and Z. These values of X, Y and Z can be different for the different objective functions. Some objective functions may be more weighted than others in a case where they are to be more prioritised. Considering this, the calculation of the generalised objective function then becomes,

$$f1 = Min\ \frac{1}{XJ_t + Y\ J_{ts} + Z\ J_{RT}} \tag{3.9}$$

Where, J_t is the mean duration of view objective function as presented in in equation, (3.1), page 58, J_{ts} is the mean number of times that a target is viewed objective function as presented in equation (3.5), Page, 63, and J_{RT} is the revisit time objective function as presented in equation (3.5), Page, 66. F1 is the generalised objective function value and X + Y + Z == 1 in increments and reductions of 0.1

The search space parameters are as in Table 3.5, the 10 targets in Table 2.11, and the physical parameters are as in Table 2.9 but with a GAST of 259.16 degrees.

For the mean duration of view, mean number of times seen and revisit schedules both actual and normalised objective functions are obtained. The normalised values are presented in Figure 3.4 and Figure 3.5. For all cases, the calculations consider that all targets have a priority of 1. Figure 3.4 shows that the maximum values for the duration of view objective function are found to be on the retrograde orbits as expected.

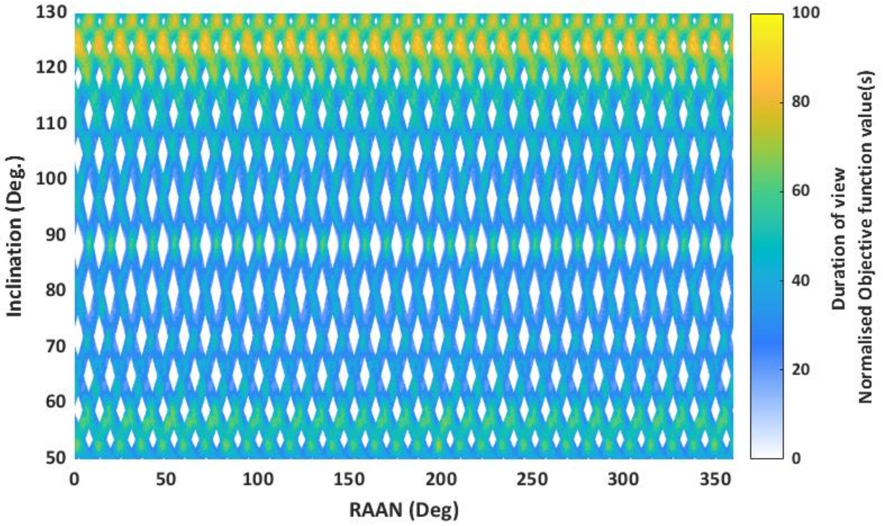


Figure 3.4: Inclination and RAAN heat plot results for orbits showing the normalised mean duration of view objective function values.

From Figure 3.5, it can be seen that for the mean number of times viewed objective function values, despite the maximum objective function (OF) value (normalised value of 100) being a retrograde orbit, there are some prograde orbits that have OF values very close to the maximum.

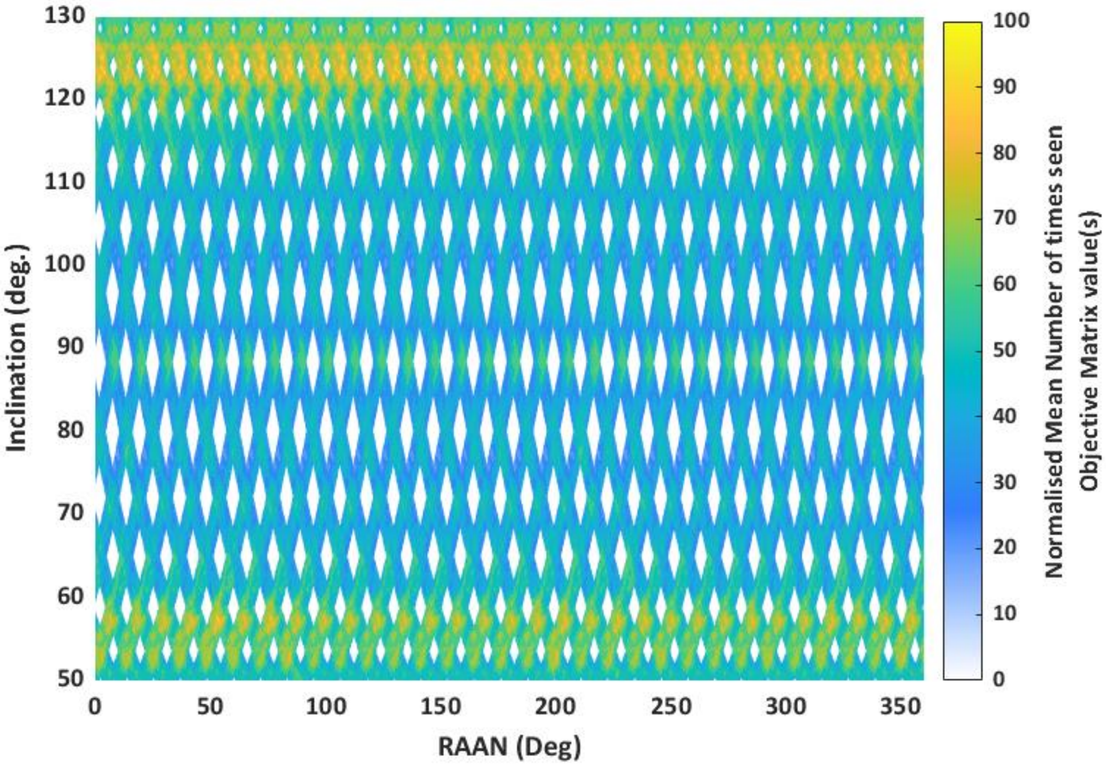


Figure 3.5: Inclination and RAAN heat plot results for orbits showing the normalised mean times of view.

For the revisit times, three different scenarios are simulated,

i. when at least one view is needed for a time of view difference less than 6-hours,

ii. when one or two views are needed for time of view difference not less than 12 hours and,

iii. when at least one view when difference between view times is greater than 24 hours.

The results from these three scenarios are presented as heat plots in Figure 3.6 ((a) – (d)) respectively. The option of 2-views in a time difference greater than 12 hours, i.e. option (c) from Figure 3.6 is selected for the multi-objective function simulation due to its diverse orbit solutions hence presenting a complex case for study. Figure 3.7 (a) presents the normalised results of the 2 views in 12 hours for 48 hours option chosen for multi-objective optimisation.

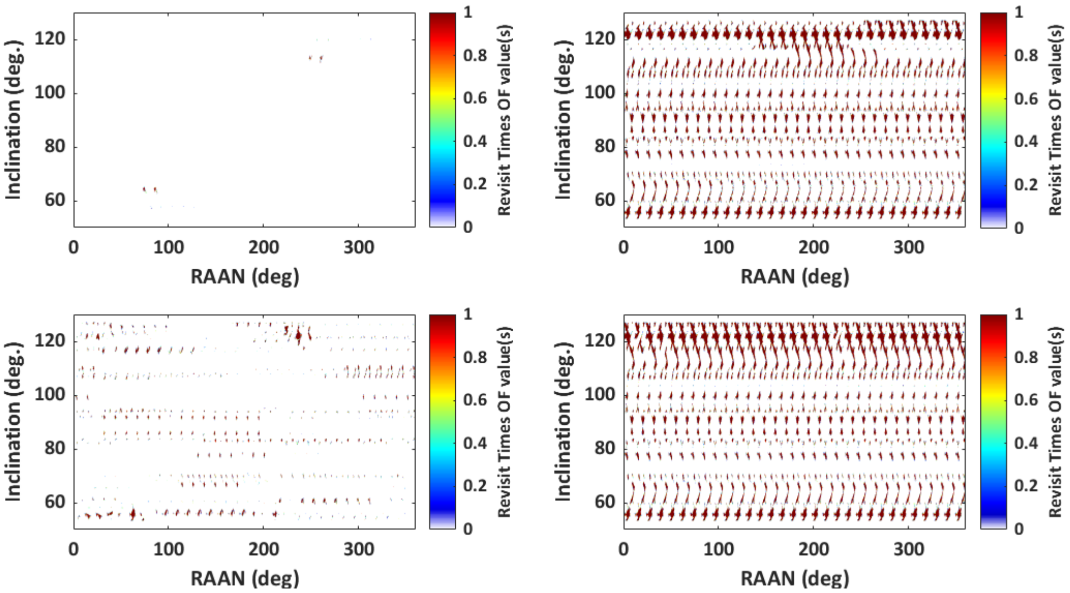


Figure 3.6: Inclination and RAAN heat plot results showing the best orbits for different revisit schedule time objective functions (OF); (a) 1 view spaced out in less than 6 hours for 48 hours, (b) at least 1 view in 12 hours for 48 hours, (c) 2 views every 12 hours for 48 hours and (d) view within 24 hours for 48 hours.

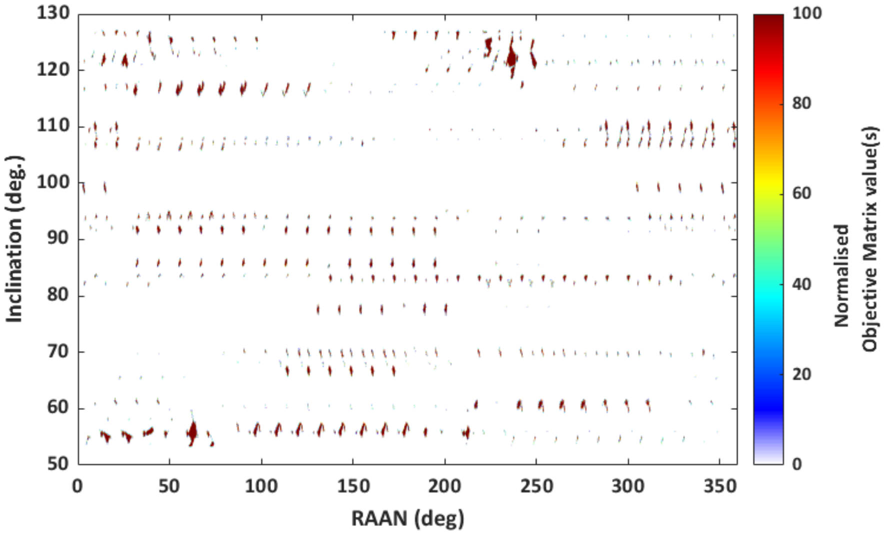


Figure 3.7: Zoomed in Inclination and RAAN heat plot results for orbits showing the best orbits for 12-hour revisit schedule showing the normalised values.

When the objective functions are simulated individually, the maximum objective function values are as in Table 3.8.

Table 3.8: Maximum objective functions for the individual objective functions for inclination 50 – 130 degrees, RAAN 0 – 360 degrees and time 0 – 172800 seconds

	Inclination (deg.)	RAAN (deg.)	Max. Objective Function
Mean Number of view times	124.20	153.42	5.00
Duration of view	124.20	256.03	108.00
12-hour revisit time	Multiple	Multiple	10.00

It can be observed that the optimum orbits for the individual objective functions are not the same and so f1, as per equation (3.9), page 67, is needed to be able to get an orbit that relates to them all. For equation (3.9), to be applied, the normalised objective functions need to be calculated. The maximum objective functions are taken to be 100 then the rest are normalised to this. When this is done, the graph of the normalised objective function values which are greater than 80 (maximum values) is presented in Figure 3.8. This shows the obits that are within 80 – 100 normalised values of each objective function. Where, $J_t N$ is the normalised mean duration of view objective function, $J_{ts} N$ is the normalised mean number of times that a target is viewed objective function and $J_{RT} N$ is the normalised revisit time objective function. From this, the insight drawn is that the orbits that have the maximum objective functions are retrograde for both mean number of times seen and duration of view.

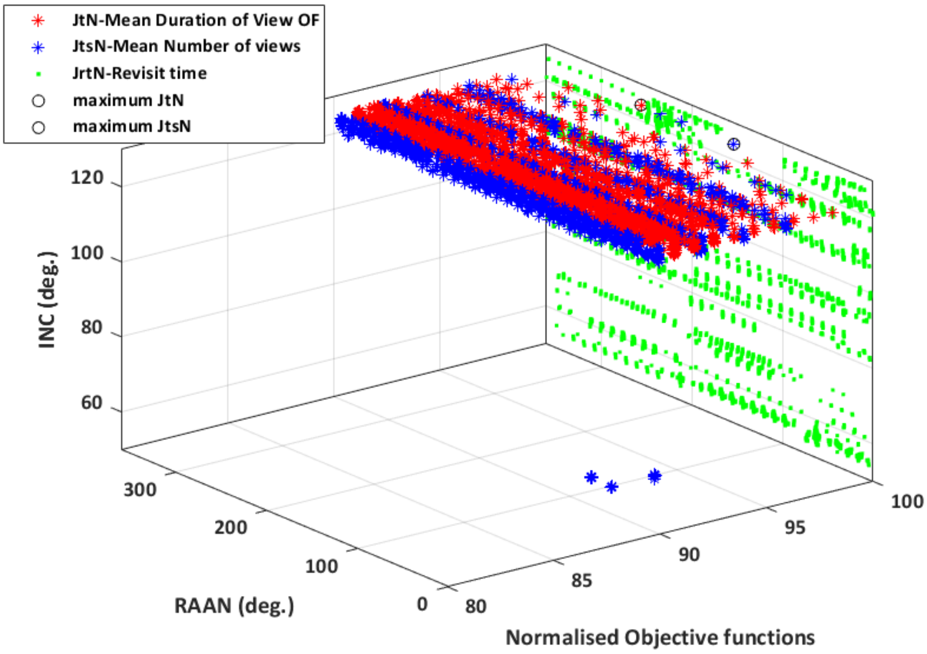


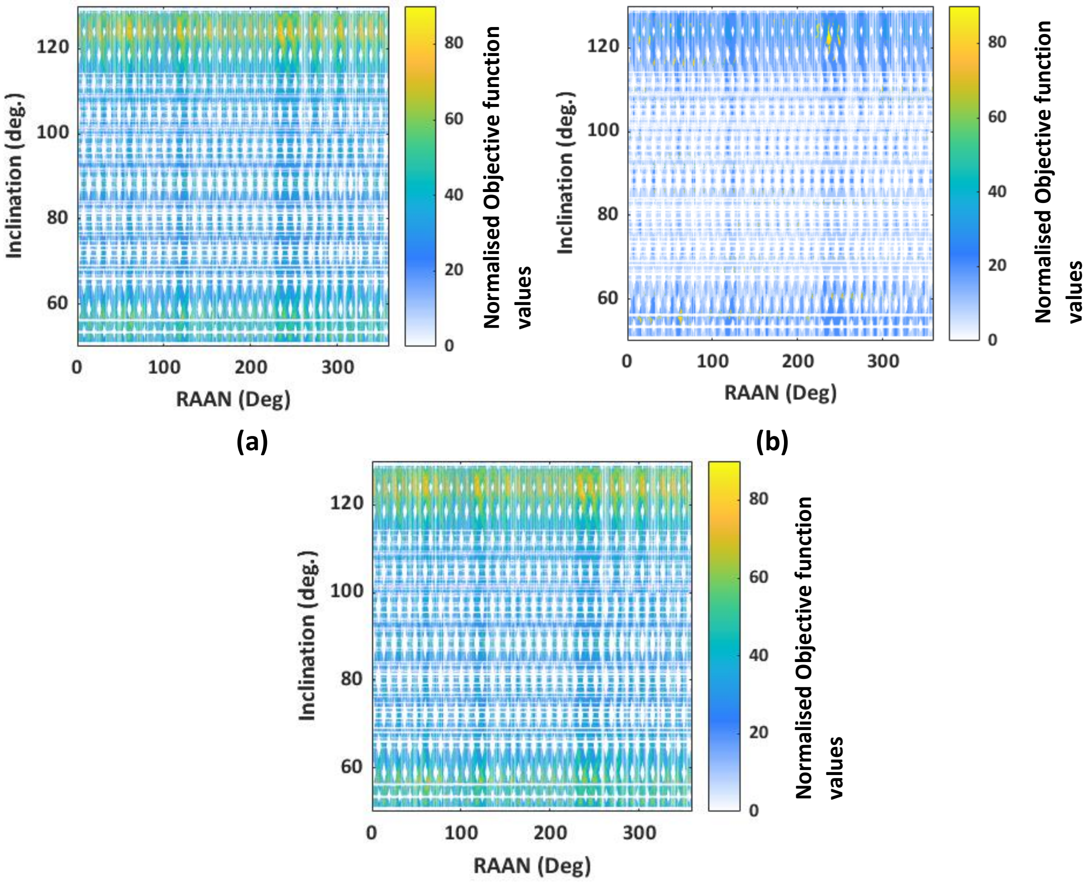
Figure 3.8: 3D plot of the maximum normalised for each individual objective function.

Using 0.1 increments from 0.1-1, the objective functions are then simulated to show the best objective function for different cases. The orbits found when changing the priority values of the objective functions are as in Table 3.9.

Table 3.9: Orbits determined for the individual optimum prioritised objective functions.

Objective function with highest priority	Inclination (Deg)	RAAN (Deg)
Mean number of times seen	123.40	235.8235
Mean Duration of view	124.00	223.2223
Revisit time	123.80	222.4222

When considering multiple objective functions, several orbits, depending on the priority values of the objective functions are found to be optimal. Normalised objective function values are used for this analysis. A heat plot of when each objective function is prioritised over the others is presented in Figure 3.9. This shows that there are no optimum objective function values between inclinations of 60 degrees and approximately 110 degrees. For the desired targets therefore, satellites on the orbits that have inclinations between 50 – 60 degrees and 110-130 degrees give more optimum orbit solutions.



(a)

(b)

(c)

Figure 3.9: Inclination and RAAN heat plot results for orbits showing the optimum orbits when objective functions have different priorities; (a) mean duration of view has higher priority, (b) revisit time (12 hours) has higher priority, and (c) mean number of times has higher priority.

The optimum orbits for the changing priority values are presented in Figure 3.10. For example, when the duration of view objective function is a priority, these orbits are indicated by the green star (*). This further verifies that the satellites on the retrograde orbits give more optimum values of the objective functions no matter which one is prioritised. For the revisit however, some prograde orbits also prove to be optimum.

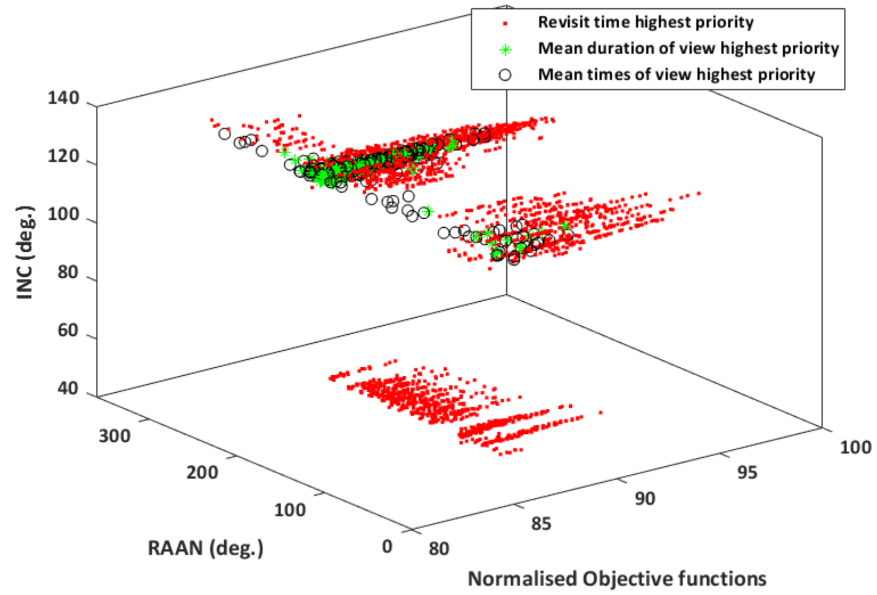


Figure 3.10: 3D plot of the Inclination, RAAN, and optimum orbits for the combined objective functions.

This analysis can be used to select the best orbit for multiple objective functions. In chapter 5 of this dissertation, a method using graph theory analysis has been developed to task optimum satellites based on different objective functions. The results of the analysis presented in this section can be used to determine the satisficing satellite orbits for the same.

3.4 Chapter Summary

The orbits determined by the developed analytical method give higher values of the studied objective functions than those obtained by previous methods. This has been highlighted in section 3.1. The developed method determined an orbit that has an optimum viewing duration for prograde orbits that is more than 30 seconds longer than the results obtained by previous numerical methods. There is a gap in literature on the analysis of retrograde orbits for objective functions such as the duration of overflight over targets. This method has facilitated this analysis due to the applicability to large search spaces. The retrograde orbits prove to have higher objective function values than prograde orbits which may be insightful to orbit designers and operators. The method is applicable to multiple objective function optimisation without the need to rerun simulations for every desired objective function when the search space is the same. The novelty in this case is contributing a method that can be used to determine orbits for multiple mission requirements and large search spaces.

Chapter 4

Addition of a modified Multi-Level Adaptive Grid

Adaptive methods have historically been used in numerical methods [82] and computational programming to efficiently get accurate or optimal solutions of certain regions of a search space. Methods such as adaptive grids have been used in Computational Fluid Dynamics to get clarity of some section(s) of a partial differential equation result as in the work of M.B. Bieterman et. al. [83]. Such methods are used to get results that have higher optimum values for dynamic methods with added efficiency. For the orbit determination method developed in this dissertation, an adaptive grid is added to reduce the time taken to get clarity and insight of the high optimum value regions of the results. The grid in this case is used to get the “optimum hotspots” of the objective functions and zooming into these regions to get ‘better’ and more optimum results faster. The initial grid used is large and this is refined into smaller sizes at the areas of interest. This chapter explains the method, the application and gives the results for each refinement stage.

The use of an adaptive grid adds value to the analytical method developed in this dissertation. This is by getting the optimum solutions related to certain objective functions by zooming into, (refining the increments of), the main areas of interest, hence, making the method have increased computational efficiency. The adaptive grid reduces the simulation time needed to identify these optimum orbits. In section 2, during the analysis of the orbits from the initial equations, it is observed that the inclination tolerance of 0.001 degrees is found to be most accurate in determining the orbit required to overfly desired targets. This is the same logic applied for the refinement of the orbits in the added adaptive grid to the developed method. Using such a refined grid in initial simulations incurs increased computational time. The adaptive method herein applies a modified type of grid referred to as the local refinement grid approach as in previous studies by scholars such as M.J. Berger et. al. [84]. For the algorithm developed in this dissertation, the grid is adapted in levels hence making it a multi-level local refinement adaptive grid method.

4.1 Literature Review of Adaptive methods:

The use of adaptive methods started decades ago with authors like M. Ciment et. al., [85] investigating the use of adaptive grids in solving boundary layer problems by formulating them as differential approximations. A. Brandt et. al., [86] extend the research by M. Ciment et. al., by studying the application of multi-level adaptive techniques to the boundary layer problem. Adaptive grid methods are used to solve elliptical, hyperbolic as in B. Kreiss et. al., [87], and many other boundary layer problems in computational fluid dynamics, (CFD), and finite element analysis, (FEA).

Just like in A. Brandt et. al., [86], the method in this work proposes a multi-level adaptive technique. As per the description by A. Brandt et. al., multi-level adaptive techniques do not work on a single grid, but with a sequence of grids hence increasing the fineness each time. A. B. Shamardan et. al., [88] modify the work of A. Brandt et. al., [86] and apply the use of multi-level adaptive methods to solve numerical Burgers' equations for initial boundary layer problems. Multi-level adaptive methods are also used in mobile trajectory clustering cases such as animal migration determination, meteorology factors such as hurricane tracking and other weather patterns amongst many others [89]. Y. Mao et. al., [89] for example explore the use of an adaptive trajectory clustering approach based on grid and density to reduce the complexity of trajectory clustering.

The refinement in this work is mainly of the search space and has a focus on orbital time, inclination, and the right ascension of the ascending node. A multi-level local adaptive grid method that refines the orbital search space i.e., inclination, RAAN and time is used to determine the optimum OF orbits.

4.2 The Multi-level Adaptive Refinement Algorithm

An added multi-level adaptive grid is developed to get more optimum results from the initially presented analytical orbit design method and reduce the time taken to determine optimum orbits. The adaptive grid is multi-level as it refines the results until a certain tolerance is reached in multiple levels. In this case, the tolerance used is in terms of time, inclination, and RAAN refinements. The refinement for the last grid is chosen to be time increments of 5 seconds, inclination increments of 0.05 degrees and RAAN increments of

0.05. Using such a refinement in the initial grid of the method would need increased computational time.

The general architecture of the multi-level adaptive grid used is as shown in Figure 4.1.

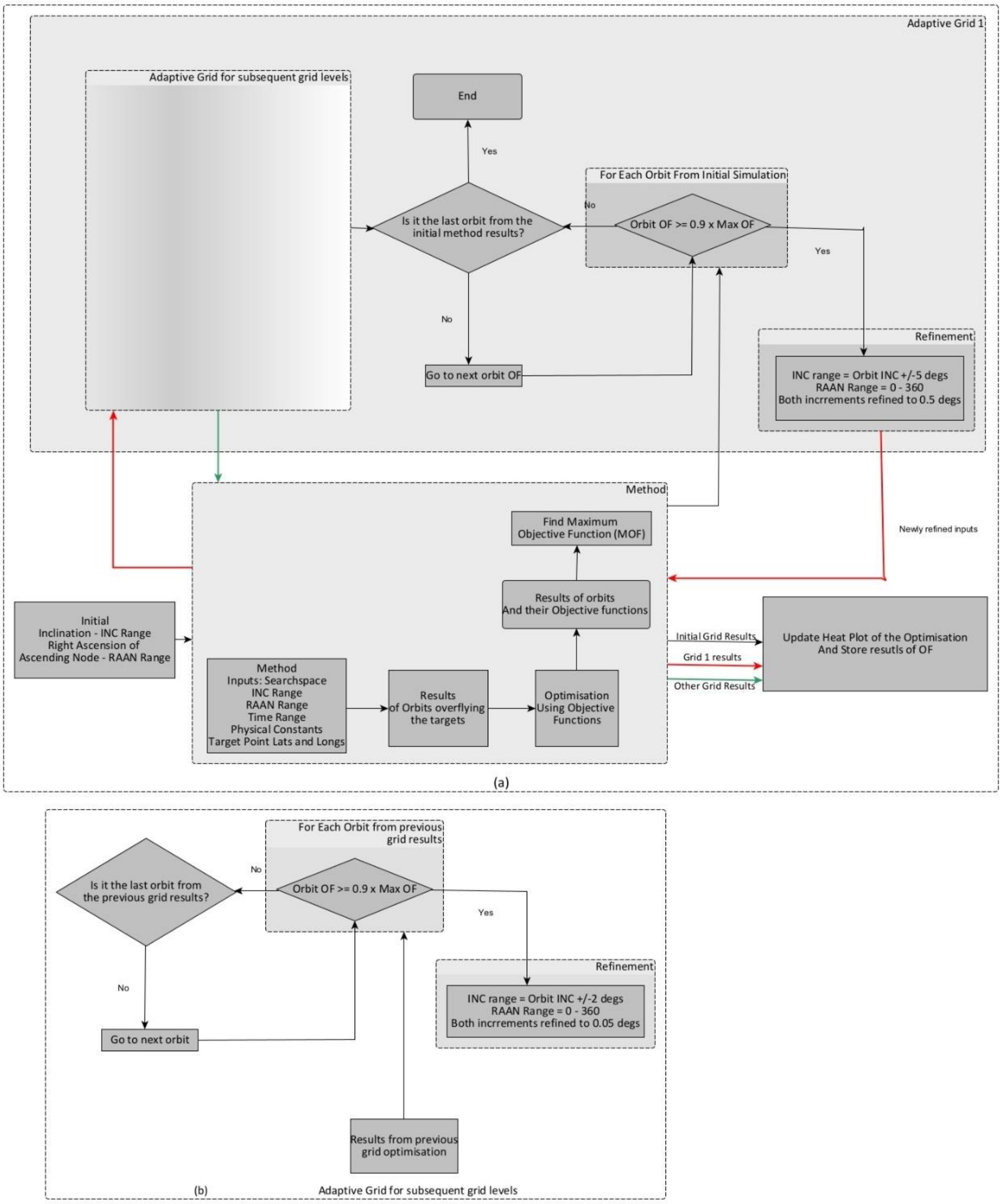

Adaptive Grid 1
Adaptive Grid for subsequent grid levels
End
Yes
For Each Orbit From Initial Simulation
No
Is it the last orbit from the initial method results?
Orbit OF >= 0.9 x Max OF
Yes
No
Refinement
Go to next orbit OF
INC range = Orbit INC +/-5 degs
RAAN Range = 0 - 360
Both incrrements refined to 0.5 degs
Method
Newly refined inputs
Find Maximum Objective Function (MOF)
Results of orbits And their Objective functions
Initial Inclination - INC Range Right Ascension of Ascending Node - RAAN Range
Initial Grid Results
Grid 1 results
Other Grid Results
Update Heat Plot of the Optimisation And Store resutls of OF
Method Inputs: Searchspace INC Range RAAN Range Time Range Physical Constants Target Point Lats and Longs
Results of Orbits overflying the targets
Optimisation Using Objective Functions
(a)
For Each Orbit from previous grid results
No
Is it the last orbit from the previous grid results?
Orbit OF >= 0.9 x Max OF
Yes
Refinement
INC range = Orbit INC +/-2 degs
RAAN Range = 0 - 360
Both incrrements refined to 0.05 degs
No
Go to next orbit
Results from previous grid optimisation
(b)
Adaptive Grid for subsequent grid levels

Figure 4.1: A chart flow architecture showing the multi-level adaptive grid method developed in this thesis. (a) shows the general grid flow architecture and (b) shows the grid level refinements, these can be as many as necessary.

The multi-level adaptive grid method algorithm uses the results from the derived analytical embarrassingly parallel orbit design method, and objective functions are then calculated to select the optimum orbits. These results are then analysed for the "peak values" and at these points, the grid is refined further in different levels as presented in Table 4.1.

Table 4.1: Multi-level (3 level) grid refinement simulation of the developed analytical method the Initial Search Space inputs to the method.

	Inclination		Right Ascension of Ascending Node (RAAN)		Time	
Grid	Range (deg.)	Increments (deg.)	Range (deg.)	Increments (deg.)	Range (sec.)	Increments (sec.)
1	50 – 90	1.0	0-360	2.0	0 – 172800	10
2	50 – 90 (for each OF peak values +/- -1 degree)	0.5	0-360	0.5	0 – 172800	10
3	50 – 90 (for each OF peak values +/- -1 degree)	0.05	0-360	0.05	0 – 172800	5

From Table 4.1, Grid 1, the increment for the inclination is 1.0 degrees, right ascension of ascending node increment is 2.0 degrees, and the increment for time is 10 seconds. With these increments the grid of the search space is very large. This being an analytical method, one of the disadvantages is lower accuracy. In such a case where the grid is this large, it is prone to even larger inaccuracies. This was observed in section 2 when different inclination tolerances are used, and different orbits are found to be viable as the FoV amongst other factors are not included in the calculations. In this section however, the FoV used is 20 degrees, and the determined orbits for all grids are based on this. Once the results are obtained from the inputs provided in Table 4.1, they may then be optimised using different objective functions. The objective function used in this case to illustrate the adaptive grid application is related to the duration of view of the targets. The algorithm can however be applied to different objective functions. The orbit results with the higher objective functions are refined and the method is again simulated around those inclinations and RAANs within the objective function *"hotspots".* The refinement used in this work is presented in Algorithm 4.1.

Algorithm 4.1: *Algorithm for the refinement of the grids (For example first grid)*

```
If $(OF)(orbit(x)) \geq 99\% \times Maximum\ Objective\ Function\ (MOF)$,
    $Inc\ range = Orbit\ Inclination \pm 1$
    $RAAN\ range = 0 - 360$
    $Increments = 0.5$
    $Input\ values\ to\ main\ method\ and\ run\ simulation\ and\ optimisation\ again$
End
```

After the results are obtained, the grid is refined, and this is done using the first refinement presented in Table 4.1, grid 2.

This refinement can be done in many levels and can be further extended to orbits within 90% of the maximum objective function for more options especially when considering multiple objective functions. In this case, however, three levels are used, and the refinement is upto inclination and RAAN increments of 0.05. This makes up a 3-level adaptive grid refinement. Compared to starting with the main method with such a refined search space however, it proved more computationally time efficient. The grid refinement can be graphically visualised in Figure 4.2. Where the higher values of the objective functions are located between 50- and 60-degrees inclinations, for example, the refinement can be seen from the grid lines to be finer around these inclinations.

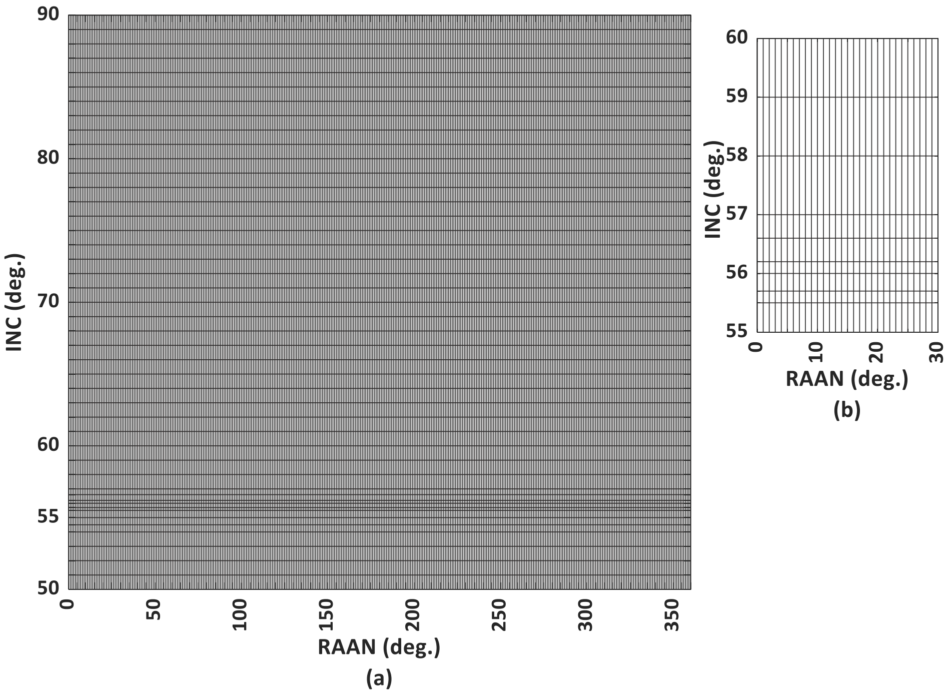


Figure 4.2: Sample grid refinement when using the optimum objective functions as presented in the adaptive grid method. (a) shows the full diagram and (b) zooms into the refined area which is between 55 deg. Inclination and 57 deg. Incination.

4.3 Results and Discussions:

The results from initial search space simulation using the inputs on Table 4.1, grid 1 are first presented. The duration of view objective function is used to optimise the orbits. From this, the results are, an optimum objective function == 89.00 at an inclination of 56.00 degrees and a RAAN of 2.00 degrees. The heat plot of this first (initial) grid results obtained is presented in Figure 4.3.

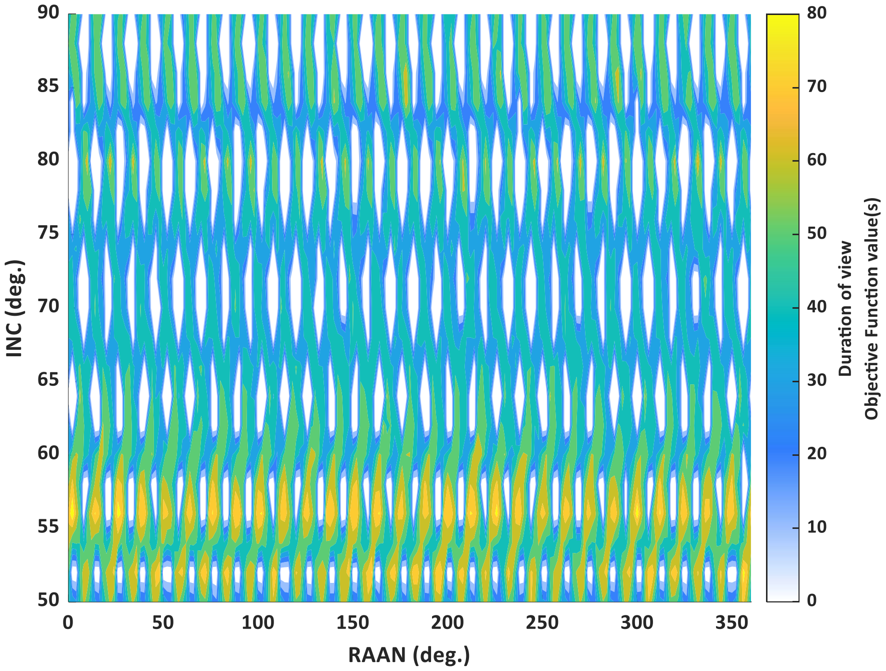


Figure 4.3: The initial calculation Grid 1 inclination and RAAN heat plot results while considering the duration of view of each orbit on each target.

The optimum OF orbit in this case is simulated using GMAT to compare to the result from the first grid. For this, the developed method obtains an objective function of 89.00 while viewing all targets whereas for the same orbit, GMAT finds an objective function of 94.34 and also viewed all targets at this orbit. This can be due to the time refinement applied as satellites in orbits that view the targets for less than 10 seconds are not considered for this grid refinement. This shows that the method while using large grids might miss some orbits that faciltate more optimum objective functions.

To obtain the results from first refinement as per Table 4.1, grid 2, the first grid refinement is done for values Inclination 55:0.5:57 which was +/-1 of the optimum inclination calculated by the initial simulation. It is worth noting that in this first refinement, the time increments are also refined to 5 seconds. This enables the method to get more possible orbit solutions as it also takes advantage of parallelising the time. On optimising the results from this for best orbit within that range, the objective function heat plot is presented in Figure 4.4.

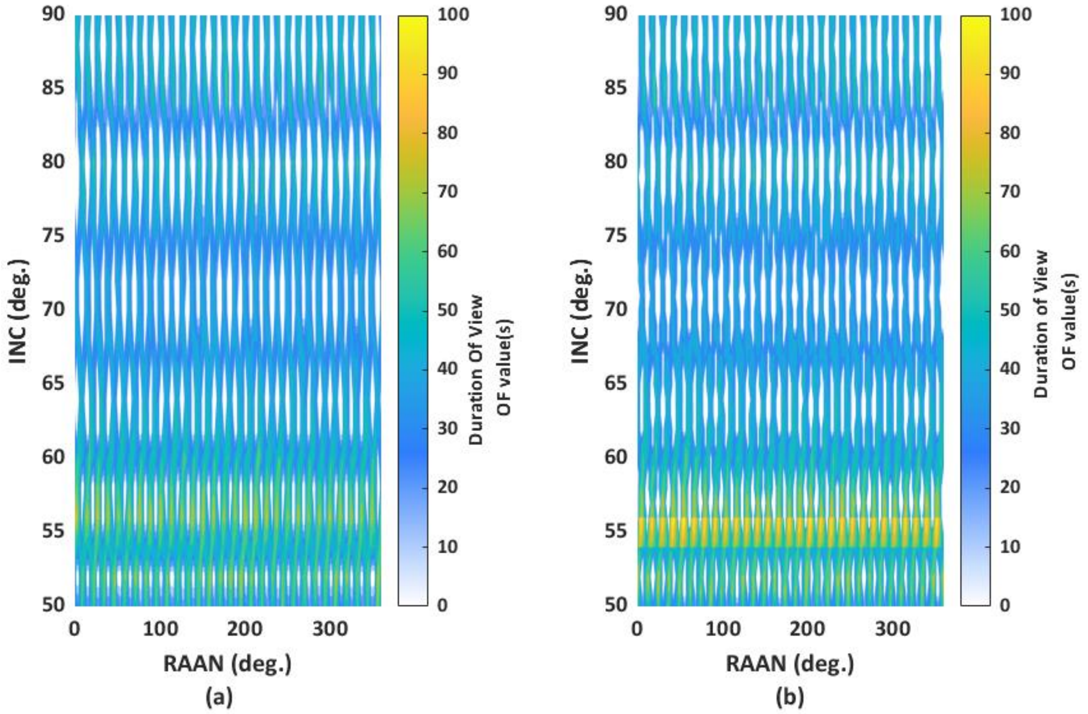


Figure 4.4: Comparison of the heat plots of the inclination and RAAN while using Adaptive grid 1 and adaptive grid 2 respectively. Plot (a) is for grid 1 (initial simulation) and graph (b) is the two combined (initial simulation and refined grid 1 – with refinement around inclination 55 degrees).

The best orbit OF for this grid size is found to be orbit 56.00 degrees inclination, 54.51 degrees RAAN, and the objective function is calculated to be 106.00. Compared to the initial grid, this result already shows an increase in the maximum objective function meaning a more optimum orbit is found. At the initial grid due to the increments of 1 degree for inclination, and RAAN, some orbits with RAANs like 54.51 are not considered, but due to the refinement, more optimum orbits are obtained.

For the second refinement done, the grid values are the ones presented in Table 4.1, grid 3. From the simulation, of values given in Table 4.1, grid 3, the objective function of the duration of view is again calculated. The heat plot of the final refinement is presented in Figure 4.5 which shows a difference between the second grid (first refinement) and the final grid (second refinement).

