POLITECNICO DI TORINO

Master's Degree in Aerospace Engineering

Master's Degree Thesis

Mission Planning Techniques for Earth Observation Satellite Constellations

Supervisor

Prof. Lorenzo CASALINO

Company Advisors

Eng. Cristoforo ABBATTISTA

Eng. Leonardo AMORUSO

Eng. Marianna CARBONE

Candidate

Diego DIMITRI

APRIL 2023

Summary

Mission Planning is a complex and critical part of space mission design. Classical techniques do not scale well in large constellations of satellites, because of the level of human involvement required in the process. Most recent constellations can count tens and hundreds of small satellites, making a higher level of automation in planning not only desirable, but necessary.

This thesis presents a brief description of modern mission planning problem modelling techniques and their solving algorithms and than studies the development of a Mission Planner capable of scheduling the tasks of a variable number of remote sensing satellites. In order to show the capabilities of the program, a case study with real and realistic satellites is presented.

Acknowledgements

I gratefully acknowledge the assistance of Planetek Italia for its valuable advice and support. In particular, I want to thank engineer Abbattista for being the first to guide me along this path, engineers Amoruso and Carbone for their invaluable help and Dr. Sanfilippo for the bureaucratic aspects.

I am also grateful to Professor Casalino for his willingness to be the Supervisor of this thesis, and I would like to extend my sincere thanks to all the professors who inspired and motivated me during my academic career, both at Politecnico di Torino and at Technische Universität München.

Special thanks go to my friends and fellow students at the universities and, above all, to my family, who have always supported and encouraged me.

Table of Contents

List of Tables

List of Figures

Acronyms

ASI Italian Space Agency

CCSDS Consultative Committee for Space Data Systems

DOD Depth of Discharge

EPS Electrical Power System

ESA European Space Agency

GA Genetic Algorithm

GEO Geostationary Earth Orbit

GPS Global Positioning System

GUI Graphic User Interface

ISS International Space Station

LEO Low Earth Orbit

LIDAR Laser Detection and Ranging

NASA National Aeronautics and Space Administration

NORAD North American Aerospace Defense Command

RAAN Right Ascension of the Ascending Node

RINEX Receiver INdependent EXchange format

SAR Synthetic Aperture Radar

SDP4 Simplified Deep-Space Perturbations-4

SEM System Effectiveness Model

SGP4 Simplified General Perturbations-4

STK Systems Tool Kit

TLE Two Line Element

TT&C Telemetry, Tracking and Command

UTC Coordinated Universal Time

WGS World Geodetic System

Chapter 1

Introduction

Mission planning is a complex but fundamental part of the space mission design. Because of its complexity, it is a time consuming procedure traditionally involving manual interventions. Traditional tools include spreadsheets, timeline tools or custom tools, hard coded to specific problems. Although such tools facilitate the work of the operator who has to plan the tasks of the various satellites, they do not replace him, requiring the figure of the man-in-the-loop, with the associated time and cost demands. The specificity of custom instruments also requires additional time and costs, especially if new needs arise during the operational phase with satellites already in orbit. Finally, very often any form of planning optimisation is totally absent or limited in these instruments. Moreover, since the number of space missions which adopt constellations of satellites is increasing, new techniques for mission planning are required, because it is not possible to manually schedule tens or hundreds of satellites. In fact, the number of satellites in orbit is going to increase more than linearly, reaching about 8000 spacecrafts in 2024 due to constellations only, the 36% of the constellations containing more than 50 elements (while another 16% has unknown size) [1].

1.1 Mission Planning and Scheduling

A definition of *Mission Planning* can be found in the book *Space Mission Analysis and Design*: “Mission planning starts before activity planning. It defines how to use resources best in accomplishing mission goals. Mission planning produces rough activity timelines across mission phases that identify the schedule and resources to complete major activities.” or, in other words, mission planning consists of “deciding what to do and when” [2].

Actually, the Consultative Committee for Space Data Systems (CCSDS) highlighted a problem about terminology concerning the terms *planning* and *scheduling* in the space operations domain [3]. In fact, although in the Artificial Intelligent Planning and Scheduling domain the difference is clear, since activities have to be selected when referring to *planning*, while their set is known and only their arrangement has to be determined in *scheduling*, in space mission design the distinction is not definite. According to the CCSDS, one of the common interpretations discriminates the concepts of *plan* and *schedule* between *high-level* and *executable*. For the purpose of standardisation, the Committee refers to *scheduling* as to *plan execution*. Nevertheless, in this thesis the terms *schedule* and *plan* are often treated as synonymous with the meaning of a list of activities collocated in a timeline. This is to facilitate comprehension and reading of the discussion, as well as to remain consistent with the cited sources. In the event of possible ambiguity, the author will be careful to specify, in accordance with the standard, the correct term being referred to.

Three Mission Planning functions are identified by the CCSDS: *Planning User*, *Planning* and *Plan Execution*.

The first function deals with the instructions given by a user, who submits requests to the Planning function and controls the process. It may require feedback throughout the mission planning. In a real system more Planning User functions could be present and might correspond to other functions within the space mission system. The Planning function is responsible for the actual mission planning. It elaborates the instructions received by the Planning User, builds the set of tasks, dividing them in events, and sends the list of events to the Plan Execution function, which is responsible for the scheduling.

1.2 Challenges for Constellations

As mentioned, the planning of the operations of a single satellite requires a significant human involvement by means of traditional techniques, because of intricate time and resource constraints. The increasing number of satellites in a constellation simply makes the human effort not enough to solve the planning problem. Automation is therefore required. Additionally, recently, more and more satellite constellations are made by nano and micro-satellites. According to the survey carried by Modenini, Curzi and Tortora [1], 32% of the constellations belong to nano-class, while the 18% to micro-class. They are economically advantageous, also thanks to the development and spread of the CubeSat standard, and allow new opportunities and levels of contingency for space missions. Constellations are indeed desirable because they are more robust and flexible, providing a greater ability to adapt to changing expected and unexpected events [4]. That means that in a single constellation there can be tens to hundreds of small satellites, each one equipped with extremely limited resources whose utilisation has to be optimised. In fact, such satellites could have power, memory or thermal constraints due to the contained dimensions and mass. Furthermore, new degrees of freedom appear in the problem, since in the same constellation there could be satellites with different payloads and able to perform crosslink.

1.3 Thesis Structure

This thesis will firstly present the possible problem modelling techniques currently used and then the most common solving algorithms in the space field. Subsequently, a mission planning software developed entirely by the author of this thesis in MATLAB environment will be presented. Finally, that software will be used to plan the daily activities of a constellation of over 20 Earth observation satellites.

Chapter 2

Problem Modelling Techniques

The mission planning problem has to be modelled in order to be solved by simulation algorithms. It can be seen as a combinatorial optimisation problem, where different tasks shall be allocated in given time slots. That kind of problem has been proven to be NP-complete [5] or even NP-hard [6, 7, 8, 9, 5], where NP stands for *Non-deterministic Polynomial-time*. A problem is said NP when a non-deterministic Turing machine (a computer that follows a set of rules which may prescribe more than one action for any given situation) can solve it in polynomial time (the algorithm's running time is upper bounded by a polynomial expression in the size of its input). Additionally, in the mission planning process, the set of tasks is not completely given, therefore it shall be firstly defined.

Zhang et al. [6] studied and classified 500 research papers regarding satellite mission planning. They propose a classification in five task planning models: *constraint satisfaction*, *integer programming*, *backpack*, *directed graph*, *multi-agent*.

2.1 Constraint Satisfaction Model

In a constraint satisfaction problem, there is a set of variables that can assume only specific values and that must obey certain constraints.

In a space mission planning problem, constraints on visibility window and slew angle are important for orbiting sensors and also the task preparation time plays a significant role.

For example, Pemberton and Galiber stated that "Satellite scheduling involves assigning a resource (or set of resources), a start time, and duration to each task in a way that satisfies the constraints on the tasks and resources" [10] and they accordingly translated the satellite scheduling problem into a constraint satisfaction model. They considered four basic objects: tasks, resources, events and their constraints. Task constraints may include continuity, disjunction or causality between tasks. Resources constraints could concern energy and on-board storage memory capacity or the payload sensor and its requirements. Event constraints are used to describe the time windows during which a task can be executed. This kind of constraints may be specified by the user, for example in case of periodic tasks.

In Section 4.6, a constraint satisfaction model is taken as starting point for the problem model of the Mission Planner.

2.2 Integer Programming Model

In a integer programming model, the problem is expressed in a mathematical form, after that it has been abstracted into quantitative data relationships, and involves some integer variables.

Valicka et al. [7] developed a deterministic mixed-integer programming ‘coverage’ model in their study on mission planning dealing with cloud cover uncertainty.

Abramson et al. [11] used a integrated approach using integer programming, network optimisation and astrodynamics to calculate optimised observation and sensor tasking plans. Their initial system model consisted of approximately 100 satellites and large number of points of interest on Earth with the objective to maximise the total science value of observations over time.