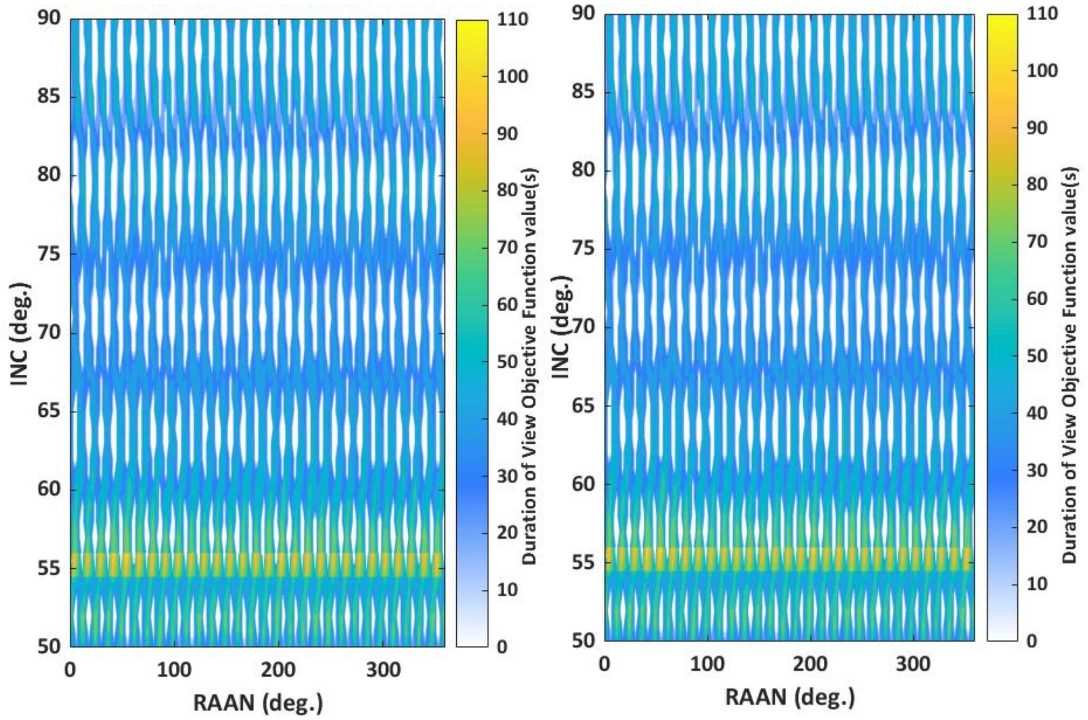


Figure 4.5: 3-level adaptive grid heat plots of the inclination and RAAN of (a) second grid, and (b) final grid.

From Figure 4.5, the maximum duration of view objective function orbit solution from the final refinement is an orbit of inclination 55.95 and RAAN 54.41. The orbit overflies all targets and with a maximum mean duration of view of 107 which is higher than the previously obtained orbits from simulation grid 1 and 2.

The general objective of finding the optimum orbits by refining the grid around the maximums is shown to be achieved. This is mainly increasing of the method efficiency in finding the higher objective function value orbits. The objective function value for the final grid orbits is higher than the other two grids. The various objective function values and the orbits obtained are presented in Table 4.2.

Table 4.2: The maximum duration of view objective function values, Inclination of this OF and RAAN from the adaptive grid multi-level refinement

Grid	Inclination (Deg.)	RAAN (Deg.)	Objective function value
1	56.00	2.00	89.00
2	56.00	54.51	106.00
3	55.95	54.41	107.00

From the presented results, the areas with the hotspots have the grid around them refined, and this gives more optimum results. Starting with a large grid then refining that to focus on those areas around the hotspots proved to reduce the computation time of obtaining the optimum values as compared to starting from a refined grid using the method developed. For further work, the developed orbit design method can be applied with other numerical methods to get more accurate results.

For the presented work, the total simulation time from the initial method grid to the final grid refinement, compared to having an initial refinement of 0.05 degrees increments for a similar search space saves on computational time. The computational time saving varies depending on different factors, such as, the specification of the computer being used, the range of objective function values being assessed (e.g. if the search spaces being refined are those that facilitate objective function values greater than 90 or greater than 99), etc. With the adaptive grid, the refinements and the search space, in this case, a third of the time was used when running the simulation with a refined grid from the beginning. This shows that using a large grid at the beginning saves on computational time. The results when using an analytical method combined with a large grid are however not as precise as for example, the maximum objective function is found to be 89.00 and the orbit determined is not the optimum for maximising the duration of view.

The method is also applied to the mean number of times in view objective function. The algorithm used is like that one used for the duration of view and all that changed is the objective function input. From this, the hotspot refinement plots are as presented on Figure 4.6.

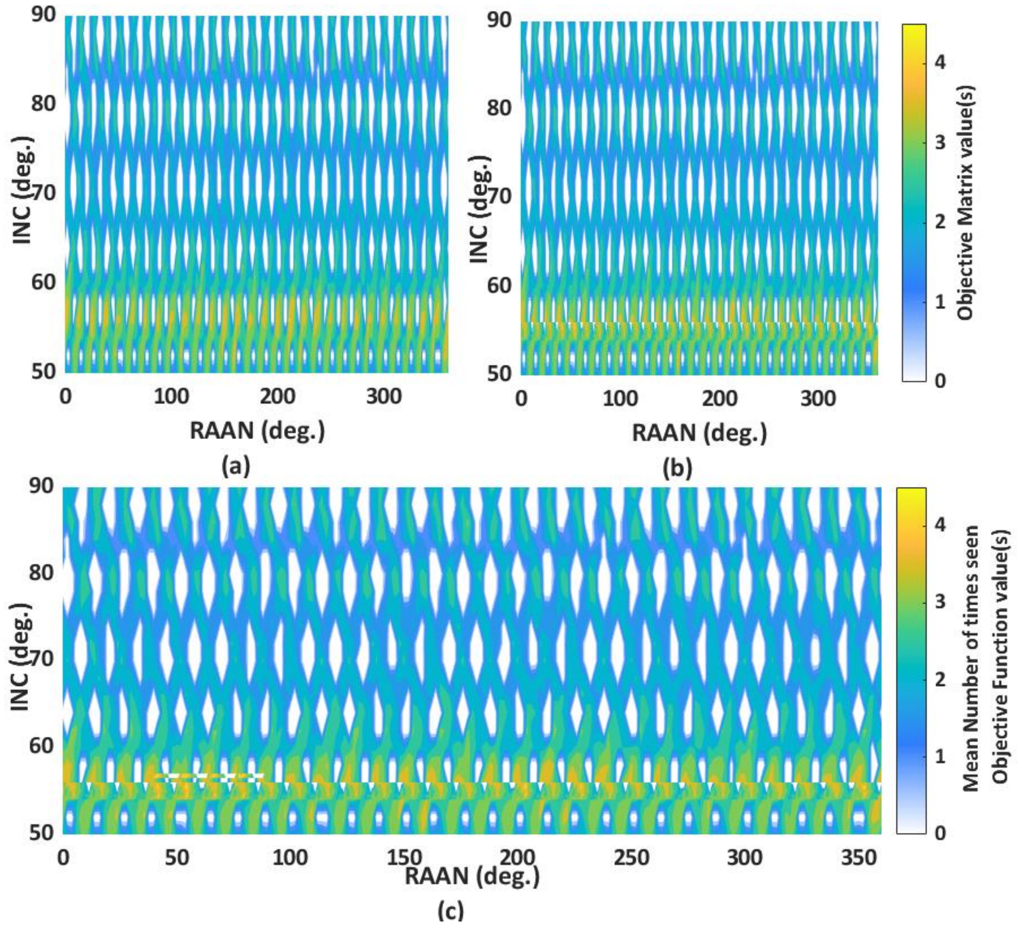


Figure 4.6: Mean number of times objective function optimisation heat plots using adaptive grid (a) level 1 (b) level 2 and (c) level 3

The values of the orbits that have the maximum mean number of times seen objective functions are presented in Table 4.3.

Table 4.3: Values of the optimum orbits found for maximum objective function of the mean number of times targets are viewed.

Grid level	Maximum OF (Mean number of times seen)	INC (Deg.)	RAAN(Deg.)
Level 1	4.20	57.00	42.00
Level 2	4.40	54.50	54.51
Level 3	4.60	56.40	53.51

The differences in the values of the maximum objective function might not seem too different for the changing grids. It is however worth noting that these are the mean number of times seen. For example, grid level 1 and grid level 2 have a difference of a mean value of 0.2 which is equivalent to 2 views. This means that the targets are viewed 2 times less by the orbit found in level 1 as compared to level 2. Optimally, the third level found an orbit that views the targets 4 times more than the first level.

4.3.1 Validation of the adaptive grid addition by comparing to previous results

The adaptive grid simulation is done using Algorithm 4.1. The refinement is 3-level, and the grids are refined according to the increments presented in Table 4.1. The targets and their priorities are as presented in Table 3.2 which are the same ones used for the validation of the developed analytical method and as was also used by W. Yao et. al., [57]. The selected orbits are those that facilitate overflight over all 10 targets as per the work by W. Yao et. al. The results from the adaptive grid obtained are as presented in Table 4.4.

Table 4.4: The maximum duration of view objective function values, Inclination of this OF and RAAN from the adaptive grid multi-level refinement

Grid	Inclination (Deg.)	RAAN (Deg.)	Objective function value
1	126.00	136.01	87.28
2	126.50	173.52	92.58
3	126.40	61.91	110.54

As per Table 4.4, a maximum value lower than the results obtained from the method without an adaptive grid is calculated using the first grid. The third grid however gives an objective function value greater than the objective function value obtained from simulating the method without an adaptive grid. This may be due to the time refinement used for the adaptive grid. To save on computational time on the method, a refinement of 10 seconds is used. This means that any times that are not factors of 10 are not analysed. For the final refinement however, a time increment of 5 seconds is used. This refines the time more hence getting results within timeframe factors of 5 seconds. The heat plots are presented in Figure 4.7 and Figure 4.8. For this simulation, using the adaptive grid was found to be 3 times faster than using the method with an initial refined grid. The computer used is a Windows 11 laptop with intel core i7, 8th Generation, and MATLAB R2023a is the software used for this simulation.

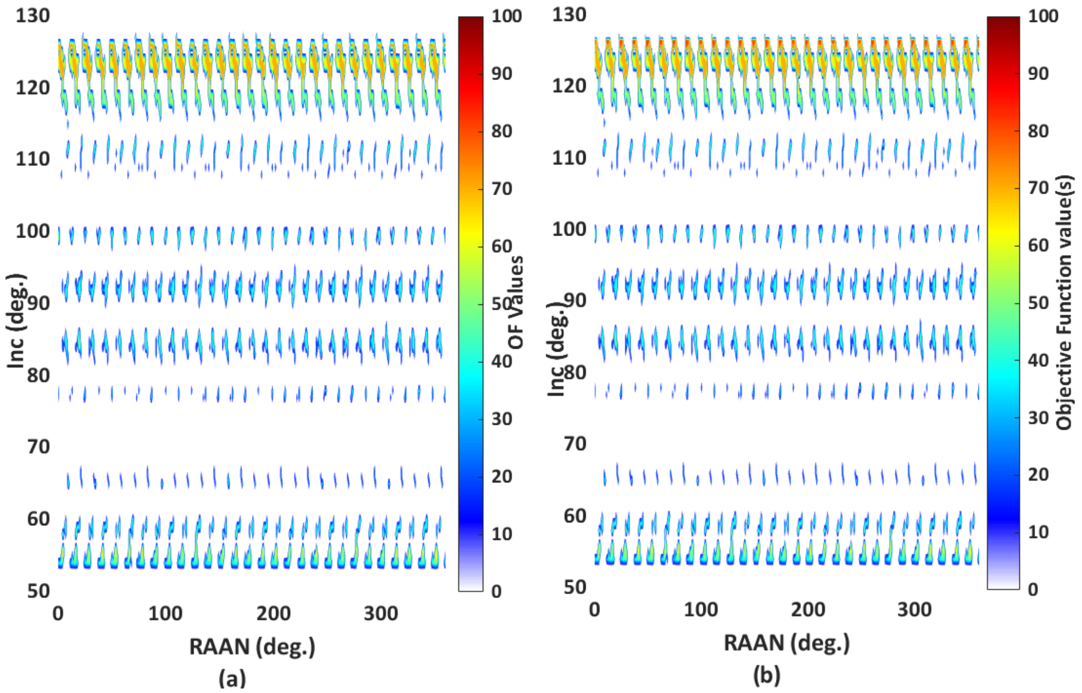


Figure 4.7: Comparison of the heat plots of the inclination and RAAN while using Adaptive grid 1 and adaptive grid 2 respectively. Plot (a) is for grid 1 (initial simulation) and graph (b) is the two combined (initial simulation and refined grid 1 – with refinement around inclination 125 degrees).

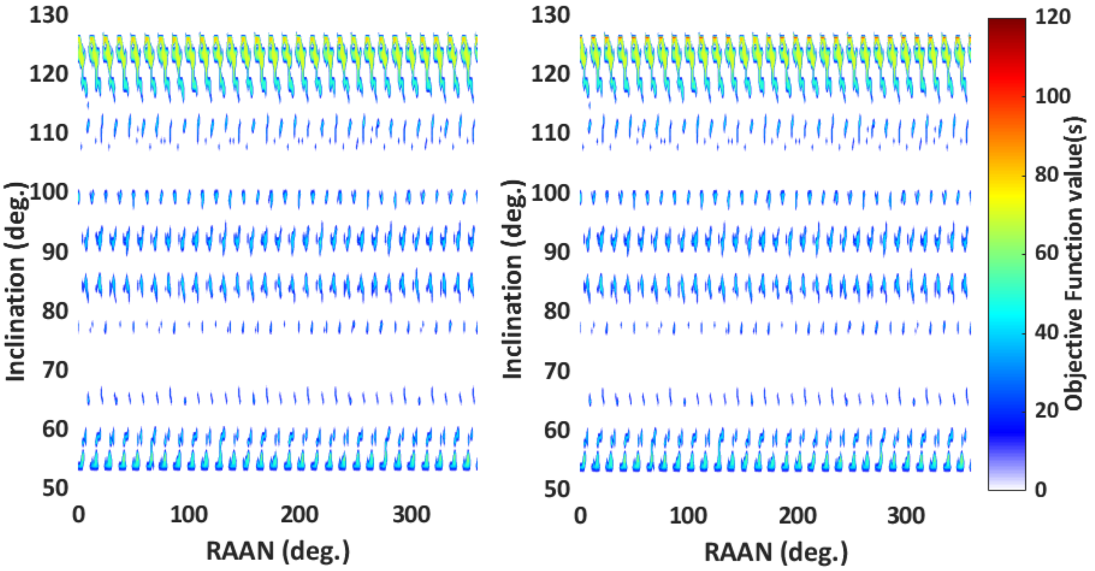


Figure 4.8: Comparison of the heat plots of the inclination and RAAN while using Adaptive grid 2 and adaptive grid 3 respectively. Plot (a) is for grid 2 (initial simulation) and graph (b) is the two combined (initial simulation and refined grid 3 – with refinement around inclination 125 degrees).

Comparing the optimum objective function orbits obtained from different methods, Table 4.5 gives a summary of the optimum objective function values from the methods, and the orbits. The Self Adaptive Differential Evolution methods are found to determine orbits with lower objective function values. Using the multi-grid adaptive method finds a higher value objective function orbit than the others.

Table 4.5: A comparison of optimum objective functions related to the duration of view calculated from orbits determined using the added adaptive grid to the developed method, developed analytical method and previously used numerical methods.

Methodology	Objective Function Value (Jt)	Inclination, deg	RAAN, deg
[57]	80.11	55.51	125.69
[53]	79.43	55.52	30.20
Developed Method	94.16	126.20	111.21
Developed method with Adaptive Grid	110.54	126.40	61.91

4.4 Chapter Summary

The advantage of adding the adaptive grid to the method is mainly increase in method efficiency to obtain optimum results. The efficiency is based on the time it takes to determine the orbits. The time reduction is based on factors such as refinement values used; for example, finer grids will take more time. This reduction also depends on the number of orbits obtained in each refinement. In this case, despite doing an analysis of both retrograde and prograde orbits, the time efficiency for the simulation increases when using the adaptive grid. When the adaptive grid is added to the developed method also, a higher value objective function orbit is determined when compared to previously used methods. This can be seen in section 4.3. This chapter also shows that the adaptive grid can be used to optimise different objective functions. Once these optimisations have been done, normalisation of each objective function can be done and the best orbits for multiple objective functions can be determined. The novelty in this section is in showing that the orbit determination method can be used in different ways and can also be used with a numerical method to increase accuracy.

Chapter 5

Satellite Tasking Using Graph Theory

In this chapter, a hybrid method is developed using combined graph theoretical approaches to query a graph database of satellites and ground targets. The graph database is queried with the aim of identifying and selecting satellites to task based on different mission requirements related to the target overflights. A graph database is a type of not only SQL (NoSQL) database designed to leverage graph theory principles to manage and query complicated relationships. It does not use the relational model of storing data in fixed tabular forms like traditional SQL models. M. Lazarska et. al. [90], show that both the querying and modification performance of graph databases are more efficient in terms of time and computational costs when compared to relational databases.

Numerical methods are frequently applied for tasking and scheduling satellites [77, 91-95]. S. Liu et. al. for example, study tasking of intelligent satellites for earth observation using a combination of a linear programming algorithm and a heuristic search algorithm [63]. The combined linear programming algorithm and heuristic search algorithm developed by S. Liu et. al. proves to have better results in terms of satellite planning profits and revenue gaps when compared to some traditional heuristic search algorithms. The method by S. Liu et. al. however has a disadvantage in terms of time consumption as they need additional computation time. Similarly, constellation and satellite orbit designs and patterns are optimised using different numerical algorithms [96]. The designs are based on different criteria such as coverage capabilities, temporal capabilities, amongst other requirements. Assigning a specific objective to a satellite is herein referred to as satellite tasking. Satellite tasking requires an analysis of all objectives and for space sustainability, minimisation of the number of satellites needed to achieve them. The satellite objectives are known as the *satellite tasks* in this chapter. The minimisation of the number of satellites requires identification of the satisficing satellites based on the tasks. “*Satisficing satellites”* refer to the satellites that give the ‘good enough’ objective performance after considering all the tasks and the trade-offs [97]. This chapter develops algorithms that combine different graph theoretical approaches to identify the satisficing satellites for multiple tasks. “*Hybrid method”* in this chapter refers to the combination of different graph analysis approaches.

In this chapter, a graph is created using the relationship between satellites in different orbits and targets desired to be over-flown after ground-track propagation within a given field of regard. The satellites and the targets are represented by nodes, and the connections between them are represented by edges, hence creating the graph. The tasks considered can either be quantifiable or non-quantifiable. In this chapter, "the quantifiable tasks" are those tasks that have a numerical value attached to them such as the amount of view time, known herein as the duration of view, and the number of overflights of a satellite over the targets. The quantifiable task values are stored as weights on the edges that connect the nodes of the graph. These task values are calculated from information that relates satellites in the various orbits, and the targets that they overfly. The information stored includes the start and end times of each view from epoch time. Non-quantifiable tasks are those that can be determined using binary analysis such as whether a satellite has overflown a target or not.

The scenarios and method developed in this chapter envisages application to tasking satellites already in space where an operator has new customer requests but could equally be applied to support system or mission design. It is a method that proves to be applicable for agile space, where the mission requirements keep changing and tasking needs to be done fast and efficiently. The applicability of the proposed method to pre-existing spacecraft populations in a case where new users emerge, or new tasks are developed gives the method an advantage. Specifically, the proposed approach is less costly for a new user than deploying new satellites, giving an additional advantage to space sustainability in terms of reducing the congestion of space. It is also a method that enhances asset value for an operator.

Before describing and developing the proposed hybrid method, a brief introduction and history of graph theory and its uses in space is first presented in this chapter. The aims, motivation and value of the study are then presented. Finally, the proposed method, its simulation, and the results as well as case study analysis are presented.

5.1 Graph Theory Literature Review:

Graph theory is an area that has been under research for centuries even dating all the way back to the seventeenth century [64]. It has been commonly used to find solutions to networking problems for cases such as traffic systems, internet connectivity, different uses in the medical field and more recently, it has been widely used by social media companies to link networks of people, amongst many other uses [98]. It is a theory that uses mathematics for analysis of graph structural properties [99]. Graphs represent various connections and interactions amongst different or similar items. There are different types of graphs, and they are classified in several ways [100]. The classifications may be based on the node characteristics, edge characteristics and the data contained on the edges as well as the structure of the graph in terms of node connections. Some of the common and relevant types of graphs are as follows,

i. Directed and undirected graphs – This classification is related to the edge direction. A directed graph is a graph where the edges have direction whereas in undirected graphs, the edges have no direction assigned to them and they are also said to be bidirectional [100-102]. The presented work mainly uses undirected graphs.

ii. Finite and infinite graphs – This depends on the number of edges and nodes. Finite graphs have a finite number of nodes and edges whereas infinite graphs have neither node nor edge limits [103]. In the presented work, finite graphs are used to develop the satellite tasking method.

iii. Weighted and unweighted graphs – This is related to the data stored on the edges. Weighted graphs have quantifiable data stored on the edges which are the weights between the connected nodes. Unweighted graph edges simply indicate the connection between two nodes [100]. Both weighted and unweighted cases are studied in the work herein presented.

iv. Bipartite graph, unipartite graphs, and multipartite graphs; these are classifications based on the node partitioning. Unipartite graphs contain no distinct node partitions, and all nodes belong to a single set. A bipartite graph contains nodes connecting two different categories of objects, for example, students to courses, airlines to airports they land at, buses to the routes they use, amongst many others. The graph nodes can be divided into two partitions. When the graph has more than two distinct node partitions it is a multipartite graph [104]. The work presented in this chapter involves a network of satellites and ground targets. This kind of graph nodes can be grouped into two whereby if say a graph G = (S, T, E), means that S are the satellite orbit nodes, T are the target nodes and E is the edges that join these nodes. In this kind of graph, no S node can be connected to another S node and likewise, no T node can be connected to another T node. The connections are therefore only between S nodes and T nodes, hence a bipartite graph.

The network structure and connection patterns may be used to determine the analysis method that is applicable. The choice of the metrics to be used for querying graphs generally depend on different factors such as the research questions, the insights needed, and the structure of the graph [105, 106]. Understanding the differences between complex, large, small, and simple networks is essential for effective choice of metrics, analysis and modelling. The differences are characterized by the type and arrangements of connections or the quantity of the connections.

Simple networks have regular structures where the nodes and edges are arranged in a predictable manner. They are deemed easier to mathematically analyse [107]. Complex networks are characterized by structures that deviate significantly from regular patterns. They have complex structures that make them more difficult to analyse [107, 108]. Large networks are characterized by a significant number of nodes and edges, making them computationally challenging to analyse and may require specialized algorithms and computational resources [109]. A small network on the other hand has relatively few nodes and edges.

These categories are however not mutually exclusive in that a network can be both large and complex, small and complex, small and simple, or large and simple. The boundary between complex and simple networks is also at times blurred as real-world networks may have the characteristics of both. As noted by various authors in the field of network science, see [110, 111], real-world networks often exhibit a hybrid nature, blending characteristics of both complex and simple network models. By recognizing the specific characteristics of different network types, researchers can select appropriate analytical tools and models to gain insights into the underlying structures and dynamics of different systems.

The study of different types of sets in graph theory has been on the rise in the more recent past [98]. These sets are used for finding solutions to problems such as matching, combinatorics, independence, and covering amongst many others [98]. As defined by Zixuan Yang et. al. [112], matching, in graph theory, refers to when there is a set of edges such that no two edges share a node. The problem of matching has been studied for decades dating back to the assignment problem, the marriage problem [113-115] and transportation problems. It has also been used in the medical field by scholars like S. Gentry et. al., [116], who study the application of maximum matching to allocate kidney donations. The Kőnig-Egerváry theorem has also been extensively used and modified to solve maximum matching problems [115]. Maximum weighted matching (MWM) is one such problem. M. Barketau et. al., [117] study bipartite graphs and focus on the minimisation of the maximum weight of subsets of a maximum matching in such graphs. M. Barketau et. al., prove that this problem is strongly NP hard, and they develop a method that find solutions that are on average of 0.5% of the optimum. Matching is applied in diverse fields including scheduling and resource allocation amongst others. In the work presented in this chapter, there is a matching of satellite nodes that have the maximum weights and jointly satisfy certain tasks.

Depending on the tasks being analysed, satellite tasking may involve finding solutions to combinatorial problems which have weighted edges connecting the nodes. An area of network science known as spectral graph theory is used to relate the adjacency matrix with the graph structure properties and it is used for graph partitioning as well as clustering as explained by Refs, [118],[119],[120]. An approach of evaluating the adjacency matrix of a network is proposed for the hybrid method. The adjacency matrix of a graph gives insight on whether nodes are adjacent or not [121] and it can be weighted or unweighted [106]. Unweighted adjacency matrices may be used to analyse the structure of a network by revealing which nodes are connected and help identify disconnected components in the network. They can be used to calculate centrality measures like degree centrality, which count the number of connections a node has [103, 106], [122, 123]. In a weighted adjacency matrix, each element represents the weight associated with the edge between two nodes. This weight can represent distance, time, cost, strength of connection, or any other relevant measure. In general, unweighted adjacency matrices give insights on the connections of a graph and hence the structure whereas weighted adjacency matrices give insights on the strength of the connections related to the weights [103, 106, 124-126] .

To study centralities, the adjacency matrix, (A), is first derived from the graph. The matrix may either be the weighted adjacency matrix or unweighted adjacency matrix, and the centralities are as described by Refs. [127-129] as:

i. Degree centrality – is the analysis of the graph in terms of the neighbours of the nodes. The formulation of this is,

$$x_i = \sum_j a_{ij} \tag{5.1}$$

Where x_i is the degree of node i and a_{ij} is the connection between node i and j. In general, degree centrality determines the importance of each node by the total of its neighbour's degree i.e.,

$$Degree\ centrality\ of\ vertex = \frac{degree\ of\ vertex}{sum\ of\ degree\ of\ all\ vertices} \qquad (5.2)$$

ii. Closeness centrality – is the analysis of node importance in terms of the closeness to all other nodes in the network. It finds the shortest path to get to all nodes in the network.

iii. Betweenness centrality – is the analysis of the fraction of the shortest paths that include the studied node. It checks more on the influence of a node based on movement between other nodes.

iv. Eigenvector centrality – is the analysis of node importance considering its neighbours. It gives an analysis of node degree and counts the number of links between the connected nodes. It uses the adjacency matrix decomposition and calculates the largest eigenvector for each node. From basic principles,

$$Ax = \lambda x \qquad (5.3)$$

i.e., a matrix, A, multiplied by the eigenvector, x, is equal to the eigenvalues, λ, multiplied by the eigenvector. The eigenvector centrality can be weighted or unweighted. The weighted eigenvectors consider the weight values and may be given as,

$$x_i = \left(\frac{1}{\lambda}\right)\sum_{j=1}^{N}(A_{ij}x_j)$$

$$\text{with } A_{ij} = \begin{cases} 1 - \text{if } i,j \text{ are connected} \\ 0 - \text{otherwise} \end{cases}$$

$$\text{or}$$

$$\text{with } a_{ij} = \begin{cases} \text{weight value if } i,j \text{ are connected} \\ 0 - \text{otherwise} \end{cases}$$

$$\text{for } a_{ij} \geq 0, \forall i,j \qquad (5.4)$$

where, x_i is the eigenvector centrality of a node i, λ is and eigenvalue of the adjacency matrix, A, and x_j is the eigenvector centrality of the neighbours of node i.

v. PageRank centrality gives an analysis of link direction and is however not analysed herein as the graph used in this chapter is an undirected graph.

The satellite-tasking hybrid method proposed in this chapter is developed using an undirected bipartite graph analysis hence the use of degree centrality and eigenvector centrality is proposed. The method however has an added advantage as it is also applicable to directed bipartite graphs as presented in section 5.4.4.

Network robustness is inversely proportional to its vulnerability to node removal see, White et. al. [130]. In their work, White et. al., state that network robustness can be quantified by calculating the numbers of node-independent paths between nodes which they suggest is equal to the connectivity. White et. al. however find that the calculation for the node-independent paths in large networks is computationally difficult and hence propose an approximate algorithm to get the lower bound on the number of node independent paths for large graphs.

Moody et. al. [131] develop a concept of structural cohesion based on network node connectivity and define structural cohesion as the minimum number of actor nodes needed to be removed from a group to disconnect the group [131-133]. Subgroups that are structurally cohesive have been used to characterise the robustness of networks as presented by R. S. Sinkovits, [134]. Cohesive subgroups are characterized by strong inter-vertex connections, even among distant nodes with no shared neighbours. However, identifying these subgroups is computationally demanding [134]. Sinkovits et. al. study a graph reduction technique based on cliques to identify cohesive subgroups. According to Cornwell et. al., in [135], though significant progress has been made in analysing one-mode networks, the development of robust methods to refine the detection of structural cohesion within two-mode networks remains a challenging problem. Cornwell et. al. propose a strategy that identifies the number of actors from one node set that may be removed before disconnecting actors in the other set but their method needs improvement on computational time.

Analysing the structural cohesion is important in this chapter as it informs on the impacts of satellite node removal or addition to target coverage. Minimisation of the number of satellites needed for maximum coverage of targets is one of the objectives of the work in this chapter. If removal or loss of a satellite node from the network results in a disconnection of a target node from the network, the satellite node needed to maintain that connection needs to be identified and included in tasking. With similarities to previous studies, such as Ref. [135], the structural cohesion in this work may be said to be the identification of the minimum number of satellite nodes needed to be removed from a network to disconnect a target node from the network.

Technically, betweenness centrality is an ideal metric for determining a networks' structural cohesion as it assesses the impact of adding or removing a node by giving insights into the structure and dynamics, such as identifying bottlenecks or critical points of failure. Removing a bridging node disrupts the shortest paths between many pairs of nodes, forcing information or flow to take longer or more circuitous routes. This significantly increases the number of shortest paths that pass through other nodes in the network, thereby boosting their betweenness centrality. Betweenness centrality relies on the shortest path but it is not normally defined for bipartite graphs [136].

The analysis of bipartite graphs can be done in different ways, and some previous methods suggest simplifying bipartite graphs and analysing them as unipartite ones [109, 135]. M. Latapy et. al. in [109] however find that analysing a bipartite graph, (which they referred to as a two-mode network due to the two sets of nodes), as a unipartite graph, which they referred to as a one-mode network, may not be applicable to some studies such as those which rely on identification of neighbouring nodes. In the work presented in this chapter, information such as the connections (degrees and neighbours) of both the target and the satellite nodes is needed for insightfully selecting the satellites to task and the graph must therefore be analysed as bipartite. A brute force algorithm is therefore the proposed approach to identify satellite nodes that are needed to maintain maximum target overflights. These satellite nodes are said to be high cohesive nodes as they strengthen the connections between nodes and act as a bridge to target nodes that have no shared satellite nodes. If for example a target is overflown by only 1 of the satellites in the network, say satellite B, for full coverage, satellite node B needs to be selected for tasking. Satellite node B is therefore said to be a high cohesive node.

In graph theory, a set is a collection of objects whereas a subset is a portion of a set [137]. The subsets used in graph theory can be classified in various ways based on the elements. A proper subset, is a set which has all its elements contained in another set which is known as a superset [121]. For this work, subsets and supersets are applicable in a case where say a satellite node A is in contact with target nodes {T1,T2,T4,T5} and satellite node B is in contact with target nodes {T1,T2,T3,T4,T5,T6,T7}, then satellite node B target connections are a superset of satellite node A target connections i.e., $A \subset B$. Likewise, satellite node A target connections are a proper subset of satellite node B target connections. For such a case, when considering maximum target coverage, satellite node B is preferred to be tasked. The proposed brute force algorithm of the hybrid method is developed to analyse the neighbours of each satellite node and determine the sets. The satellite nodes connected to similar sets of target nodes are then identified with an aim of having all the target nodes connected to at least one satellite node hence maximum coverage.

G. Lohmann et. al. [138] find that the use of eigenvector centrality, is a computationally time efficient tool for their analysis of the connectivity patterns in functional magnetic resonance data for the human brain. From these findings by G. Lohmann et. al., the approach to tasking satellites in this work proposes the use of the eigenvector centrality to analyse the weighted satellite-target graphs.

Satellite tasking can be said to be a dynamic problem especially when considering satellites that are already in orbit, as the desired tasks may change and have different properties. A requirement may, for example, be to task satellites that overfly a certain target twice in 12 hours. Such a task requires development of unweighted graph querying methods. The method proposed in this chapter proves to be applicable to multiple tasks relating to both weighted and unweighted graphs.

5.2 Graph Theory in Space:

With the complexity of modelling current space networks, whether satellite to satellite (Inter-Satellite Links (ISLs)), satellite to target, or the general space system architectures, graph theory has proven to be a useful way of giving great insights into this [139-141]. The use of graph analysis has been efficient and on the rise with the increase in number of satellites and constellations being launched annually. This efficiency is seen especially now with more opportunities to develop federated satellites and constellations, inter-linking satellites, and trying as much as possible to reduce the costs of operation while reducing the data latency and maximising data transfer [142].

C. J. Lowe et. al. , [142], present a method for data routing using satellite networks and develop an efficient routing of data by exploiting information on the future contact schedules between network nodes which proves to be beneficial in terms of network performance. C.

J. Lowe et. al., develop this in [143], by introducing a contact graph scheduling method aimed at minimizing latency between data arrival, delivery and pickup. C. J. Lowe et. al., improve on the previous work by combining both scheduling and routing solutions which increase fulfilment of requests and delivery latency is minimised.

Other graph theory applications include minimum cost or maximum flow optimisation, which have also been applied to identify effective ground stations [144]. In [145], M. B. Larsen et. al. use graph theoretical quantities to describe a satellite formation that is tethered. M. B. Larsen et. al., use graph theory and Lagrange mechanics to allow a broad class of formations and validate their method for general tethered satellite formations. For other space applications, H. Fan et. al., [146] use graph theory to study how to improve the timelines on task scheduling of multi-satellite, multi-task scenarios. H. Fan et. al., propose a multi-layer network aggregation method for tasks that have similar and relevant properties.

Determination of satellites that give optimum results to various tasks related to overflying specific targets especially for agile space systems, can be classified as a complex network problem where a network between the satellites and ground targets needs to be developed and queried. C. N. McGrath et. al., [147] present a graph approach for analysis of responsive satellites that are manoeuvrable. The work by C. N. McGrath et. al., show that better insights can be drawn from their analysis using graph theory as compared to previously used techniques. The work presented in this chapter shares similar advantages especially in the robust determination of satisficing satellites for various tasks.

The use of centrality and eigenvalues to analyse graphs has been applied to many real-world sectors. An example of this is the study of eigenvector centrality to detect a community of influential disease spreaders which is presented by R. A. Clark et. al., [148]. R. A. Clark et. al., introduce a non-linear relationship between time and probability of disease spreading into network dynamics and alter the Laplacian matrix to highlight the reason why nodes with higher degrees are the most influential spreaders. R. A. Clark et. al., conclude that eigenvector-based selections optimise the initial rate of infection, or the average rate of infection or produced the fastest time to full infection. R. A. Clark et. al. [139], develop this further to find the relative influence of ground stations in terms of receiving data using eigenvectors of the adjacency matrix. By first modelling the space systems as a flow network, R. A. Clark et. al., [139], use the dominant eigenvectors of the adjacency matrix of the network to identify influential communities of ground stations so that users can make informed decisions on the ground station selection. In a similar way, this work aims to identify groups of satellites that can be used to achieve both single and multiple tasks optimally/sub-optimally.

Satellite tasking and scheduling has been done using different heuristic algorithms and other algorithms aimed at selecting the most urgent tasks then scheduling them. X. Zhao et. al. [149] for example, find a need to address limitations of efficiently generating run-time task configuration for multi-mission satellites that are autonomous. X. Zhao et. al. develop a novel ontology-based method for task configurations focusing on satellite clusters. X. Zhao et. al. develop a mechanism that creates a solution space tree and finds the optimal solution of task configuration by having constraints from ontology then solving the constrained optimization problem. In the same way, this work addresses tasking optimum satellites for different tasks while minimising the number of satellites needed. Z. Shen., et. al. [150], identify a gap related to the optimization techniques for tasking and the work herein aims to address this. Z. Wang et. al [151], investigate the use of graph neural networks for multi-satellite scheduling which gives better quality solutions, and the results are generated quickly compared to meta-heuristic algorithms. The proposed tasking method aims to share these advantages.

5.3 Motivation and Introduction

The main aim of the work presented in this chapter is increasing the value of assets already in space. This increase of value is achieved by tasking existing space population with different tasks as they arise, or as new users emerge. It is aimed at lowering the costs of obtaining needed data when a new need arises or when the data is needed by new space users. It is also aimed at increasing the profit of the assets already in space for the owners. With these aims achieved, the rise in space congestion may also be reduced despite the emergence of new users hence promoting space sustainability.

The work presented in this chapter develops a hybrid method that uses different graph theory metrics and approaches to identify, select and task satellites. The method aims to analyse the graph and identify satellites that have optimum or near optimal performance of single and multiple tasks. The analysis of the nodes of graphs give a full scale view of solutions and hence better insights especially on optimality of nodes based on the weight values[147]. This work takes advantage of this characteristic of graphs.

In the context of complex networks requiring multi-objective optimization, a comprehensive analysis necessitates the consideration of both edge weights associated with the nodes and the topological relationships defined by node-to-node connections. Multi-objective satellite tasking is a highly combinatorial problem that requires different analysis approach of the graph depending on the task being analysed. The hybrid method proposed herein demonstrates that employing graph theory for satellite tasking by node identification and selection offers valuable insights. These insights include identifying nodes crucial to the graph's structure and determining optimal nodes based on the aggregate edge weights associated with each node. Leveraging these insights enables the selection of satisficing satellites that effectively optimize all defined objectives, surpassing the limitations of solely selecting satellites that optimize a single task.

Using the satellite-target graph approach, this chapter examines three key tasks, which necessitate distinct graph theoretical approaches for analysis, as detailed below,

i. The target coverage task – The objective for this is to have maximum or full target coverage while minimising the number of satellites tasked. It equates to an analysis of the structural cohesion of the graph. A brute force algorithm considering the neighbours of nodes is developed to identify the satellite node(s) which if removed will disconnect a target node from the network.

ii. The duration of view task – this focuses on identifying satellite nodes that maximise observation time over the targets. This requires a graph theory approach for calculating and identifying optimal satellite nodes based on edge weights. For instance, if satellite node M observes target Q twice, with viewing durations of 20 and 30 seconds respectively, the edge connecting M to Q is assigned a weight of 50 seconds, representing the cumulative observation time. Within this chapter, this value is referred to as the 'node edge weight.' If satellite node M has no other target observations apart from the stated two of Q, its *total edge weight* with respect to duration of view is 50 seconds.

iii. The number of times viewed task – this is also represented as a weight on the edge connecting two nodes together. In the case of satellite node M in point (ii) above, the total edge weight of node M in relation to the number of times seen task is 2.

This chapter is further divided as follows; the development of the proposed method by generating a simple graph and using distinct graph theory approaches to analyse two different tasks is presented in section 5.4 (5.4.1 – 5.4.3). The justification of the use of the two approaches to gain insights on the two tasks is then presented in section 5.4.4 and a larger network is then analysed in section 5.4.5. Section 5.5 then gives an analysis of a complex network and applies the two-graph theory approaches previously developed, proving the reliability of the approaches in complex network analysis while analysing an additional metric. Section 5.6 shows the combination of the studied graph theory approaches and hence introduces the developed hybrid method, which is given the name, the Multi-Tasking Proposed Method (MTPM). Section 5.6.3 gives further application of the MTPM and shows how the MTPM algorithms can be easily edited to evaluate a new task. A task related to the time from epoch is introduced and the analysis is presented in section 5.6.4.

5.4 Development of The Satellite Tasking Method

To develop the proposed satellite tasking method, graphs are generated of satellites to be tasked and the targets that they are desired to overfly. The graphs are generated using satellites in orbits that have had their ground-tracks propagated within a given field of regard (FoR), which can be done using the method presented in chapter 2 of this dissertation. The field of regard is the area that can be seen by the satellite assuming its pointing nadir (downwards). The graphs generated are made up of satellite nodes and target nodes with the edges between them indicating an overflight. Based on different tasks related to the overflight of the targets, graph theory methods are proposed for identification of the satellites that optimise the performance of the tasks. *Optimisation of the task performance* refers to maximising the values for the quantifiable tasks (weighted graph analysis such as duration of view) and achieving the task for the non-quantifiable tasks which rely on logical analysis such as coverage. Information obtained from the ground track propagation while considering the field of regard that relates the satellites, and the target overflights is stored on the edges. The graphs are analysed to identify the satellites that give satisficing performance of the desired task(s).

Satellite-target networks can be simple, small, large, complex, or a mixture of any. In this chapter, a *"simple network"* is used to refer to a network that is small, simple and can be analysed using single metrics and even visual assessments can suffice. A *"complex network"* on the other hand refers to a larger network which needs more detailed analysis to give insights on the network, and it is difficult to visually assess.

Different approaches and metrics may be used to give insights into how the satellites perform the tasks. Whether the tasks are quantifiable i.e. use weighted graphs or not may influence the metrics that can be applied to query the graph.

5.4.1 Graph Architecture

Once the ground-track of a satellite in orbit is propagated within a given field of regard, data such as the start time and the end times of each view of the desired targets is obtained and stored on the edges of the graph. The tasks to be analysed may include the duration of view per visit, the number of times each target is viewed within a given span of time, and the revisit time for each target by the satellite. Values related to the tasks may be calculated from the start and end times then attached to the edges as weights without the need of redoing the propagation which gives this method an added advantage. These weights can then be used for the weighted adjacency matrix for graph analysis as is presented herein.

To show an example of the proposed network, and how it is implemented in a real-world scenario, an illustrative graph is generated. A single satellite in orbit is propagated within a given field of regard and the desired overflight targets are 4. After propagation, the general graph architecture is modelled by generating the graph, Figure 5.1. The satellite in orbit is found to overfly only three of the four targets and this is seen in Figure 5.1, as one of the targets has no edges to the satellite. The view start time and end time over each target by the satellite is stored on the edges as is observed in Figure 5.1.

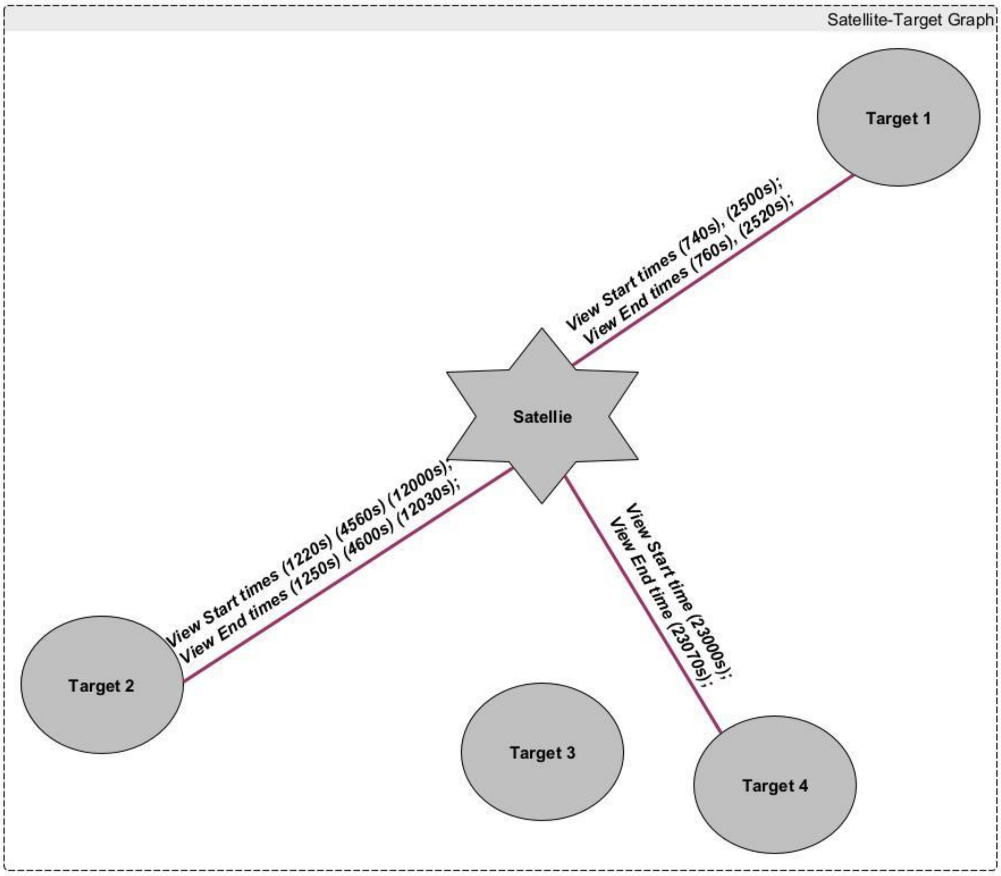

Satellite-Target Graph
Target 1
View Start times (740s), (2500s);
View End times (760s), (2520s);
Satellie
View Start times (1220s) (4560s) (12000s);
View End times (1250s) (4600s) (12030s);
View Start time (23000s);
View End time (23070s);
Target 2
Target 3
Target 4

Figure 5.1: Satellite-Target Graph for a single satellite and four targets with edges containing information related to the overflights

For simple networks such as the one presented in Figure 5.1, visual assessments may be sufficient to give insights but when the networks become larger or more complex, further mathematical analysis is needed to provide the same insights. Some of the mathematical methods are studied herein, and combined to give insights that will aid in making the decisions on which satellites to task.

Apart from using visual assessments, it is possible to query and analyse simple networks using individual metrics such as degree centrality and still get insights on the structure and node edge weight optimality. For large or complex networks however, which this work proposes a use in, more than one metric may be needed to gain both the structural and node edge weight insights.

Understanding the structure of the network i.e., the structural cohesion [152], is vital in network studies as the loss or addition of a node may bring significant changes to the network. Satellites have an operating lifespan, hence, the impact of losing a satellite from the network is insightful to both an operator and a user because it encourages more informed satellite tasking decisions. In terms of the edge weights, the total weights of the edges connected to a node, herein referred to as *node weight*, reflects how a node performs quantifiable tasks, and this is also needed for the tasking decision.

The first proposed approach for the hybrid method involves the development of an algorithm that gives an analysis of the structural cohesion of the graph. This algorithm is developed and studied in section 5.4.2.

5.4.2 Satellite Tasking, Task 1: Maximum Target Coverage Using the Minimum Number of Satellites

In graph theory, a vertex cover refers to a case where for a subset S, of nodes, all the edges in the graph have at least one end point in S [153]. The minimum vertex cover is the smallest vertex cover [112, 154]. Methods that would be applicable to determining the minimum number of satellite nodes that are in contact with all the target nodes may include finding the minimum vertex cover, and more generally, an analysis of the subsets and supersets within the graph. The latter can be developed into an analysis of the *'neighbours'* of the satellite nodes which form the first algorithm developed herein and referred to as the "*neighbours' algorithm*". Neighbours in this case refer to the direct connections of a node. Because this study involves a bipartite graph, the neighbours of each satellite node will be the target nodes that are in contact with it and the neighbours of a target node are the satellite nodes in contact with it.

The proposed neighbours' approach is a brute force algorithm used to identify the satellite nodes that when removed reduce the target coverage. The algorithm is developed to determine and identify the minimum number of satellite nodes that satisfy the requirement of full coverage or maximum coverage, and this is presented as Algorithm 5.1. Combinations of satellite nodes are also analysed, and, in this case, they are referred to as *unions*. The unions of satellite nodes are analysed to determine which combinations meet the requirement of being in contact with all the target nodes at least once. Being in contact with all target nodes in this work is referred to as *full coverage*. The sequence of the developed algorithm is,

i. Find and store the neighbours of each satellite node,

ii. from this, the satellite nodes that are individually in contact with all the target nodes are identified and stored separately (N.B when there is a single satellite node in contact with all the target nodes, the minimum number of satellites needed to fulfil the full coverage requirement is one),

iii. the satellite nodes that are individually in contact with all the target nodes are eliminated and an analysis of the remaining satellite nodes is done by making unions of each of the remaining nodes one to another,

iv. each union of two satellite nodes that are in contact with all the target nodes is selected and stored. If no single or two satellites are in contact with all target nodes, the algorithm makes a union of the satellite nodes with other individual satellite nodes, and this is repeated until the minimum number of satellite nodes that are in contact with all, or the maximum number of target nodes is achieved.

Algorithm 5.1: Algorithm to find minimum satellite nodes and satellite node combinations for contact with all the target nodes

```
Input: Graph G(S (satellite),T (target), E (edge))
Output: Single Nodes and node pairs belonging to S with E ==10
Create function: creates a call function for the neighbours algorithm
Initialization of variables: set i, full_set_nodes and pair_nodes to zero
for counter_n <= numel(S) do //loop through all satellite nodes and get the neighbours
    Neighbours1 – store all target sets for all orbit nodes
    if numel(T)==numel(Neighbours1) do// finds satellite nodes in contact with all (T)
        Satellite nodes – gives all satellite nodes that are in contact with all targets
    End
    for 1<counter_n2<= numel(S) do//loop through all satellite nodes>1
        Neighbours2 – store all target node sets for all alternate satellite nodes
        Union – computes the unions between Neighbours1 and Neighbours2
        if numel (T) == numel (Union) do//finds the orbit pairs in contact with all (T)
            i = i+1
            N_pairs((i),:)=[Satellite_nodeID,Satellite_nodeID2];
        End
    End
End
single satellite = [Satellite nodes]
```

satellite pairs = [N_pairs]

Application of Algorithm 5.1 on a satellite-target graph provides insights on the minimum number of satellites needed for maximum or full coverage. The results from the algorithm can also be used to give insights to the satellite nodes that if removed from the network will result in less target coverage.

The simple networks used to develop the hybrid satellite tasking method extend the architecture presented in Figure 5.1 from a single satellite to a multi-satellite graph. For the simple networks, a relationship between three satellites and four targets is used to generate the graphs. The details of the ground target locations used for this simple case analysis and proposed method development are presented in Table 5.1. The satellites in the various orbits are presented in Table 5.2 where the orbits of the 3 satellites are presented in terms of their inclination and Right Ascension of Ascending node.

Table 5.1: The targets for simple case analysis of graph theory use in satellite tasking.

Node ID	Node name	Latitude (Deg)	Longitude (Deg)
‘Target 1’	'Moscow'	55.5	37.4
‘Target 2’	'London'	51.3	0.1
‘Target 3’	'Washington, D.C'	38.5	-77
‘Target 4’	'Sydney'	-33.5	151.1

Table 5.2: The in-orbit satellites for simple case analysis of graph theory use in satellite tasking

Node Name	Inclination (Deg)	Right Ascension of Ascending Node (Deg)
'Satellite 1'	50	2
'Satellite 2'	52	0
'Satellite 3'	54	0

From the ground-track propagation of the satellites in the orbits, Table 5.2, while considering the satellite field of regard, the graph presented in Figure 5.2 is generated. In Figure 5.2, the target nodes are named using the Node IDs presented in Table 5.1 and the satellites are labelled using the Node Names in Table 5.2. Figure 5.2 gives a representation of two possible scenarios. For the first scenario, all edges represent contact between the nodes, even the dotted ones i.e., two satellites (satellite 1 and satellite 3) are connected to three targets, and satellite 2 is connected to two targets. The second scenario is illustrative of a case when the dotted edges are no longer connected i.e. satellite 1 has no contact with target 2 and 3, and satellite 3 has no contact with target 4.

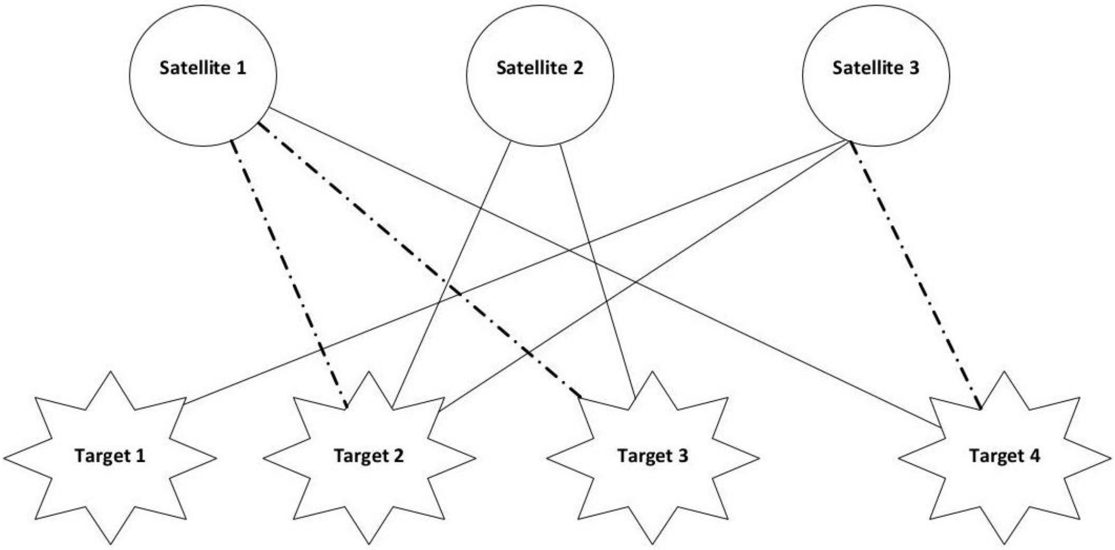


Figure 5.2: Satellite and target network architecture representing 2 scenarios used for the simple network study where for scenario 2, the dotted edges indicate no contact.

In the simple networks presented in Figure 5.2, no satellite node is in contact with all the target nodes. For those networks, it can be seen from visually assessing the graphs the minimum number of satellites needed for maximum coverage of the targets, (which in this case is full coverage), is two for scenario 1 and 3 for scenario 2. The results from running Algorithm 5.1 are presented and analysed to verify the methods ability to find the minimum number of satellites needed either for maximum coverage or for full coverage. The results of larger networks are presented in sections 5.5 – 5.6 of this chapter.

For the simple network presented in Figure 5.2, it is observable that the set of two satellites needed for full coverage for scenario 1 must include satellite 3. This is however not visually assessable for more complex networks and so methods that are efficient and give an accurate and clear analysis are needed. Algorithm 5.1 is used for this analysis and the results obtained are presented in Table 5.3 for both the first scenario and second scenario in Figure 5.2. As expected, the results show that the minimum number of satellites needed to overfly all targets is two satellites for scenario 1 and for scenario 2 is all 3 satellites. This result can be confirmed through visual assessment of Figure 5.2.

Table 5.3: Results for minimum number of satellite nodes needed to overfly all targets.

Parameter	Scenario 1	Scenario 2
Minimum number of satellites needed to view all targets	2	3
Number of 2-satellite sets viewing all targets	2	0
Possible satellite combinations to overfly all targets at least once	[satellite 1, satellite 3] [satellite 2, satellite 3]	[satellite 1, satellite 2, satellite 3]

The results on Table 5.3 show that Satellite 3 appears in both unions of scenario 1 indicating that it is *"important"* to the network and may be said to be a critical node. Satellite nodes that appear in many sets when considering the maximisation of coverage in this work are said to be high cohesive nodes as without them the coverage is reduced. Without satellite node 3 for example, only 3 of the 4 targets will be overflown. In the case of the simple network for example, Satellite 3 is the only satellite node connected to target node 1. It can therefore be concluded that, satellite nodes common to multiple sets/unions are the important satellite nodes in that, inclusion of those satellite nodes for the tasks ensure that target nodes with few connections are considered in the tasking hence this is the metric proposed to identify the high cohesive nodes. Not including such satellites can be taxing to the network and such information is important for the decision-making process.

Methods that may further relay other structural information and even consider the weighting of the network are also needed to aid in making informed tasking decisions. The method presented so far has only considered binary graph where the existence or non-existence of edges between nodes is assessed.

5.4.3 Satellite tasking, Task 2: Maximisation of target observation time:

In this section, the task related to the total target(s) observation time by a satellite, referred to as "the duration of view", is analysed. The proposed approach aims to identify the satellite(s) that maximise the sum of the duration of view of all the targets that they overfly. A satellite's total observation time over each target that it overflies is calculated from the propagation data (already stored on the edges when creating the graph), and the values are the edge weights related to the duration of view. The algorithm proposed for this task is based on the weighted adjacency matrix of the graph derived from the edge weights. An analysis can be done using various methods to identify the satellite nodes that have maximum total node edge weights.

As per the findings of G. Lohmann et. al., [155], eigenvector centrality gives computationally efficient insights on the structure of the network. Eigenvector centrality is the proposed approach studied in this section to give insights on the node optimality. Node optimality herein refers to identifying the nodes that maximise on the sum of node edge weights. Unweighted eigenvector centrality gives insights on the importance of nodes based on the importance of their connections and weighted eigenvector centrality considers weighted strength of a nodes connections [123, 156]. The satellite nodes that have optimum task performance (maximum sum of node edge weights), need to be identified. From definition, the satellite nodes with higher eigenvector centralities are connected to the targets that are viewed for the maximum durations hence maximising on the task. *"Optimum task performance satellites"* in this work refers to those satellite nodes that have the highest total sum of edge weight values, for example in terms of duration of view, the satellite nodes that maximise the observation time of all the targets that they overfly.

For unweighted degree centrality, the number of connections is the primary indicator of importance. It however does not consider the quality or strength of connections based on edge weights nor the structural position of a node [122, 123]. The degree of the satellite nodes can however be used to inform on the number of target nodes the satellites are in contact with. The analysis of both the weighted and unweighted adjacency matrices are done in this section so that clear conclusions can be drawn on which centralities give the desired results and insights for tasking. For the simple network studied therefore, the degrees of the nodes and the weighted and unweighted eigenvector centrality of the graph are assessed with an aim of identifying the satellite nodes that individually optimise the duration of view of the targets. The choice to use the eigenvector and degree centralities is further justified and explained in section 5.5.

From a given satellite and target network the adjacency matrix (A) of the graph can be derived. For the simple network being analysed in this case, the nodes of the graph are 7, and so a 7×7 matrix is obtained. In the case of unweighted analysis, the adjacency matrix derived is a binary matrix, where a is the element on the ith column and jth row,

$$for\ a_{ij} \geq 0, \forall i, j \qquad (5.5)$$

To model the bipartite adjacency matrix in terms of the partite (different parts), the adjacency matrix can be partitioned i.e., divided into two parts. This gives a 4×3 matrix. If the two parts of the bipartite graph are x and y, the partitioned adjacency matrix can be given as per ref. [157],

$$A = (a_{ij})_{x \times y} \qquad (5.6)$$

where $a_{ij} = 1$ if node x and y are connected and a_{ij} = 0 if not.

The adjacency matrix of the separate parts of the bipartite graphs may then be given as,

$$A^T A = B^x \textit{ and } AA^T = B^y \qquad (5.7)$$

where y and x represent the different parts of the bipartite graphs and so the B matrices can be studied for the individual part centralities

When the graph edge weights are considered however, the adjacency matrix is represented using the weight values of the connections. In terms of the duration of view task, the weight of the matrix is related to the total sum of duration of view that a satellite overflies each individual targets.

Based on the first scenario of the illustrative architecture presented in Figure 5.2, page 106, the weighted graph presented in Figure 5.3 is obtained. The graph edges, defined as $E(G)$, where G is the graph, contain weights which give the total duration that a satellite in orbit is overflying each target. The colourmap on the graph indicates the degree of each node which represent the number of connections they have. Satellite 1 and Satellite 3 nodes have connections to 3 target nodes hence a degree of 3. Target node Moscow has only one satellite overflying it and so has a degree of 1. No satellite node has 4 edges nor a degree of 4 and so in this case, and as previously noted, there is no single satellite that views all targets.

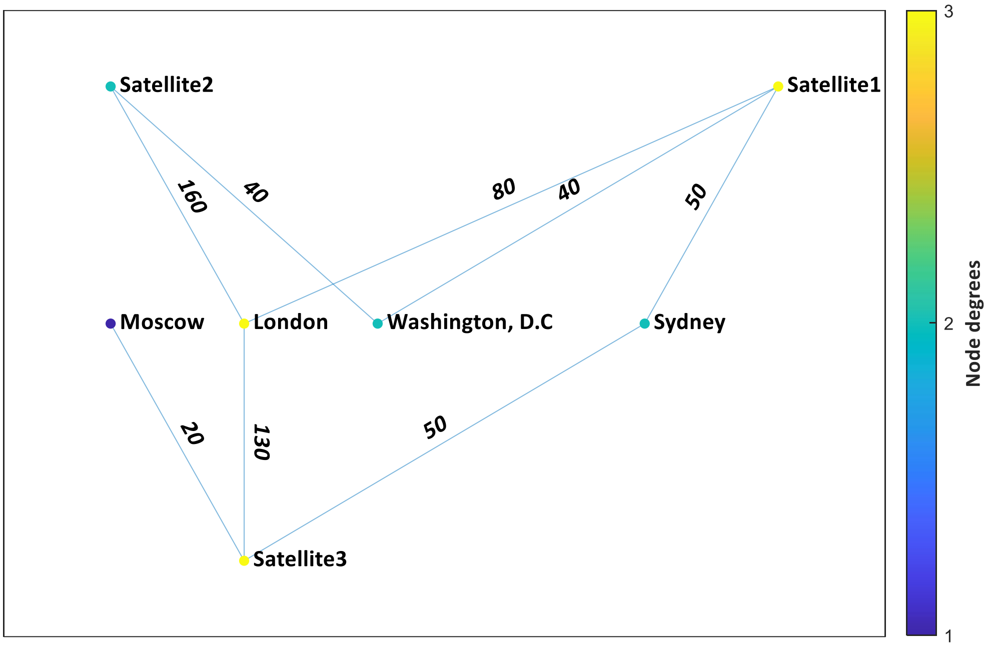


Figure 5.3: Simple case Graph for Scenario 1 with weights related to the duration of view indicated on the edges and the colour bar indicating the degree of each node.

In this chapter, the "*weighting of a node*" or "*node weight"* refers to the total sum of edge weights connected to that node. This represents the total sum of the duration of view in this section. Satellite 2 in Figure 5.3 for example has a total sum of the duration of view equal to 160+40 = 200s and hence a node weight of 200. The eigenvector centralities are analysed using the adjacency matrix derived from the graph. The unweighted and weighted adjacency matrices are presented, in equations (5.8) and (5.9) respectively.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.8}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 80 & 40 & 50 \\ 0 & 0 & 0 & 0 & 160 & 40 & 0 \\ 0 & 0 & 0 & 20 & 130 & 0 & 50 \\ 0 & 0 & 20 & 0 & 0 & 0 & 0 \\ 80 & 160 & 130 & 0 & 0 & 0 & 0 \\ 40 & 40 & 0 & 0 & 0 & 0 & 0 \\ 50 & 0 & 50 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{5.9}$$

For the unweighted case, it is expected that satellite nodes with more connections to target nodes which have multiple connections will give high unweighted eigenvector centrality values. In the weighted case the strength of the connections, based on the weight of the node and in this case the duration of view, is influential in that, a satellite with many connections but connected to a target node with a low weight will have a lower weighted eigenvector centrality than one that connects to fewer targets that have higher weights. These expectations are observed when the eigenvector centralities of the graph, Figure 5.3, are calculated. The results obtained are presented in Figure 5.4 and Table 5.4, and a discussion based on these results is given.

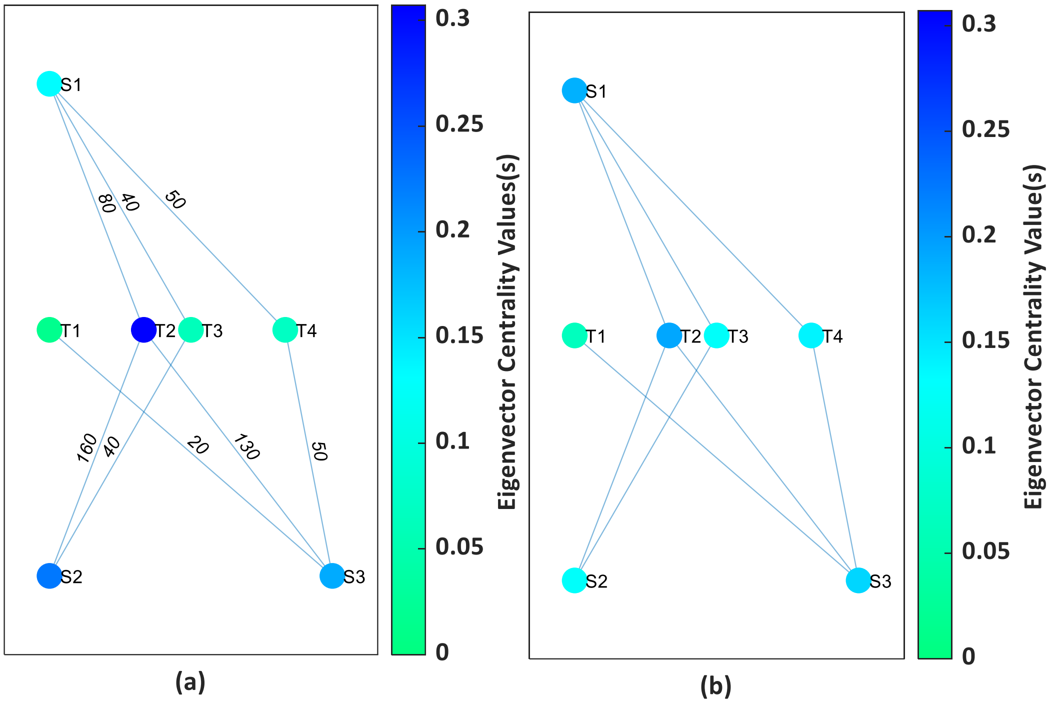


Figure 5.4: Eigenvector centrality values related to the duration of view task (a) weighted and (b) unweighted.

The results graphically presented in Figure 5.4 show both the (a) weighted and (b) unweighted eigenvector centrality values highlighted using a colourmap. The values are tabulated on Table 5.4 for better clarity and analysis.

Table 5.4: Weighted and unweighted centralities for satellites and targets from simple network, scenario 1 simulation.

Node Name	Weighted Eigenvector Centrality score	Unweighted Eigenvector Centrality Score	Node degree	Total duration viewed (sec)
Moscow (T1)	0.016	0.065	1	20
London (T2)	0.307	0.192	3	370

Washington, D.C., (T3)	0.062	0.127	2	80
Sydney (T4)	0.070	0.140	2	100
Satellite 1 (S1)	0.132	0.186	3	170
Satellite 2 (S2)	0.224	0.129	2	200
Satellite 3 (S3)	0.189	0.161	3	200

From Table 5.4, the unweighted eigenvector centrality results are observed to be closely related to the degree of the node. To verify this, the correlation of the data is calculated and presented in Table 5.5. The correlation shows that the p-value between the unweighted eigenvector centrality and the duration of view is below the chosen significance level of 0.05. Though this indicates a correlation significance, the correlation coefficients and the p-values generally indicate that the unweighted eigenvector centrality is more related to the number of edges than to the node weights (total sum of duration of view) which is expected as it is considering the binary adjacency matrix related to the node connections.

Table 5.5: Unweighted eigenvector centrality correlation calculations

Unweighted Eigenvector Centrality Vs	Node Degree	Total Duration of View
Correlation	0.964	0.795
P-Value	4.5e-4	0.032

From Figure 5.4 (b), it is observed that if a satellite node connects to a high unweighted eigenvector centrality target node, it also increases the unweighted eigenvector centrality of that satellite node. Therefore, if a satellite node has the same degree as another satellite node but lower unweighted eigenvector centrality, it indicates that the satellite node is connected to a target node that has a lower node degree. Node S1 is observed to have the highest score in terms of the unweighted eigenvector centrality than the other satellites. S1 and S3 have the same number of connections, but S3 has a notably lower centrality as one of its connections is the lowest centrality target T1. From these results therefore, satellite nodes with equal node degrees to others but lower unweighted eigenvector centralities may be further analysed for structural cohesion insights.

If selecting a satellite to task based on the unweighted eigenvector centrality therefore, the satellite(s) to task based on the results on Table 5.4 while considering maximisation of the unweighted eigenvector centrality would be satellite node S1. In terms of duration of view however, this satellite does not give the maximum total sum of the duration of view and so this shows that the unweighted eigenvector centrality does not give the desired clear insights while considering quantifiable tasks for undirected graphs.

From the data on Table 5.4, the correlation coefficients of the weighted eigenvector are also calculated and presented in Table 5.6. As expected, the correlation coefficient and the p-value indicate that the weighted eigenvector centrality is more related to the node edge weights than the number of edges attached to the node.

Table 5.6: Weighted eigenvector centrality correlation values

Weighted Eigenvector Centrality Vs	Node Degree	Total Duration of View
Correlation	0.698	0.972
P-Value	0.081	2.49e-4

The weighted eigenvector centrality considers the strength of the nodes' connections i.e., the sum of total edge weights (total sum of duration of view) of the neighbouring nodes as expected. For example, S2 and S3 have the same total sum of the duration of view value of 200 seconds, but S3 has a notably lower centrality as one of its connections has the lowest weighted eigenvector centrality, i.e., target T1. Satellite 1, S1, has the lowest weighted eigenvector centrality score as it has the lowest sum of the duration of view, and this can be observed on the graph given in Figure 5.4 (a). If selecting the satellite to task based on the weighted eigenvector centrality therefore, satellite S2 would be selected as it has the highest value of the weighted eigenvector centrality. Selecting S2 would indeed maximise on the duration of view hence showing that relying on the weighted eigenvector centralities gives direct insights on the node optimality of weighted graphs.

Insights such as which satellite node(s) will enable full coverage and in turn the general structure of the network are however not directly drawn from the eigenvector centrality values of an undirected graph. Moreover, the results show that due to the influence of the node's neighbours, satellite nodes connected to the target nodes with low values may be overlooked for tasking if selection is based only on the eigenvector centralities values. This justifies the need for a hybrid method that makes use of multiple graph theory approaches to be able to task satellites with different mission objectives.

5.4.3.1 Hybrid Method

In the simple network first scenario presented in Figure 5.3 page 110, when considering tasking satellites that have the highest values of the sum of the duration of view and contribute to maximum or full coverage, a combination of the results from the developed Algorithm 5.1 and eigenvector centralities can be used. If tasking based on both the full coverage hence structure of the network and the highest weights related to the tasks, the results from the developed neighbours' brute force algorithm and eigenvector centralities on scenario 1 indicate that the satisficing satellites to task are satellites 2 and 3. Further analysis of the hybrid method is studied and analysed in section 5.6 when analysing complex networks. All the algorithms developed for the proposed hybrid method are presented in Appendix F. Additionally, the weighted eigenvector centrality analysis is seen to be both computationally and time efficient in identifying the satellites to be tasked based on node weight optimality.

A justification of the use of the hybrid method for satellite identification and tasking is presented in section 5.4.4. Moreso, the justification of the use of Algorithm 5.1 to identify high cohesive nodes. Eigenvector centrality is used to identify optimum satellite nodes related to weight related tasks and this is comparable to the use of the 1st Principal Eigenvector.

5.4.4 Justification of Using the Proposed Hybrid Method for Satellite Tasking

The main aim of the work in this chapter is to develop a method for tasking satellites using graph theoretical approaches on satellite-target networks. The two tasks studied so far are related to the target coverage, and the total observation time of the targets by the satellites that overfly them. This research gives an analysis of the two distinct satellite tasking problems using two complementary graph theory approaches. The objective is to develop a hybrid methodology that leverages insights from both approaches to identify satellite nodes that achieve satisfactory performance for tasks simultaneously. To validate the effectiveness of this proposed hybrid approach compared to single-method analyses, the Principal Eigenvectors (PEVs) of the simple network presented in Figure 5.3, page 110 is studied.

The first, second, and third principal eigenvectors (sometimes called dominant eigenvectors) in graph theory have been found to offer valuable insights into the structure and properties of a network, and they capture different aspects including the node community influence, the structural influence and the node global influence amongst others. Scholars like, Clark. et. al. [158] analyse the 1st 2nd and 3rd PEVs to determine influential nodes and node communities in a system. Herein, the 1st, 2nd and 3rd PEVs are calculated and analysed to investigate the insights that can be drawn in relation to the tasks studied in section 5.4.2 and 5.4.3. This work investigates the applicability of node community identification techniques to satellite tasking. The primary objective is to determine if these techniques can effectively identify critical satellite nodes. These critical nodes are herein defined as those nodes that:

i. Maintain Network Cohesion i.e., identifying satellite nodes whose removal would disconnect target nodes from the network.

ii. Optimise Node Performance: Assessing the overall satellite node optimality based on the total edge weights. This considers factors like the duration of view, where satellite nodes exhibiting the longest cumulative duration of view over the targets they observe are identified.

This investigation aims to validate the use of both eigenvector centrality and Algorithm 5.1, here after referred to as *the hybrid method*, in identifying the critical satellite nodes.

For bipartite graphs, a method to analyse the principal eigenvectors by using singular value decomposition (SVD) of the adjacency matrix is proposed by X. Shuang et. al., in [157]. X. Shuang et. al., use an analysis of the SVD method, as compared to using eigenvector centrality and develop two methods of identifying influential spreaders in bipartite networks which they name, the SVD-rank, and the SVDA-rank. In the latter, they use an augmented network where two ground nodes are added to the network. The method by X. Shuang et. al., prove to be robust to bipartite networks. As will be presented in the appendix however, the method by X. Shuang et. al., present a result correlating to the one obtained from the 1st principal eigenvector analysis and in turn the eigenvector centrality.

Clark. et. al. [158], demonstrate that the first, second, and third Principal Eigenvectors (PEVs) are effective in identifying the most influential nodes and communities within a network. This study investigates the potential application of this approach to identify nodes such as satellite S3 by assessing their significance within the overall network structure. The 1st PEV is expected to give insights on the importance of a node in relation to the node degree for unweighted cases and in terms of the total sum of the edge weight values when using the weighted case. The 1st PEV is directly related to the eigenvector centrality of the node. The 2nd and 3rd PEVs on the other hand are expected to give insights into community and subcommunity nodes in a graph. Analysing the 2nd and 3rd PEV is expected to identify nodes with similar values and nodes that are more interconnected with each other.

With reference to the adjacency matrices of the simple network scenario presented in Figure 5.3, page 110, the 1st, 2nd and 3rd PEVs are simulated. Both weighted and unweighted PEVs are analysed to assess their insights into the network structure. A key objective is to determine if PEV analysis, independent of Algorithm 5.1, can predict potential disruptions to target coverage. Specifically, can the analysis of PEVs anticipate situations where the loss of a satellite would lead to less target coverage and consequently disruption of the network's overall coverage structure?

Figure 5.5 gives the graphs and the results of the unweighted adjacency matrix' PEVs. For the unweighted case, Figure 5.5 (a), shows the graph being used for the analysis where compared to Figure 5.3 page 110, Moscow is T1, London T2, Washington D.C T3, Sydney T4 and Satellite 1 is S1, Satellite 2 S2, and Satellite 3 S3. The graph of the 1st PEV, (V1) against the 2nd PEV, (V2) is presented in Figure 5.5 (b) which shows the position of the nodes in relation to these PEVs. Figure 5.5 (c) then shows the 2nd PEV against the 3rd PEV (V3) and Figure 5.5 (d) give a 3-dimensional representation of the 3 PEVs.

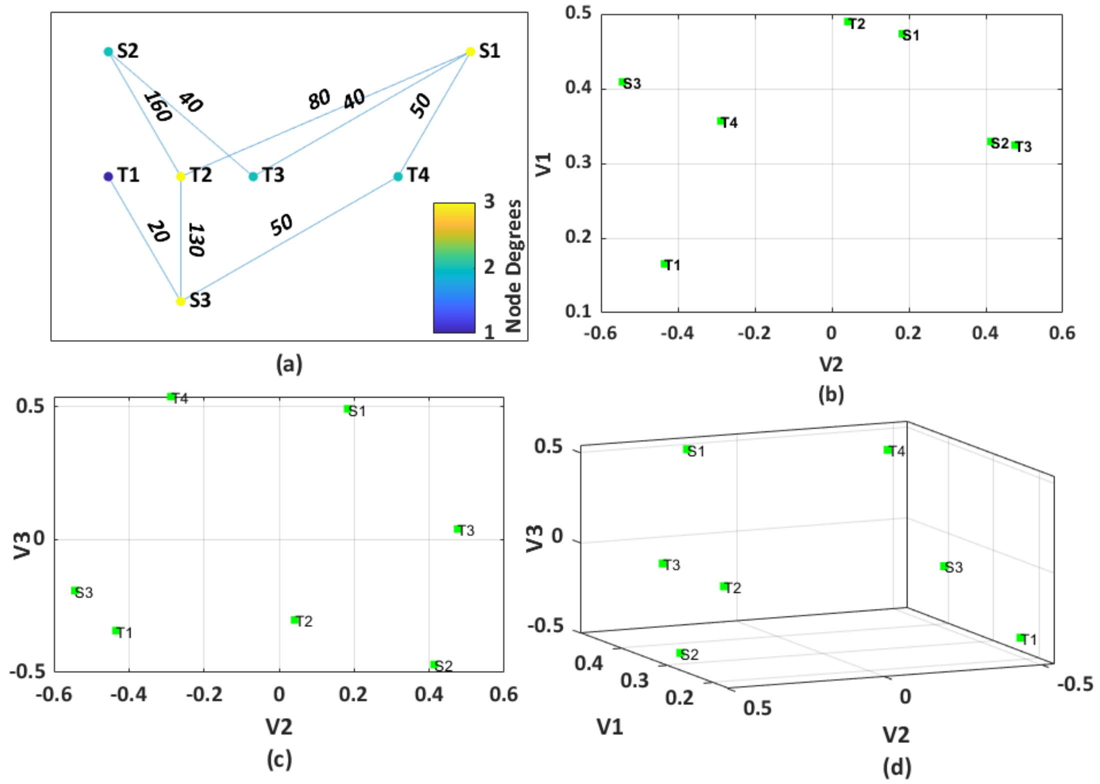


Figure 5.5: Plot of (a) the satellite and target network (b) unweighted 1st (V1) and 2nd (V2) PEVs, (c) unweighted 2nd (V2) and 3rd (V3) PEVs, and (d) 3D plot of unweighted 1st, 2nd, and 3rd PEVs

The magnitude of the first principal eigenvector (V1) is observed to correlate strongly with eigenvector centrality as expected. Figure 5.5 (b), demonstrates that satellite S1 exhibits a higher V1 value compared to the other two satellites. This aligns with the unweighted eigenvector centrality results presented in Figure 5.4, page 111.

While Figure 5.5 (a) visually indicates that both S1 and S3 have three connections, S3 is connected to the lowest-degree target node (T1), resulting in a lower V1 value for S3 compared to S1. V2 is expected to reveal graph structure and is known to be the basis of spectral clustering approaches. Figure 5.5 (b) and (c), demonstrate that, since T1 is solely connected to S3, V2 and V3 indicate a proximity between these two nodes. This suggests a potential community comprising S3, T4, and T1 based on their interconnections. However, the insights derived from V2 and V3 can be limited. For instance, without prior knowledge (such as from Figure 5.5 (a)), V2 might erroneously suggest that S3 has no connection to T2. Furthermore, while V2 and V3 offer some structural insights, they are less explicit than the neighbours' algorithm approach, which directly identifies a node's connections.

In the context of this, small, undirected network, the unweighted adjacency matrix's 2nd and 3rd PEVs provide a general structural overview. For example, they might indicate that removing S3 would isolate T1. However, for more nuanced connection insights, the PEV analysis may be inconclusive. The interpretation of the results from the proposed hybrid method offers a more straightforward and insightful approach compared to relying solely on principal eigenvectors. This finding emphasizes the value of the hybrid method for analysing and identifying critical nodes in a network.

Figure 5.6 shows the weighted 1st, 2nd, and 3rd PEVs, from this it is again evident based on V1 that satellite node S2 has greater importance based on the value of the total edge weights connected to it as well as the total edge weights of the neighbouring nodes. Figure 5.6 (b), suggests a community comprising S2, T2, and T3, as all exhibit negative V2 values. While S1 also connects to T2 and T3, its positive V2 value places it in a distinct cluster. Similarly, Figure 5.6 (c), indicates a sub-community involving S3, T1, and T2, characterized by their positive V3 values. However, the connection between T4 and S3 remains less evident based solely on V3 analysis.

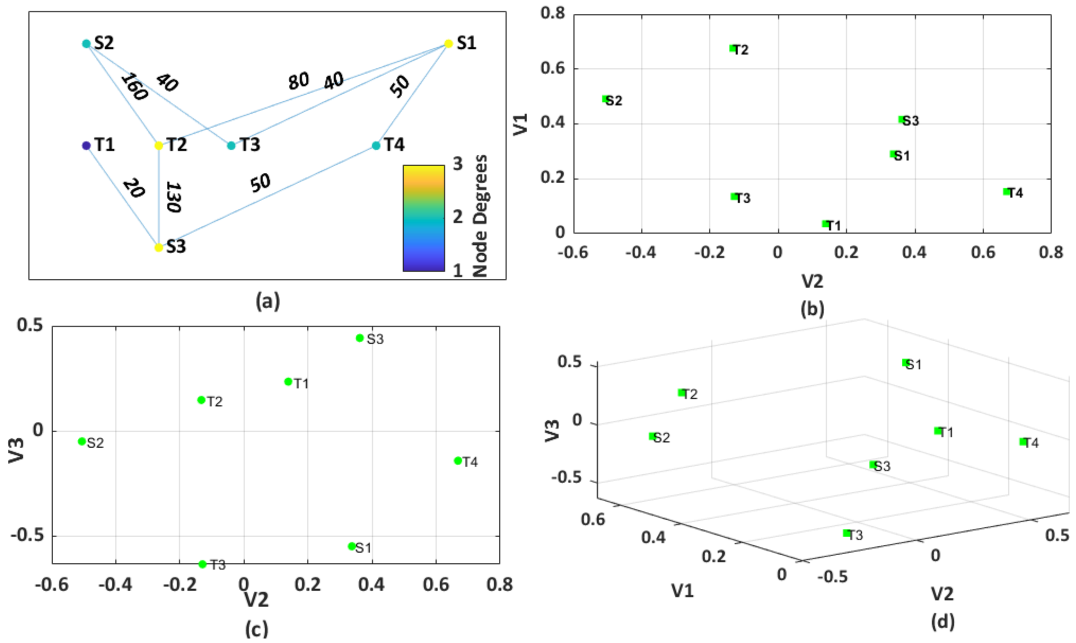


Figure 5.6: Plot of (a) satellite-target network, (b) Weighted 1st (V1) and 2nd (V2) PEVs, (c) Weighted 2nd (V2) and 3rd (V3) PEVs, and (d) 3D plot of 1st, 2nd, and 3rd PEVs

Without the visual reference of Figure 5.6 (a), accurately determining the connectivity of other satellites would be challenging, potentially leading to erroneous conclusions. These limitations, particularly in a small, undirected network, underscore the potential challenges of relying solely on PEV analysis for comprehensive network understanding. This observation further supports the rationale for employing the proposed hybrid method for effective satellite tasking decisions.

To further investigate these dynamics, a directed graph is constructed based on the satellite-target network presented in Figure 5.3, page 110. This directed graph, depicted in, Figure 5.7 is then analysed to determine the centralities and singular value decomposition of the out-degree satellite nodes. On Figure 5.7, the node degrees are going outwards from the satellite nodes to the target nodes hence out-degree. This represents a directed graph where the direction of data flow is defined. The results of this analysis are summarized in Table 5.7.