Zhang et al. [6] give a example of simplified satellite mission planning problem:

$$\text{Max}\sum_{i\in T}\sum_{k\in S} p_{i,k} \times \left(\sum_{t=1}^{N_{i,k}} x_{i,k,t}\right) \tag{2.1}$$

s.t.

$$\forall i \in T, \sum_{k\in S}\sum_{t=1}^{N_{i,k}} x_{i,k,t} \leq 1 \tag{2.2}$$

$$\forall i,j \in T, i \neq j, \forall k \in S, y_{i,j} + y_{j,i} + 1 \geq \sum_{t=1}^{N_{i,k}} x_{i,k,t} + \sum_{t=1}^{N_{j,k}} x_{j,k,t} \tag{2.3}$$

$$\forall i,j \in T, i \neq j, t_i - t_j + \sum_{k\in S}\sum_{t=1}^{N_{i,k}} x_{i,k,t} \times (e_{i,k} + s_{i,j,k}) \leq Z \times (1 - y_i, j) \tag{2.4}$$

where T is the task set, S the satellite set, $N_{i,k}$ the number of time windows allowed when task i uses resource k, $p_{i,k}$ the profit of satellite k performing task i, $e_{i,k}$ the time for satellite k to perform mission i, $s_{i,j,k}$ the conversion time to continue task j after task i is completed, Z the length of time in the planning cycle, $x_{i,k,t}$ the Boolean variable stating whether the task i is executed in the t-th window of resource k, $y_{i,j}$ the Boolean variable stating whether on the same resource task i is executed before task j. "The objective function 2.1 ensures the maximum total benefit of task execution, the constraint condition 2.2 ensures that a task can only be executed once at most, the constraint condition 2.3 logically specifies the sequence of tasks executed on the same resource, and the constraint condition 2.4 ensures that there can be no intersection during task execution; that is, two tasks cannot be executed on the same resource at the same time" [6].

2.3 General Problem Model

2.3.1 Backpack Model

The backpack problem, also known as knapsack problem, is a classical combinatorial optimisation problem in which a virtual backpack with a given capacity has to be filled by a finite number of items, each with associated value and weight. The problem consists of performing the best selection of items in order to maximise the value, without exceeding the backpack capacity.

A *relaxation mechanism* can be employed: during the search, the capacity constraint may be violated by the current configuration; in that case, the configuration is immediately repaired by suppressing the elements which have the worst value-weight ratio, until the capacity constraint is satisfied.

In the space mission domain, there is a number of tasks to be distributed to a finite number of satellites, which have limited amount of resources. Each task requires consumption of power and memory, giving a observation income.

Vasquez and Hao [8] considered a special instance of a multidimensional knapsack problem, with a single backpack constraint and three additional logical constraints. The knapsack constraint states that the sum of the sizes of the taken images cannot exceed the maximal on-board storage capacity. The logical constraints refer to the non-overlapping of two observations and the minimal transition time between two successive observations on the same sensor and to the limitations on instantaneous data flow through the satellite telemetry resulting from simultaneous observations on different sensors.

2.3.2 Directed Graph Model

The directed graph model, also known as Bayes network, is a probabilistic model in which the conditional dependence structure is represented by a graph.

The nodes (or *vertices*) are random variables whose mutual relationships are represented by arrows (or *edges*) connecting them.

Wu et al. [5] converted a multi-satellites observation scheduling into an acylic directed graph model. Each vertex was corresponding to a possible observation

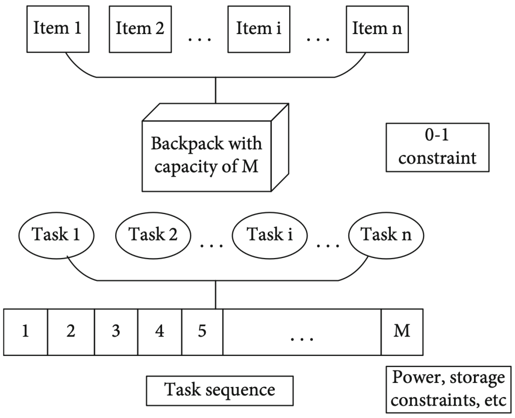


Figure 2.1: Comparison between satellite mission planning problem and backpack problem model [6].

task and the directed edges revealed the possible order of different tasks.

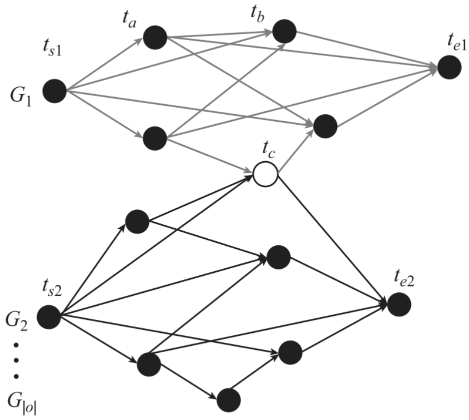


Figure 2.2: Acyclic directed graph model of satellites observation scheduling [5].

2.3.3 Multi-Agent Model

In a multi-agent model, the system is modelled as a set of autonomous decision-making entities called *agents*. Each agent individually assesses its situation and makes decisions on the basis of a shared set of rules. It is a bottom-up modelling method, where reactive and active entities interact with each other. Differently from the other models, the agent-based model has a certain degree of intelligence and autonomy and is, thus, more responsive in dynamic scheduling.

Liu et al. [12] used an agent-based system to create a distributed scheduling algorithm which uses a game-negotiation mechanism for a distributed earth observation constellation. Satellites are rational players that play a cooperative game with the goal of scheduling the largest number of tasks. Being each satellite a independent agent, the mission scheduling model can run on-board, using crosslinks to exchange information across the constellation. The reasons that pushed to this solution were the computation and memory cost of centralised methods, the time delay between satellites and ground stations when emergencies occur and the trend of decentralising system management and control.

Skobelev et al. [13] used a multi-agent approach where each schedule is designed as a flexible network of connected demand and resource agents. The plan is built by self-organised separate agents, competing and collaborating in order to find the solutions profitable for each of them and for the system as a whole.

Van Der Host and Noble [14] created a market-based model where tasks are allocated with a reverse, sealed-bid auction mechanism. If a node has a task to outsource, it assumes the role of *auctioneer* and announces the task to its direct neighbours. If a satellite in the auction community possesses the necessary resources, it submits a bid to the auctioneer. At every step along the way, the intermediate nodes aggregate all bids and only re-transmits the best one. The transmitted bid is increased by a ‘commission’ factor. Nearby under-utilised nodes will be thus preferred. In that way the allocation process executes in parallel across the network, leading to efficient and adaptive task allocation at a global level.

Princeton Satellite Systems [15] developed an agent-based software architecture for autonomous distributed systems. It was based on a message passing architecture, i.e. the communication between agents is done by means of messages passing through message centres which register and validate agents, process messages and pass them to registered agents. The benefits were several: increased level of autonomy on-board the spacecrafts; flight software flexibility and adaptability, because the agents could be dynamically loaded; improved reliability of spacecraft and fleets of spacecraft by incorporating fault detection at both high and low levels; reduced need for large ground support organizations. The same system was used by Schetter et al. [16] for satellites in close formation. They identified four high-level tasks: performing science (imaging), formation maintaining and control, cluster reconfiguration and cluster upgrade. Four types of agents were thus present in the system: the lowest intelligence level could only receive and execute commands and tasks from other spacecrafts or ground stations; the intermediate levels could either local plan their own tasks or interact with other satellites, keeping and updating an internal representation of the cluster; the highest level of intelligence is capable of monitoring all spacecraft-level agents, planning for the constellation as a whole.

Chapter 3

Solving Algorithms

The problem models depicted in the previous chapter can be solved in different ways. Some solving strategies have better performances than others, depending on the characteristics of the problem.

In this chapter the most common division in *exact*, *heuristic* and *meta-heuristic* algorithms is examined, with a brief introduction to more modern methods based on artificial intelligence.

3.1 Exact Algorithms

Exact algorithms can accurately find the optimal solution of a problem by dividing it into simpler sub-problems. Examples of exact algorithms are branch and bound algorithm, dynamic programming and cutting-plane method.

In *branch and bound algorithms*, the problem is branched into simpler solvable sub-problems. The algorithm searches for solutions by exploring these branches in order. To make the search more efficient, upper and lower bounds can be estimated for each branch and thus all branches with a bound which is worse than the best solution found so far can be excluded.

Also in *dynamic programming* the problem is subdivided into smaller sub-problems. Those problems are then solved optimally, using recursive steps if necessary, and the solutions are finally used to construct a optimal solution to the original problem.

The *cutting-plane method* iteratively refines a feasible set of solutions by means of linear inequalities, named *cuts*. It works by solving a non-integer linear problem, then checking if the found optimisation is a integer solution. If it is not, a new constraint is added such that cuts off the non-integer solution, but does not cut off any other points in the feasible set. That process is repeated until the optimal integer solution is found.

Valicka et al. [7] used a branch and bound algorithm to solve the problem presented in Section 2.2.

Usually, exact algorithms are not employed in mission planning problems, because they are poorly computationally efficient in large-scale problems [6]. In fact, exact algorithms explore every feasible solution to a certain problem, implying long computation times.