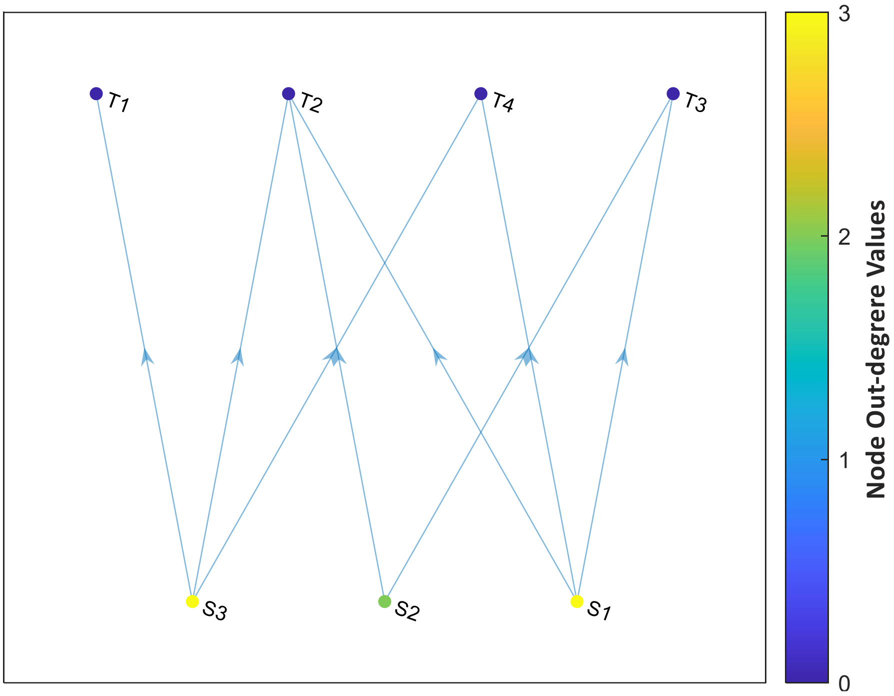


Figure 5.7: Out-degree graph of directed satellite-target simple case graph

In the directed graph, as with undirected graphs, the out-degree and in-degree provide insights into a node's connectivity. Table 5.7 confirms this expectation, reflecting the number of outgoing connections for each node.

The calculated weighted in-degree and out-degree centralities exhibit a strong correlation with eigenvector centralities, given their reliance on the graph's adjacency matrix. By analysing the weighted centrality measures, deeper understanding of a node's position and influence within the directed network structure can be obtained.

Table 5.7: Out-degree Centralities and Singular Value Decomposition analysis

Node	Node Name	Node Out degree	Total Node Out Edge Weight	Out degree Weighted Centrality	Weighted SVD Analysis
'T1'	'Moscow'	0	0	0.00	0.00
'T2'	'London'	0	0	0.00	0.00
'T3'	'Washington, D.C'	0	0	0.00	0.00
'T4'	'Sydney'	0	0	0.00	0.00
'S1'	'Satellite 1'	3	170	0.24	0.41
'S2'	'Satellite 2'	2	200	0.41	0.70
'S3'	'Satellite 3'	3	200	0.35	0.59

Using the principal eigenvectors (PEVs), proves to give some insights related to the node degrees and edge weighting and more so the 1st PEV in terms of the undirected graph and the SVD analysis in terms of the directed graph. Even for the directed case, however, for a simple network, PEV analysis provides unclear predictions regarding the impact of node removal on network structure. This limitation underscores the need for the proposed hybrid methodology. The hybrid approach effectively addresses this by providing valuable insights into the critical satellite nodes.

The proposed hybrid methodology demonstrates adaptability to directed graphs. By modifying the neighbours' algorithm to consider 'successors' (nodes reachable via outgoing edges) within the directed graph (Figure 5.7), and incorporating outdegree weighted centrality, the methodology effectively identifies optimal satellite nodes for tasking.

This modified approach yields results consistent with Scenario 1 in Table 5.3, page 107, identifying two sets of two satellites ({S1, S3} and {S2, S3}) capable of achieving full coverage. By prioritizing critical satellite nodes, the hybrid method selects the satisficing set {S2, S3} for tasking.

This successful application to both directed and undirected graphs highlights the robustness and clarity of the results obtained through the proposed hybrid methodology.

5.4.5 Small network with increased targets and satellites

A graph of 11 satellites, in 11 different orbits and 10 targets is generated after ground track propagation considering the field of regard of the satellite. The graph for this is presented on Figure 5.8 where the top nodes are 10 target nodes, the bottom nodes are the satellite nodes. The colormap associated with the nodes represents their respective degrees within the graph.

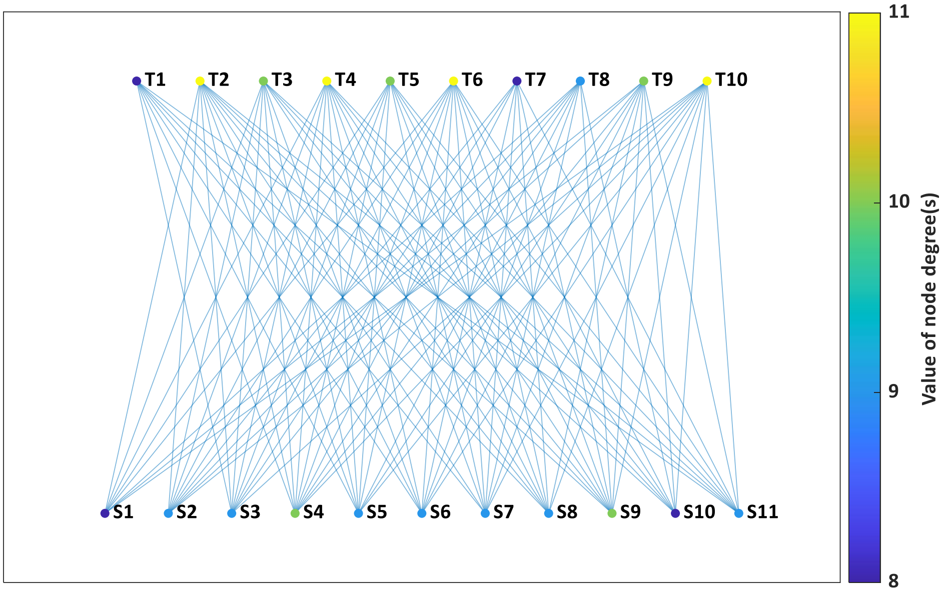


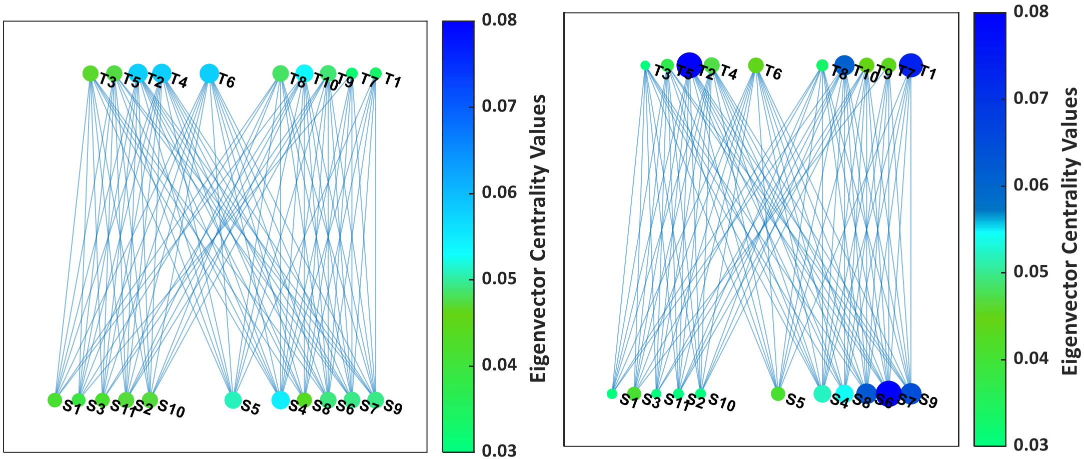
Figure 5.8: Graph of 11 satellite nodes (S1 – S11) and 10 target nodes (T1 – T10)

Algorithm 5.1 is applied to the network in Figure 5.8. The analysis reveals that two satellite nodes, S4 and S9, maintain contact with all ten target nodes for unrestricted overflight period. However, when considering overflights exceeding ten seconds per visit, only satellite S4 provides full coverage. For overflights exceeding 10 seconds per visit, therefore, the network achieves maximum coverage (i.e., all targets observed) and the minimum of one satellite (S4) is necessary for complete coverage.

Furthermore, excluding S4, eighteen distinct two-satellite combinations (unions) are identified that collectively achieve full coverage. Satellite S9 appears in eight of these combinations, indicating that it is a critical node for full coverage. S6 exhibits the second-highest criticality. While these findings suggest prioritizing S4 or S9 for tasking, the optimal selection of the union should ultimately depend on other mission objectives.

The presented case of 10 targets and 11 satellites can however be considered a simple small network as it can be assessed visually from the graph. A visual inspection of Figure 5.8 corroborates the validity of Algorithm 5.1.

To further analyse the network in Figure 5.8, eigenvector centralities are computed for the duration-of-view task. Figure 5.9 presents the resulting graphs for both weighted (Figure 5.9 (b)) and unweighted (Figure 5.9 (a)) cases, while considering views exceeding ten seconds. Satellite S7 exhibits the highest eigenvector centrality in the weighted case, while S4, as expected, has the highest score in the unweighted case.


T3 T5 T2 T4 T6 T8 T10 T9 T7 T1
S1 S3 S11 S2 S10 S5 S4 S8 S6 S7 S9
0.08
0.07
0.06
0.05
0.04
0.03
Eigenvector Centrality Values

(a) (b)

Figure 5.9: Eigenvector Centralities (a) unweighted (b) weighted

The results presented in Table 5.8 further confirm that, based on weighted eigenvector centrality (reflecting total edge weights), satellite node S7 is the optimal choice for maximising the duration of view. While satellite nodes S9 and S6 offer the same duration of view, S9 exhibits a higher weighted eigenvector centrality due to its connections to high-centrality target nodes.

The maximum/optimal duration of view achievable by a single satellite is 890 seconds, provided by S7. Employing the proposed hybrid approach indicate satellite node sets N {S9, S7} and N {S6, S7} ensures full target coverage with a near-optimal average duration of view of (730+890)/2 = 810 seconds. Considering satellite node S9's connectivity to high-centrality target nodes however, the preferred satisficing satellite set is N {S9, S7}.

Table 5.8: Eigenvector Centralities for 10 target 11 orbit graph

Satellite	Unweighted Eigenvector Centrality	Weighted Eigenvector centrality	Node Degree	Node Edge Weights
S1	0.042	0.028	7	410
S2	0.047	0.029	8	380
S3	0.039	0.041	7	510
S4	0.055	0.052	10	640
S5	0.051	0.041	9	490
S6	0.049	0.062	9	730
S7	0.049	0.080	9	890
S8	0.044	0.054	8	590
S9	0.049	0.065	9	730
S10	0.047	0.029	8	380
S11	0.042	0.027	7	310

An analysis of the PEVs for the unweighted graph results are presented in Figure 5.10. The unweighted V1 values directly correspond to the node degrees, revealing that targets T1 and T7 possess the lowest degrees. However, the PEV analysis provides limited insights into the specific satellite-target connections. Notably, the PEVs do not explicitly indicate the impact of satellite loss on the network connectivity. This limitation further reinforces the necessity of employing the proposed hybrid method for effective satellite selection and tasking.

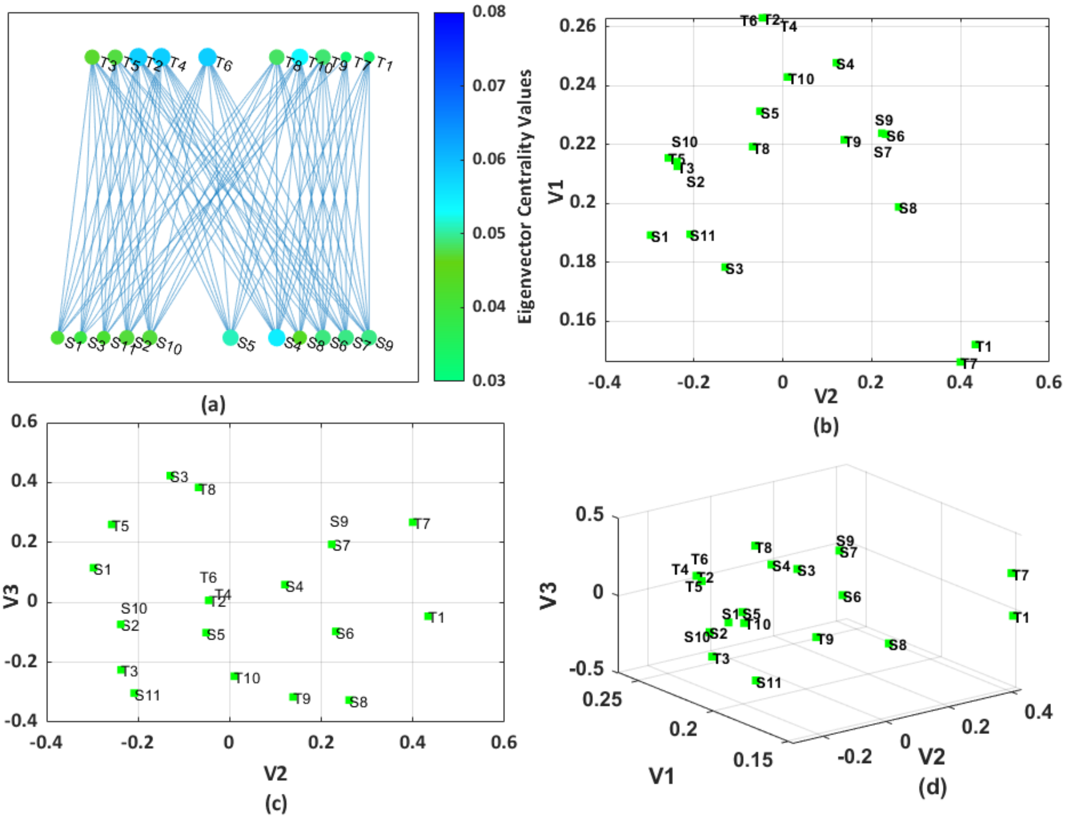


Figure 5.10: Plot of (a) Unweighted eigenvector centralities, (b) Unweighted 1st (V1) and 2nd (V2) principal eigenvectors, (c) 2nd (V2) and 3rd (V3) principal eigenvectors, and (d) 3D plot of 1st, 2nd, and 3rd principal eigenvectors

The weighted PEVs presented in Figure 5.11 demonstrate that the 1st PEV (V1) can effectively identify satellite nodes with optimal performance with respect to the task, in this case, maximising the duration of view. However, the 2nd and 3rd PEVs Figure 5.11 (b) – Figure 5.11 (d) provide limited insights into the specific satellite node combinations required to achieve maximum target coverage, a crucial aspect of structural cohesion in this work. While identifying optimal task satellites for larger networks using PEV analysis can be valuable, this information is also obtainable through weighted eigenvector centrality analysis. These observations further underscore the necessity of employing the proposed hybrid method for effective satellite selection and tasking.

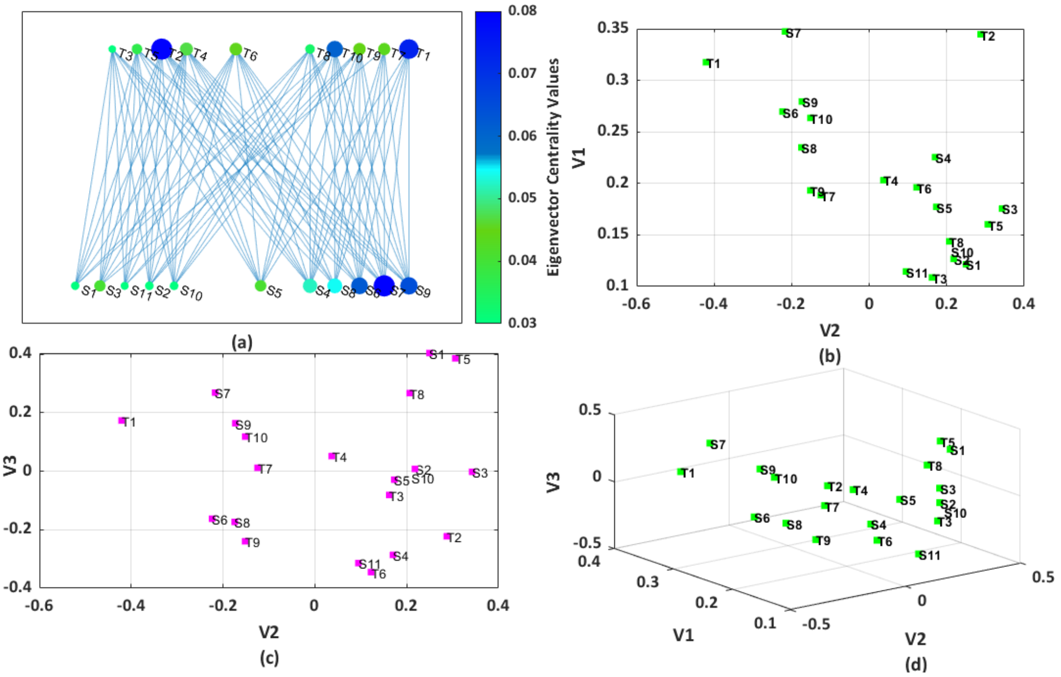


Figure 5.11: Plot of (a) Weighted eigenvector centralities , (b) Weighted 1st (V1) and 2nd (V2) PEVs, (c) Weighted 2nd (V2) and 3rd (V3) PEVs, and (d) 3D plot of 1st, 2nd, and 3rd PEVs

5.5 Complex Satellite-Target Graph Analysis Related to Increased Number of Satellites

To demonstrate the applicability of the proposed hybrid satellite tasking method to larger and more complex scenarios, analysis is further extended to a network comprising 50 satellites and 10 targets. The analysis in this section aims to,

i. Determine the minimum number of satellites required for full coverage, (to overfly all targets at least once) and identify the satellite nodes that if removed would limit this,

ii. Identify the satellites with the highest cumulative duration of view across all observed targets,

iii. Identify and select the satellite nodes that maximise on number of times of overflight across all targets (node optimality based on the sum of overflights),

iv. and determine a satisficing set of satellite nodes by considering trade-offs among the criteria defined in (i) - (iii) while minimizing the number of tasked satellites.

The network of the 50 satellites and 10 targets is depicted in Figure 5.12. The colormap to the right of the graph represents the node degrees, where a degree of 50 indicates the maximum number of connections. Given the network structure, this maximum degree must belong to a target node, as the highest possible degree for a satellite node is limited to 10 (the number of targets). The graph presented in Figure 5.12 is used for analysis in this section to demonstrate the robustness and efficiency of the proposed hybrid method in determining optimal satellite selections for maximizing coverage and weighted task performance.

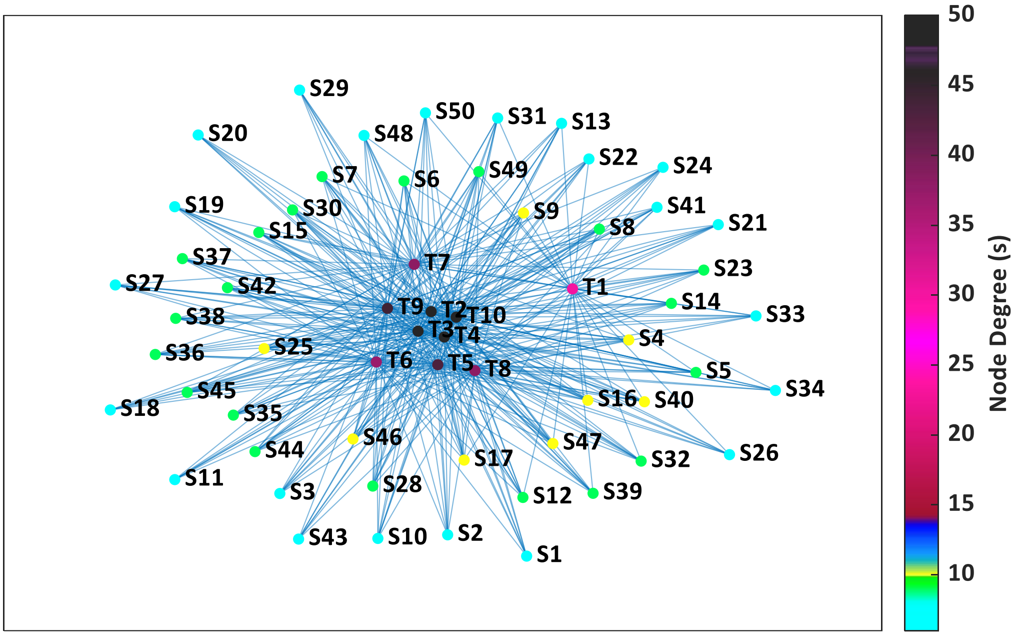


Figure 5.12: Network of the *50* satellites (S1 – S50) and 10 target points (T1 – T10) used for the complex network simulation and analysis.

5.5.1 Minimum Number of Satellites Needed to Overfly All Targets.

For the network, Figure 5.12, Algorithm 5.1 is employed to determine the minimum number of satellite nodes required to effectively maximise coverage while minimising the number of satellites to task. The analysis reveals that eight out of the 50 satellites each in a distinct orbit individually provide full coverage of all ten targets within a 2-day repeat orbit.

This finding is corroborated by a visual inspection of Figure 5.12, where satellite nodes with a node degree of 10 (represented by yellow) are readily identifiable. These nodes, by definition, have connections to all ten targets. Consequently, the minimum number of satellites necessary for full coverage in this scenario is one. This result demonstrates the continued suitability of Algorithm 5.1 for determining the minimum number of satellites needed for maximum coverage, even as the number of satellites and targets increase.

Upon excluding the eight satellites that individually provide full coverage, the analysis identifies 473 distinct two-satellite combinations (referred to as "unions") capable of achieving full coverage. The evaluation of unions is further explored in Section 5.6, as they are particularly relevant when considering multi-tasking scenarios for the satellites.

5.5.2 Analysis of Node Optimality Related to Duration of View of the Targets.

The tasking method presented in this work proposes the combined use of Algorithm 5.1 and the eigenvector centrality of the network's adjacency matrix. Eigenvector centrality is proposed to determine optimum satellite task performance, particularly for tasks such as the duration of view that are influenced by the network's weighted properties.

This section investigates both the unweighted and weighted eigenvector centralities and additionally, the weighted degree centralities and gives a comparison to further justify the use of eigenvector centrality. The goal is to further justify the robustness and suitability of eigenvector centrality as the primary metric to assess the node optimality for both small networks and complex, large networks. It is expected that both weighted and unweighted satellite node eigenvector centralities are significantly influenced by the connected target nodes i.e., eigenvector centralities reflect the node importance in relation to the importance of its neighbours.

Using the 50 satellites and 10 targets network, presented in Figure 5.12, the analysis is focused on overflights exceeding 10 seconds, i.e., if a satellite overflies a target for less than 10 seconds, it is not considered as an overflight of the target and is therefore not included in the analysis. In the present case of 50 satellites and 10 targets for example, Algorithm 5.1 identifies the number of individual satellites that view all targets with view durations greater than 10 seconds to be 4 satellites. Additionally, 587 2-satellites sets (unions) also facilitate full coverage.

The number of satellite nodes in contact with all the target nodes in this section is key to analysing the unweighted degree centrality and unweighted eigenvector centrality, as they are both influenced by the number of edges. This means that the 4 satellites that have full coverage, are expected to have the highest values for both unweighted eigenvector and degree centralities.

The results of the weighted and unweighted eigenvector centrality analysis for the 50 satellites are presented in Figure 5.13. It is notable that significant disparities are observed particularly in the nodes that have maximum values for each of the centrality measures.

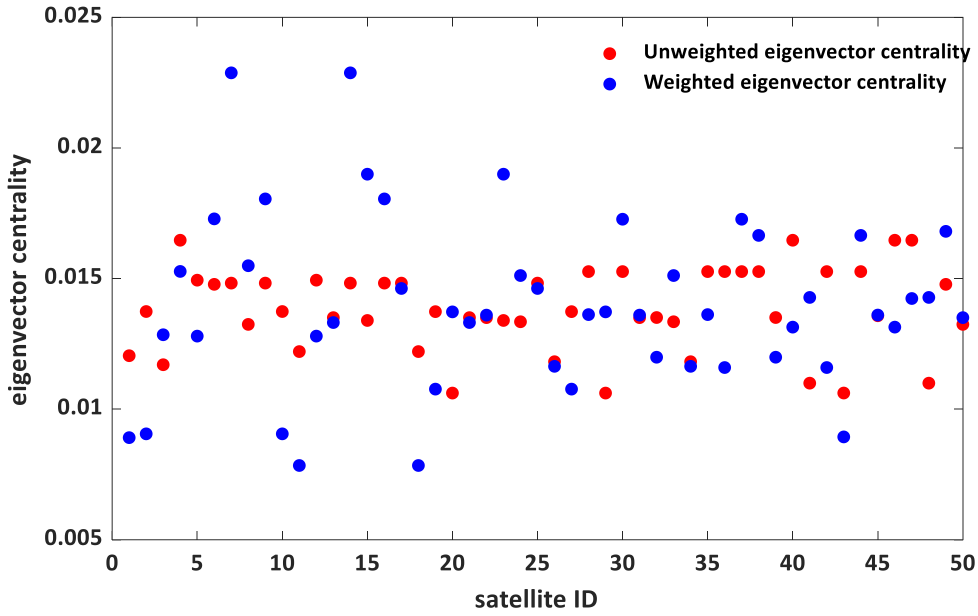


Figure 5.13: Graph of weighted and unweighted Eigenvector centrality analysis for the 50 satellite nodes, based on the duration of view weighting.

Figure 5.13, indicates that in terms of the weighted eigenvector centrality, satellite 7 and 14 have the highest scores. These two satellites have the longest total duration of view of the targets that they overfly which is 890 seconds each. They are also both in contact with nine target nodes, one of them being target node T2. Target node T2 has the highest total duration of view amongst the target nodes which is 5420 seconds while the second highest total duration of view is target T1 with 3440 seconds. This shows that nodes connected to T2 will exhibit high values of the eigenvector centralities. In general, based on the analysis of weighted eigenvector centralities, the mentioned factors make satellite 7 and 14 have weighted eigenvector centralities significantly higher than the other satellite nodes.

Satellite 7 and 14 are however not the highest in terms of unweighted eigenvector centralities where satellite 4, 40, 46 and 47 have the highest values. These are highest because the unweighted eigenvector centrality is closely related to the degree of the nodes. The four satellites, 4, 40, 46 and 47 all have degrees of 10, i.e., they overfly all the targets, but they have a low total duration of time of overflight. These are 640 seconds, 530 seconds, 530seconds and 590 seconds respectively. In terms of the weighted eigenvector centrality therefore, despite being connected to target 2, they are also connected to weaker target nodes such as T6 which has a total duration of view of 1960 seconds, hence lowering their general weighted eigenvector centrality values.

The weighted degree centrality analysis also gives results closely related to the weighted eigenvector results analysis. Mapping the normalised values of the two results on each other, however, shows that there are some differences. This is presented in Figure 5.14. The difference between the two results is because weighted degree centrality only considers the actual total duration of view weight values of the nodes whereas weighted eigenvector centrality considers the weight of the neighbours of the nodes i.e. the total sum of edge weights of the connections is included in eigenvector analysis.

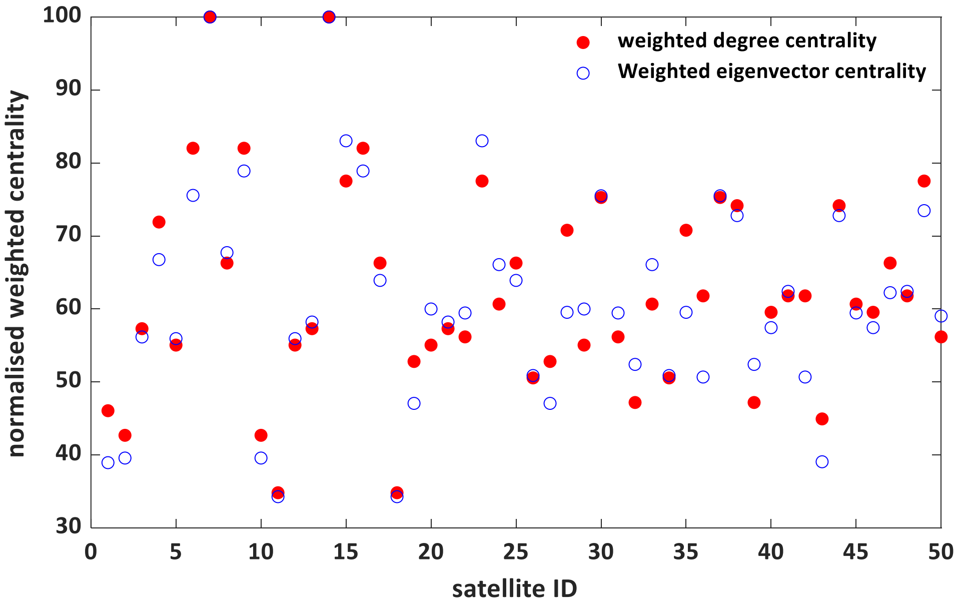


Figure 5.14: Graph of the weighted normalised eigenvectors and degree centralities showing the differences between the two for 50 satellites analysis.

From Figure 5.14, taking satellite 28 for example, there is a significant difference between the weighted degree centrality and the weighted eigenvector centrality. This is again attributed to the connection of satellite 28 to 9 target nodes which exclude the second highest duration of view value target, T1. From this analysis, for the duration of view task, satellite nodes 7 and 14 have the highest centrality scores in terms of both weighted degree centrality and weighted eigenvector centrality. Based on the analysis and results showing that the weighted eigenvector centrality gives different selections based on the neighbouring nodes however, it is proposed to add the weighted degree centrality analysis to the hybrid method. This addition is to enable an operator to obtain more robust and informed satellite selections that are purely based on the individual node weight without the influence of the neighbouring node. Although with a small difference, for example, eigenvector centrality shows that the second-best pair of satellites to be tasked based on the total duration of view should be satellite 15 and 23. These satellites however have lower duration of views than satellites such as satellite 6, 9 and 16. This is due to the connection of the three latter satellites to target nodes with lower centralities. It would therefore be ideal for a decision maker to be aware of such insights before selecting the satellites to be tasked. This addition is more profound in multi-tasking and is further analysed in section 5.6.

Nevertheless, both weighted degree centrality and eigenvector centrality analysis have led to the conclusion that satellite 7 and 14 are the best satellites for maximisation of the total sum of the duration of view. For purposes of determining and tasking the satellite(s) that optimise the duration of view task, therefore, the weighted eigenvector centrality has again proved to be a suitable metric for even large, complex networks.

5.5.2.1 Principal Eigenvector analysis for the Duration of View

An analysis of the weighted and unweighted 1st, 2nd and 3rd PEVs are done based on Figure 5.12. Using the developed hybrid method gives clear insights on the node optimality based on the tasks that consider the node weights as well as identifying the satellite nodes that ensure maximum coverage of the targets with minimum number of satellites.

From Figure 5.15, the unweighted 1st principal eigenvector shows that target node T1 is the least overflown. It is however not clear which satellites overfly the target. This means that insights on which satellite node when removed will reduce coverage cannot be drawn from the 1st PEV, V1. The 2nd and 3rd unweighted PEV on the other hand give insights on which satellites overfly common targets. From Figure 5.15 (b) for example, satellite nodes S34 and S26 have the same value of V2 and V3 and are found to overfly the same targets. This however does not give conclusive insights on which satellites nodes when removed will result in less coverage.

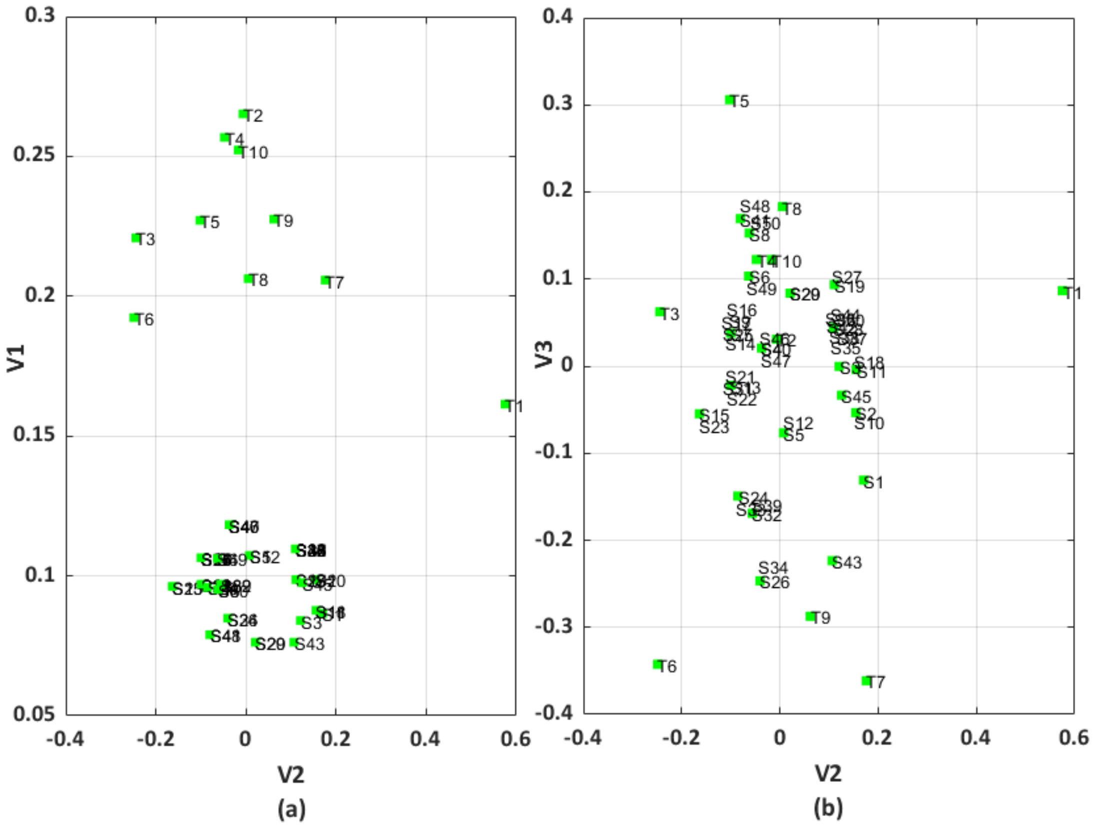


Figure 5.15: Plot of (a) Unweighted 1st (V1) and 2nd (V2) eigenvectors, (b) Unweighted 2nd (V2) and 3rd (V3) eigenvectors

From calculating the 1st, 2nd and 3rd weighted principal eigenvectors, the plot, Figure 5.16 is generated.

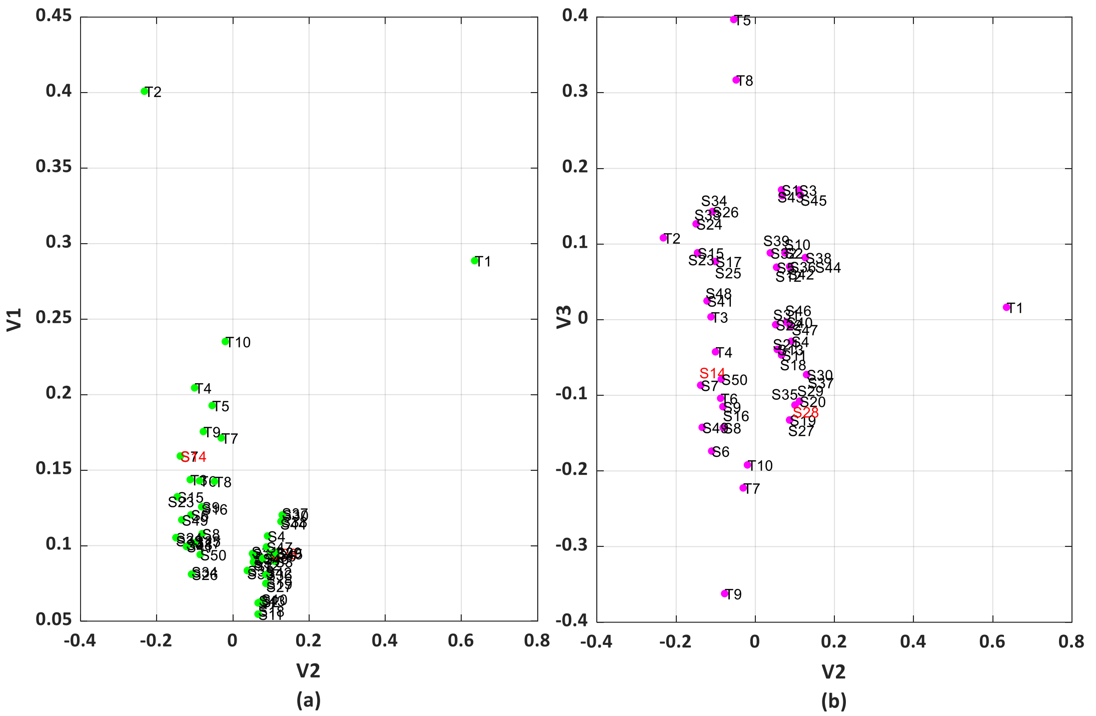


Figure 5.16: Plot of (a) Weighted 1st (V1) and 2nd (V2) PEVs, (b) Weighted 2nd (V2) and 3rd (V3) PEVs

Figure 5.16 (a), indicate that the satellites with the longest duration of view are found to be satellite S14 and S7 just as in the eigenvector centrality analysis. The two satellites, S14 and S7 however do not overfly target T3 and this is not evident either from Figure 5.15 nor from Figure 5.16. This shows that the hybrid method gives more insights related to the node optimality and the structural cohesion hence better-informed tasking decisions even for larger, complex networks. For the case of analysing a satellite-target network for maximisation of target coverage which requires structural cohesion analysis, the hybrid method developed gives clearer insights compared to using PEVs. When considering the maximisation of the total sum of the edge weights connected to a node, the eigenvector centrality and in turn the 1st principal eigenvector gives clear insights when considering the neighbours of the node.

5.5.3 Analysis to Find Optimum Satellites for Number of Times Seen Objective Function.

For the 50 satellite network, in terms of the number of times that the satellites view the targets, the centralities are calculated and the results for both eigenvector centrality and degree centrality are presented in Figure 5.17 and Figure 5.18 respectively. These results again show some similarities in the weighted degree and eigenvector graph shape. There are however some differences due to eigenvector centrality not only considering the number of times that a satellite views a target but also the neighbouring targets' number of times seen value.

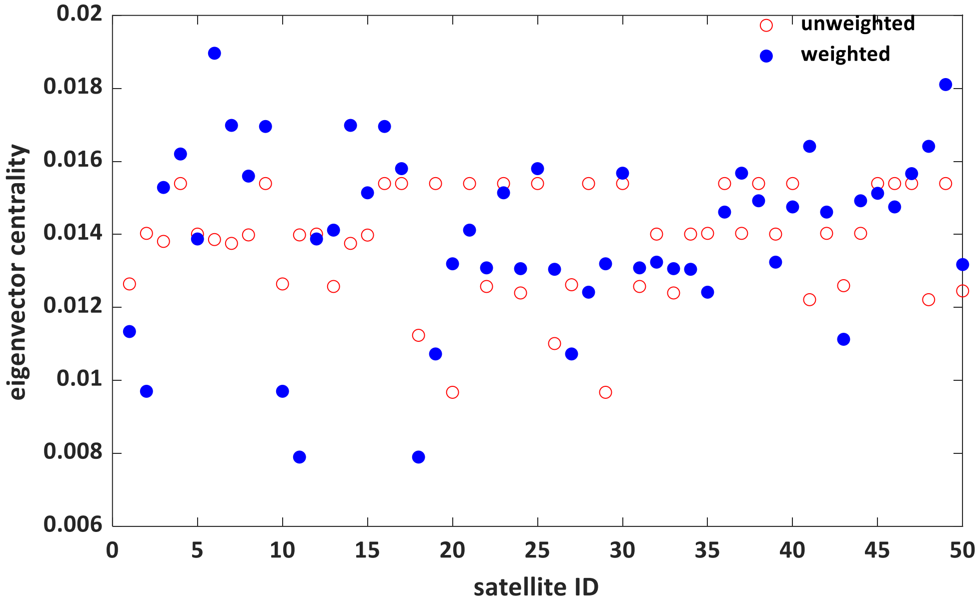


Figure 5.17: Graph of the weighted and unweighted Eigenvector centrality analysis of 50 satellite nodes based on the number of times that a satellite views all the targets.

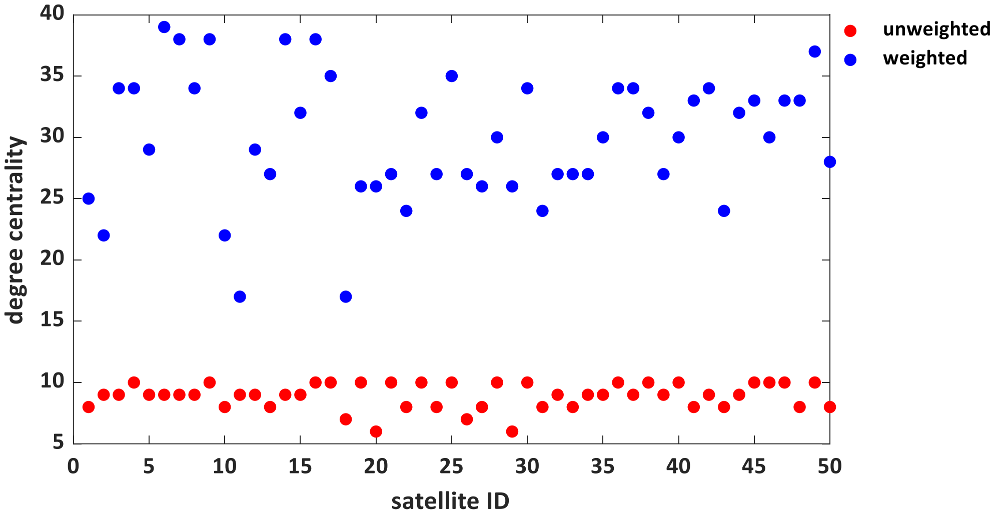


Figure 5.18: Comparison of the weighted and unweighted Degree centrality based on the number of times that a satellite overflies the target nodes.