3.2 Heuristic Algorithms

The term *heuristic* refers to any approach to problem solving or self-discovery that relies on strategies based on past experiences. A heuristic algorithm gives a feasible solution to a problem, trying to minimise the search time, but the result is not necessarily the optimal solution.

Heuristic algorithms are strongly problem-dependent, i.e. there is not a general algorithm structure, but they are specific to the examined problem.

3.3 Meta-heuristic Algorithms

Quoting Osman, "a metaheuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions" [17].

In the following subsections the most employed algorithms in the satellite planning domain [6] are described.

3.3.1 Evolutionary Algorithm

The evolutionary algorithm is a population-based model inspired by nature. It is called *population-based* because it constantly deals with a set of different solutions rather than a single one per iteration. It takes inspiration from the mechanisms of evolution and natural selection. Starting from a initial population, the evolutionary algorithm applies some operators in each iteration. They are typically of two kinds: *recombination* (or crossover) and *modification* (or mutation). The *crossover* operator randomly selects two individuals of the current population and creates a new individual by randomly choosing its characteristics between the ones of the parents. The *mutation* operator produces new individuals by applying random changes to the selected single parent.

Each individual is associated to a value, named *fitness*. The algorithm proposes the mechanism of the *survival of the fittest*, which is present in the evolutionary theories of natural sciences. Therefore, individuals with a higher fitness value have more chances to ‘survive’ and be part of the population of the next iteration, whose number of entities is usually, but not always, kept fixed.

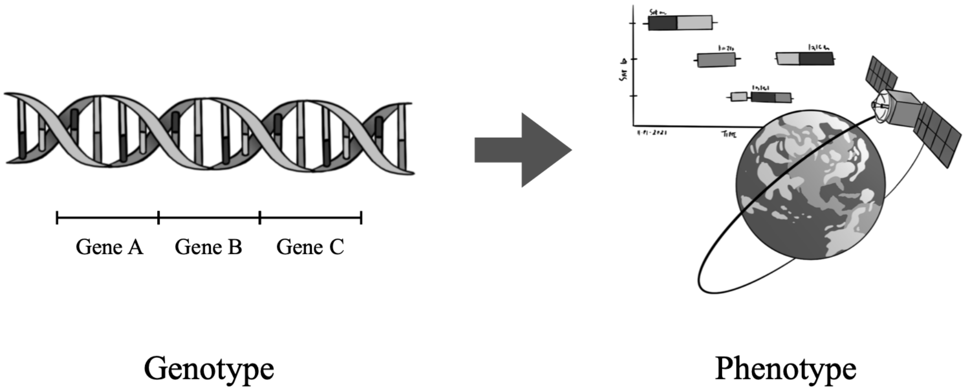


Figure 3.1: Example of genotype and phenotype representation.

Genetic Algorithm

One example of evolutionary algorithm is the genetic algorithm, where the solutions acquire the form of a sequence of characters, called *genes*, in analogy to the DNA structure. In this context, individuals are named *genotypes*, whereas the solutions that are encoded by them are named *phenotypes*.

The genetic algorithm is one of the most employed methods and it is often used as basis for comparison in research papers when suggesting new solving algorithms [5, 12, 18, 19].

A example of genetic algorithm can be found in Section 4.7.2.

3.3.2 Swarm Intelligence Algorithm

The swarm intelligence algorithm is another technique taken from the mechanisms of nature. It is inspired by animals that live in groups, like insects, birds or herds, and whose main activity is foraging. Those animals cooperate and exchange information in order to find food more quickly. The algorithm simulates that behavior for the search of solutions for optimisation problems.

Ant Colony Optimisation

The ant colony optimisation algorithm, as the name suggests, is inspired by the behavior of ants. Those insects, while searching for food, release pheromones on the ground. When a individual is deciding what direction to go, it will prefer paths with higher concentrations of pheromones.

The algorithm is, thus, divided into two parts: *path construction* and *pheromone update*. During the construction phase, ants navigate the search space, moving from one node to the neighbour one according to a probability function depending on the pheromone variable. In the update phase, a global procedure updates pheromone values, considering the additional pheromones the ants released on the edges of their paths.

Iacopino et al. [18] applied an ant colony optimisation algorithm in the mission planning of a disaster monitor constellation. They built a multi-agent architecture with ant-like agents, where the assignment problem was modelled by a binary chain. The spacecraft camera was a binary reusable resource strictly dependent on a depletable resource, the on-board storage memory. The imaging was an activity which consumed memory, whereas the downlink produced memory. They showed that in such problem the ant colony algorithm had better performances than a genetic algorithm dealing with dynamic scenarios and constellations.

A hybrid form of the ant colony optimisation algorithm was employed also by Wu et al. [5] to solve the problem presented in Section 2.3.2. Their model could deal both with common and emergency tasks, outperfoming simulated annealing and genetic algorithms.

Particle Swarm Optimisation

In the swarm particle optimisation algorithm, a population of individuals called *particles* moves in steps in a search space. At each step the velocity of each particle is updated according to a function that links the velocity to the value that the objective function has in the current point.

The source of inspiration of the algorithm is the behavior of flocks of birds and swarms of insects. In fact each individual is attracted to the best locations that it or any of the other particles has previously found.

An example of swarm particle algorithm was employed by Chen et al. [20], in order to optimise the solutions obtained by a planner based on a bid auction mechanism and which was able to deal with new and urgent tasks.

3.3.3 Local Search Algorithms

Differently from the previous methods, local search algorithms are not population-based, but rely on a single point search, i.e. they work on single solutions. They are called *trajectory methods* since during the process they describe a trajectory in the search space starting from a initial solution. Lemaître et al. [9] showed that this type of algorithms have better performances than greedy algorithms, dynamic programming algorithms and constraint programming approaches, although the latter one is more flexible.

Tabu Search Algorithm

The tabu search algorithm uses a short term memory in order to escape local minima. It relies on the creation of the history of the search, called *tabu table* or *tabu list*, which takes track of the last visited solutions, forbidding moves toward them. That process avoids recursive cycles, forcing the algorithm to even accept up-hill moves, i.e. moves that brings the trajectory to less valuable solutions, but allow the algorithm to escape stagnation points.

According to Blum and Roli, “the implementation of short term memory as a list that contains complete solutions is not practical, because managing a list of solutions is highly inefficient. Therefore, instead of the solutions themselves, solution *attributes* are stored” [21]. *Attributes* can be the differences of the values of the solutions, specific components or the moves themselves. Each category has a personal tabu list and the set of the attributes and the relative list define the tabu conditions applied in the search path. In order to overcome the possibility that a good quality solution is avoided because of the tabu list, an aspiration criteria (the conditions that are used to select the allowed set of possible moves, even when they should be avoided according to the tabu conditions) might be adopted.

Two mechanisms are often introduced into the method: *intensification* and *diversification*. By *intensification*, the search focuses more on regions of the space, or characteristics of solutions, that are promising according to the search history. *Diversification* undertakes to explore regions that significantly differ from regions previously visited.

Vasquez and Hao [8] applied a tabu search algorithm in the problem already mentioned in Section 2.3.1, whereas Bianchessi et al. [22] developed a tabu search algorithm for a Multiple Satellite and Multiple Orbit Problem, introducing upper bounds based on column generation, in order to asses the quality of the solutions.

Simulated Annealing Algorithm

The basic idea of the simulated annealing algorithm is to allow up-hill moves in the trajectory. This strategy is meant to avoid stagnation in local minima, aiming at the search of a global minimum.

The process simulates the annealing treatment of metals and glass. The algorithm starts by generating a initial solution and by initialising the *temperature* parameter. At each iteration, a new solution is sampled and accepted depending on the values of the function and the temperature, which decreases after each step. As metals and glass recrystallise when cooled, in the same way the probability of accepting up-hill moves decreases with the temperature parameter. In this way, at the beginning the algorithm explores more easily the search space, slowly converging to a minimum as temperature value decreases. The cooling schedule assumes a crucial role in the performance of the algorithm, exactly as it does in the heat treatment of material science.

Wu et al. [5] presented an adaptive simulated annealing algorithm aggregated with a dynamic task clustering strategy. They employed three mechanisms: adaptive temperature control, tabu-list to avoid short-term revisiting, combination of two neighborhood structures.

3.4 Artificial Intelligence Algorithms

As stated by Zhang et al., "at present, the use of artificial intelligence method to solve the task planning problem is in its infancy" [6].

Li et al. [19] employed artificial intelligence to improve on-board scheduling. Genetic algorithms are computationally demanding and therefore they are not always suitable to run on-board. Nevertheless, on-board scheduling is desirable, in order to give the system robustness and fast adaptability. A neural network was trained offline, in ground stations, based on the optimised historical schedules obtained by the genetic algorithms, and then the method was tested on simulations, showing enhanced performances.

Chapter 4

Mission Planner Development

After presenting in the previous chapters the different mission planning techniques currently employed in the space domain, this thesis wants to present a original example of Mission Planner, developed in MATLAB environment. The aim of aforementioned planner is to be able to handle constellations of collaborative satellites, i.e. a certain number of spacecrafts which, despite having personal resources, share mission goals and collaborate with each other to achieve them. The planner shall be easily scalable and flexible to the architectures desired by the user.

Even though the software could handle different types of constellations, in order to reduce complexity and at the same time to reach an adequate level of fidelity to the reality, the focus of the planner is on remote sensing constellations.

4.1 Introduction to the Mission Planner

The Mission Planner allows the user to create a model of the space mission and to solve the planning problem. The mission model is made of a constellation of satellites equipped with limited resources that shall observe a given set of targets and transfer the acquired data to the ground segment, respecting certain constraints. Therefore the planning problem consists in scheduling the target observations and the downlinks in order to accomplish that mission.

There are four different types of objects: satellites, targets, ground stations and constraints. In the following sections it will be presented in more detail how these entities are translated into the model and what are the inputs the user shall insert to shape the model to the real scenario.

In particular, satellites are treated as independent agents that greedily choose their own activities according to a meta-heuristic algorithm. Scheduled activities must fulfil some conditions and they provide a payoff based on their characteristics. Therefore, for each iteration, the genetic algorithm that has been chosen for the Mission Planner firstly creates a population of feasible solutions and then selects and modifies them, with the crossover and mutation mechanisms, according to that fitness value. The best solution in the population is finally selected as scheduled task for the satellite. In order to make the constellation collaborative, a coordination mechanism is then responsible for the optimal selection of activities to resolve any conflicts.

The output of the algorithm is the mission schedule, which is a table containing the Source, the Target, the Start Time, the End Time and the Duration of every single scheduled activity. The obtained plan can also be adapted to further needs that may arise during the plan execution and which the user can communicate to the software.

4.2 Scenario Creation

The space mission simulation is performed using MATLAB's *Aerospace Toolbox* [23]. This toolbox has a object, called `satelliteScenario`, that represents a 3D arena consisting of satellites, ground stations and their interactions. The function which creates the scenario requires the start time, the end time and the sample time as inputs. In the planner, two distinct scenarios are generated. The first one is used for the propagation of satellites' orbits. The second one is where the actual planning takes place. The reason for that is to reduce the temporal size of the simulation concerning the plan execution, since the planning process is computationally demanding.

4.3 Satellite Model

The satellites are introduced into the scenario by the creation of a `Satellite` object. This object is then enriched by other variables stored in a table like Table 4.1. The variables are used to model the payload, the communication system and to define energy and storage memory on-board capacity.

4.3.1 Orbit Definition and Propagation

The toolbox allows different ways of defining the orbit of a satellite: by the ephemeris; the Keplerian elements defined in the Geocentric Celestial Reference Frame; the System Effectiveness Model (SEM) almanac file; the Two Line Element (TLE) file;

Satellite	
Name	A string containing the name of the satellite.
Keplerian elements	The orbital elements which define the satellite's orbit.
Payload	A string containing the typology of the sensor of the payload.
Night	A Boolean variable which states whether the sensor can work during night or not.
DataStorage	The maximum capacity of the on-board memory storage in Gb.
PayloadPower	The power required by the payload to perform a observation in W.
PayloadDataRate	The bit rate of the data obtained by an observation in Mbps.
SensorApertureTime	The duration of the aperture and closure of the sensor in seconds.
ObsTime	The duration of the payload acquisition in seconds.
MaxSensorTime	The maximum duration of a observation in seconds.
BasePower	The base power consumption of the satellite bus in W.
SolarCharging	Average power production of the solar array in W.
EnergyStorage	The maximum capacity of the on-board batteries in Wh.
DOD	The depth of discharge.
Frequency	A string containing the band the satellite uses for communications.
LinkPower	The power consumption of the communication system during downlink in W.
LinkMinDuration	Minimum duration of a downlink in seconds.
LinkMaxDuration	Maximum duration of a downlink in seconds.
LinkDataRate	Bit rate of the data transfer during downlink in Mbps.

Table 4.1: Required inputs for a satellite.

the Receiver INdependent EXchange format (RINEX) navigation message output, in case of GPS or Galileo constellations.

The Mission Planner can alternatively accept Keplerian elements and TLE files.

The six Keplerian elements uniquely define the orbit and the satellite's position within the orbit. They are [2]:

- *semi-major axis a*: half of the length of the longest diameter of the ellipse;
- *eccentricity e*: describes the shape of the ellipse. It is the ratio between the distance to the focus and the distance to the directrix;
- *inclination i*: the angle between the angular momentum vector h and the unit vector in the Z-direction;
- *right ascension of the ascending node (RAAN) Ω*: the angle from the vernal equinox to the ascending node (the point where the satellite passes through the equatorial plane moving from south to north). Right ascension is measured as a right-handed rotation about the pole, Z;

- *argument of periapsis* ω: the angle from the ascending node to the eccentricity vector measured in the direction of the satellite's motion. The eccentricity vector points from the centre of the Earth to perigee with a magnitude equal to the eccentricity of the orbit;

- *true anomaly* ν_o: the angle from the eccentricity vector e to the satellite position vector r_o, measured in the direction of satellite motion.

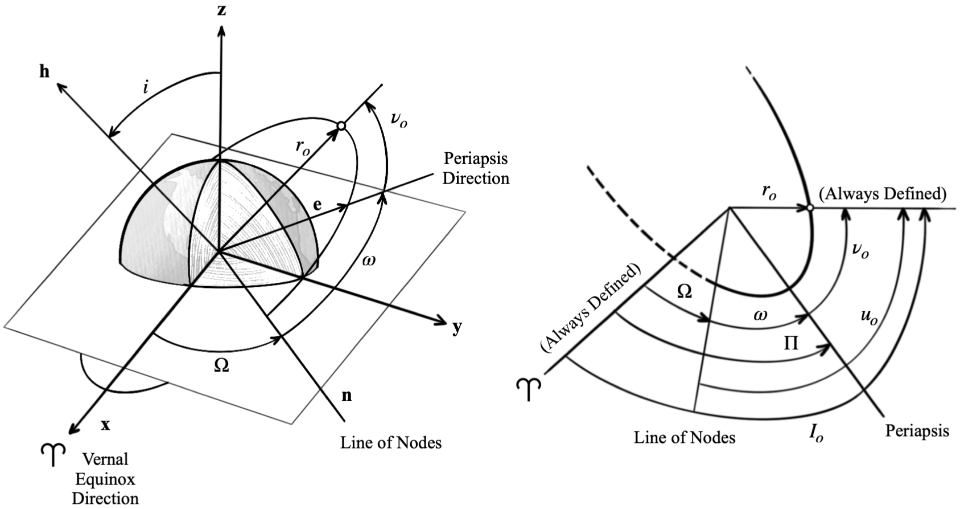
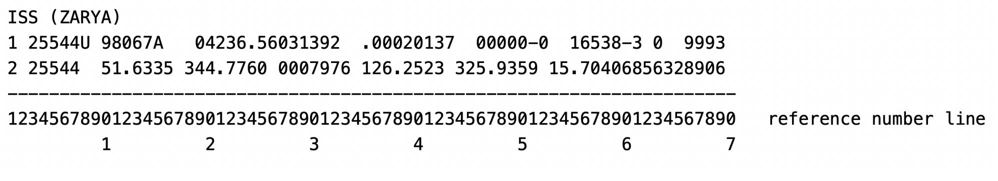


Figure 4.1: Definition of the Keplerian elements of a satellite in a elliptic orbit.

The Two Line Element (TLE) is a data format encoding a list of orbital elements of a Earth orbiting object for a given point in time, the *epoch*. It was created by the North American Aerospace Defense Command (NORAD) and the National Aeronautics and Space Administration (NASA) and it became a *de facto* standard.

A example of TLE file is the one in Figure 4.2 and the description of the format can be found in Table 4.2.



```
ISS (ZARYA)
1 25544U 98067A   04236.56031392  .00020137  00000-0  16538-3 0  9993
2 25544  51.6335 344.7760 0007976 126.2523 325.9359 15.70406856328906
----------------------------------------------------------------------
1234567890123456789012345678901234567890123456789012345678901234567890   reference number line
         1         2         3         4         5         6         7
```

Figure 4.2: A example of Two Line Element (TLE) file for the ISS [24].

Not considering the dedicated propagators for ephemeris, GPS and Galileo,

Line 0		
Columns	Example	Description
1-24	ISS (ZARYA)	The common name for the object based on information from the Satellite Catalog

Line 1		
Columns	Example	Description
1	1	Line Number
3-7	25544	Satellite Catalog Number
8	U	Elset Classification
10-17	98067A	International Designator
19-32	04236.56031392	Element Set Epoch (UTC) Note: spaces are acceptable in columns 21 and 22
34-43	.00020137	1st Derivative of the Mean Motion with respect to Time
45-52	00000-0	2nd Derivative of the Mean Motion with respect to Time (decimal point assumed)
54-61	16538-3	B* Drag Term
63	0	Element Set Type
65-68	999	Element Number
69	3	Checksum

Line 2		
Columns	Example	Description
1	2	Line Number
3-7	25544	Satellite Catalog Number
9-16	51.6335	Orbit Inclination (degrees)
18-25	344.7760	Right Ascension of Ascending Node (degrees)
27-33	0007976	Eccentricity (decimal point assumed)
35-42	126.2523	Argument of Perigee (degrees)
44-51	325.9359	Mean Anomaly (degrees)
53-63	15.70406856	Mean Motion (revolutions/day)
64-68	32890	Revolution Number at Epoch
69	6	Checksum Back to top

Table 4.2: Description of the Two Line Element (TLE) Format [24].

three types of orbit propagators are already available in the toolbox: Two-Body-Keplerian, Simplified General Perturbations-4 (SGP4) and Simplified Deep-Space Perturbations-4 (SDP4).