For both the weighted eigenvector and the weighted degree centrality, satellite ID 6 is identified as the optimum satellite for number of times seen but does not overfly all the targets. For the unweighted values, as expected, the satellite nodes that maximise node degree and hence coverage have the highest values. As before, using the weighted eigenvector centrality proves to be a robust method to selecting satellite nodes that optimise on the tasks related to the node weights. Addition of the weighted degree centrality gives better and clearer insights as to whether the selected satellites individually optimise the task, or it must rely on the weight of the neighbouring nodes.

5.5.3.1 Satisficing Satellites selected

Depending on the given mission requirements, two tasks, for example the duration of view task and the minimum number of satellites required to maximise on coverage, can be modelled as multi-objective/ multi-tasking problem. An operator may need the minimum number of satellites that overfly all targets as well as optimise the duration of view. This requires a selection of satellites that achieve both missions in a near optimum way. In the presented network, Figure 5.12, page 125, the satisficing satellites for the duration of view and the full coverage tasks are identified to be a 2 – Satellite union, i.e., satellite node 14, which is in contact with 9 targets excluding target node 3 and satellite node 28 which is in contact with target node 3 but excludes target node 1. Satellite 28 has a higher weighted degree centrality than its eigenvector centrality and this is due to a contact with target node T3 which has a low weighted eigenvector centrality. The average normalised weighted eigenvector and weighted degree centrality scores of the two satellite nodes are given in Table 5.9. The orbits of these two satellite nodes which are selected to optimise on the two mission requirements are also presented in Table 5.9.

Table 5.9: Suggested satellite pair to maximize on duration of view task as well as view of all targets from the graph analysis.

Satellite	14	28	Tasking both satellites
Target nodes in contact	[T1, T2, T4, T5, T6, T7, T8, T9, T10]	[T2, T3, T4, T5, T6, T7, T8, T9, T10]	Full coverage
Weighted Eigenvector Centrality (normalised)	100	59.5454	79.77
Weighted Degree Centrality (normalised)	100	70.7865	85.39
Orbit Inclination (Deg)	56.00	50.00	
Orbit RAAN (Deg)	2.001	8.003	

The combined results from selecting and tasking satellite 14 and 28 is, full coverage of targets, average normalised weighted eigenvector centrality of 79.77 and average normalised weighted degree centrality of 85.39. In relation to the duration of view, tasking the union of satellite 14 and 28 ensures a total duration of view equating to approximately 1520 seconds within 2-days. This brings in the study of multiple task optimisations for the tasking of satellites using the proposed hybrid method and is studied further in section 5.6 where optimum multi-task analysis for satellite tasking is presented.

5.5.4 Application of the hybrid method to a scenario with more targets than satellites

Given a scenario with 202 targets and 135 satellites in orbit, the hybrid methodology is used to determine the minimum number of satellites that will give maximum total durations of view as well as achieve the maximum coverage of the targets in a span of two days. The 202 targets are presented on Figure 5.19.

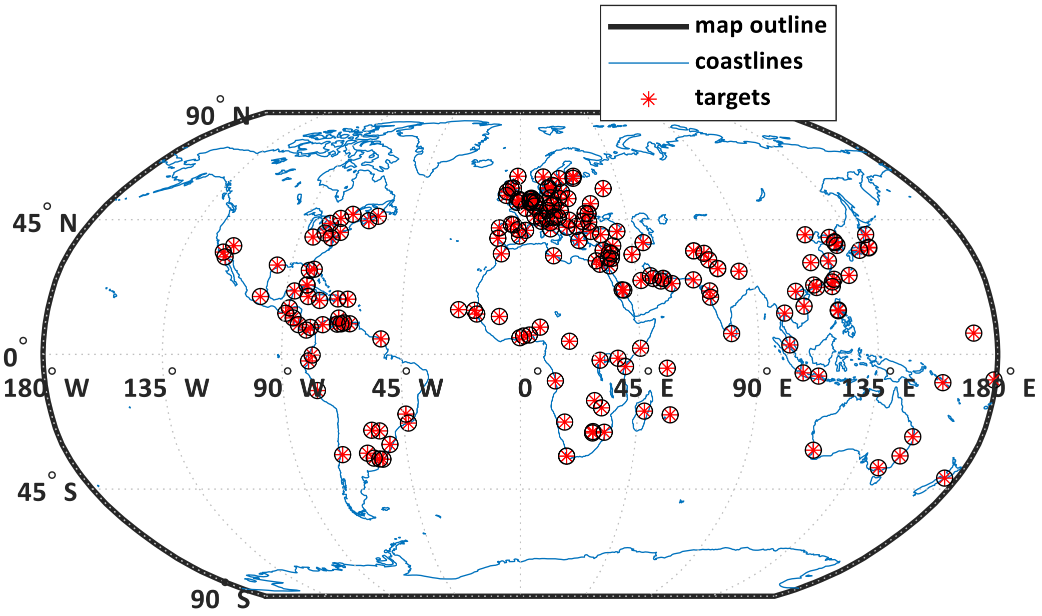


Figure 5.19: Map plot of the target points to be used for tasking satellites using the developed hybrid method

For the coverage task, when the hybrid method's, Algorithm 5.1 is applied to the 202 targets and 135 satellites graph, none of the satellites are found to individually overfly all the targets within the given time period of 2-days. 125 unions of 2 – satellites are however found to overfly all the targets.

When simulated for the duration of view task, two satellites overfly the targets with the maximum duration of view. These are satellites 5 and 48 of which overfly 135 and 160 of the 202 targets respectively. Together, the two satellites, 5 and 48 overfly 182 targets out of the 202. There is therefore no full coverage for the longest duration of view satellites.

From the hybrid method, to have full coverage and get the maximum possible duration of view while minimising the number of satellites, the optimum sets of 2-satellites are determined and presented in Table 5.10.

Table 5.10: Combined normalised values to get full coverage from sets of 2-satellites

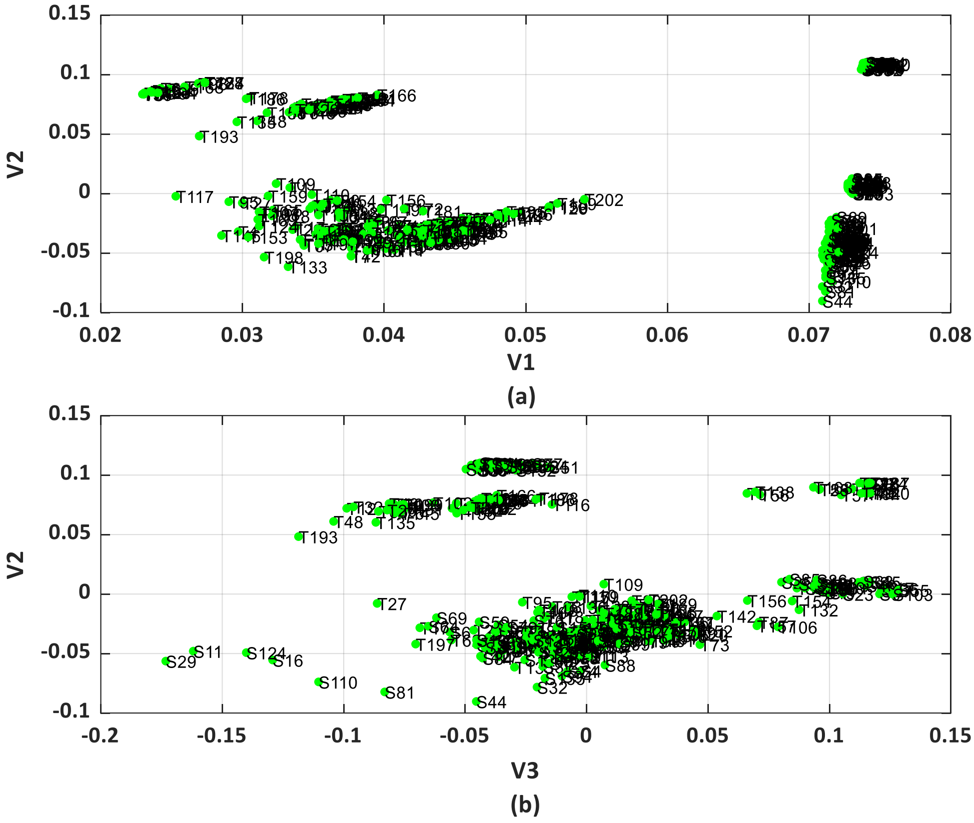
combined normalised duration of view	satellite 1	satellite 2
96.73	49	44
96.44	61	49
96.95	66	49
95.03	76	49
97.33	90	49
96.27	102	49

From Table 5.10., satellite 49 is a critical satellite node as it appears in all the satellite sets for full coverage. For satellite selection based on maximisation of the duration of view, and full coverage, satellite node set {90 49} is identified as the satisficing set from the hybrid method.

A directed graph of the 202 targets and 135 satellites in orbit is generated and the PEVs of the Laplacian are calculated. The Laplacian is used in this case as the directed graph has an asymmetric adjacency matrix. The Laplacian if the difference between the adjacency matrix and the out-degree diagonal matrix.

$$L = D - A$$

Where L is the Laplacian matrix, D is the out-degree diagonal, and A is the adjacency matrix. It is observed that the results and insights are not as clear as using the proposed hybrid method. The communities are more evident in terms of the 2nd and 3rd PEVs but in terms of the network's structural cohesion, the insights need further analysis to be conclusive. The results of the directed graph PEVs are as presented in Figure 5.20.



(a)

(b)

Figure 5.20: Principal Eigenvectors of the Laplacian of the directed graph of 202 targets and 135 satellites (a) V1 – 1st PEV and V2 – 2nd PEV (b) V3 – 3rd PEV and V2 – 2nd PEV

5.6 Case Study: Multi-Task Analysis Using Proposed Satellite Tasking Methodology

Once the optimum satellite nodes for individual tasks have been identified, the graph can be further queried and analysed to determine combinations of satellite nodes that, after considering trade-offs between optimal nodes for different tasks, yield *satisficing*, multiple-task performance. These combinations are referred to as *satisficing satellite unions*. This means that they satisfy multiple tasks in a near optimum way. Near optimum in this case means that when selecting satellites to be tasked, satellites that may not necessarily perform any of the tasks in an optimum way but perform all desired tasks in a near optimum way can be selected. For instance, Satellite A might excel in duration of view but underperform in the number of times seen. Conversely, Satellite C might have a high number of times seen but a low duration of view. A more balanced approach could involve Satellite B, which offers a compromise between these two metrics. Alternatively, a combination of Satellites A and C could be considered to optimize performance across both metrics. Different algorithms can be used to determine the satellites to task for multiple objectives. Y. Zhang et. al. [159], for example used a pareto-optimal search to determine communities in a bipartite graph while considering both structural cohesiveness and the weighting values. The proposed hybrid method can be used for tasking satellites that optimise both single and multiple tasks, and it shares some of the advantages of the previously used multi-objective function optimisation method presented in chapter 3 of this dissertation.

The results obtained from the single task optimisation formulate the starting point for the multi-task optimisation developed. For the duration of view and number of times seen tasks, both weighted eigenvector and degree centralities are calculated and normalised. The satellite nodes with the best performance in both tasks are identified and selected. The 50 satellite graph presented in Figure 5.12 page 125 is used to illustrate the application of the hybrid method. The eigenvector centralities of the adjacency matrix considering both duration of view tasks and number of times seen tasks are first calculated. The eigenvector centrality scores for each of the tasks are then normalised.

An analysis of the normalized values is conducted, involving a sensitivity analysis on task priorities. This involves adjusting the priorities assigned to the two tasks and recalculating the weighted sum of their normalized values. For instance, if the initial priorities are equal (1:1), the normalized values are averaged. By varying these priority values, different combinations of satellite nodes are evaluated to identify optimal solutions under various prioritization schemes. This means that if the normalised centrality for duration of view is 60 and the normalised centrality for the number of times seen is 80, the combined centrality will be (1/2 *60) +(1/2 *80) = 70. For the 2:3 priority ratio on the other hand, if the same centralities are applied, i.e., 60 and 80, the combined centrality will be (2/5 * 60) + (3/5*80) = 24+48 = 72.

The result for the analysis done using this change in priority of the tasks is presented in Figure 5.21 which shows how the optimum satellites related to the two tasks are determined. In Figure 5.21, the priorities of the two tasks are varied according to the ratios on the legend (Duration of view: Number of times Seen) and this shows that the total normalised eigenvector centralities related to the two tasks vary as the priorities change. Satellite 7 and 14 are the optimum satellites for most of the changing priorities but when the duration of view has a 20% priority compared to the number of times seen which has 80% however, the

6th satellite is the optimum for tasking (this is seen on the magenta line i.e., Duration of view: Number of times seen = 1:4).

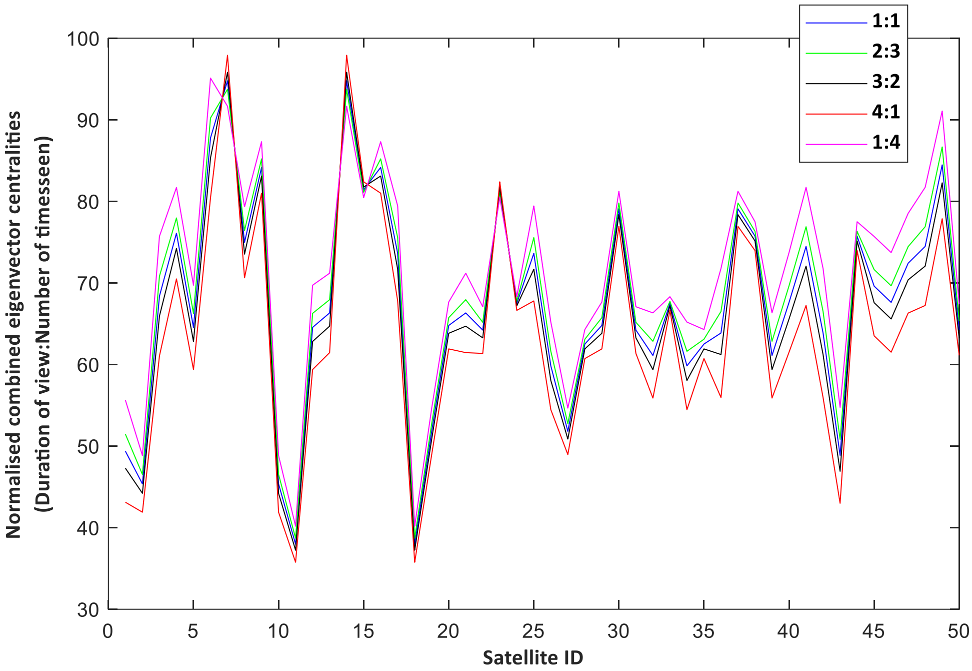


Figure 5.21: Optimisation of both duration of view objective function and mean number of times viewed when their priorities are different.

From these results, the satisficing multi-task satellites that solely depend on the weighting can be determined by identifying the satellite nodes with the highest combined centrality values in the network. These satellite orbits will ensure that multiple tasks are achieved and in a near-optimal way. As previously observed, using the eigenvector centralities alone may overlook some important insights on the satellite tasking especially in relation to the structural cohesion. A more efficient and informed method of satellite task selection is realised by the inclusion of Algorithm 5.1 and in this section, the degree centralities are also integrated into the proposed hybrid method. This newly developed proposed method is herein known as the Multi-Tasking Proposed method (MTPM) and is formulated as follows,

i. The graph is created after propagation of satellite orbit ground-tracks within a given field of regard,

ii. The single satellite node(s) or node sets that maximise on the node connectivity which in this case is represented by the target coverage are determined from the simulation of Algorithm 5.1,

iii. Each satellite and satellite set eigenvectors and degree centralities are then calculated and normalised,

iv. The best satellite node(s)/ node sets are then selected as those found to have highest weighted centrality values and together satisfy maximum coverage.

The full algorithms used for these calculations, satellite tasking selections and determination are presented in the appendix section F.

5.6.1 Multi-tasking of Satellites Using the MTPM

This work employs centrality measures, specifically degree and eigenvector centrality, to analyse satellite-target networks. However, traditional methods for identifying influential nodes in bipartite graphs while considering both node cohesion and the weighting values often lack formal justification for identification see Y. Zhang et. al. [159]. To address this limitation, a multi-tasking approach to identify optimal satellite nodes that simultaneously satisfy three tasks is proposed. The tasks are:

i. The least number of satellites needed to overfly all targets at least once,

ii. The satellite nodes that optimise the duration of view of the targets,

iii. The satellite nodes that optimise viewing the targets as many times as possible.

The algorithm proposed to identify the satisficing satellite combinations is a modified multi-task optimisation analysis which incorporates the normalised centrality results from each task. The summarised algorithm for this is presented in Algorithm 5.2.

Algorithm 5.2: Optimisation of the combination of satellites

```
$\boldsymbol{Inputs}$: $Normalised\ Objective\ function\ centralities\ (degree\ and\ eigenvector -$
$both\ weighted\ and\ unweighted)$
$\boldsymbol{Outputs}$: $Optimal\ satellites$
$Normalise\ all\ the\ centrality\ values$
$\boldsymbol{for}\ i = 1{:}\,numel(satellites)$
        $\boldsymbol{if}\ for\ all\ centralities\ (i) \geq 80$
                $fprintf\ (Satellite\ Id)$
                $Optim[\quad] = satellite\ (i)$
        $\boldsymbol{end}$
$\boldsymbol{end}$
Optimal satellites = Optim.
```

For the 50 satellites and 10 targets network, which is presented in Figure 5.12, page 125, Algorithm 5.2 is simulated. The results obtained from this simulation when the combined normalised centralities are kept at a minimum value of 80, i.e., $for\ all\ centralities\ (i) \geq 80$, the satisficing satellites are found to be satellite 7 and 14. These satellites however do not view all the targets as desired. They both do not overfly target 3.

Compared to the previous analysis using the weighted eigenvector centrality analysis, these are the same satellite nodes that were selected as the optimal satellites. When the brute-force neighbours' algorithm, Algorithm 5.1, is simulated, the satellite node that is appearing in most sets/unions is considered to be the node holding the structure of the network together, i.e., acting as a bridge and ensuring structural cohesion.

From the results of simulating Algorithm 5.2, if the satellite nodes with a minimum combined normalised values of 40 are considered, i.e., for all centralities $(i) \geq 40$, the simulation obtained 12 satellite nodes. When simulated for full coverage, unions of 2 satellite nodes were found to fulfil achieve full coverage. Amongst the 12 satellite nodes, 29 unions of 2-satellite nodes were found to be in contact with all the target nodes.

From the 29 unions of 2-satellites, centralities are calculated to get the optimum normalised combinations for the unions. This results in the satisficing unions of satellites being union number 21 and 29, and near-optimal unions are union number 5, 8, 9 and 12. The satellite IDs, satellite orbit Inclinations and satellite orbit right ascensions of union 21, 29, 12 and 5 are presented in Table 5.11.

Table 5.11: Optimal satellite pair data from the simulated orbit network

Satellite Pairs	1 (21)	2 (29)	3 (12)	4 (5)
Satellite 1 ID	49	49	37	30
Satellite 2 ID	14	7	7	14
Satellite 1 orbit Inclination (Deg)	56.00	56.00	52.00	52.00
Satellite 2 Orbit Inclination (Deg)	56.00	56.00	56.00	56.00
Satellite 1 orbit RAAN (Deg)	12.005	12.005	2.001	8.003
Satellite 2 Orbit RAAN (Deg)	2.01	2.00	8.004	2.01

The satisficing satellites obtained show that for the fulfilment of the tasks, satellites at inclinations of 56 degrees and 52 degrees achieve the mission requirements optimally. This work was done with a satellite orbit ground track propagation limited to an orbit range of 50-

60 degrees inclination with increments of 1 degree. With a more refined grid, these satisficing satellites would be more varied.

Nevertheless, the proposed method developed, MTPM, from a combination of different graph theory techniques and developed algorithms is seen to be a valid method for determining satellites to task. The proposal of tasking satellites using graph theory has been proven to be able to attain the desired objectives efficiently in both time and computational efficiency while considering tasks that can be weighted and those that do not depend on weight. The application of the method is extended to a wider range of targets and satellites making the network larger and more complex as will be presented in the sections following.

5.6.2 Multi – Tasking of Satellites which are in Orbits at Different Altitudes.

The satellite tasking results presented in this chapter have been based on satellites that have been on 2-day repeat ground track orbits. In this section, satellites in different orbital altitudes are considered for the analysis. The satellites are to overfly 10 desired target points. The satellite orbit ground tracks are propagated using the analytical method presented in chapter 2. The MTPM, is then used in this section to determine the satisficing satellites to task for the optimum performance of the duration of view and the number of times seen as well as obtain maximum target coverage.

The 10 targets used are presented in Figure 2.13, page 44. The satellite orbital space parameters are then presented in Table 5.12.

Table 5.12: Orbital Search Space parameters of where the satellites are located.

Parameter	Range	Units
Inclination	50-60	Deg
Right Ascension of Ascending Node	0-5	Deg
Altitude	300 – 800	km
Time (From Epoch)	0 – 172800	Sec

The proposed Multi-Tasking Problem Model (MTPM) is simulated using graph theory-based algorithms detailed in Appendix F. The results and steps of the simulations are as follows,

i. The first step takes the bulk of the work as it involves the creation of the graph. In this case, 1500 satellites in different altitudes have their ground-tracks propagated and the graphs of the satellites and the targets are generated using Algorithm F.1.

ii. From the graph, the minimum number of satellites needed to ensure full coverage, can be queried. This is done using Algorithm 5.1. In this case, the minimum number of satellites needed to overfly all the targets is found to be 1 satellite. This may be due to the small number of targets required to be overflown.

iii. The weighted and unweighted degree and eigenvector centrality of the graph for both tasks, of duration of view and number of times seen are then calculated and normalised using Algorithm F.3 and Algorithm F.5.

iv. From this normalisation, the combined sets/unions of satellites that have high normalisation values and ensure full coverage are determined using Algorithm F.6.

From this analysis, unions of 2-satellites that are determined to be optimum for the normalised centralities are selected. This is because none of the satellites that have full coverage optimise on both duration of view and number of times seen tasks. It is observed that there are no optimum or near optimum satellites at the lower altitudes (300-500) km. Table 5.13 presents the unions of satellites that are found to maximise on both tasks.

Table 5.13: Maximum normalised combined objective function satellite orbits (satisficing 2-satellite unions at different altitudes selected)

	Satellite 1 INC (Deg)	Satellite 2 INC (Deg)	Altitude (m)
Union 1	56.8	53	610000
Union 2	53.4	52.6	670000
Union 3	52.6	53.4	680000

From the simulation of 1500 satellites at altitudes between 300-800 km and inclinations between 50-60 degrees, 6 satellites are found to perform the 3 tasks in a near optimal way. Using the MTPM, the analysis proved to be efficient for giving insights to this large, complex network of 1500 satellite nodes and 10 target nodes. This shows that the proposed method of satellite tasking using graph theory analysis is valid for different network structures. To further validate this, a complex satellite network using an increased number of targets is presented in section 5.6.3.

5.6.3 Case Study 2: Complex Network Analysis for Satellite Tasking with Increased Number of Targets.

The MTPM is aimed at tasking of satellites that are already in orbit hence maximising on assets in space. The algorithms presented and developed for use in determining satellites to task is further validated when used in this section to determine and task over 3000 satellites overflying 202 targets that optimally satisfy given tasks. The 202 target locations to be overflown are presented in Figure 5.19, page 134.

In this section, the application of the MTPM is presented considering the determination of,

i. the minimum number of satellites needed to overfly all the targets in the span of 48 hours from epoch,

ii. best satellites to task that optimise the total duration of view and the mean number of times seen of the desired targets within the 48 hours i.e. multi-task optimisation considering both duration of view of the targets and number of times a satellite views the targets.

iii. satellites that view the most targets within different amounts of time after epoch, the first has been set to within 500 seconds of Epoch. This is to show that tasking of the satellites that can view targets within a given amount of time can be achieved using graph theory.

In this case, the satellite orbit range details are as presented in Table 5.14.

Table 5.14: Satellite orbital parameters used for propagation.

	Inclination (Deg)	RAAN (Deg)	Altitude (km)
Range	0-180 (10-degree increments)	0-360	300-700

For aim (i), approximately 3500 satellites are analysed using the first section of the method which uses Algorithm 5.1. From this analysis, the minimum number of satellites found to overfly all the targets at least once within 2-days from epoch is 2 satellites. Multiple unions of 2-satellites or more are found to achieve the task of full coverage within the same period.

For aim (ii), the weighted centralities are calculated and the satellites with the highest values are determined based on the individual task of the duration of view and the mean number of times they view the targets. The selected satellites are those found to have normalised centrality values above 90 for each of the individual quantifiable tasks. The satellites that maximise on the duration of view are presented in Figure 5.22 (b). The duration of view satellite orbits presented in Figure 5.22 (b) show that the satisficing satellites to possibly task for optimisation of the duration of view are more than 10 satellites and Figure 5.22 (a) gives similar insights for the number of times seen. Using all the satellites found to optimise the tasks in this case would mean tasking many satellites. Minimisation of the number of satellites to be used is hypothetically more economical. An analysis of the combined satellite nodes that satisfy both tasks is used to minimise the satellites to task. To optimise both tasks, 4 satellite nodes are found to have the normalised values greater than 90 for both tasks. These are as presented in Figure 5.22 (c) which shows the satellites that have near-optimum normalised centrality values related to both tasks and their orbital values are presented in Table 5.15.

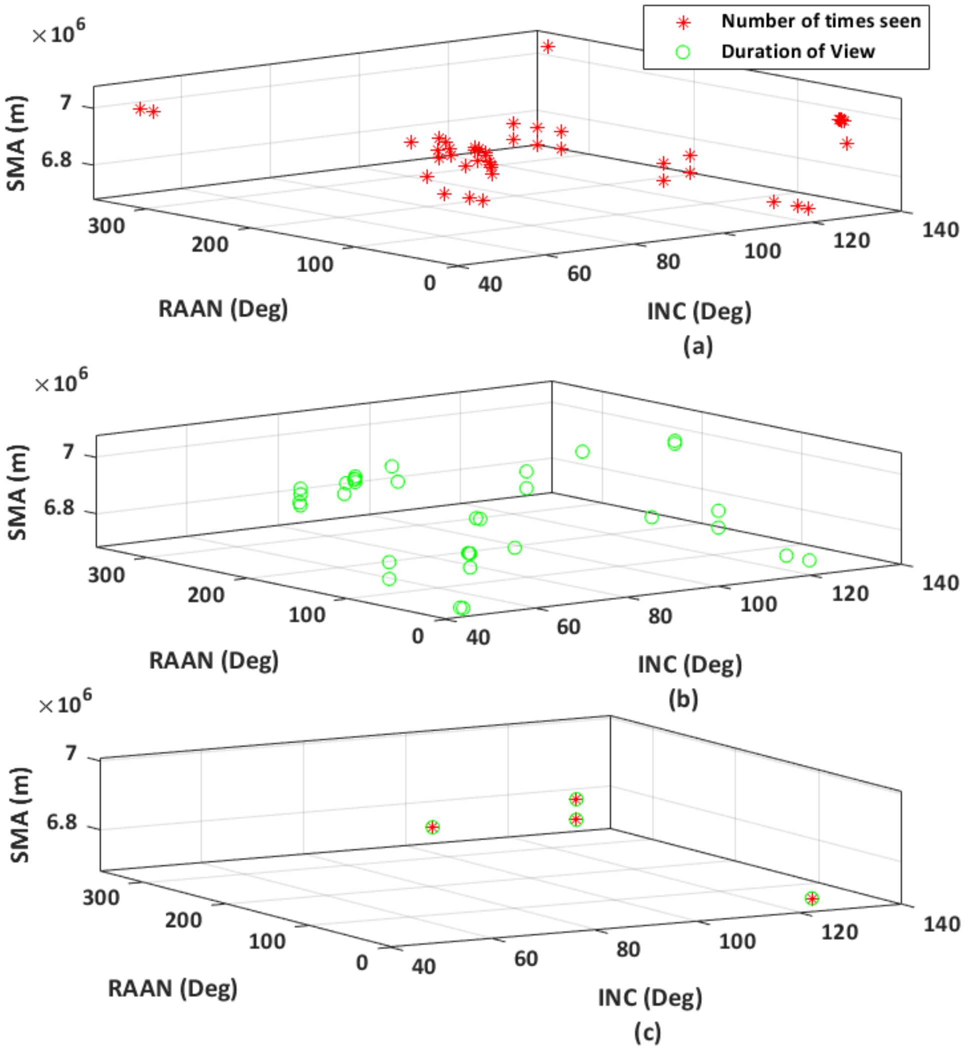


Figure 5.22: (a) Orbits of the number of times viewed satellites nodes with normalised values above 90. (b) Orbits of the duration of view satellites nodes with normalised values

above 90 (c)Satellite tasking based on two objective functions - number of times of view and duration of view.

Table 5.15: Satellites tasked for both duration of view and number of times seen optimisation.

	Inclination (Deg)	RAAN (Deg)	SMA (m)
Satellite 1 Orbit	130.00	330.032	6788000
Satellite 2 Orbit	130.00	330.032	6728000
Satellite 3 Orbit	130.00	47.0038	6678000
Satellite 4 Orbit	50.00	11.0002	7008000

These results show the application of the MTPM to task minimum number of satellites while ensuring full coverage of the targets. It also validates the use of the MTPM to task satellites based on multi-task optimisations using combined graph theory metrics, and particularly the centralities. The next section, section 5.6.4, validates the robustness of the method by developing it further by using it to task satellites that overfly the targets based on desired times from epoch.

5.6.4 Development of Satellites Tasking Algorithm Based on Time Elapsed from Epoch Time.

To further show the robustness of using graph theory to task satellites, the third (iii) task presented in section 5.6.3, page 143, where the time from epoch is to be considered is presented in this section. It is simulated using the graph obtained after propagation of the ground tracks within a given field of regard. The analysis for this is complex and selection is done using a modification of the previously given algorithms but with a change of task to consider the time from Epoch. The modified algorithm is presented as Algorithm 5.3.

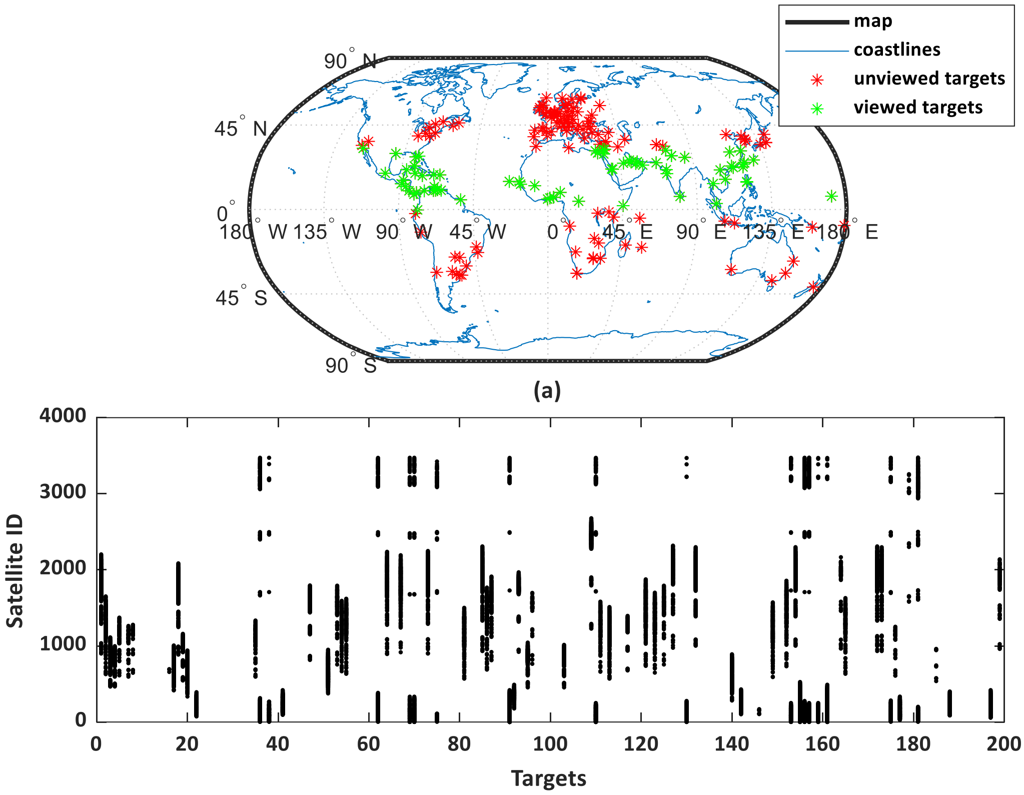
Algorithm 5.3 Tasking of satellites within a certain time from Epoch

```
input: propagated satellite-target graph
output: target visit times from epoch and satellite tasking
for each satellite and target, node edge weight (target visit time),
    if edge weight <= 500 (or preferred time from Epoch)
        Store satellite-target nodes and edge weight (time from epoch)
    End
```

```
End
store optimum satellite nodes to achieve desired time from epoch
```

Algorithm 5.3 determines the satellites that overfly various targets within a given epoch time. This can be useful for emergency cases and uses low computational time to identify the satellites that can be assigned for such a task. The method can also be used to task satellites to revisit targets after a specified minimum/maximum amounts of time. For example, if a target is overflown at 26000 seconds after epoch, a desire may be to overfly the target again within 4000 seconds, i.e., before 30000 seconds after epoch. Using Algorithm 5.3, satellites that satisfy such a task are determined.

Given 202 targets and multiple satellites in orbit, approximately 3500, a satellite-target graph is created after ground track propagation of the satellites in orbit. Using this graph, Algorithm 5.3 is simulated for 500 seconds after epoch. The results obtained are as presented in Figure 5.23. The results presented in Figure 5.23 (a) shows a world map visual of the targets which were found to be overflown after 500 seconds. Figure 5.23 (b) then shows the satellites that may be tasked to overfly those targets.



(a)

(b)

Figure 5.23: Plot of targets viewed and those not viewed within 500 seconds of epoch time with (a) showing the ground targets viewed and not viewed, (b) shows both the target ID and the satellites at different orbits that view the targets.

From Figure 5.23 (a), it is observed that less than half of the targets are viewed within 500 seconds of epoch. The results show that for the orbits that the 3500 satellites are on, only 78 targets are viewed within 500 seconds. 124 of the 202 targets are not viewed within this time from Epoch. The analysis also determines that 2707 of the 3500 satellites view at least 1 of the targets within 500 seconds whereas 793 of the satellites do not view even one target within 500 seconds.

From the results obtained, it can be observed that not all targets can be viewed within 500 seconds after epoch by the propagated satellite ground tracks. Some targets on the other hand can be viewed by multiple satellites within 500 seconds from epoch. Any of these 2707 satellites can therefore be tasked to collect data/images from the targets viewed in this period. They are all possible satellites to task for emergency situations in those target areas. To assign fewer satellites for this duty however, further optimisation depending on the other tasks may be done.

By modifying Algorithm 5.3 and simulating the 3500 satellite and 202 target graph, satellites that can overfly the targets a specified number of times within a given time from epoch can also be determined. Considering a case where tasking of satellites that overfly any of the targets at least twice within a given period is desired for example, the developed method may be used. In this case, if a satellite overflies any of the targets twice in a span of less than or equal to 7 hours, the satellite is considered for tasking. Algorithm 5.3 is modified and simulated for this case and the results from this are presented in Figure 5.24 which shows the satellites that can be tasked as well as the targets that those satellites overfly. From Figure 5.24 (a), it is observed that not all targets can be viewed twice within the selected time span by the satellites. Also, Figure 5.24 (b) shows that some satellites will view multiple targets twice within the span of 7 hours after the first view.

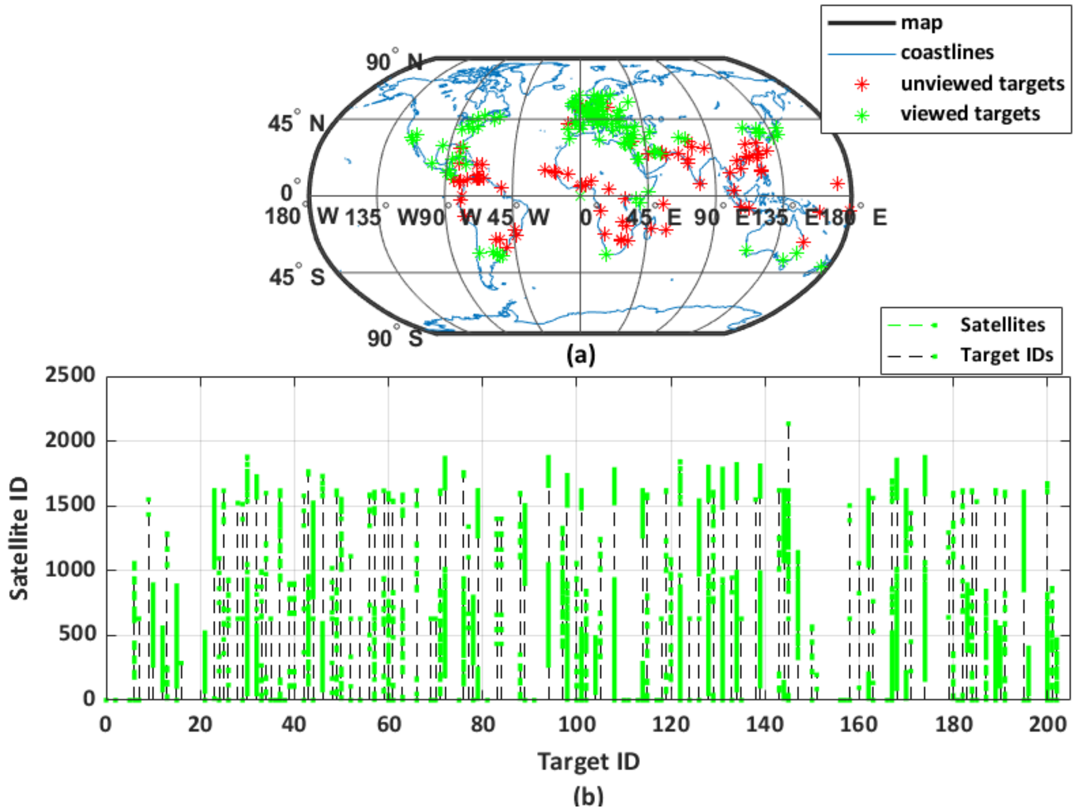


Figure 5.24: Views within 7 hours of epoch, (a) shows the targets viewed and (b) shows the satellites that can be tasked to view targets twice within 7 hours of epoch and the targets viewed.

The results presented in Figure 5.24, reveal that 790 satellites can overfly 121 targets out of the 202 targets twice in a span of 7 hours (25200 seconds). 101 targets were however not overflown twice in 7-hours, though some satellites are found to overfly more than one target twice. The number of targets viewed by each of the satellites is presented in Figure 5.25 and the orbital values of the satellites that view most targets within the desired span are presented in Table 5.16. Three satellites are observed to view the most targets.

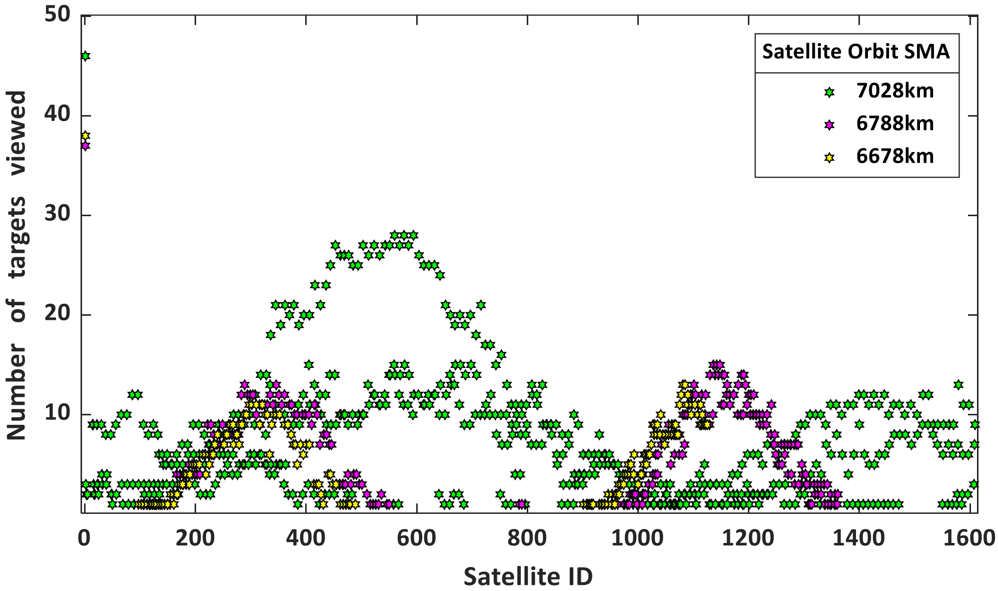


Figure 5.25: Number of targets that each satellite views at least twice within a span of 7 hours.

Table 5.16: Satellite Orbits that view most targets twice in the 7-hour period.

Number of targets viewed	Satellite Obit INC (Deg)	Satellite Orbit RAAN (Deg)	Satellite Orbit SMA (km)
46	40.00	0.00	7028
37	50.00	10.00	6788
38	60.00	0.00	6678

Some targets have more than one satellite viewing them multiple times and this data is presented in Figure 5.26 and Table 5.17.