The Two-Body-Keplerian propagator assumes that the gravity field of the Earth is spherical and neglects all other sources of orbital perturbations. It is thus the least accurate.

The SGP4 propagator accounts for secular and periodic orbital perturbations caused by the oblateness of Earth and atmospheric drag, while the SDP4 accounts for third-body interactions (solar and lunar gravity) and it is thus the more accurate among them. The default orbit propagator for `satelliteScenario` is SGP4 for satellites whose orbital period is less than 225 minutes and SDP4 otherwise. The TLE files requires SGP4 and SDP4, because that data representation is specific to Simplified General Perturbations models. The accuracy that can be obtained with the SGP4 orbit model for a LEO satellite is on the order of 1 km within a few days of the epoch of the element set [25].

The Mission Planner distinguishes between the epoch from which the propagation starts and the start time of the planning request, in order to give more flexibility to the user.

When a satellite constellation is added, the planner firstly propagates the orbit starting from the epoch of the orbital elements or of the TLE files to the start time of the planning request using SGP4 or SDP4, depending on the orbital period. Then, the user can choose between extending the propagation to the stop time with SGP4 (or SDP4) or using a numerical propagator. The numerical propagator is the one available in Simulink's *Aerospace Blockset* [26]. The block uses spherical harmonics for Earth, adding increased fidelity by including higher-order perturbation effects accounting for zonal, sectoral, and tesseral harmonics. That option is not available in the standalone app version of the Mission Planner as it requires Simulink.

Every propagator accumulates errors of the order of the kilometer each day. Plan start times that are too distant from the epoch of the TLE files should therefore be avoided. Anyway the errors in the orbit are only one of the problems related to long term mission planning. Abramson et al. believe that planning activities over extended periods of time at a high level of detail is both futile and impractical. "Futile because detailed actions planned on the basis of a specific prediction of the future may become obsolete well before they are to be executed due to an inability to accurately predict the future. Impractical because the computational resources required to develop detailed plans over extended periods of time may be prohibitive either in cost or availability or both" [11].

4.3.2 Payload Model

Payloads on a satellite can be of several types, e.g. observation, communication, navigation. In this thesis only observation payloads involved in remote sensing are considered, but the implementation of other kinds of payloads could be one of the future developments of the software.

A payload sensors shall point to a subject, collect the electromagnetic radiation, convert them into signals and data and process them into information. Earth observation satellites employ wavelengths that can pass through the atmosphere without being excessively absorbed, like visible, infrared and radio wavelengths.

Payload sensors can be passive or active. A passive system records the radiation naturally reflected or radiated by the subject. An active system uses its own source of energy to measure the reflected radiation. Examples of passive sensors are cameras, whereas examples of active sensors are SAR and LIDAR.

In the software, the parameters of the payload are the typology, the power consumption, the acquisition time and the data rate.

There are several constraints on the sensor utilisation. Firstly, the electrical power must be sufficient for the observation. Then, there must be enough available memory aboard. Passive payloads do not operate during night, since they need solar light. There is also a opening and closure time of the sensor, which may include average slew manoeuvre time, and a maximum operating duration, which may be affected by temperature or any other limitations.

4.3.3 Electrical Power System

The Electrical Power System (EPS) model simply consists of a variable which represents the power storage capacity of the on-board batteries. Its behaviour is approximated to be linear, in order to lighten the model and especially to keep it easily adjustable and flexible to user's necessities.

When the satellite is in direct sun condition, the stored energy increases linearly, according to an averaged power recharging rate due to the solar array. The batteries are only allowed to discharge of a certain percentage, the so-called depth of discharge (DOD), in order to limit deterioration. This value is chosen by the user and it is usually 60-75% for GEO and maximum 30% for LEO.

In this model there are three sources of power consumption: the satellite bus, the payload and the communication system. For each of them an averaged power consumption rate is required as input from the user. The satellite bus includes all the subsystems that are necessary for the spacecraft to work, but that are not directly involved in imaging and downlink activities. It constantly requires electrical power, while payload and communication system only consume energy when their activities are scheduled.

4.3.4 Thermal Constraints

In the space environment, spacecrafts are exposed to external heating sources, like the Sun radiation, the albedo (the solar heating reflected by the planet) and the planets' infrared radiations, as well as internal heat, due to the dissipation of electrical energy of onboard components. The only way an object can emit heat is through its own radiation. The thermal control system can be passive or active. Passive thermal control maintains component temperatures without using powered equipment, while active thermal control requires electrical energy.

Although the design of the spacecraft should already maintain the components in the limits of their operational and survival temperatures, the thermal constrains are important also for the mission planning process because scheduling an activity which involves subsystems that dissipates heat could possibly lead to overheating.

Therefore, during the development of the software, a MATLAB script which computes the temperature of the satellites during the mission was written. Such function approximated the spacecraft to a homogeneous sphere, with a certain specific heat, absorptivity and emissivity. That approximation often gave unrealistic results and did not show the real behaviour of the subsystems. Moreover, it added complexity and computational time to the algorithm.

Aware that a more accurate thermal model would have meant unacceptable slowness of the algorithm, it has thus been decided to express the thermal constraints by means of other parameters: the maximum operating duration of the payload sensor and of the communication system. That is certainly an approximation which underestimates the capabilities of the satellite, since the duration while in eclipse is the same as in sunlight, but it is sufficient to avoid heating issues during the plan execution, because it is assumed that the maximum operating duration is one of the known design values.

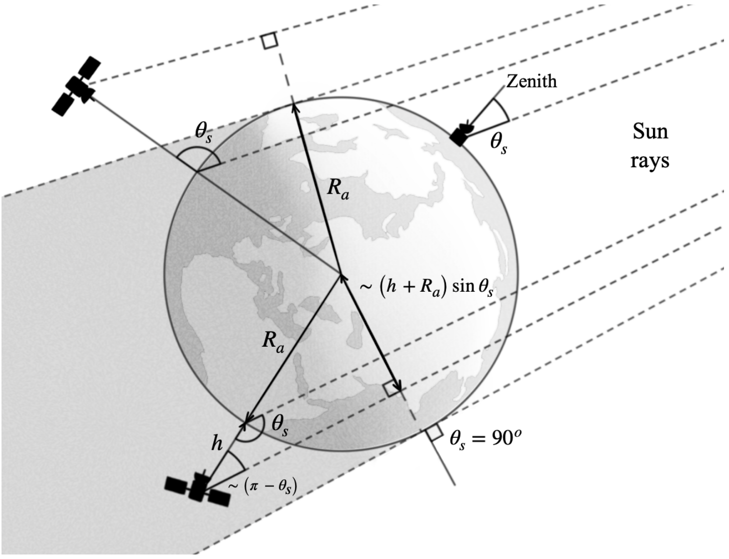


Figure 4.3: Geometric relationship between solar zenith angle and eclipse.

4.3.5 Eclipse Time

As mentioned, the information whether the satellite and the observation target are under direct sun light is important for EPS and sensors.

Reda and Andreas [27] published a step by step procedure for implementing an algorithm to calculate the solar zenith and azimuth angles in the period from the year -2000 to 6000, with uncertainties of ±0.0003°. That algorithm has been implemented in a MATLAB script.

Assuming that the celestial bodies are spherical in shape, Ortiz-Longo and Rickman [28] developed a method to identify shadow terminator points. A planet generates two conical projections: the *umbra* (the darkest part of the shadow) and the *penumbra* (the partial shade). To simplify, in this thesis a object is considered in eclipse even while in penumbra.

Therefore, from the solar zenith angle it is determined if an object is in eclipse if these two conditions are satisfied:

$$\theta_s > 90^o \tag{4.1}$$

$$(h + R_e) \sin\theta_s < R_e - (h + R_e) \cos\theta_s \tan\alpha_p \tag{4.2}$$

where θ_s is the solar zenith, h is the altitude of the object, R_e is the Earth radius and α_p is the penumbral subtended angle, which can be calculated with

$$\alpha_p = \sin^{-1} \frac{D_s + 2R_e}{2\delta_{e-s}} \tag{4.3}$$

where D_s is the diameter of the Sun and δ_{e-s} is the Earth to Sun distance.

4.3.6 On-Board Data Storage

Each satellite is equipped with a finite data storage capacity. The memory decreases if a observation occurs and increases if data are transferred by a downlink to a ground station. The storage cost of Telemetry, Tracking and Command (TT&C) data is neglected, because this is in general orders of magnitude lower in volume than observation data.

The maximum value of storage memory is an input from the user. The minimum is zero. No imaging can occur while the available memory variable is zero; similarly, no downlink is scheduled if the on-board data storage is completely empty, i.e. the memory variable is equal to the maximum data storage capacity.

4.3.7 Communication System

The satellites are equipped with a communication system with the purpose of exchanging data with the ground stations. These data can be either TT&C data or mission data. As outlined in the previous section, the first ones are neglected, but it is still important to keep in mind that they are transmitted only when a link between satellite and ground station occurs.

The quantity of data represents the total information that the communication system shall send. It is proportional to the data rate, the quantity of information per unit time. The time duration is given by the difference between the duration of the visibility window and the time needed to initiate the link, approximately 1-2 minutes. Moreover, the link does not start as soon as satellite and ground station are in line of sight, but it requires a minimum elevation angle to overcome atmospheric losses. Both data rate and minimum elevation angle are inputs from the user.

As mentioned before, also the communication system needs electrical power, so the user shall insert the required subsystem power.

4.4 Ground Stations Model

Ground stations are defined by their geographical position, expressed by longitude, latitude and altitude, and by the minimum elevation angle needed to start the downlink (Table 4.3), where *altitude* is the height above the World Geodetic System 84 (WGS 84) ellipsoid in meters. Each ground station can only link to a satellite per time. If multiple links are simultaneously possible, additional identical ground stations shall be created by the user. It is also possible to specify the frequencies supported by the available antennas. The stations are introduced into the scenario as `GroundStation` objects.

Ground station	
Name	A string containing the name of the ground station.
Latitude	Geographical latitude in degrees.
Longitude	Geographical longitude in degrees.
Altitude	The height above the WGS 84 ellipsoid in meters.
MinElevationAngle	The minimum elevation angle, in degrees.
Frequency	A string containing the bands the station supports.

Table 4.3: Required inputs for the ground segment.

4.5 Targets Definition

Also targets are modelled as `GroundStation` objects with longitude, latitude and altitude. They have additional information stored in a table like Table 4.4: a priority factor, which indicates the importance of the target; the type of observation; the minimum and the maximum elevation angles. Those are all inputs from the user.

Establishing the priority of tasks is important because in some situations some target observations could be in conflict and it is fundamental to decide which one should be scheduled later or even be left unscheduled.

The type of observation is necessary to decide which satellite can observe the target in order to extract the desired information from the acquired data.

The minimum elevation angle is a constraint given by the payload sensor and the required performances, while the maximum elevation angle could be relevant to the purpose of the observation, since it could be important that the target is observed only under a given inclination.

Target	
Name	A string containing the name of the target.
Priority	A integer number expressing the priority of the target. Note: higher the value, higher the priority.
Type	The type of observation required.
Latitude	Geographical latitude in degrees.
Longitude	Geographical longitude in degrees.
Altitude	The height above the WGS 84 ellipsoid in meters.
MinElevationAngle	The minimum elevation angle, in degrees.
MaxElevationAngle	The maximum elevation angle, in degrees.

Table 4.4: Required inputs for the target definition.

4.6 Planning Problem Model

In Chapter 2 several problem modelling techniques have been depicted. For the purpose of the development of a mission planner, taking some papers as inspiration [5, 10, 12, 13, 29], a constraint satisfaction model has been created as starting point. Indeed, a observation has the following constraints:

- *window constraint*: the satellite is in line of sight with the target only in some visibility windows. The observation must happen exclusively in that time;
- *elevation angle constraint*: sensors usually have a minimum elevation angle that must be guaranteed in order to work. Moreover, some satellite applications might also require a maximum elevation angle;
- *memory constraint*: the imaging activity produces a quantity of data which must be stored in the on-board memory. Therefore, sufficient memory must be available in order to perform the payload acquisition;

- *energy constraint*: the satellite must have sufficient electrical energy in order to turn on the payload and perform the observation;
- *payload constraint*: the satellite's payload sensor must be compatible with the type of the requested observation;
- *light constraint*: some sensors require sun light to work or to have good performances. Such type of observations must be scheduled only when the target is not in shadow;
- *duration constraint*: a maximum duration for consecutive payload acquisitions is given. It shall account for thermal constraints too;
- *target uniqueness*: each target shall be scheduled only once. If more than one image of the same target is desired, it must be entered multiple times.

A downlink has similar constraints, but also some differences:

- *window constraint*: to settle a link, the ground station and the satellite must be in line of sight, just as a observation;
- *elevation angle constraint*: the link requires a minimum elevation angle;
- *memory constraint*: the downlink activity transfers data from the on-board memory to the ground segment. If there are no data stored in the satellite, the downlink has no sense. That is the case because in this implementation TT&C data are neglected;
- *energy constraint*: the communication system needs a certain amount of electrical energy which the satellite must have;
- *duration constraint*: minimum and maximum duration for a link is given;
- *ground station availability*: ground stations can usually link only to a finite number of satellites. While scheduling a downlink, the planner must take into consideration this aspect.

To support the constraint satisfaction model, the system has been modelled as a multi-agent system, where the entities are the satellites, the ground stations and the observation targets. Satellites are *active* agents, whereas ground stations and targets are *passive*. This model, as mentioned in Section 2.3.3, has the advantage of being responsive and easily adaptable to different kinds of missions.

Satellites can thus choose between scheduling the imaging of a target or the downlink with a ground station. The available renewable resources are storage memory and energy.

Another constraint is given by the *non-simultaneity* of scheduled actions, i.e. a satellite can not simultaneously perform multiple actions.

4.6.1 Payoff Function

In order to establish the quality of a solution in comparison with the other ones, a function that computes a payoff is needed.

In their model, Abramson et al. [11] considered several options for calculating the science value of observations: total observation time, number of observations and the quality (a function of sensor type, range, slant angle) of the observations. However, their payoff concerned the total solution, while for the Mission Planner a function which gives a fitness value to each scheduled activity is needed. Therefore the elements which have been chosen to affect the payoff are the quantity of transferred data and the position in the timeline for downlinks and the priority, the quality and again the position in the timeline for observations.

The function shall produce a payoff only if the given solution satisfies the aforementioned constraints; if it is infeasible, it shall generate a penalty. In fact there are three ways to deal with infeasible solutions according to Blum and Roli [21]. The most simple action is to *reject* them, but it could be difficult in some cases. Therefore, the strategy of *penalising* infeasible solutions in the payoff function is sometimes more appropriate or even unavoidable. The third possibility consists in trying to *repair* a infeasible solution. As mentioned, the second method has been preferred.

Regarding downlinks, the goal is to gather the data transfer in few links, in order to reduce the costs related to the use of ground stations but also to reduce their workload, since, as highlighted by Curzi, Modenini and Tortora [1], constellations management will be one of the future challenges due to the increasing number of satellites in orbit. Nevertheless, payload acquisition data age and in some cases they shall be transferred as soon as possible to keep their science value. The function shall thus take advantage of the whole duration of the visibility windows and shall try to schedule downlinks as early as possible. A way to estimate the quantity of transferred data is to consider the duration of the data transfer in comparison to the time needed to initiate the link. In fact, since that latter quantity is fixed, because it is decided by the user, a greater percentage of data transfer during a link means a longer downlink and thus a greater amount of data which is transferred to the ground segment. Concerning the time slot in the schedule, the starting option was to link a earlier scheduled activity to a greater payoff. The drawback of such solution was that the algorithm would prefer two consecutive links to different ground stations rather than one single link of the same total duration, while the second option is clearly better. Therefore, the function does not associate a payoff with time, but a penalty, which becomes larger the later the activity is scheduled in the plan. For consistency, the same considerations concerning the position in the timeline hold for observations. The link between payoff and priority of the target is immediate: more important scheduled targets shall have higher payoff. The quality of the payload acquisition is instead estimated by the maximum elevation angle the satellite reaches during the activity. The closer that value is to the maximum value chosen by the user, the higher is the fitness value of the observation, since it is supposed that it is the best operational condition for the sensor.

Therefore, the payoff u of the k-th activity for observations is equal to

$$u_k = w_p \frac{p_i}{p_{max}} + w_\epsilon \frac{\epsilon_{k,max}}{\epsilon_i} - w_{t1} \frac{t_k^{start}}{t_{max}} \tag{4.4}$$

and for downlinks it is

$$u_k = w_l \frac{1 - \Delta t_j}{\Delta t_k} - w_{t2} \frac{t_k^{start}}{t_{max}} \tag{4.5}$$

where p_i is the priority factor of the i-th target; p_{max} is the highest priority; $\epsilon_{k,max}$ is the maximum elevation angle during the k-th activity; ϵ_i is the maximum possible elevation angle for the i-th target; t_k^{start} is the start time of the activity; t_{max} is the stop time of the whole plan request; Δt_k is the duration of the activity; Δt_j is the minimum duration of the link for the j-th satellite; w_p, w_ϵ, w_l, w_{t1} and w_{t2} are respectively the weights for the priority factor, the elevation, the transfer duration and the position in the timeline for observations and downlinks.

The values of the weights are the product of further considerations. Observations shall always be preferred to downlinks, when feasible. Therefore the weight of the priority factor shall be at least one order of magnitude bigger than the one concerning the link. Moreover, a link should be positive in most cases, because data need to be transferred even in the worst conditions. The weight related to time has then been chosen ten times smaller than the one concerning the duration. The same proportion is valid also for observations. The relationship between elevation weight and time weight is the following: an increment of elevation of 10% shall be equal to an increment of 2% in time, i.e. in a plan of 24 hours, in order to have an increment of 10% in the quality of the observation, the maximum permissible delay is 28.8 minutes.