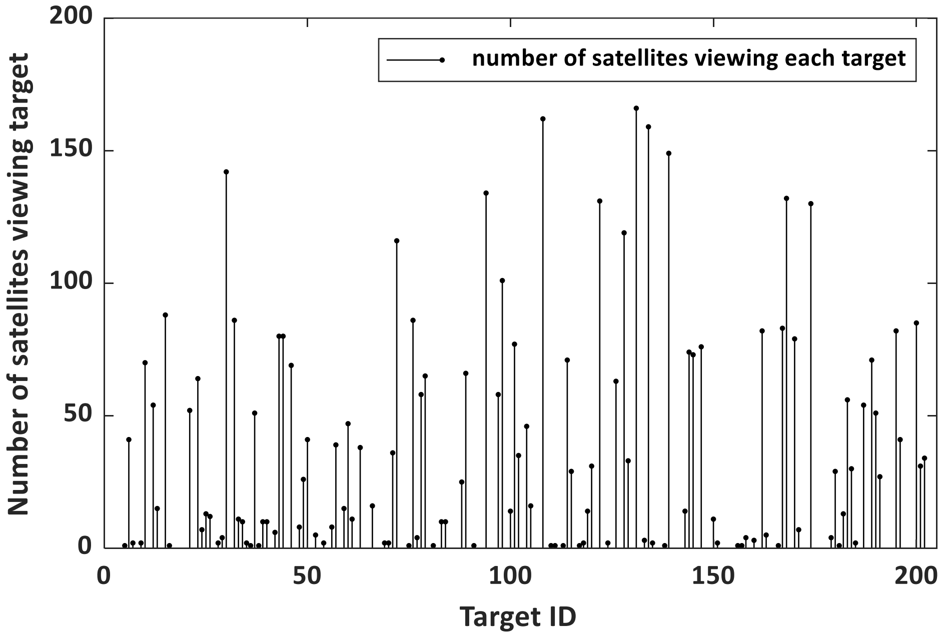


Figure 5.26: Numbers of satellites that view each target twice in the 7-hour period analysed.

The targets that have most satellites to choose from for the 7-hour revisit tasking are as presented in Table 5.17. The targets are between latitude +/- 30 degrees and 50 degrees. This may justify why the optimum satellites are also between inclinations of 40 degrees and 60 degrees.

Table 5.17: Targets that can be viewed within a 7-hour period by most satellites.

Target description	Latitude (Deg)	Longitude (Deg)
'Luxembourg'	49.61	6.13
'Melbourne'	-37.81	144.96
'Paris'	48.85	2.35
'Plzen'	49.74	13.37

From the 500 seconds after epoch analysis and the 7-hour revisit, more insights that can be drawn using the method results are such as, the greater the time needed before the first visit, the less the number of satellites needed to ensure full coverage. This can be seen as even with 3500 satellites; not all targets are overflown within 500 seconds.

To further show the insights that can be drawn from the results obtained using this method, 4 scenarios are analysed from the results obtained from Algorithm 5.3. These are:

i. The percentage coverage of the targets within a span of 500 seconds.

ii. The percentage coverage of the targets using the same number of satellites analysed at 1000 seconds after epoch, and a span of 2000 seconds.

iii. The percentage coverage when the number of satellites is increased for 500, 1000 and 2000 seconds.

iv. The minimum amount of time needed for 100% coverage and with how many satellites.

The result for this analysis is presented in Figure 5.27. This shows the number of satellites and the target coverage percentage when considering 500, 1000, 2000 and 5000 seconds after epoch.

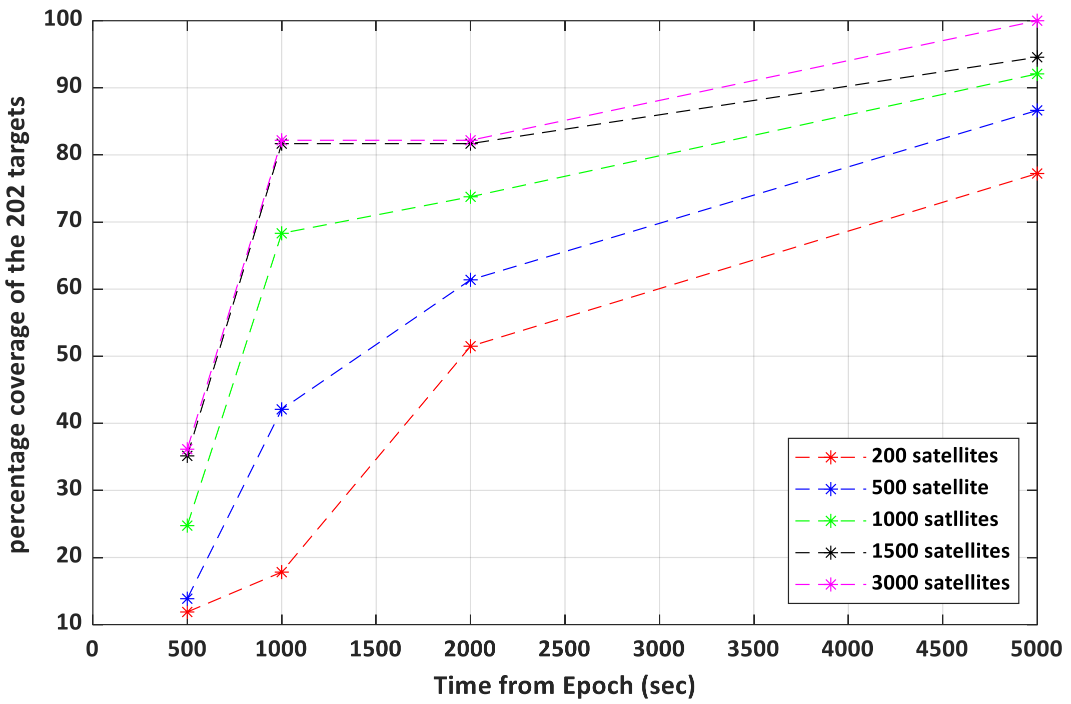


Figure 5.27: Target coverage after 500 seconds, 1000 seconds, 2000 seconds, and 5000 seconds.

From Figure *5.27*, not all the targets can be viewed within 2000 seconds after epoch with the satellites that have been propagated. There is however 35% coverage after 500 seconds, and approximately 83% coverage after both 1000 seconds and 2000 seconds.

For full coverage, within 500 seconds after epoch, the propagated satellites showed that more than 3000 satellites are needed. From Figure *5.27*, also, it is observed that for 5000 seconds after epoch, the minimum number of satellites needed to be tasked for full coverage are approximately 3000 satellites.

From the analysis carried out in this section, satellite tasking by modelling satellites and targets as nodes then using the ground propagation data as edges between the nodes hence creating graphs, then applying various graph theory techniques to gain specific task insights is seen to be valid and applicable to single and multiple tasks. The study in this chapter proposes the use of the developed brute force neighbours' algorithm as the method to give structural insights on the graphs related to satellite node connections and hence target coverage. The algorithm is integrated with the analysis of the weighted and unweighted adjacency matrices of the graphs to increase efficiency of the method in terms of insights. These matrices are used to calculate the degree and eigenvector centralities to determine satellites that optimise on quantifiable tasks. This is seen to give insights to which satellites to task based on multiple tasks or single tasks, as well as simple and complex networks. The tasking methods proposed prove to be robust, and both time and computationally efficient. The method algorithms have also been seen to be easily edited for different tasks as per the task relating to number of overflights within a given period. From the results obtained from simulating the developed algorithms which are fully presented in Appendix F, and hence the MTPM, insights drawn are seen to be clear concise and efficient for identifying the satellites to task.

5.7 Chapter Summary

This chapter introduces and develops a non-rigid method for storing data using graph databases then applying graph theoretical methods on the graphs to obtain required information. The novelty lies in having a graph database of satellites and the targets that they overfly. The information related to the overflights is stored on the nodes of the graph making it more robust and non-rigid in terms of intuitive data modelling and complex relationship analysis. The satellite tasking using graph theory offers greater flexibility and scalability especially for large numbers of satellites and targets as has been presented in section 5.5. Different scenarios have been presented as case studies where the graphs that the method has been applied to have been simple and small, as well as large and complex, proving the robustness of the method.

Chapter 6

Conclusions

This section gives a summary of the key findings, answers to the research questions and contributions of the research presented in this dissertation. It outlines the conclusions drawn from the conducted studies based on the gaps identified in literature, summarised as research questions and issues in section **Error! Reference source not found.** of this d issertation.

6.1 Research Summary:

The introduction of this dissertation identified some gaps in literature and the limitations of some traditional orbit design, constellation determination and satellite tasking methodologies. This research proposed to fill in some of these gaps as a contribution to knowledge. One gap that this study aimed to fill was, the need to develop new, fast, and robust satellite orbit design and determination methodologies that are based on localised target overflight. The work herein also aimed to develop an orbit design method that would enhance the efficiency of multi-objective function optimisations to determine satellite orbits. The research developed a target based analytical orbit design methodology that gives fast and effective orbit insights that can be useful to orbit designers. Additionally, an adaptive grid based on desired objective functions was proposed to increase the efficiency of the developed analytical orbit design method, and the search for more optimum objective function value orbits by reducing the computation time.

A novel methodology, grounded in graph theory was presented to optimize satellite task allocation. This approach aimed to identify optimal satellite configurations that balance multiple tasks, such as maximizing observation duration and minimizing resource utilization. By considering both the structural properties of a satellite-target network and the relative importance of different tasks, this method offered a more comprehensive and efficient solution to satellite tasking. This method considered both the structural cohesion of the graph and the weighting, which is the numerical value related to either single or multiple tasks.

6.2 Conclusions of the study

The key findings and contributions of the research presented in this dissertation include:

a. Development of a fast fully analytical, so-called *embarrassingly parallel* orbit design method based on desired ground target areas which included the first zonal harmonic of the central body perturbations, J_2 was done in section 2. As the end of the decade of action is fast approaching, the use of satellites to monitor and obtain informed decisions especially in relation to SDGs has significantly increased. One of the goals is to ensure satellite coverage in areas that do not currently have coverage which include remote and rural areas. Satellites are used to supplement information used by emergency services during evacuations, monitoring natural disasters and monitoring general development amongst many other uses. A gap of a fast and robust target-based orbit design method was identified in literature for such missions.

   The developed method incorporated the field of view of the instrument on board (see section 2.2.1.2) and used an *embarrassingly parallel* analytical approach (see section 2.2.2) which increased the robustness of the method especially when optimising multiple objective functions (see section 3.3).

   The orbits determined from the method were optimised using mission requirements, referred to as objective functions. The proposed methodology shifted from the traditional patterns previously used to design satellite orbits, determine constellations, and optimise different objective functions. The method proved to be fast and gave results within reasonable accuracy. To verify the accuracy of the method, analysis of the determined optimum orbits was done against propagation from a third-party software in section 2.3.3.

   Numerical methods have been proven to be more accurate in orbit determination than analytical methods. However, these accuracies come at the addition of computational time and costs in general. When the results of orbit design and optimisation using the developed method and a previously used improved differential evolution method were compared, the developed method results gave orbit solutions that had greater values of the objective function hence global optimum results (see section 3.1).

b. Development of a computationally efficient method that addresses the scalability challenges associated with large search space analysis, facilitating the acquisition of relevant insights. For large search spaces, previously used numerical methods take a long time to arrive at insights such as the ones that have been shown in this dissertation (see section 2.3 for insights and section 3.1 for comparison to previously used numerical methods). The orbit design method developed in this dissertation proved to be fast, robust, and can propagate large search spaces. From the analysis simulated throughout section 2 to 4, the developed method gave a range of useful insights on the various orbit search spaces which included both prograde and retrograde orbits. This was an added advantage as it gave a variety of orbit options to optimise hence addressed the gap related to optimisation of large search spaces.

c. Due to the use of an *embarrassingly parallel* approach, the developed method results allowed for multi-objective function optimisations without a need to re-run the orbital simulations (see section 3.3). For the same search space parameters, targets and physical constants, the method was found to be efficient and robust for multiple objective function optimisations. For some previously used methods, when there is a change in objective functions the full simulations must be re-run. As presented in section 3.3, the method was used to obtain satisficing orbits based on importance of the objective functions. This addressed the gap identified in multi-objective optimisation of satellite orbits.

d. An improvement of the developed analytical method's efficiency by adding a modified multi-level local refinement adaptive grid to the method was presented in section 4. Identifying the critical regions of interest, the addition of the adaptive grid reduced the computation burden of the developed orbit design method. This showed that the method can be implemented with other algorithms and the contribution in this case, was the efficient identification of the critical search space areas, hence reducing computational complexities. When the grid was added to the method, compared to finding the optimum orbits with a refined grid in the initial simulation, the computational time was reduced by almost a third (for this work) and more optimum orbits were determined (see section 4.3.1).

e. Development of a novel graph-based method for satellite tasking. Using a satellite-target graph database and graph theoretical analysis, multi-objective function satellite tasking was achieved and presented in section 5. The method was aimed at tasking pre-existing space populations, leading to cost reduction and space sustainability. The combination of different graph theoretical techniques led to the development of a hybrid method for satellite tasking. The techniques focused on the network's structural cohesion and the node weight analysis (see sections 5.4.2, 5.4.3, 5.4.3.1). The structural cohesion was defined as the identification of the minimum number of satellite nodes needed to be removed to disconnect a target node from the network. This was in relation to the target coverage as minimising the number of satellites to be tasked to get maximum coverage was considered a mission objective. The node weight analysis was focused on tasks that could be assigned a numerical value such as the observation period of the targets. The weighted adjacency matrices were then used to identify the nodes that maximise on the various weight values. The developed hybrid method proved to be insightful for scalable analysis of options that also involved multiple ground targets and satellite options (see 5.6.3).

   The developed method analysed the satellite-target graphs and identified the optimum satellites based on either single or multiple mission requirements, here in referred to as tasks. This method can be implemented for both newly designed satellite orbits, and pre-existing satellites in orbit. The studies presented in chapter 5 demonstrated that satellite can be tasked from an analysis of propagated orbit ground tracks within a given field of regard and targets using graph theory. The use of graph theory for satellite tasking was found to be insightful, computationally efficient and robust. The developed method can be automated using the algorithms presented in Appendix F. Such a system would be a great addition to the space sector as operators may save on time used to task their satellites.

   To validate the developed hybrid method, an option of using the principal eigenvectors of a graph to determine critical nodes was proposed (see 5.4.4, 5.4.5, 5.5.2.1). An analysis encompassing the first, second, and third principal eigenvectors (PEVs) was conducted. However, this analysis failed to provide the required insights, specifically, for determining graph structural cohesion. Conversely, the developed hybrid method demonstrated high computational efficiency and robustness in multi-task satellite determination and tasking, yielding clear and concise analytical results.

The developed hybrid method proved to be applicable to diverse mission requirements (see section 5.6).

Chapter 7

Future work

For future work, an increase of accuracy of the developed method would be a valuable addition; this can be done by including the effect of perturbations such as the atmospheric drag to the method. In the analysis of the analytical orbit design method, it was highlighted that including the secular Earth perturbations to the calculations made it more accurate. The consideration of solar radiation pressure and atmospheric drag should also be further analysed. In this dissertation, an adaptive grid was included to increase the efficiency of the method. However, an added numerical analysis on the higher objective function value regions may increase the accuracy even further. Therefore, the proposal is to use the method alongside a well-developed numerical algorithm for efficient computation and increased accuracy of the results.

The prevalence of prograde orbital trajectories in space missions is largely attributable to the inherent launch complexities associated with achieving retrograde orbits. However, this research advocates for an increased use of retrograde orbits. In the context of escalating concerns regarding space sustainability, retrograde orbits present a viable supplementary strategy for active debris removal (ADR) operations. Furthermore, they offer a potential solution for mitigating congestion within heavily populated prograde low Earth orbit (LEO) satellite constellations. For coverage of some ground targets, retrograde orbits proved to be more optimal than prograde ones. Chapter 3 demonstrated that retrograde orbits exhibit improved performance for both target observation duration and revisit frequency objective functions. Retrograde orbits, when strategically implemented minimise collision risk due to lower population densities of satellites in these trajectories. This should however be studied with caution on collision possibilities. This work therefore proposes further investigation into the strategic use of retrograde orbits for enhanced satellite mission objective function optimisation with collision avoidance by more research on space situation awareness (SSA).

The satellite tasking method may be further developed to include dynamic networks. In this work, the developed hybrid method is mainly studied for static satellite-target networks. A proposal to make the algorithms publicly available via html front-end is also given. This means putting the algorithms into an application so that users may get access i.e. creating a web interface (using HTML) that allows anyone to interact with and use the algorithms through a standard web browser. This will be a valuable tool that can be used for both single and multiple tasking of pre-existing space populations.

7.1 Closing Remarks

This dissertation advances obit design methodologies through the development of a novel, analytical-based and embarrassingly parallel algorithm, enabling rapid solution derivation. This innovation contributes to the expansion of analytical methodologies for orbit design and determination providing a significant contribution to the field of orbit mechanics. The method has proven to be robust in giving useful engineering and design insights related to the accurate overflight of targets. Additionally, the method can be used to give fast and initial insights to large satellite orbit search spaces and efficient multi-objective function optimisations. The addition of an adaptive grid enhances computational time efficiency of the developed method and subsequently increases the overall efficiency.

Another primary contribution of the work in this dissertation is the development of a novel graph-theoretic approach for adaptive satellite tasking, which enables the reassignment of on-orbit satellites to accommodate revised mission parameters or user-driven objectives. This methodology contributes to enhanced space sustainability through improved asset use and fulfilment of novel operational demands by optimising the satellite tasking. With the growing number of new satellite users, the developed satellite tasking methodology can also be an asset used together with the developed orbit design method to redesign on-orbit satellite trajectories and maximise on their usage. This may aid in taking a step forward towards solving the Sustainable Development Goals (SDG) using satellite technology. Therefore, the presented methods are a contribution to knowledge and if adapted, will be valuable to achieving the goals of the decade of action.

The methods in this dissertation are an addition to knowledge in disparate ways but may be inter-linked and used by orbit designers at different stages of satellite orbit mission designs.

References

[1] Z. Song, X. Chen, X. Luo, M. Wang, and G. Dai, "Multi-objective optimization of agile satellite orbit design," *Advances in Space Research,* vol. 62, no. 11, pp. 3053-3064, 2018, doi: 10.1016/j.asr.2018.08.037.

[2] M. E. Avendaño, J. J. Davis, and D. Mortari, "The 2-D lattice theory of Flower Constellations," *Celestial Mechanics and Dynamical Astronomy,* vol. 116, no. 4, pp. 325-337, 2013, doi: 10.1007/s10569-013-9493-8.

[3] J. Chen, "Research on Enhanced Orbit Prediction Techniques Utilizing Multiple Sets of Two-Line Element," *Aerospace,* vol. 10, p. 532, 06/03 2023, doi: 10.3390/aerospace10060532.

[4] A. Liu, X. Xu, Y. Xiong, and S. Yu, "Maneuver strategies of Starlink satellite based on SpaceX-released ephemeris," *Advances in Space Research,* vol. 74, no. 7, pp. 3157-3169, 2024/10/01/ 2024, doi: https://doi.org/10.1016/j.asr.2024.06.038.

[5] Y. A. Moiseev and N. V. Emelyanov, "Ephemeris Theories JPL DE, INPOP, and EPM," *Astronomy Reports,* vol. 68, no. 11, pp. 1098-1118, 2024/11/01 2024, doi: 10.1134/S1063772924700938.

[6] M. Pu, J. Wang, D. Zhang, Q. Jia, and X. Shao, "Optimal small satellite orbit design based on robust multi-objective optimization method," *Aerospace Science and Technology,* vol. 70, pp. 339-350, 2017/11/01/ 2017, doi: https://doi.org/10.1016/j.ast.2017.08.016.

[7] J. Sang, B. Li, J. Chen, P. Zhang, and J. Ning, "Analytical representations of precise orbit predictions for Earth orbiting space objects," *Advances in Space Research,* vol. 59, no. 2, pp. 698-714, 2017, doi: 10.1016/j.asr.2016.10.031.

[8] B. Tosun, "Computation of Satellite Orbits by Using Numerical Methods," 05/25 2020. [Online]. Available: https://www.researchgate.net/publication/345378873_Computation_of_Satellite_Orbits_by_Using_Numerical_Methods.

[9] O. Abdelkhalik, "Initial Orbit Design from Ground Track Points," *Journal of Spacecraft and Rockets,* vol. 47, no. 1, pp. 202-205, 2010, doi: 10.2514/1.44857.

[10] H. Guo *et al.*, "Measuring and evaluating SDG indicators with Big Earth Data," *Science Bulletin,* vol. 67, no. 17, pp. 1792-1801, 2022/09/15/ 2022, doi: https://doi.org/10.1016/j.scib.2022.07.015.

[11] H. Guo, "Big Earth data facilitates sustainable development goals," *Big Earth Data,* vol. 4, no. 1, pp. 1-2, 2020/01/02 2020, doi: 10.1080/20964471.2020.1730568.

[12] I. Ruiz-Martínez and J. Esparcia, "Internet Access in Rural Areas: Brake or Stimulus as Post-Covid-19 Opportunity?," *Sustainability,* vol. 12, no. 22, p. 9619, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/22/9619.

[13] N. Crisp, S. Livadiotti, and P. Roberts, "A Semi-Analytical Method for Calculating Revisit Time for Satellite Constellations with Discontinuous Coverage," 2018, doi: 10.48550/arXiv.1807.02021.

[14] S. J. Morgan, C. N. McGrath, and O. L. de Weck, "Optimization of Multispacecraft Maneuvers for Mobile Target Tracking from Low Earth Orbit," *Journal of Spacecraft and Rockets,* vol. 60, no. 2, pp. 581-590, 2023, doi: 10.2514/1.A35457.

[15] X. He, H. Li, L. Yang, and J. Zhao, "Reconfigurable Satellite Constellation Design for Disaster Monitoring Using Physical Programming," *International Journal of Aerospace Engineering,* vol. 2020, pp. 1-15, 2020, doi: 10.1155/2020/8813685.

[16] H. Sun, W. Xia, Z. Wang, and X. Hu, "Agile Earth Observation Satellite Scheduling Algorithm for Emergency Tasks Based on Multiple Strategies," *Journal of Systems Science and Systems Engineering,* vol. 30, no. 5, pp. 626-646, 2021/10/01 2021, doi: 10.1007/s11518-021-5506-4.

[17] R. Sandau, K. Brieß, and M. D'Errico, "Small satellites for global coverage: Potential and limits," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 65, no. 6, pp. 492-504, 2010/11/01/ 2010,doi:https://doi.org/10.1016/j.isprsjprs.2010.09.003.

[18] D. Paikowsky, "What Is New Space? The Changing Ecosystem of Global Space Activity," *New Space,* vol. 20, 03/03 2017, doi: 10.1089/space.2016.0027.

[19] A. Golkar and I. Lluch i Cruz, "The Federated Satellite Systems paradigm: Concept and business case evaluation," *Acta Astronautica,* vol. 111, pp. 230-248, 2015/06/01/ 2015, doi: https://doi.org/10.1016/j.actaastro.2015.02.009.

[20] J. L. Stern and P. T. Grogan, "Federated Space Systems' Trade-Space Exploration for Strategic Robustness," *Journal of Spacecraft and Rockets,* vol. 59, no. 4, pp. 1240-1254, 2022, doi: 10.2514/1.a35103.

[21] C. Circi, E. Ortore, and F. Bunkheila, "Satellite constellations in sliding ground track orbits," *Aerospace Science and Technology,* vol. 39, pp. 395-402, 2014, doi: 10.1016/j.ast.2014.04.010.

[22] T. Li, J. Xiang, Z. Wang, and Y. Zhang, "Circular revisit orbits design for responsive mission over a single target," *Acta Astronautica,* vol. 127, pp. 219-225, 2016/10/01/ 2016, doi: https://doi.org/10.1016/j.actaastro.2016.05.037.

[23] T. M. Hackett, S. G. Bilén, D. J. Bell, and M. W. Lo, "Geometric Approach for Analytical Approximations of Satellite Coverage Statistics," *Journal of Spacecraft and Rockets,* vol. 56, no. 5, pp. 1286-1299, 2019, doi: 10.2514/1.A34267.

[24] B. A. Conway, "A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems," *Journal of Optimization Theory and Applications,* vol. 152, no. 2, pp. 271-306, 2012/02/01 2012, doi: 10.1007/s10957-011-9918-z.

[25] C. N. McGrath and M. Macdonald, "General Perturbation Method for Satellite Constellation Reconfiguration Using Low-Thrust Maneuvers," *Journal of Guidance, Control, and Dynamics,* vol. 42, no. 8, pp. 1676-1692, 2019, doi: 10.2514/1.G003739.

[26] M. H. Lane and F. R. Hoots, "General perturbations theories derived from the 1965 lane drag theory," AEROSPACE DEFENSE COMMAND PETERSON AFB CO OFFICE OF ASTRODYNAMICS, 1979.

[27] J. Cheng, G. G. Yen, and G. Zhang, "A grid-based adaptive multi-objective differential evolution algorithm," *Information Sciences,* vol. 367-368, pp. 890-908, 2016, doi: 10.1016/j.ins.2016.07.009.

[28] J. J. Pocha, "Orbit Propagation," in *An Introduction to Mission Design for Geostationary Satellites*, J. J. Pocha Ed. Dordrecht: Springer Netherlands, 1987, pp. 138-163.

[29] R. Flores, B. M. Burhani, and E. Fantino, "A method for accurate and efficient propagation of satellite orbits: A case study for a Molniya orbit," *Alexandria Engineering Journal,* vol. 60, no. 2, pp. 2661-2676, 2021/04/01/ 2021, doi: https://doi.org/10.1016/j.aej.2020.12.056.

[30] D. Brouwer, "Solution of the problem of artificial satellite theory without drag," *Astronomical Journal,,* vol. Vol. 64, , p. p. 378 (1959) 1959, doi: 10.1086/107958.

[31] Y. Kozai, "The motion of a close earth satellite," *The Astronomical Journal,* vol. 64, p. 367, November 01, 1959 1959, doi: 10.1086/107957.

[32] A. Rocha, "Numerical Methods and Tolerance Analysis for Orbit Propagation," San José State University, 2018.

[33] D. A. Vallado, *Fundamentals of astrodynamics and applications*. Springer Science & Business Media, 2001.

[34] V. Morand, J. Dolado, H. Fraysse, F. Deleflie, J. Daquin, and C. Dental, "Semi-Analytical Computation of Partial Derivatives and Transition Matrix Using STELA Software," in *6th European Conference on Space Debris*, L. Ouwehand, Ed., 2013, vol. 6, no. 1: ESA, p. 8. [Online]. Available: https://conference.sdo.esoc.esa.int/proceedings/sdc6/paper/138. [Online]. Available: https://conference.sdo.esoc.esa.int/proceedings/sdc6/paper/138

[35] F. R. Hoots and R. L. Roehrich, "Models for propagation of NORAD element sets," Aerospace Defense Command Peterson AFB CO Office of Astrodynamics, 1980.

[36] X. He and H. Li, "Analytical solutions for Earth discontinuous coverage of satellite constellation with repeating ground tracks," *Chinese Journal of Aeronautics,* 2021, doi: 10.1016/j.cja.2021.11.012.

[37] T. Savitri, Y. Kim, S. Jo, and H. Bang, "Satellite Constellation Orbit Design Optimization with Combined Genetic Algorithm and Semianalytical Approach," *International Journal of Aerospace Engineering,* vol. 2017, pp. 1-17, 2017, doi: 10.1155/2017/1235692.

[38] G. Zhang, X. Cao, and D. Mortari, "Analytical approximate solutions to ground track adjustment for responsive space," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 52, no. 3, pp. 1366-1383, 2016, doi: 10.1109/taes.2016.140644.

[39] G. Dai, X. Chen, M. Wang, E. Fernández, T. N. Nguyen, and G. Reinelt, "Analysis of Satellite Constellations for the Continuous Coverage of Ground Regions," *Journal of Spacecraft and Rockets,* vol. 54, no. 6, pp. 1294-1303, 2017, doi: 10.2514/1.A33826.

[40] T. C. Co, C. Zagaris, and J. T. Black, "Responsive Satellites Through Ground Track Manipulation Using Existing Technology," *Journal of Spacecraft and Rockets,* vol. 50, no. 1, pp. 206-216, 2013, doi: 10.2514/1.A32263.

[41] A. E. H. Elbeltagy, A. Youssef, A. Bayoumy, and Y. Elhalwagy, "Fixed ground-target tracking control of satellites using a nonlinear model predictive control," *Mathematical Modelling of Engineering Problems,* vol. 5, no. 1, pp. 11-20, 2018, doi: 10.18280/mmep.050102.

[42] C. Gao, D. Yang, X. Hong, B. Wang, and B. Zhang, "Performance Analysis of Ground Target Detection Utilizing Beidou Satellite Reflected Signals," *Sensors (Basel),* vol. 19, no. 9, May 9 2019, doi: 10.3390/s19092163.

[43] M. J. Nadoushan and N. Assadian, "Repeat ground track orbit design with desired revisit time and optimal tilt," *Aerospace Science and Technology,* vol. 40, pp. 200-208, 2015, doi: 10.1016/j.ast.2014.11.007.

[44] E. Ortore, M. Cinelli, and C. Circi, "A ground track-based approach to design satellite constellations," *Aerospace Science and Technology,* vol. 69, pp. 458-464, 2017/10/01/ 2017, doi: https://doi.org/10.1016/j.ast.2017.07.006.

[45] D. Casanova, M. Avendano, and D. Mortari, *Optimizing Flower Constellations for Global Coverage*. 2012.

[46] S. Huang, C. Colombo, and F. Bernelli-Zazzera, "Multi-criteria design of continuous global coverage Walker and Street-of-Coverage constellations through property assessment," *Acta Astronautica,* vol. 188, pp. 151-170, 2021/11/01/ 2021, doi: https://doi.org/10.1016/j.actaastro.2021.07.002.

[47] M. Xu and L. Huang, "An analytic algorithm for global coverage of the revisiting orbit and its application to the CFOSAT satellite," *Astrophysics and Space Science,* vol. 352, no. 2, pp. 497-502, 2014, doi: 10.1007/s10509-014-1939-2.

[48] S. Y. Ulybyshev and A. A. Lysenko, "Design of Satellite Constellations for Operational Global Monitoring with a Daily Repeat of Flight Track," *Cosmic Research,* vol. 57, no. 3, pp. 204-212, 2019, doi: 10.1134/s0010952519030080.

[49] P. Zong and S. Kohani, "Optimal Satellite LEO Constellation Design Based on Global Coverage in One Revisit Time," *International Journal of Aerospace Engineering,* vol. 2019, pp. 1-12, 2019, doi: 10.1155/2019/4373749.

[50] M. Zuo, G. Dai, L. Peng, and M. Wang, "An envelope curve-based theory for the satellite coverage problems," *Aerospace Science and Technology,* vol. 100, 2020, doi: 10.1016/j.ast.2020.105750.

[51] Z. Song, G. Dai, M. Wang, and X. Chen, "A Novel Grid Point Approach for Efficiently Solving the Constellation-to-Ground Regional Coverage Problem," *IEEE Access,* vol. 6, pp. 44445-44458, 2018, doi: 10.1109/ACCESS.2018.2864181.

[52] Z. Song, X. Hu, M. Wang, and G. Dai, "Judgement Theorems and an Approach for Solving the Constellation-to-Ground Coverage Problem," *Mathematical Problems in Engineering,* vol. 2018, pp. 1-10, 2018, doi: 10.1155/2018/5097324.

[53] Y. Chen, V. Mahalec, Y. Chen, R. He, and X. Liu, "Optimal satellite orbit design for prioritized multiple targets with threshold observation time using self-adaptive differential evolution," *Journal of Aerospace Engineering,* vol. 28, no. 2, p. 04014066, 2015.

[54] G. Zhang and J. Sheng, "Impulsive Ground-Track Adjustment for Assigned Final Orbit," *Journal of Spacecraft and Rockets,* vol. 53, no. 4, pp. 599-609, 2016, doi: 10.2514/1.A33447.

[55] H. W. Lee, S. Shimizu, S. Yoshikawa, and K. Ho, "Satellite Constellation Pattern Optimization for Complex Regional Coverage," *Journal of Spacecraft and Rockets,* vol. 57, no. 6, pp. 1309-1327, 2020, doi: 10.2514/1.A34657.

[56] C. Zhang, J. Jin, L. Kuang, and J. Yan, "LEO constellation design methodology for observing multi-targets," *Astrodynamics,* vol. 2, no. 2, pp. 121-131, 2018, doi: 10.1007/s42064-017-0015-4.

[57] W. Yao, J. Luo, M. Macdonald, M. Wang, and W. Ma, "Improved Differential Evolution Algorithm and Its Applications to Orbit Design," *Journal of Guidance, Control, and Dynamics,* vol. 41, no. 4, pp. 936-943, 2018, doi: 10.2514/1.G003214.

[58] S. Ghorbanpour, Y. Jin, and S. Han, "Differential Evolution with Adaptive Grid-Based Mutation Strategy for Multi-Objective Optimization," *Processes,* vol. 10, no. 11, 2022, doi: 10.3390/pr10112316.

[59] H.-D. Kim and J.-D. Seong, "Multi-objective Heuristic Design Approach for SAR Mission for Monitoring Local Target Area," *International Journal of Aeronautical and Space Sciences,* vol. 20, no. 2, pp. 518-524, 2019, doi: 10.1007/s42405-018-0129-9.

[60] J. Osei-kwakye, F. Han, A. A. Amponsah, Q. Ling, and T. A. Abeo, "A hybrid optimization method by incorporating adaptive response strategy for Feedforward neural network," *Connection Science,* vol. 34, no. 1, pp. 578-607, 2022, doi: 10.1080/09540091.2021.2025339.

[61] M. Li, W. Du, and F. Nian, "An Adaptive Particle Swarm Optimization Algorithm Based on Directed Weighted Complex Network," *Mathematical Problems in Engineering,* vol. 2014, pp. 1-7, 2014, doi: 10.1155/2014/434972.

[62] M. Wang, G. Dai, and M. Vasile, "Heuristic Scheduling Algorithm Oriented Dynamic Tasks for Imaging Satellites," *Mathematical Problems in Engineering,* vol. 2014, p. 234928, 2014/07/17 2014, doi: 10.1155/2014/234928.

[63] S. Liu and J. Yang, "A Satellite Task Planning Algorithm Based on a Symmetric Recurrent Neural Network," *Symmetry,* vol. 11, no. 11, p. 1373, 2019, doi: 10.3390/sym11111373.

[64] N. Biggs, E. K. Lloyd, and R. J. Wilson, *Graph Theory, 1736-1936*. Oxford University Press, 1986.

[65] K. H. Rubinder Mannan, Clément Albinet, Giuseppe Ottavianelli, Philippe Goryl, et al., "ESA's Earthnet data assessment pilot: paving the way for new space players," in *SPIE Remote Sensing, 2019,* , Strasbourg, France, 2019, doi: 10.1117/12.2532818.short. [Online]. Available: https://dx.doi.org/10.1117/12.2532818.short

[66] F. İnce, "Nano and Micro satellites as the Pillar of the "New Space" Paradigm," *Journal of Aeronautics and Space Technologies,* vol. 13, no. 2, pp. 235-250, 07/28 2020. [Online]. Available: https://jast.hho.msu.edu.tr/index.php/JAST/article/view/420.

[67] J. E. Harries, "Satellite meteorology: An introduction. By Stanley Q. Kidder and Thomas H. Vonder Haar. Academic Press. 1995. x + 466 pp. ISBN 0 12 406430 2," *Quarterly Journal of the Royal Meteorological Society,* vol. 128, no. 581, pp. 1037-1037, 2002, doi: 10.1002/qj.200212858118.

[68] M. Macdonald and V. Badescu, *The International Handbook of Space Technology*. 2014, pp. 1-731.

[69] D. Vallado and P. Crawford, *SGP4 Orbit Determination*. 2008.

[70] M. Pu, D. Wang, Y. Wu, J. Wang, and X. Shao, "Multi-objective Optimization Method For Repeat Ground-track Orbit Design Considering the Orbit Injection Error," *Journal of Aerospace Technology and Management,* vol. 10, 2018, doi: 10.5028/jatm.v10.913.

[71] P. R. Escobal, *Methods of Orbit Determination*. J. Wiley, 1965.

[72] S. Sivanandam, S. Deepa, S. Sivanandam, and S. Deepa, *Genetic algorithms*. Springer, 2008.

[73] E. Alba and M. Tomassini, "Parallelism and evolutionary algorithms," *TEVC,* vol. 6, no. 5, pp. 443-462, 2002, doi: 10.1109/TEVC.2002.800880.

[74] M. Herlihy, *The art of multiprocessor programming*. Morgan Kaufmann, 2012.

[75] A. Malapert, J.-C. Régin, and M. Rezgui, "Embarrassingly parallel search in constraint programming," *Journal of Artificial Intelligence Research,* vol. 57, pp. 421-464, 2016.

[76] J.-C. Régin, M. Rezgui, and A. Malapert, "Embarrassingly Parallel Search," Springer Berlin Heidelberg, 2013, pp. 596-610.

[77] M. Xiong, W. Xiong, and Z. Liu, "Evolutionary Multiobjective Satellite Range Scheduling With Learning-Guided Population Generation," *IEEE Access,* vol. 10, pp. 84664-84679, 2022, doi: 10.1109/ACCESS.2022.3197637.

[78] S. Zhao, Y. Wang, Y. Zhang, and K. Zheng, "Research Progress and Prospect of Satellite Constellation Optimization Design," *J. Phys.: Conf. Ser,* vol. 2457, no. 1, p. 12045, 2023, doi: 10.1088/1742-6596/2457/1/012045.

[79] C. M. A. Deccia, D. N. Wiese, and R. S. Nerem, "Using a Multiobjective Genetic Algorithm to Design Satellite Constellations for Recovering Earth System Mass Change," *Remote Sensing,* vol. 14, no. 14, 2022, doi: 10.3390/rs14143340.

[80] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization,* vol. 26, no. 6, pp. 369-395, 2004/04/01 2004, doi: 10.1007/s00158-003-0368-6.

[81] A. Shirazi, J. Ceberio, and J. A. Lozano, "Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions," *Progress in Aerospace Sciences,* vol. 102, pp. 76-98, 2018, doi: 10.1016/j.paerosci.2018.07.007.

[82] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics,* vol. 53, no. 3, pp. 484-512, 1984/03/01/ 1984, doi: https://doi.org/10.1016/0021-9991(84)90073-1.

[83] M. B. Bieterman, J. E. Bussoletti, C. L. Hilmes, F. T. Johnson, R. G. Melvin, and D. P. Young, "An adaptive grid method for analysis of 3D aircraft configurations," *Computer Methods in Applied Mechanics and Engineering,* vol. 101, no. 1, pp. 225-249, 1992/12/01/ 1992, doi: https://doi.org/10.1016/0045-7825(92)90024-E.

[84] M. J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics," *Journal of Computational Physics,* vol. 82, no. 1, pp. 64-84, 1989/05/01/ 1989, doi: https://doi.org/10.1016/0021-9991(89)90035-1.

[85] M. Ciment, "Stable Difference Schemes with Uneven Mesh Spacings," vol. 25, no. 114, p. 219, 1971, doi: 10.2307/2004917.

[86] A. Brandt, "Multi-Level Adaptive Solutions to Boundary-Value Problems," vol. 31, no. 138, p. 333, 1977, doi: 10.2307/2006422.

[87] B. Kreiss, "Construction of a curvilinear grid," *SIAM Journal on Scientific and Statistical Computing,* vol. 4, no. 2, pp. 270-279, 1983.

[88] A. B. Shamardan and Y. M. A. Essa, "Multi-level adaptive solutions to initial-value problems," *Korean Journal of Computational &amp; Applied Mathematics,* vol. 7, no. 1, pp. 215-222, 2000, doi: 10.1007/bf03009939.

[89] Y. Mao, H. Zhong, H. Qi, P. Ping, and X. Li, "An Adaptive Trajectory Clustering Method Based on Grid and Density in Mobile Pattern Analysis," *Sensors,* vol. 17, no. 9, p. 2013, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/9/2013.

[90] M. Lazarska and O. Siedlecka-Lamch, "Comparative study of relational and graph databases," pp. 000363-000370, 2019, doi: 10.1109/Informatics47936.2019.9119303.

[91] X. Wang, G. Wu, L. Xing, and W. Pedrycz, *Agile Earth observation satellite scheduling over 20 years: Formulations, methods and future directions*. 2020.

[92] Z. Chang, A. P. Punnen, Z. Zhou, and S. Cheng, "Solving dynamic satellite image data downlink scheduling problem via an adaptive bi-objective optimization algorithm," *Computers & Operations Research,* vol. 160, p. 106388, 2023/12/01/ 2023, doi: https://doi.org/10.1016/j.cor.2023.106388.

[93] X. Feng, Y. Li, and M. Xu, "Multi-satellite cooperative scheduling method for large-scale tasks based on hybrid graph neural network and metaheuristic algorithm," *Advanced Engineering Informatics,* vol. 60, p. 102362, 2024/04/01/ 2024, doi: https://doi.org/10.1016/j.aei.2024.102362.

[94] J. Wang, G. Song, Z. Liang, E. Demeulemeester, X. Hu, and J. Liu, "Unrelated parallel machine scheduling with multiple time windows: An application to earth observation satellite scheduling," *Computers & Operations Research,* vol. 149, 2023, doi: 10.1016/j.cor.2022.106010.

[95] X. Wang, S. Li, and Y. She, "Concept design and cluster control of advanced space connectable intelligent microsatellite," *Acta Astronautica,* vol. 141, pp. 1-7, 2017, doi: 10.1016/j.actaastro.2017.09.024.

[96] N. Hitomi and D. Selva, "Constellation optimization using an evolutionary algorithm with a variable-length chromosome," in *2018 IEEE Aerospace Conference*, 3-10 March 2018 2018, pp. 1-12, doi: 10.1109/AERO.2018.8396743.

[97] H. A. Simon, *Models of man; social and rational* (Models of man; social and rational.). Oxford, England: Wiley, 1957, pp. xiv, 287-xiv, 287.

[98] A. Casado, S. Bermudo, A. D. López-Sánchez, and J. Sánchez-Oro, "An iterated greedy algorithm for finding the minimum dominating set in graphs," *Mathematics and*

Computers in Simulation, vol. 207, pp. 41-58, 2023/05/01/ 2023, doi: https://doi.org/10.1016/j.matcom.2022.12.018.

[99] E. W. Weisstein. "Graph Theory." MathWorld. https://mathworld.wolfram.com/GraphTheory.html (accessed 06/06/2023, 2023).

[100] R. R. Korfhage, "CHAPTER 2 - Undirected Graphs," in *Discrete Computational Structures*, R. R. Korfhage Ed.: Academic Press, 1974, pp. 37-100.

[101] R. R. Korfhage, "CHAPTER 4 - Directed Graphs," in *Discrete Computational Structures*, R. R. Korfhage Ed.: Academic Press, 1974, pp. 130-165.

[102] E. W. Weisstein. "Directed Graph." MathWorld. https://mathworld.wolfram.com/DirectedGraph.html (accessed 08/06/2023, 2023).

[103] D. West, *Introduction to Graph Theory (2nd Edition)*. 2000.

[104] S. Akbari and S. Bahramian, "Complete Multipartite Graphs and their Null Set," *Electronic Notes in Discrete Mathematics,* vol. 45, pp. 67-72, 2014/01/15/ 2014, doi: https://doi.org/10.1016/j.endm.2013.11.014.

[105] A. E. Ezugwu *et al.*, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Eng. Appl. Artif. Intell.,* vol. 110, no. C, p. 43, 2022, doi: 10.1016/j.engappai.2022.104743.

[106] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications* (Structural Analysis in the Social Sciences). Cambridge: Cambridge University Press, 1994.

[107] C. G. S. Freitas, A. L. L. Aquino, H. S. Ramos, A. C. Frery, and O. A. Rosso, "A detailed characterization of complex networks using Information Theory," *Scientific Reports,* vol. 9, no. 1, p. 16689, 2019/11/13 2019, doi: 10.1038/s41598-019-53167-5.

[108] L. d. F. Costa, M. A. Rodrigues Tognetti, and F. N. Silva, "Concentric characterization and classification of complex network nodes: Application to an institutional collaboration network," *Physica A: Statistical Mechanics and its Applications,* vol. 387, no. 24, pp. 6201-6214, 2008/10/15/ 2008, doi: https://doi.org/10.1016/j.physa.2008.06.034.

[109] M. Latapy, C. Magnien, and N. D. Vecchio, "Basic notions for the analysis of large two-mode networks," *Social Networks,* vol. 30, no. 1, pp. 31-48, 2008/01/01/ 2008, doi: https://doi.org/10.1016/j.socnet.2007.04.006.

[110] M. Newman, *Networks: An Introduction*. Oxford University Press, 2010.

[111] A.-L. Barabasi, "The New Science of Networks," *J. Artificial Societies and Social Simulation,* vol. 6, 03/01 2003, doi: 10.2307/20033300.

[112] Z. Yang, H. Lu, and Q. Yu, "Critical independent sets of König–Egerváry graphs," *Discrete Applied Mathematics,* vol. 318, pp. 1-5, 2022/09/15/ 2022, doi: https://doi.org/10.1016/j.dam.2022.04.014.

[113] M. D. Sarder, "Chapter 7 - Assignment and transshipment problems with linear programming," in *Logistics Transportation Systems*, M. D. Sarder Ed.: Elsevier, 2021, pp. 169-195.

[114] J. Munkres, "Algorithms for the Assignment and Transportation Problems," *Journal of the Society for Industrial and Applied Mathematics,* vol. 5, no. 1, pp. 32-38, 1957. [Online]. Available: http://www.jstor.org/stable/2098689.

[115] G. Katona and D. Sza´sz, "Matching problems," *Journal of Combinatorial Theory, Series B,* vol. 10, no. 1, pp. 60-92, 1971/02/01/ 1971, doi: https://doi.org/10.1016/0095-8956(71)90067-0.

[116] S. Gentry, M. A. Mankowski, and T. S. Michael, "Maximum matchings in graphs for allocating kidney paired donation," *Operations Research for Health Care,* vol. 25, p. 100246, 2020/06/01/ 2020, doi: https://doi.org/10.1016/j.orhc.2020.100246.

[117] M. Barketau, E. Pesch, and Y. Shafransky, "Minimizing maximum weight of subsets of a maximum matching in a bipartite graph," *Discrete Applied Mathematics,* vol. 196, pp. 4-19, 2015/12/11/ 2015, doi: https://doi.org/10.1016/j.dam.2015.01.008.

[118] M. Dehmer, *Structural Analysis of Complex Networks*. 2011.

[119] A. Bhattacharya, S. Friedland, and U. N. Peled, "On the First Eigenvalue of Bipartite Graphs," *The Electronic Journal of Combinatorics,* vol. 15, no. 1, 2008, doi: 10.37236/868.

[120] Y.-F. Chen, H.-L. Fu, I.-J. Kim, E. Stehr, and B. Watts, "On the largest eigenvalues of bipartite graphs which are nearly complete," *Linear Algebra and its Applications,* vol. 432, no. 2-3, pp. 606-614, 2010, doi: 10.1016/j.laa.2009.09.008.

[121] E. W. Weisstein. "Adjacency Matrix." Wolfram Research. https://mathworld.wolfram.com/ProperSubset.html (accessed 02/07/2024, 2024).

[122] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks,* vol. 1, no. 3, pp. 215-239, 1978/01/01/ 1978, doi: https://doi.org/10.1016/0378-8733(78)90021-7.

[123] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social Networks,* vol. 32, no. 3, pp. 245-251, 2010/07/01/ 2010, doi: https://doi.org/10.1016/j.socnet.2010.03.006.

[124] P. Chunaev, *Community detection in node-attributed social networks: a survey*. 2019.

[125] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang, "Complex networks: Structure and dynamics," *Physics Reports,* vol. 424, no. 4, pp. 175-308, 2006/02/01/ 2006, doi: https://doi.org/10.1016/j.physrep.2005.10.009.

[126] Y. Yang, S. Peng, D.-S. Park, F. Hao, and H. Lee, "A Novel Community Detection Method of Social Networks for the Well-Being of Urban Public Spaces," *Land,* vol. 11, no. 5, p. 716, 2022. [Online]. Available: https://www.mdpi.com/2073-445X/11/5/716.

[127] M. G. Everett and S. P. Borgatti, "The centrality of groups and classes," *The Journal of Mathematical Sociology,* vol. 23, no. 3, pp. 181-201, 1999, doi: 10.1080/0022250x.1999.9990219.

[128] V. Latora and M. Marchiori, "A measure of centrality based on network efficiency," *New Journal of Physics,* vol. 9, pp. 188 - 188, 2004.

[129] K. C. Yang, B. Aronson, M. Odabas, Y. Y. Ahn, and B. L. Perry, "Comparing measures of centrality in bipartite patient-prescriber networks: A study of drug seeking for opioid analgesics," (in eng), *PLoS One,* vol. 17, no. 8, p. e0273569, 2022, doi: 10.1371/journal.pone.0273569.

[130] D. R. White and M. E. J. Newman, "Fast Approximation Algorithms for Finding Node-Independent Paths in Networks," 2001.

[131] J. Moody and D. White, "Social Cohesion and Embeddedness: A Hierarchical Conception of Social Groups," *American Sociological Review,* vol. 68, 08/11 2002.

[132] J. Torrents, *Structural Cohesion: Visualization and Heuristics for Fast Computation with NetworkX and matplotlib*. 2015, pp. 67-76.

[133] D. Zhang, H. Men, and Z. Zhang, "Assessing the stability of collaboration networks: A structural cohesion analysis perspective," *Journal of Informetrics,* vol. 18, no. 1, p. 101490, 2024/02/01/ 2024, doi: https://doi.org/10.1016/j.joi.2024.101490.

[134] R. S. Sinkovits, J. Moody, B. T. Oztan, and D. R. White, "Fast determination of structurally cohesive subgroups in large networks," *Journal of Computational*

Science, vol. 17, pp. 62-72, 2016/11/01/ 2016, doi: https://doi.org/10.1016/j.jocs.2016.10.005.

[135] B. Cornwell and J. Burchard, "Structural cohesion and embeddedness in two-mode networks," *The Journal of Mathematical Sociology,* vol. 43, no. 4, pp. 179-194, 2019/10/02 2019, doi: 10.1080/0022250X.2019.1606806.

[136] M. Newman, *Networks*. OUP Oxford, 2018.

[137] E. W. Weisstein. "Subset." Mathmatica. https://mathworld.wolfram.com/Subset.html (accessed 17/07/2024, 2024).

[138] G. Lohmann *et al.*, "Eigenvector Centrality Mapping for Analyzing Connectivity Patterns in fMRI Data of the Human Brain," *PloS one,* vol. 5, p. e10232, 04/27 2010, doi: 10.1371/journal.pone.0010232.

[139] R. A. Clark, C. N. McGrath, and M. Macdonald, "Dynamical Influence Driven Space System Design," Cham, 2022: Springer International Publishing, in Complex Networks & Their Applications X, pp. 27-38.

[140] D. C. Arney and A. W. Wilhite, "Modeling Space System Architectures with Graph Theory," *Journal of Spacecraft and Rockets,* vol. 51, no. 5, pp. 1413-1429, 2014, doi: 10.2514/1.A32578.

[141] A. M. Khidhir, A. M. Ali, and S. M. Aziz, "Application of width distance on semi–star link satellite constellation," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 24, no. 3, pp. 797-807, 2021/04/03 2021, doi: 10.1080/09720529.2020.1843265.

[142] C. J. Lowe and M. Macdonald, "Resource-Considerate Data Routing Through Satellite Networks," *Journal of Aerospace Information Systems,* vol. 14, no. 8, pp. 472-482, 2017, doi: 10.2514/1.I010423.

[143] C. J. Lowe, R. A. Clark, C. N. McGrath, and M. Macdonald, "A delay-tolerant network approach to satellite pickup and delivery scheduling," *Ad Hoc Networks,* vol. 151, p. 103289, 2023/12/01/ 2023, doi: https://doi.org/10.1016/j.adhoc.2023.103289.

[144] I. del Portillo, B. G. Cameron, and E. F. Crawley, "A technical comparison of three low earth orbit satellite constellation systems to provide global broadband," *Acta Astronautica,* vol. 159, pp. 123-135, 2019/06/01/ 2019, doi: https://doi.org/10.1016/j.actaastro.2019.03.040.

[145] M. B. Larsen, R. S. Smith, and M. Blanke, "Modeling of tethered satellite formations using graph theory," *Acta Astronautica,* vol. 69, no. 7, pp. 470-479, 2011/09/01/ 2011, doi: https://doi.org/10.1016/j.actaastro.2011.05.014.

[146] H. Fan, Z. Yang, X. Zhang, S. Wu, J. Long, and L. Liu, "A novel multi-satellite and multi-task scheduling method based on task network graph aggregation," *Expert Systems with Applications,* vol. 205, p. 117565, 2022/11/01/ 2022, doi: https://doi.org/10.1016/j.eswa.2022.117565.

[147] C. N. McGrath, R. A. Clark, and M. Macdonald, "Novel concept of satellite manoeuvre planning using graph theoretical techniques," *Advances in Space Research,* vol. 67, no. 11, pp. 3775-3784, 2021, doi: 10.1016/j.asr.2020.06.008.

[148] R. A. Clark and M. Macdonald, "Identification of effective spreaders in contact networks using dynamical influence," *Appl Netw Sci,* vol. 6, no. 1, p. 5, 2021, doi: 10.1007/s41109-021-00351-0.

[149] X. Zhao, Z. Wang, Y. Cui, and G. Zheng, "Novel Ontology-Based Method for Generating Satellite Cluster's Task Configuration," *Journal of Aerospace Information Systems,* vol. 17, no. 2, pp. 86-96, 2020/02/01 2019, doi: 10.2514/1.I010731.

[150] Z. Shen *et al.*, "A Survey of Next-generation Computing Technologies in Space-air-ground Integrated Networks," *ACM Comput. Surv.,* vol. 56, no. 1, p. Article 23, 2023, doi: 10.1145/3606018.

[151] Z. Wang, X. Hu, H. Ma, and W. Xia, "Learning multi-satellite scheduling policy with heterogeneous graph neural network," *Advances in Space Research,* vol. 73, no. 6, pp. 2921-2938, 2024/03/15/ 2024, doi: https://doi.org/10.1016/j.asr.2023.12.036.

[152] J. Moody and D. R. White, "Structural cohesion and embeddedness: A hierarchical concept of social groups," *American Sociological Review,* vol. 68, no. 1, pp. 103-127, 2003, doi: 10.2307/3088904.

[153] D. G. Harris and N. Narayanaswamy, "A faster algorithm for Vertex Cover parameterized by solution size," in *Symposium on Theoretical Aspects of Computer Science*, 2022.

[154] S. Li, J. Wang, J. Chen, and Z. Wang, "An Algorithm for Minimum Vertex Cover Based on Max-I Share Degree," *JCP,* vol. 6, pp. 1781-1788, 08/01 2011, doi: 10.4304/jcp.6.8.1781-1788.

[155] G. Lohmann *et al.*, "Eigenvector Centrality Mapping for Analyzing Connectivity Patterns in fMRI Data of the Human Brain," *PLOS ONE,* vol. 5, no. 4, p. e10232, 2010, doi: 10.1371/journal.pone.0010232.

[156] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology,* vol. 2, pp. 113-120, 1972.

[157] X. Shuang, P. Wang, and C.-X. Zhang, "Identification of influential spreaders in bipartite networks: A singular value decomposition approach," *Physica A: Statistical Mechanics and its Applications,* vol. 513, 09/01 2018, doi: 10.1016/j.physa.2018.09.005.

[158] R. A. Clark, C. N. McGrath, and M. Macdonald, "Identifying effective sink node combinations in spacecraft data transfer networks," *Applied Network Science,* vol. 7, no. 1, 2022, doi: 10.1007/s41109-022-00473-z.

[159] Y. Zhang, K. Wang, W. Zhang, X. Lin, and Y. Zhang, *Pareto-optimal Community Search on Large Bipartite Graphs*. 2021, pp. 2647-2656.

Appendix A: A Comparison of Optimum Results for Retrograde and Prograde Orbits.

Despite the high launching costs and lack of enough ideal areas that can be used to launch into retrograde orbits, these orbits prove to have more optimum results for different objective functions. In this case the same objective functions of number of times seen and duration of view are presented and compared. Generally, retrograde orbits are found to have better performance in both. The inclination, RAAN and time increments used to simulate the results presented here are 0.05 degrees, 0.05 degrees and 10 seconds respectively. The search space and target values are identical to the ones presented in chapter 2 section 2.3.

The heat plots for the mean duration of view and the mean number of times seen are as presented in Figure A.1 and Figure A.2.

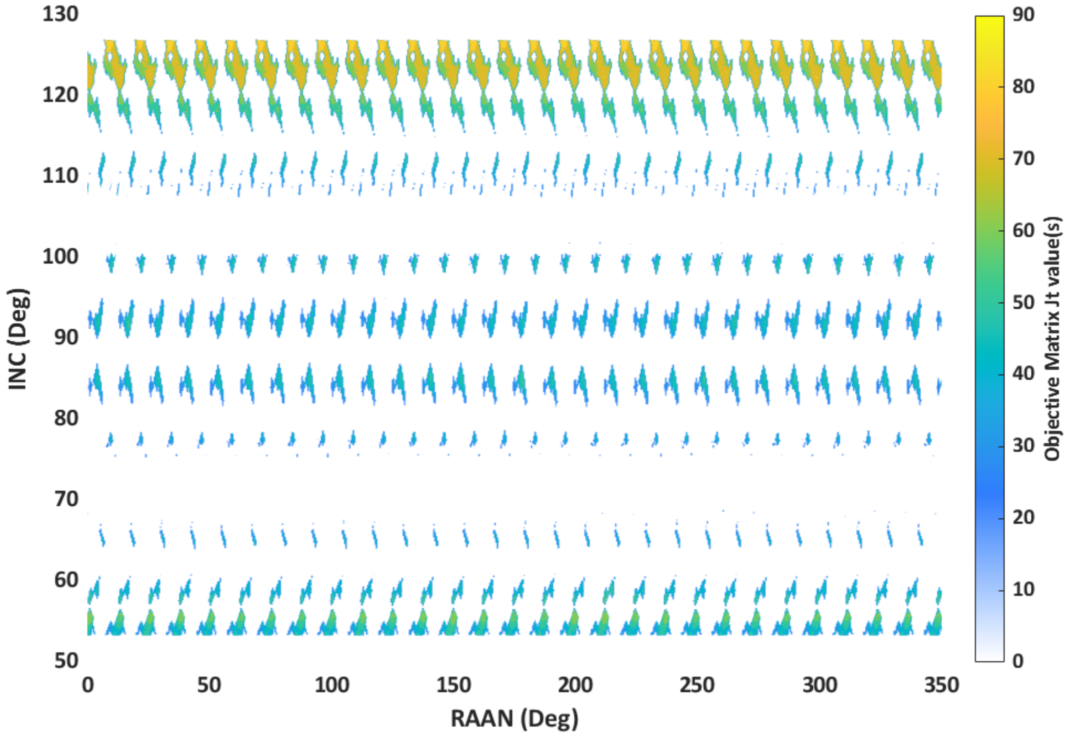


Figure A.1: duration of view objective matrix values figure of both retrograde and prograde orbits that view all 10 targets at least once.

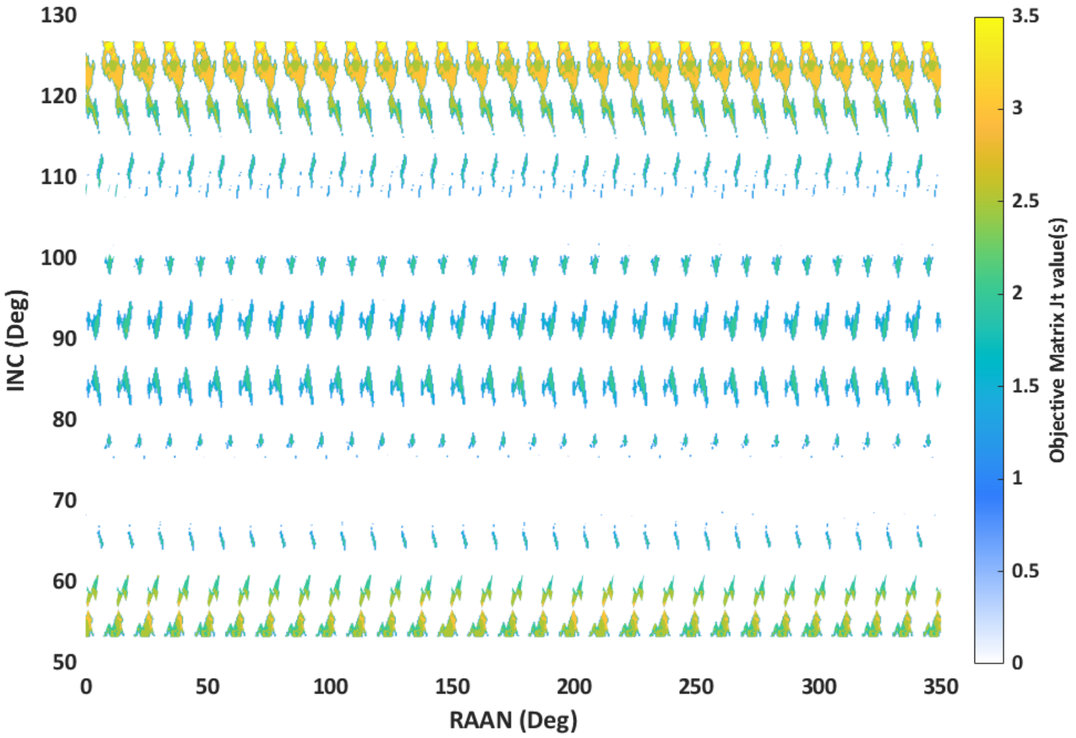


Figure A.2: Mean number of times targets are seen objective matrix values figure of both retrograde and prograde orbits that view all 10 targets at least once.

The maximum values for the objective functions of both the retrograde and prograde values are presented in Table A.1.

Table A.1: Maximum mean values of duration of view and number of times seen for retrograde and prograde orbits:

	Value of objective functions	Inclination (Deg.)	RAAN (Deg.)
Duration of view objective function	84.47	55.55	54.20
Number of times in view objective function	3.42	55.50	55.10
Duration of view objective function	94.16	126.20	111.21
Number of times in view objective function	3.73	126.40	209.68

The values of the retrograde orbits are higher than the ones for prograde, both for duration of view and number of view times of the targets.

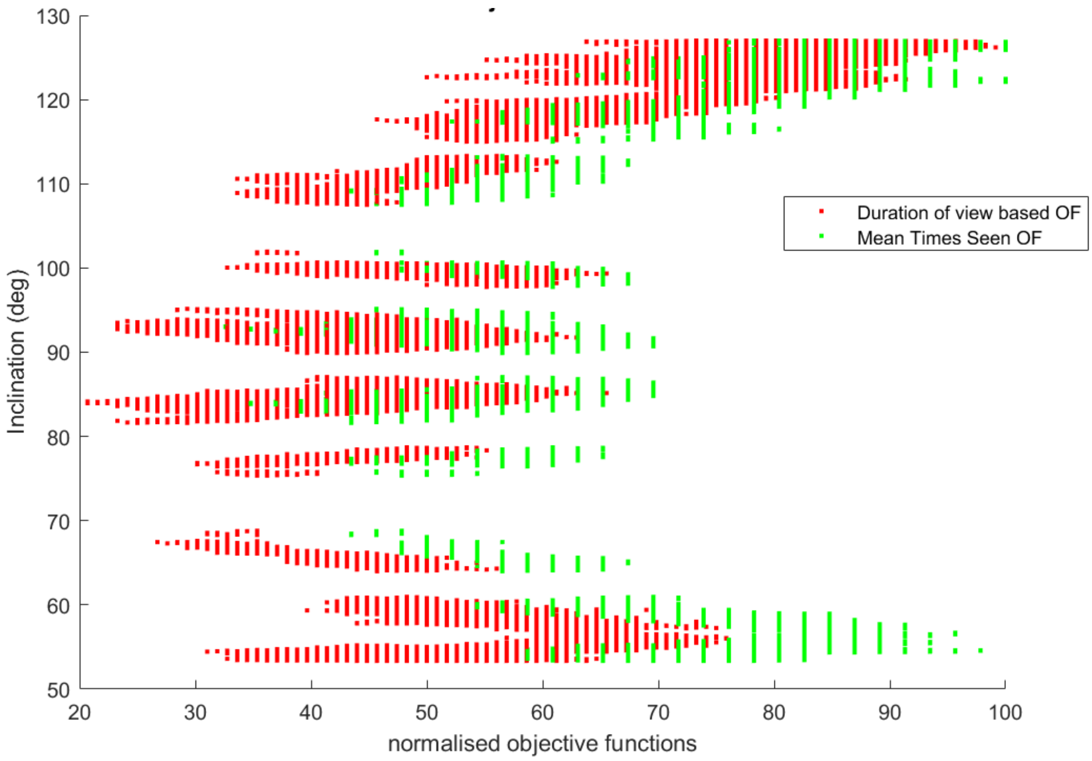


Figure A.3: Graph showing inclinations of all determined orbits against the normalised objective functions.

Figure A.3 shows the normalised objective functions of the mean duration of view and mean number of times seen. The duration of view of the prograde orbit as expected is much less than the ones for the retrograde orbit.

This was further analysed in terms of priorities of the objective functions. From this, the graphs presented in Figure A.4 were produced.

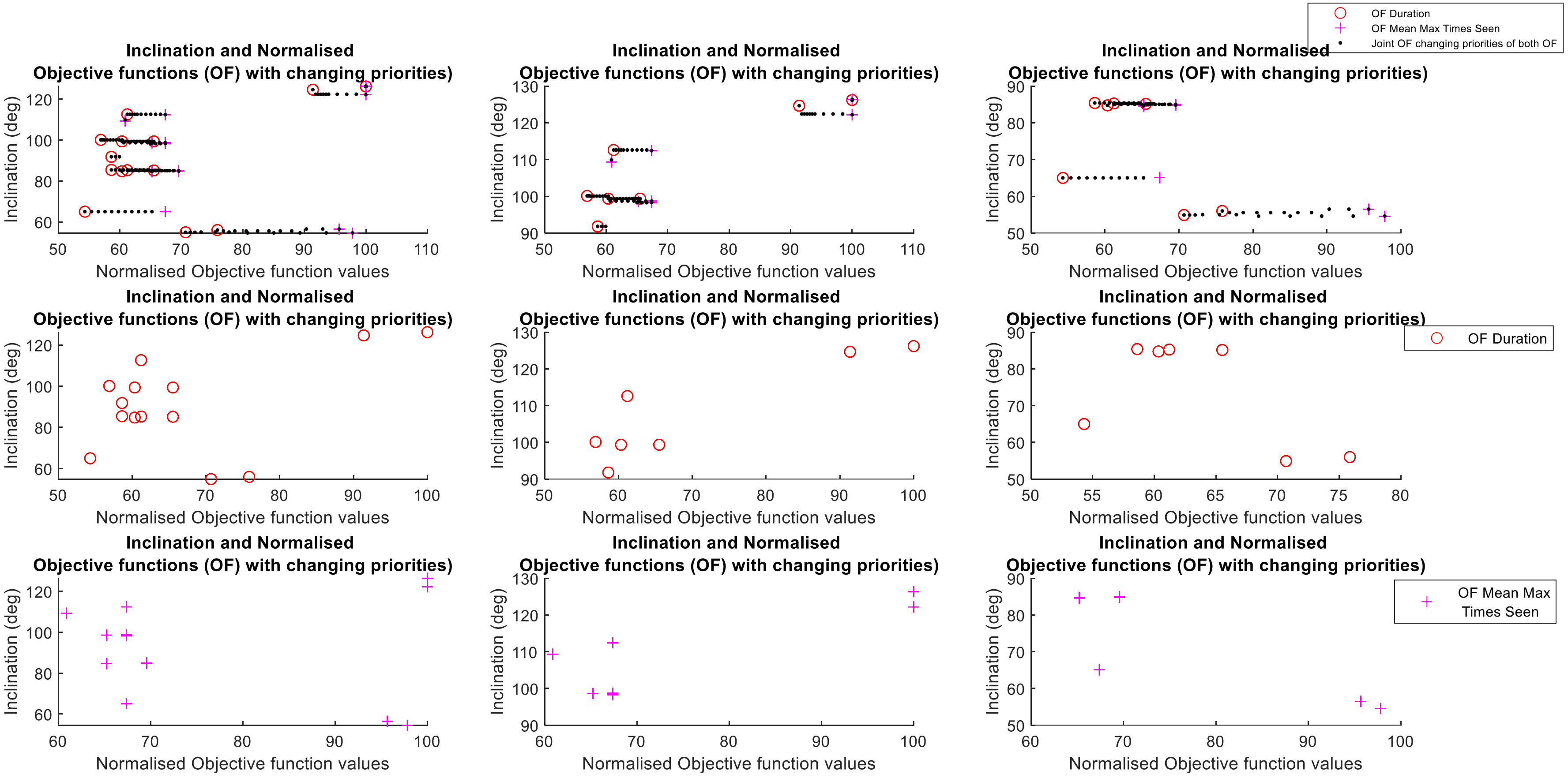


Figure A.4: Changing priority values of normalised objective functions to show the advantages of retrograde orbits over prograde orbits.

Figure A.4 shows that retrograde orbits are better in terms of performance than prograde orbits.

Appendix B: Extended Analysis of Inclusion of J_2 Perturbation to the Proposed Analytical Method

This appendix highlights an extended analysis of the use of the orbit design method developed in section 2 of this dissertation when the J_2 perturbations are included in the method and when not. Some previous analytical methods did not consider any secular perturbations with an aim to keep the analysis as simple and analytical as possible. This as has been seen before, increases the errors that already challenge the use of analytical methods. This section therefore gives a general justification of the use of J_2.

Firstly, the analysis of incorporating J_2 in the method and not has been highlighted in chapter 2. Further analysis is however given herein. It has already been highlighted that the number of orbits and the number of times of viewing a target differ. This shows that using the method without perturbations gives some erratic orbits which may not be applicable in real-life situations. When an analysis was done on the mean duration that each orbit views the targets, the orbits that had the longest mean duration is presented in Table B.1 and the heat plot is presented in Figure B.1.

Table B.1*:* Best Mean Duration of view orbit comparison when including J_2 in the method and when no perturbation is included.

Method used	Mean Duration of View	Best Orbit INC, deg	Best Orbit RAAN, deg
Including J_2	193.33 seconds	56.60	353.04
Without J_2	190.00 seconds	57.20	183.02

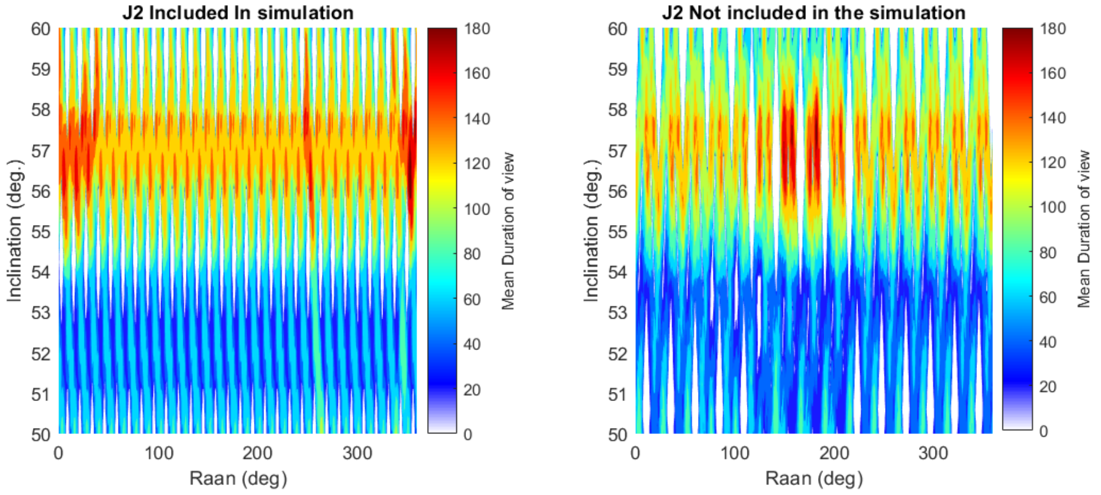


Figure B.1: Mean duration of view analysis of method including J_2 and with no perturbations.

For the further analysis of the number of times each orbit it viewed, the results presented in chapter 2 produced the following graphs.

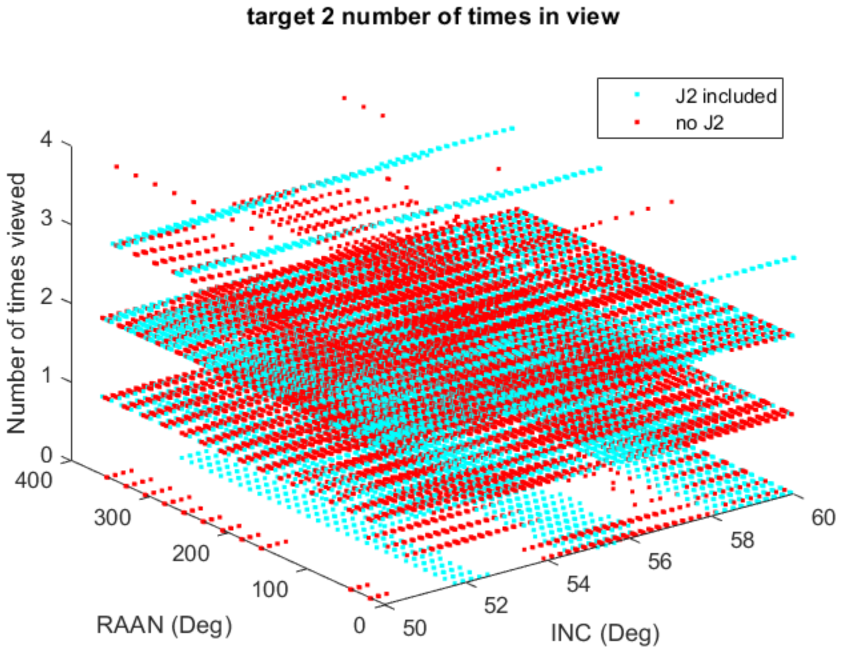


Figure B.2: Target 2 number of times in view for when $J_2$is included and when J_2 is not included.

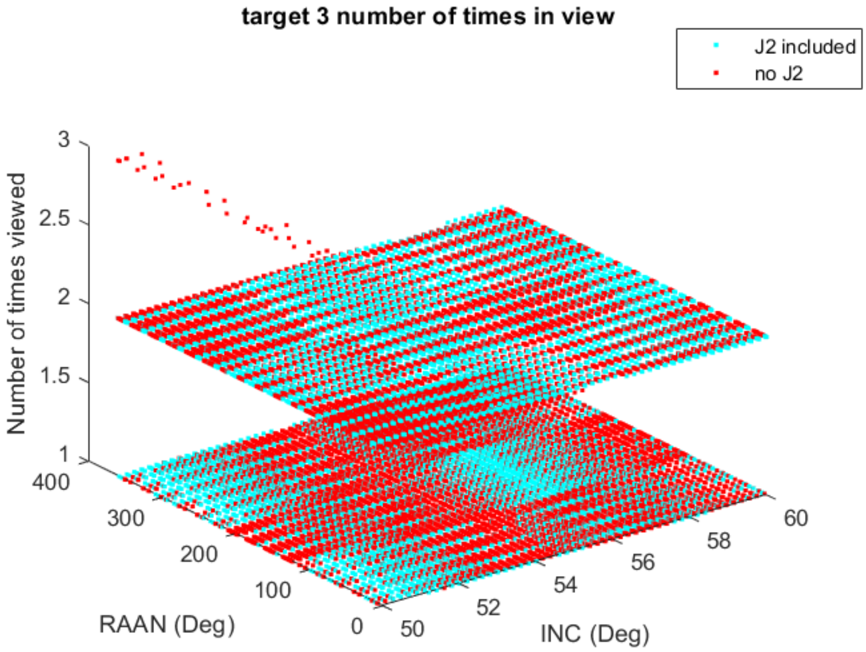


Figure B.3: Target 3 number of times in view for when J_2 is included and when J_2 is not included.

Analysis of results for orbit with inclination 56.6 degrees and RAAN 353.04 degrees was further carried out. This showed that the best mean duration of view orbit calculated when including J_2 was also obtained by the analysis when not including J_2. The orbit was calculated to have a mean duration of view of 126.67 seconds when $J_2$was not included. In general, the maximum duration of view for the targets are as on Table B.2.

Table B.2: duration of view differences in results using J_2 and without

Maximum duration of view	With J_2 on method (Sec.)	Without J_2 on method (Sec.)
Target 1	365	350
Target 2	210	160
Target 3	140	140

The orbit calculated to have mean duration of view, when J_2 is included is used to analyse the actual differences when J_2 is included on method, on GMAT.

The GMAT simulation of orbit 56.60 degrees inclination and 353.04 degrees RAAN results are presented in Table B.3 which shows that when the method includes J_2 perturbations the results are closer to the GMAT simulations than for simulations when J_2 is not included.

Table B.3: Comparison of mean maximum duration of view results from GMAT simulation and proposed analytical method with J_2 and without J_2.

Number of times orbit is viewed per target	(GMAT simulation)	Analytical method with J_2	Analytical method with no J_2
Target 1	5	5	4
Target 2	3	3	2
Target 3	2	1	1

First to note is that the calculated orbit is a repeat ground track orbit that repeats after 2-days periods, and this is confirmed by the simulation on GMAT. For the calculation that does not include J_2 however, the orbit does not repeat. This is presented in Figure B.4. Also, in all cases, the orbit overflies ALL the 3 targets used (as presented in Table 2.15).

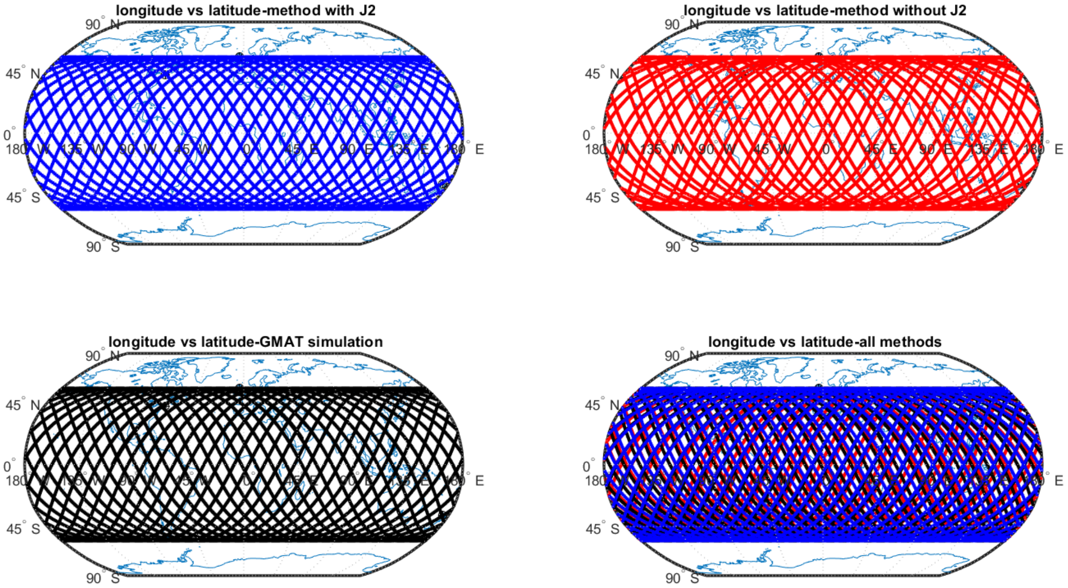

longitude vs latitude-method with J2
longitude vs latitude-method without J2
longitude vs latitude-GMAT simulation
longitude vs latitude-all methods
90° N
45° N
0°
45° S
90° S
180° W
135° W
90° W
45° W
0°
45° E
90° E
135° E
180° E

Figure B.4: Ground-track difference of developed method with J$_2$, without J$_2$ and GMAT

The ground tracks of the simulation when J_2 is included and when J_2 is not included clearly show the errors incurred assuming that all secular perturbations are neglected. For this method therefore, the inclusion of the perturbation increases the accuracy of the method.

Appendix C: Calculation of Minimum Elevation Angle for GMAT Analysis and Comparison with Proposed Analytical Method Results.

As presented by Refs. [53] the relationships between the satellite, the ground target and the Earth centre are herein presented. The diagram for this relationship is presented in Figure C.1.

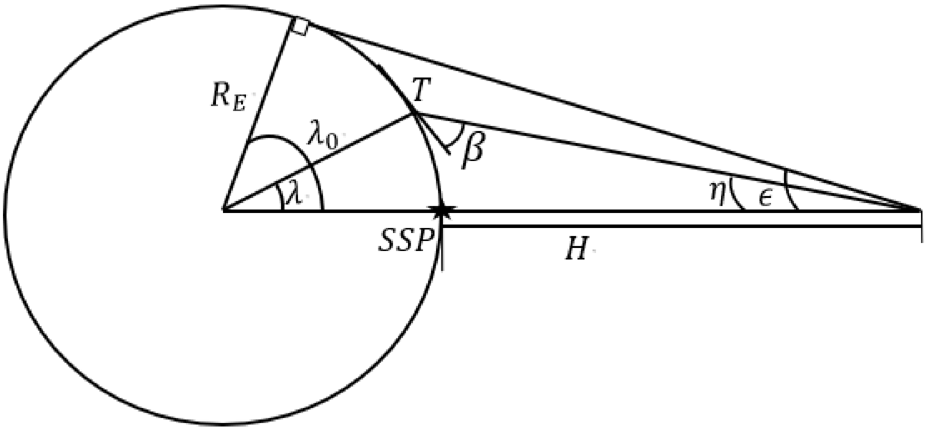


Figure C.1: Relationship of satellite view angles to get the FoV and minimum elevation angle to use in numerical method simulations.

From the diagram, the half effective FoV angle, η is dependent upon the instrument onboard the satellite and is therefore given by the mission. In this case, it is given as 20 degrees. It is measured from the SSP to the target, T, and determines the minimum value of the ground elevation angle. For propagation using third party software, this value is required for increased accuracy of computation. The equations previously presented in section 2 can be used for this calculation, but an alternative method is presented. The angular radius of the Earth,ϵ and the Earth central angle can be calculated from the radius of the Earth and the semi-major axis using the mnemonic trigonometric function ratios, SOHCAHTOA.

$$sin\,\epsilon = cos\,\lambda_0 = \frac{R_E}{R_E+H} \qquad (C.1)$$

From this, the minimum elevation angle can be calculated from,

$$cos\,\beta = \frac{sin\,\eta}{sin\,\epsilon} \qquad (C.2)$$

The effective Earth central angle, λ, can then be calculated from previously presented equations as,

$$sin\,\lambda = \frac{D\,sin\,\eta}{R_E} \qquad (C.3)$$

where,

$$sin(90 + \beta) = \frac{a\,sin\,\eta}{R_E} \qquad (C.4)$$

And

$$D = R_E\,cos(90 + \beta) + acos\,\eta \qquad (C.5)$$

With this calculation, the minimum elevation angle used to simulate the orbit and targets it overflies is 67.816 degrees. Also,

$$\eta + \lambda + \beta = 90\,deg$$

Which confirms this calculation.

Appendix D: Further Ground Track, Latitude and Longitude Error Analysis of Proposed Analytical Method Using Third Party Software.