Therefore, selecting arbitrarily $w_p = 1000$, we obtain $w_\epsilon = 20$, $w_l = 100$, $w_{t1} = 100$ and $w_{t2} = 10$.

4.7 Solving Algorithm

The solving algorithm lets the single satellites choose the activities and then acts as a coordinator to select the best choices at constellation level. It takes inspiration from the game-negotiation mechanism of Liu et al. [12], presented in Section 2.3.3, but deviates from it in different aspects. Their planning process accounts for observations only, leaving links to ground stations in a second phase, outside the plan. On the contrary, in the Mission Planner here depicted, satellites can choose between scheduling a observation or a downlink. As mentioned in Section 4.6.1, assigning time slots for links will be more and more critical as the number of satellites in orbit will increase in future and that aspect has thus to be taken into account. Additionally, Liu's architecture uses crosslinks to exchange information between agents, but since crosslink is not always available on constellations, the Mission Planner here developed relies on a centralised scheduling, but offers also the possibility for independent and distributed planning when dealing with emergencies.

4.7.1 Coordination Mechanism

The overall goal of planning is to schedule the imaging of a set of observation targets and the transfer of the acquired data to the ground segment. Satellites' goal is to have the highest payoff by scheduling activities.

During each iteration, each satellite can greedily choose the best move between scheduling a observation or scheduling a downlink. The 'best' move is evaluated by a meta-heuristic algorithm according to the payoff function described in Section 4.6.1. At the end of each iteration, the coordination algorithm seeks conflicts and solves them by preferring the overall best solution. The list of unscheduled targets is then updated and a new iteration starts. The process ends when no more convenient moves are possible for each satellite or the maximum iteration number is reached. That procedure is a compromise between finding the optimal solution with reference to the entire constellation and reducing the calculation time, as well as maintaining the independence of the agents.

The greed mechanism at the root of the algorithm may not seem the best, because in some circumstances it might lead to bad decisions at constellation levels. Consider this example: during a iteration, two satellites select the same target, while a third satellite selects another target that could have been selected with higher fitness value by one of the former two satellites; in that way, the second target will not be scheduled with the best payoff at constellation level, because it is assigned while the most suitable satellite was busy with another target, which at the end may not even be assigned to it. A solution could be to perform multiple iterations for the whole planning process, relying on its heuristic behaviour in order to select the best solution afterwards. Unfortunately that solution is extremely time consuming and often not worth it, since the improvement are small. Another solution is to let the satellites make more than one selection per iteration, so that the coordination mechanism can take into consideration more possibilities and consequences. That requires again additional time and it is sometimes counter-productive. In fact, satellites can schedule both unrelated and dependent activities. For the coordination mechanism, dependent activities produces higher payoff, because removing one activity, which has a fitness value, causes the cancellations of more dependent activities with additional linked fitness values. Therefore, a satellite could ‘steal’ a target from a more suitable satellite only because the first one has more dependent activities linked to that observation. The last solution is to allow satellites to change tasks which have been already scheduled, implementing a tabu list to prevent loops. The trade-off between higher payoff and longer computational time led to the choice of using that solution only to dynamic scheduling. As it will be described in Section 4.8, the ability of adapting to new high priority tasks is important in that case, but it is excessively time consuming for the static planning, due to the larger amount of activities to be scheduled.

Algorithm 1 Mission scheduling algorithm.

```
input Satellites Set, Targets Set, Ground Stations Set.
output Overall Scheduling Plan.
procedure
    for each iteration do
        for each satellite, simultaneously do
            Receive information about unscheduled targets and ground stations;
            Calculate the payoff and select the activity by GA;
        end for
        Resolve conflicts within the constellation;
        Update the Plan;
    end for
end procedure
```

4.7.2 Genetic Algorithm

During a iteration, each satellite has to make three choices: *what* to do between observing a target or transferring data, *when* starting doing that and for *how long*. Those three pieces of information can be translated into a triplet of integer numbers: the window ID, which includes both the action and the time allocation; the start ID, which indicates at what precise point of the visibility window the activity actually starts; the duration, encoded in the number of simulation time steps.

The suggested solving algorithm when dealing with integers is the genetic algorithm (GA). In this case, the individuals have three genes, representing the window ID, the start ID and the number of time steps. MATLAB has a function for the GA included in the *Global Optimization Toolbox*.

The algorithm starts by creating the initial population.

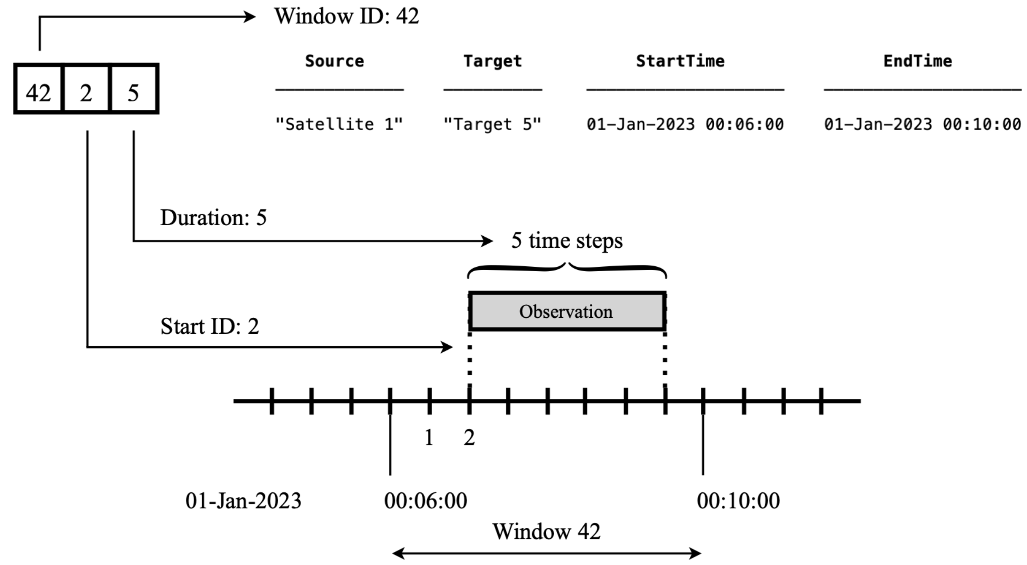


Figure 4.4: Information which is contained in a individual of the genetic algorithm.

Then the algorithm creates a sequence of new populations, using at each step the individuals in the current generation. Firstly it computes the fitness value of each member. The fitness function is the one described in Section 4.6.1. Individuals with best fitness values are selected as *parents* for the following generation. There are three kinds of *children*: *elite*, *crossover* and *mutation*. *Elite* children are the individuals in the current generation with the best fitness values and they automatically survive to the next generation. The GA creates *crossover* children by combining pairs of parents. At each coordinate of the child vector, the crossover function randomly selects a gene at the same coordinate from one of the two parents and assigns it to the child. *Mutation* children are created by randomly changing the genes of individual parents, adding a random vector from a Gaussian distribution. Finally, the algorithm replaces the current population with the children to form the next generation.

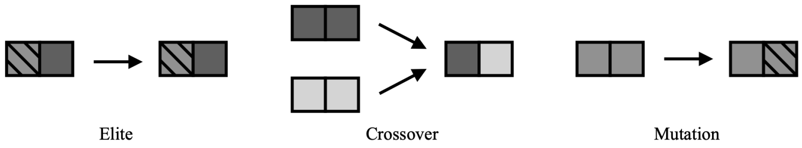


Figure 4.5: Types of children in a genetic algorithm.

When the problem has integer constraints, the software modifies all generated individuals to be feasible with respect to those constraints.

The algorithm stops when one of the stopping criteria is met. Those criteria are: maximum number of generations is reached; maximum running time in seconds is reached; the value of the fitness function for the best point in the current population is less than or equal to a given desired value; the average relative change in the fitness function value over the maximum number of stall generations is less than the function tolerance; there is no improvement in the objective function during an interval of time in seconds equal to the maximum stall duration.

The possible values that the genes can have are limited. The lower bound is the vector [1; 0; 1]: 1 is the minimum amount of windows required to schedule an activity; 0 means that the activity starts as soon as the window starts; 1 is the minimum number of time steps an activity can have. The upper bound is given by the total number of available windows for the satellite and by the maximum duration, in time steps, of the longest available window. Then the fitness function avoids that the solution exceeds the selected window's time limits.