From the analysis on section 2.3.3, comparing the method results from a direct third-party software analysis, there are some errors in the longitude simulations. This section further analyses the errors between the two ground-track simulation methods. The orbit used for this analysis is 55.2 degrees inclination and 150.074 degrees RAAN. The simulation for the orbits to get the ground tracks used an analytical orbit propagator where the SMA was constant. With this regard the ground track results have errors. For calculating the orbits however, the method updated the SMA according to J_2, and the orbital inclination. This explains why the GMAT results for the found orbits were more accurate than for the ground track simulations. This shows that the short coming for the errors found are more in the ground track simulation than the method used. The starting period for both was found to be 5879.2 seconds. By the 4th minute however, the GMAT simulation of the ground track is updated to 5877.87 seconds. For the analytical ground track, this is not updated and hence the differences in Longitudes simulated for the ground-track of the given orbit. These are some of the shortcomings of using analytical propagators as compared to numerical propagators.

When this was redone using a semi-analytical ground-track simulator, the results of the two ground-tracks were much closer as shown below in Figure D.1.

The respective longitude and latitude covariance and correlations both gave positive results. The ground plots, the ground plot errors as well as the relationships between the two values are presented.

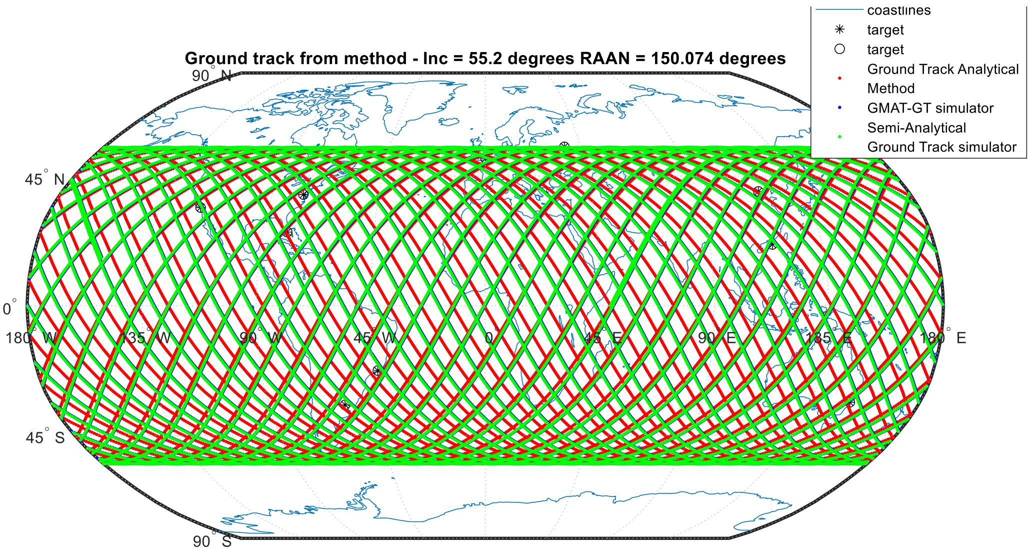

Ground track from method - Inc = 55.2 degrees RAAN = 150.074 degrees
90° N
45° N
0°
45° S
90° S
180° W
135° W
90° W
45° W
0°
45° E
90° E
135° E
180° E
coastlines
target
target
Ground Track Analytical Method
GMAT-GT simulator
Semi-Analytical Ground Track simulator

Figure D.1: Ground-track simulation using analytical method, numerical method, and semi-analytical method.

The error analysis for this then gives the following longitude and latitude error plots.

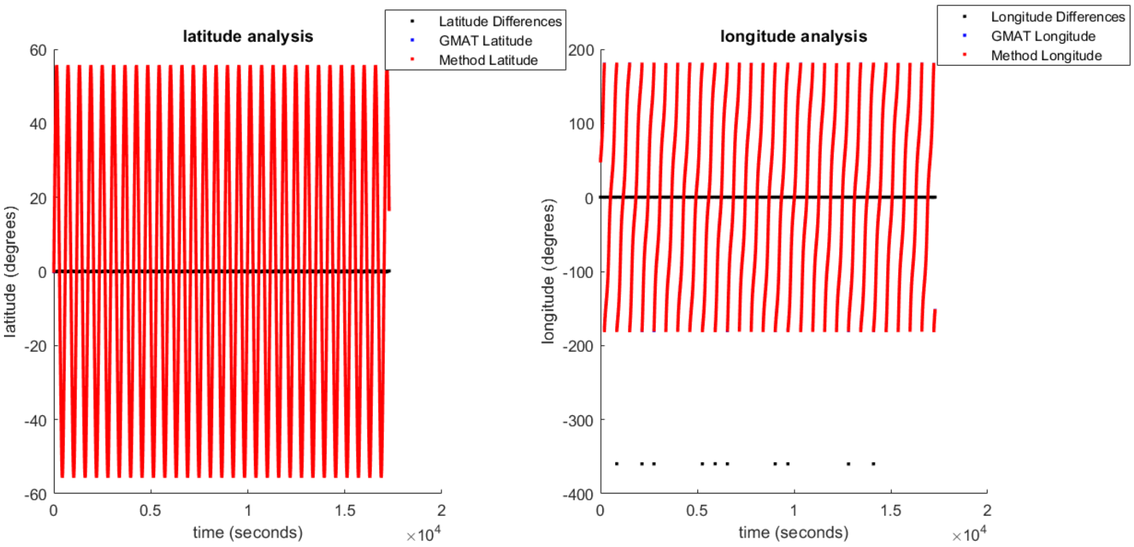


Figure D.2: Longitudes and latitudes when using a semi-analytical ground-track simulator and GMAT ground-track simulator.

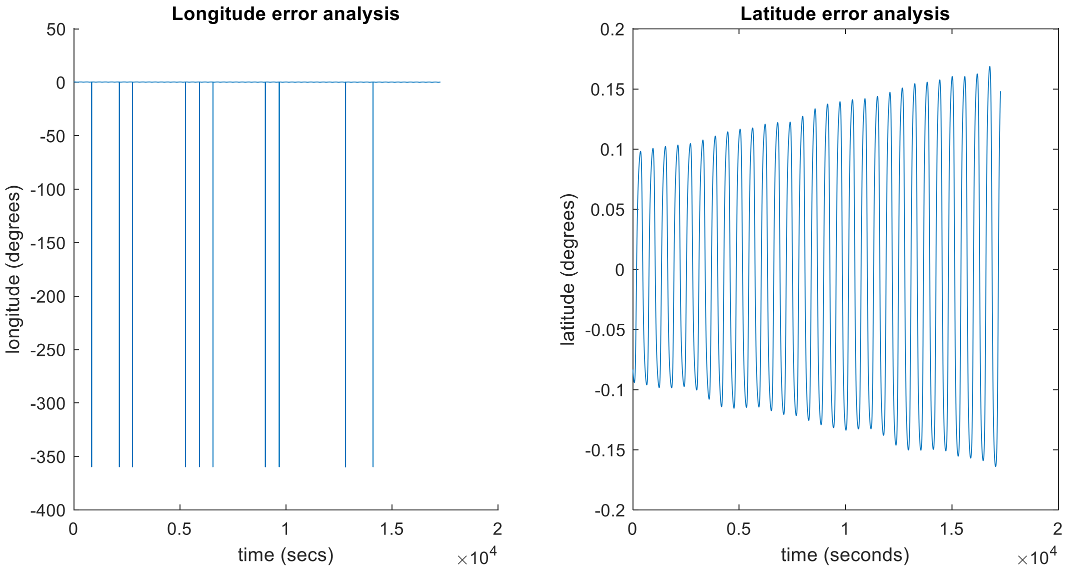


Figure D.3: Longitudes and latitudes when using a semi-analytical ground-track simulator and GMAT ground-track simulator.

Table D.1: Covariance and correlation between the values of the semi-analytical ground-track and the numerical GMAT ground-track

Covariance Longitude		Covariance Latitude	
10872.87	10835.24	1397.64	1398.59
10835.24	10872.56	1398.59	1399.56
Correlation Longitude - 0.997		Corelation Latitude - 0.999	

With all these different simulations however, the orbit found by the developed method proved to overfly all the targets at least once and from the GMAT simulation in particular, the duration of view had a less than 10 second difference. This shows that the method does work as it should and though there might be some errors due to the analytical nature as well as the increments used for the simulation, it is a method that is helpful for initial orbit designs.

Appendix E: Adaptive grid using multiple objective functions.

Section 3 presents an adaptive grid method used with the developed analytical method. The adaptive grid reduces computation time and refines the grids where the maximum objective functions are. In this appendix, the adaptive grid is used to optimise two objective functions, and the results are presented using the algorithms presented in section 3.

The duration of view and the mean number of times optimised in section 3 are simulated using an orbit with inclination 50-90 degrees, RAAN 0-360 degrees, time 0-172800 seconds in increments of 1 degree, 1 degree and 10 seconds respectively.

These orbits were set to overfly the 10 targets at least once each. The results from the simulations are presented in Figure E.1 – Figure E.3.

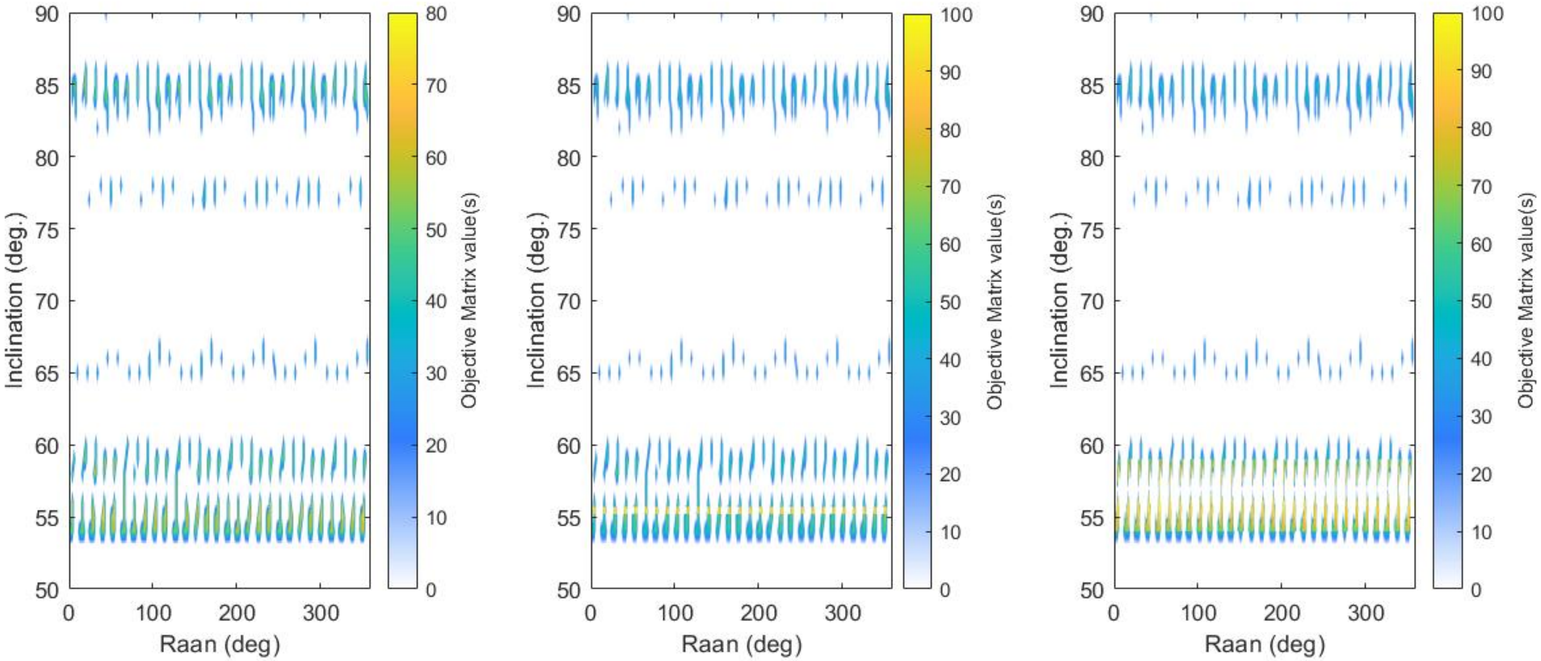


Figure E.1: Duration of View Objective function with refined grids. First figure shows the first grid when the refinement has not yet been done, second is for grid two (first refinement) and third is for grid 3 (second refinement)

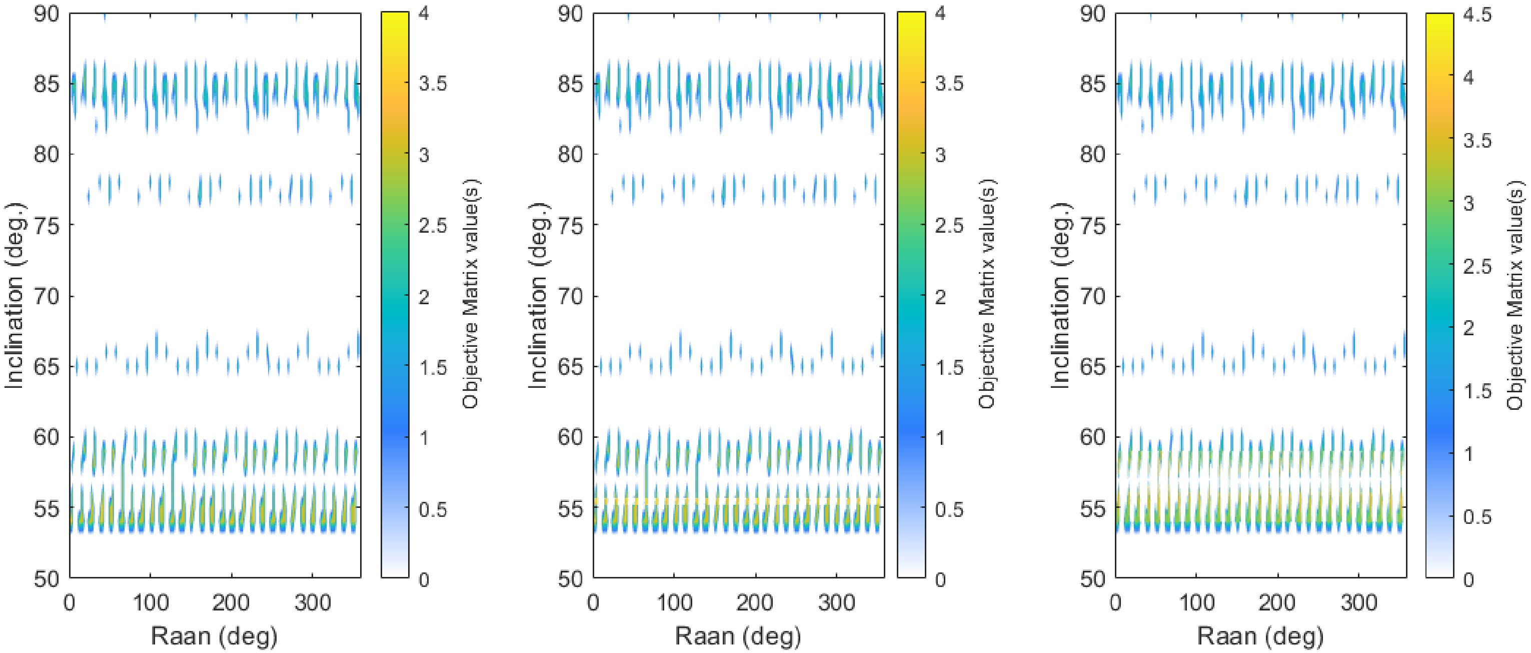


Figure E.2: Mean number of times seen Objective function with refined grids. First figure shows the first grid when the refinement has not yet been done, second is for grid two (first refinement) and third is for grid 3 (second refinement)

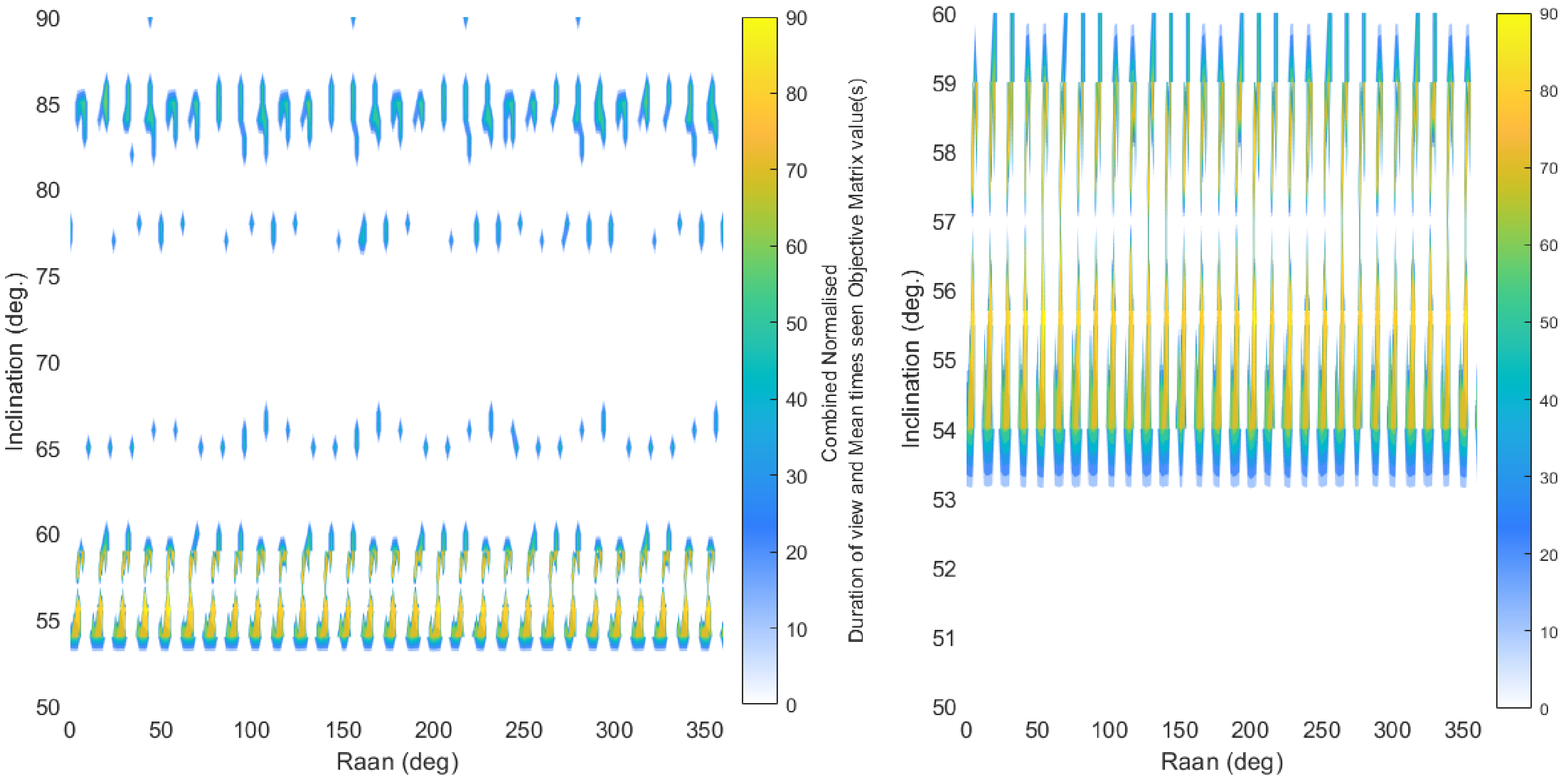


Figure E.3: Combined Normalised Objective functions heat plot

Appendix F: A Breakdown of all the Algorithms Used for Satellite tasking, Chapter 5.

A series of algorithms were used for the optimisation of multi-objective functions to determine tasking of satellites. These algorithms are presented herein. The first algorithm is the one used to create the graph.

Algorithm F.1**: **Graph creation

```
Input: data from propagation file
Output: graph of satellites and ground targets
Define target nodes, T
Define satellite nodes, S
%create graph
function graph(G) = inputs(data from propagation file)
G = graph;
G = addnode(G,T)
Weights = data from propagation method do// select the optimisation data from
propagation file
G = addnode(G,S)
G = addedges(G,S,T,weights)
End
graph = G
```

The Algorithm F.1 gives the algorithm used to create a graph from the propagation data.

Algorithm F.2: Algorithm to find minimum satellite nodes and satellite node combinations

```
Input: Graph G(S,T,E)
Output: Nodes and pairs belonging to S with E ==10
Create function: creates a call function for the neighbours algorithm
Initialization of variables: set i, full_set_nodes and pair_nodes to zero
```

```
for counter_n <numel(S) do //loop through all satellite nodes and get the
neighbours
    Neighbours1 – store all target sets for all satellite nodes
    if numel(T)==numel(Neighbours1) do// finds satellite nodes in contact
    with all (T)
        Satellite nodes – gives all satellite nodes that are in contact with all
        targets
    End
    for 1<counter_n2<numel(S) do//loop through all satellite nodes>1
        Neighbours2 – store all target node sets for all alternate satellite
        nodes
        Union – computes the unions between Neighbours1 and
        Neighbours2
        if numel (T) == numel (Union) do//finds the orbit pairs in contact
        with all (T)
            i = i+1
            N_pairs((i),:)=[Satellite_nodeID,Satellite_nodeID2];
        End
    End
End
single satellite = Satellite nodes
satellite pairs = N_pairs
```

Algorithm F.**2** was used to find the neighbours of each satellite node then identify the satellite nodes and satellite node pairs that are in contact with all target nodes.

Algorithm F.3: Basic algorithm to get the optimum satellites based on one objective function.

```
input: G(S,T,E)
output: (centralities)
function (centralities(:,:)) = inputs // create function for centrality calculations
eigenC (:,:) = eigencentrality(G);
eigenCw(:,:) = eigencentrality weighted(G);
End
```

```
centralities = [eigenC, eigenCw]
Repeat for degree centrality and partitioned graphs of the original bipartite graph
```

Algorithm F.**3** calculates the centralities of the satellite nodes and stores them for further analysis.

Algorithm F.4: Eigenvector and SVD algotithm

```
input: G(S,T,E)
output: (Eigenvectors, Singular Values)
function (Eigen_vectors(:,:),svd(:,:)) = inputs
A = adjacency (G);
B1 = A*A(transpose);
B2 = A(transpose)*A;
Eigen = eigenvector(A);
Eigen_1 = eigenvector(B1);
Eigen_2 = eigenvector(B2);
SVD = svd(A);
SVD_1 = svd(B1);
SVD_2 = svd(B2);
End
[Eigenvectors,svd] = [Eigen,Eigen_1,Eigen_2;SVD,SVD_1,SVD_2];
```

Algorithm F.**4** calculates the eigenvectors of the adjacency matrix both joint bipartite graphs and the different partite eigenvectors and SVD’s. The principal values were then stored for further analysis.

Algorithm F.5 :Normalisation of the centralities

```
Inputs: Individual Objective function centralities
Output: Optimal Satellites
function Optim = inputs(Objective function centralities)
normalise each set of centrality values
for I = 1:numel(satellite nodes)
    if for all centralities (i)>= 80
        Optim[] = satellite node(i)
        End
End
Optim[] = Optim
```

Algorithm F.**5** calculates the normalised values of the objective function centralities and stores the optimum satellite nodes.

Algorithm F.6: Optimisation of the combined centralities to best task the satellite combinations

```
Inputs: (O),normalised_centralities_OF1,normalised_centralities_OF2 (for each orbit nodes), N_pairs, 1_Satellite(from neighbours algorithm)
Outputs: Optimum nodes for combined OF1 and OF2
function $Optim\_const = Opt\_combination(inputs)$
for n = 1:numel(N_pairs)
    Opt_OF1(n) = normalised_centralityOF1(node2)+normalised_centralityOF1(node1)
    Opt_OF2(n) = normalised_centralityOF2(node2)+normalised_centralityOF2(node1)
End
Normalise each set of the centrality values
for i = 1:numel(satellite nodes)
    if $for\ all\ centralities\ (i) \geq 70$
        $Optim\_combination[\quad] = satellite\_pair(i)$
    End
End
Optim_combination[] = optim_combination;
```

Algorithm F.6 gives the optimisation of the combined objective function values and stores the optimum satellite nodes that can be used to get the tasks performed optimally.

Algorithm F.7: Summary of algorithms

```
Use create graph algorithm
function neighbour = neighbours(G)
function (centralities(:,:)) = inputs(G,G1,G2)
function (Eigen_vectors(:,:),svd(:,:)) = inputs(G,G1,G2)
function Optim = inputs(centralities)
function $Optim\_satellites = Opt\_satellites(Optim, neighbours)$
fprintf(Optim_satellites)
plot(Optim_satellites)
```

All these algorithm results can be produced from the Algorithm F**.7** which evaluates and simulates all the functions. This method can be made into a data base that simply in one click operators find the optimum satellites needed to achieve the mission objectives.

When these algorithms are simulated using 1000 possible satellites and 10 targets, 141 of the 1000 satellites are found to view all 10 targets at least once. When this was simulated for satellite pairs using the three different requirements earlier presented in chapter 5 of this dissertation, the satellite pairs that are found to have optimal values that satisfy the mission requirements and have high optimums are 9,512 pairs.

From these, 22 satellites were found to have the highest optimal values for all three requirements. These are satellites were mainly on orbits that have inclination values of 52 degrees and 56 degrees but with different RAAN values. As before, the reason for this can be because the orbit design method considered orbits with INC increments of 1 degree. For more refined cases, the range of orbits obtained in terms of inclination may be larger.

Appendix G: A Comparison of Using the Partitioned Adjacency Matrix for the Bipartite Graph Analysis and the Unpartitioned, Full Adjacency Matrix While Implementing the Singular Value Analysis.

For the graph theory method to identify constellations, it was proposed to compare the bipartite graph approach to the approach of directly analysing the centralities while ignoring the partitions. The results for these are herein presented.

Principal eigenvectors while considering the partitions:

When the adjacency matrix was partitioned as illustrated in equation G.1, i.e.,

$$A^T A = B^x \text{ and } AA^T = B^y \qquad (G.1)$$

The principal eigenvector of the partition compared to the full matrix and eigenvector centrality results were simulated. From this, the patterns of all were similar and are as presented in Figure G.1.

The method by which the adjacency matrix is computed should however be considered. For this, the adjacency matrix A used to compute the eigenvector centrality for example must be a square matrix and so in this case, if the partitions are not applied, is a 60x60 adjacency matrix. A general computation of the adjacency matrix in the case of the 50 satellites and the 10 targets for example, would be a 10x50 matrix in which case the computation of the eigenvectors would not be possible. For this, the option would be to either consider the partition as per equation G.1 or use the singular value decomposition of the matrix.

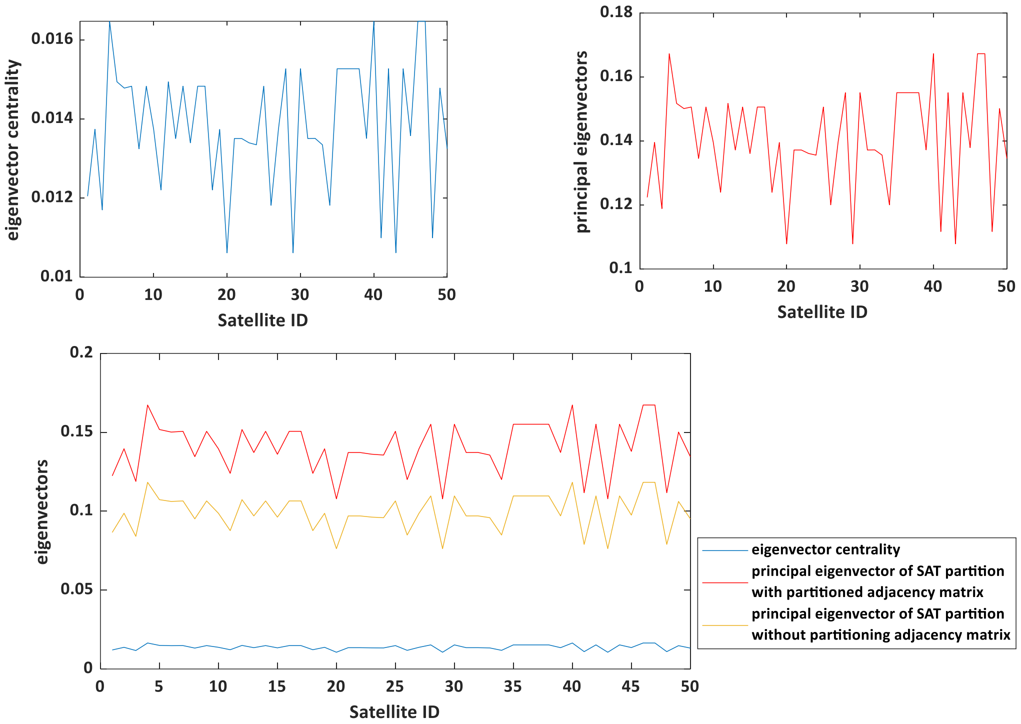


Figure G.1: A comparison graph of the eigenvector analysis of satellite nodes while using partitioned adjacency matrix and when using the full adjacency matrix.

The same analysis is done on the target nodes and Figure G.2 is simulated.

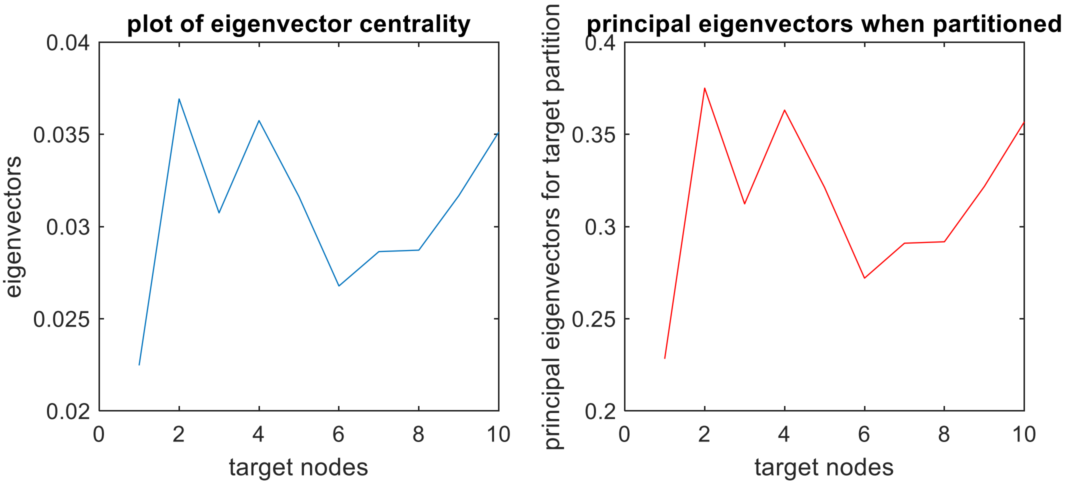


Figure G.2: A comparison graph of the eigenvector analysis of target nodes while using partitioned adjacency matrix and when using the full adjacency matrix.

For the 50 satellites, using the eigenvector analysis of the 60x60 matrix and using the eigenvector analysis of the partitioned graph gives similar trends. Using the eigenvector analysis of the main adjacency matrix, 60x60, in this case does not incur any noticeable losses. When the 10x50 matrix is analysed using singular value decomposition, the analysis of the left singular values results in the same trend. This is presented in Figure G.3 for the satellite nodes and Figure G.4 for the target node analysis.

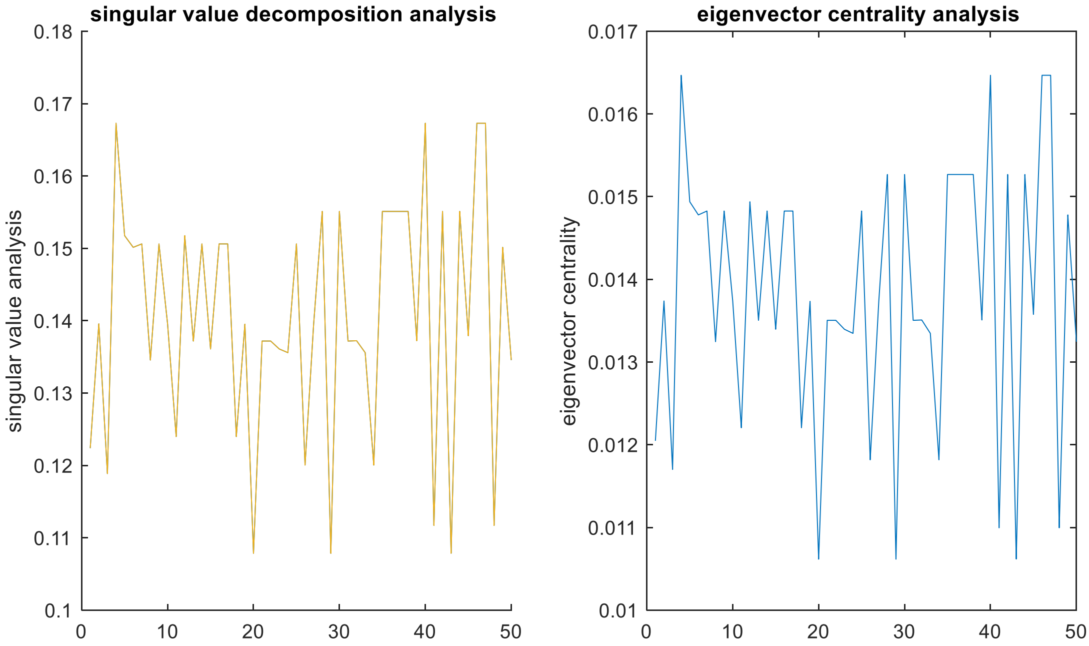


Figure G.3: A comparison graph of the eigenvector analysis and singular value decomposition of satellite nodes while using partitioned adjacency matrix and when using the full adjacency matrix.

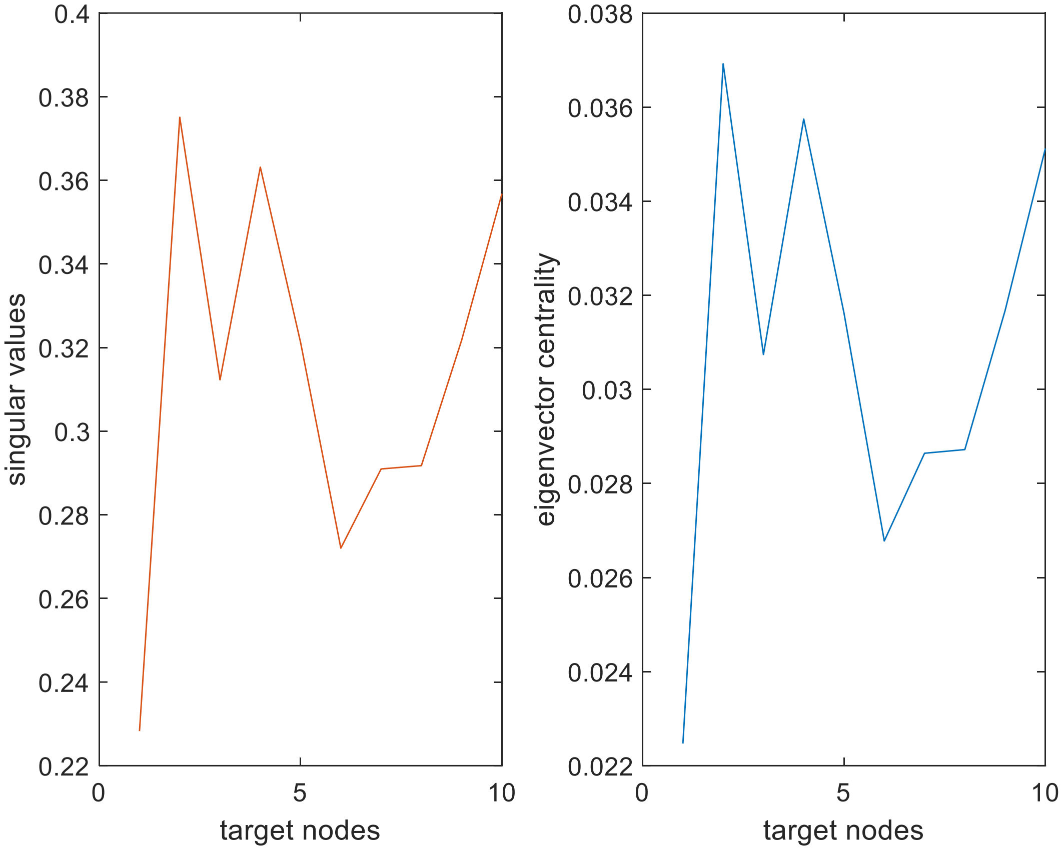


Figure G.4: A comparison graph of the eigenvector analysis and singular value decomposition of target nodes while using partitioned adjacency matrix and when using the full adjacency matrix.

To further validate this, a graph network of 1000 satellite nodes and 10 target nodes was analysed, and the results are as presented in Figure G.5 and Figure G.6.

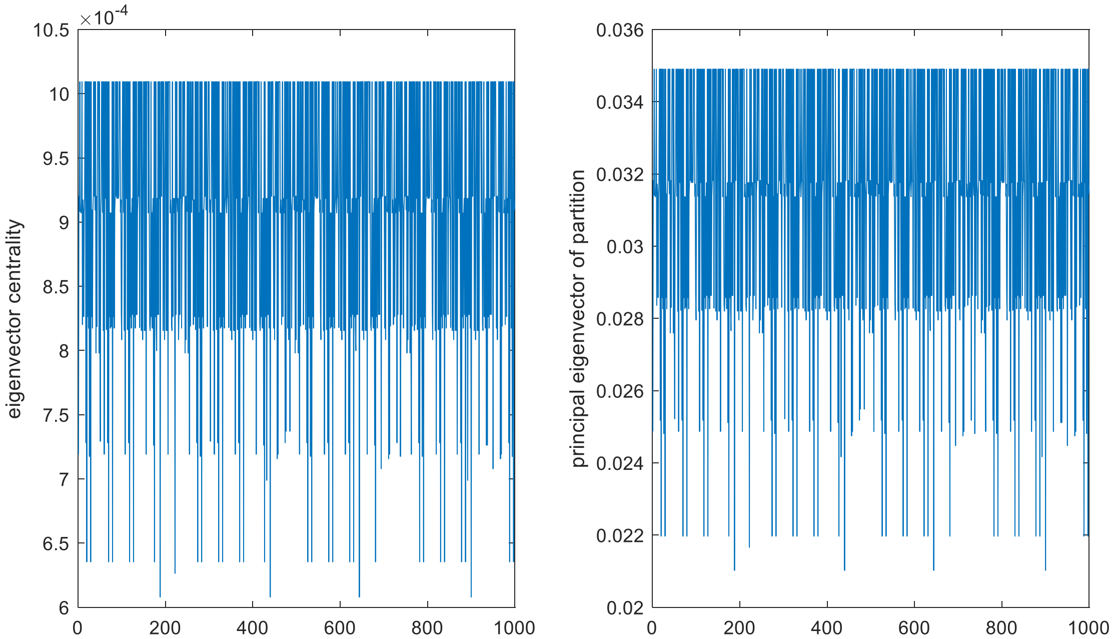


Figure G.5: Satellite node analysis for 1000 orbits considering the eigenvector and principal eigenvector analysis while using partitioned adjacency matrix.

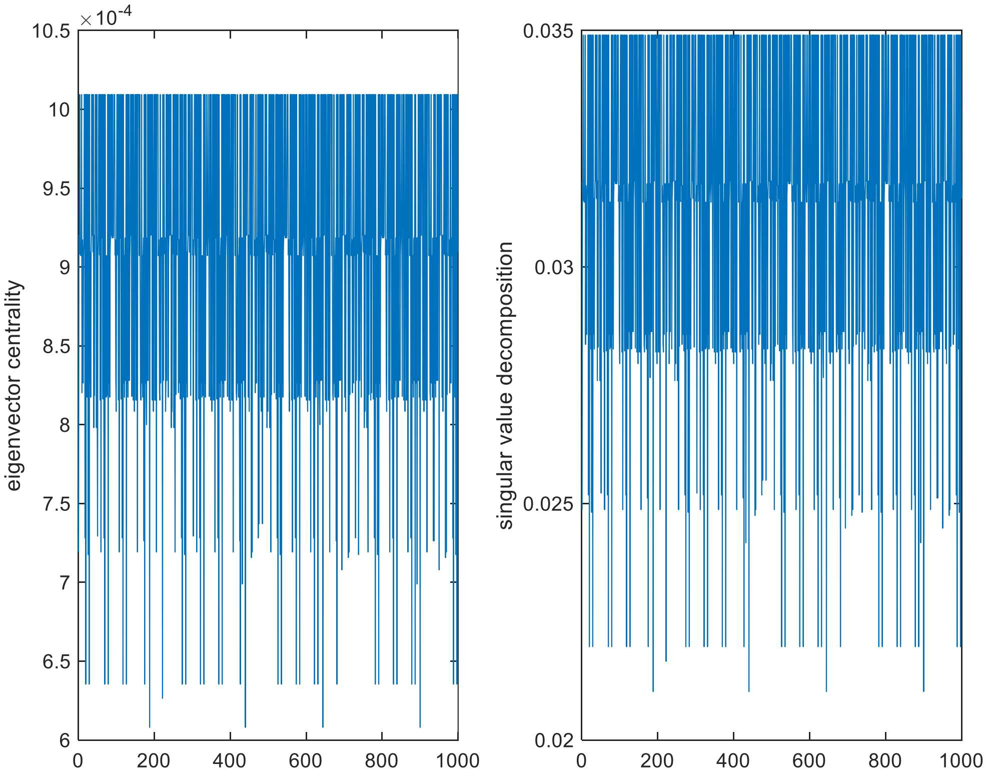


Figure G.6: Satellite node analysis for 1000 orbits considering the eigenvector centrality and the singular value decomposition analysis.

This analysis gives further validation of the method and algorithms used in section 5 for satellite tasking for simple and complex satellite-target networks.