Algorithm 2 Genetic Algorithm.

```
input Satellites Set, Targets Set, Ground Stations Set.
output Windows ID, Start ID, Duration.
procedure
    Create a random feasible initial population;
    while stopping criteria are not met do
        Compute individuals' fitness values;
        Select parents;
        Select elite children;
        Produce crossover and mutation children;
        Replace the population;
        Modify the generated individuals to be feasible.
    end while
end procedure
```

As shown in Chapter 3, the genetic algorithm is one of the most used optimisation algorithms, but better methods have been often proposed. The problem with those alternatives is that they require a change to the problem model, in order to make those algorithms feasible to the shape of the solution. This is not recommended if the aim of the planner is to be flexible and adaptable. As mentioned in Section 3.1, exact algorithms are not efficient in large-scale problems. Meta-heuristic algorithms are more rigorous than heuristic algorithms and are thus usually preferred. Among all the meta-heuristic algorithms, as described in Section 3.3.2, ant colony optimisation algorithms have shown in different cases the best performances. In the reported examples, the planning problem was modelled either as a Bayes network or as a chain of Boolean variables. Since all the visibility windows are computed, such method could be used for the Mission Planner. Nonetheless, that would reduce the time granularity of the solutions, because the algorithm could only decide whether a whole window is scheduled or not, without the possibility of scheduling the activity in the middle of the visibility window. In that way, some scheduling opportunities could be missed or some activities could be not scheduled, because the duration of the window would be excessive for the satellite's resources.

4.8 Dynamic Scheduling

The kind of mission planning which has been considered heretofore is called *static* planning. In static planning, the environmental information is known beforehand and the environment does not change during the mission. Although this kind of planning is easier, it is not practical, since the hypothesis that the environment will not change is generally not true and it is not possible to know the whole environmental information before the start of the mission. In fact, the plan execution may be susceptible to further constraints, due to planned restrictions or unexpected events. The schedule might have to adapt to those constraints, without excessively altering the rest of the plan. *Dynamic* planning can deal with environmental changes under condition of incomplete information.

The additional constraints can be divided into two categories: *systematic* and *random*. The first one can be considered as an addition to static planning, because its constraints are known well in advance by the user. The latter is completely part of the dynamic planning, because it deals with unexpected events that can not be predicted.

4.8.1 Systematic Constraints

Examples of systematic constraints can be time limitations on a observation target, unavailability of a ground station, maintenance or station keeping operations of a satellite. Those constraints are known beforehand and can thus be inserted by the user in a table. The program asks for the subject of the new constraint, the type of the subject (satellite, target or ground station), the start time and the end time. Furthermore, a multiple time constraint on a target can be used to express a precise time limitation on the observation. As mentioned before, this kind of constraints can be inserted during static planning, as well as they can be added afterward.

In the first case, new time constraints are simply considered during the scheduling process, without any particular difference. In the second case, instead, the planner applies the new constraints to the current plan, removing both activities which are directly affected by the constraints and activities which are no longer feasible after the change of the plan, e.g. a observation which requires an amount of on-board memory previously cleared by a downlink which has been now removed by effect of the new constraints. Subsequently, a new planning process starts, with the same rules but new additional time constraints, in order to assign the unscheduled targets and plan potential new downlinks.

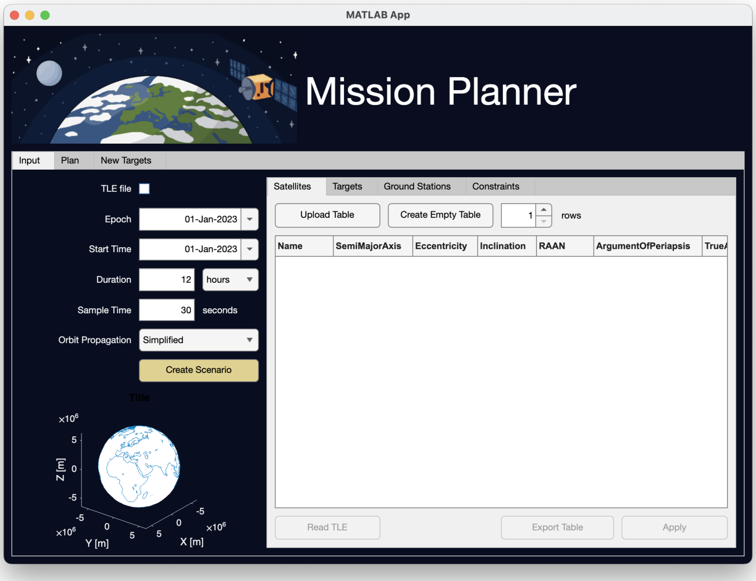
4.8.2 Random Constraints

Examples of random constraints can be a failure in a subsystem or in performing a downlink, cloud coverage of the target or any other events that can not be predicted well in advance. According to Pemberton and Greenwald [30], there are four contingencies which require dynamic planning: targets of opportunity, resource changes, new (short notice) tasking and problem parameter uncertainty. Targets of opportunity are interesting events that the constellation could detect during the plan execution, but which were not taken into consideration by the original plan. In this case, "the schedule must be flexible enough to permit automatic changes during execution" not to lose those opportunities. It may happen that "some of the assigned resources become unexpectedly unavailable", because of external events like solar flares or weather. In order to deal with resource changes, dynamic planning is necessary. The "ability to rapidly accommodate new, high priority tasks without going through a completely new cycle of scheduling and uplinking" is surely part of dynamic scheduling, but it can also be considered part of the systematic constraints previously analysed. Environmental uncertainty, as mentioned, creates problem parameter uncertainty, that could lead to a failure of an activity. Remaining tasks should thus be modified accordingly.

Three possibilities have been debated for dealing with random constraints. The first possibility is to re-schedule the activities on-board of the satellite where the failure in the plan has occurred. The problem is then processed and solved entirely on the satellite, to whom all the new tasks are entrusted. Although that is the simplest solution, it does not take advantage of the whole potential of the constellation. Another way to re-plan the mission is to convey the anomaly to the ground segment through the first scheduled downlink, re-plan the mission and send the new commands to the whole constellation during the subsequent links. In fact, as mentioned in Section 4.3.7, the transfer of TT&C data has been considered for the schedule. The third possibility is to share the information among the whole constellation by means of crosslinks and to let the satellite re-plan their tasks accordingly. The planning algorithm allows that since it is partially based on the work of Liu et al. [12] who considered exactly that eventuality. The downside is the need of an architecture in the space segment which is able to perform crosslinks. The Mission Planner shall choose between those three possibilities according to the pros and cons of each solution and the amount of time involved between computing and executing the new tasks.

In case of short notice tasking, the software asks the user for the new targets and for the desired start time of the new planning process and consequently the maximum duration of the algorithm, after which the computation will stop. The goal is to obtain a suitable plan that can be communicated to the spacecrafts by means of the already scheduled links. Therefore, the GA will assign a penalty to every solution which will not be consequent to the new start time and that does not have a preexisting link prior the new scheduled tasks. As mentioned in Section 4.7.1, the GA is helped by a tabu list in order to avoid loop generated by same payoff values of different possible solutions.

For failures in the plan execution, the algorithm is the same, but the satellite responsible of the failed task can schedule a task without having a preexisting link, since it is already aware of the situation and does not need to communicate to the ground segment.



TLE file □

Epoch 01-Jan-2023

Start Time 01-Jan-2023

Duration 12 hours

Sample Time 30 seconds

Orbit Propagation Simplified

1 rows

Name	SemiMajorAxis	Eccentricity	Inclination	RAAN	ArgumentOfPeriapsis	TrueA

Figure 4.6: Homepage of the mission planner, designated to the inputs.

4.9 Graphic User Interface

Although the script has been thought with the aim of being integrated into a preexisting software in future, a Graphic User Interface has been developed for the purpose of this thesis. It is a standalone app created with MATLAB App Developer.

The first page concerns the inputs. It requires the creation of a scenario, defining the start time, the duration and the sample time, as well as the type of orbit propagator. It asks also for the epoch, which is omitted in case of TLE files, since the information is already included in them. The lists and characteristics of satellites, targets, ground stations and constraints can be created starting from a empty table or can be uploaded from .xlsx files. The tables can subsequently be exported and saved. After a scenario has been created, a 3D-plot of planet Earth appears. That allows the user to see the ground stations, the targets and the satellites' orbits each time a table is applied. When TLE files are used, the software will firstly ask for the .txt file and then for a .xlsx file containing the additional needed information.

The second page is the one concerning the outputs. There, when all the inputs have been inserted and confirmed, it is possible to start the planning algorithm and obtain the schedule and a plot of it. Potential errors and the names of unscheduled targets are displayed in a text box below the schedule. Additionally it is possible to plot the available on-board memory and the electrical energy during the mission plan. The schedule is shown in a table containing the Source, the Target, the Start Time, the End Time and the Duration of each scheduled activity. The table can be saved as .xlsx file and the matrix containing the scheduled activities can be exported as .txt file. The matrix can be also be uploaded before starting the planning process in order to change a preexisting plan. During each activity, a loading icon will appear to show to the user that the algorithm is running.

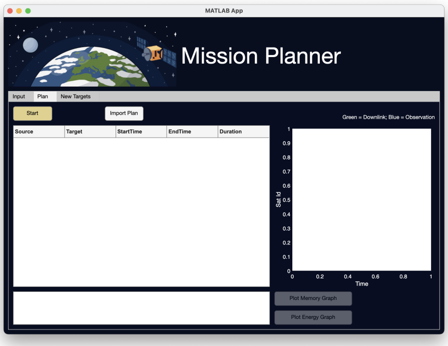
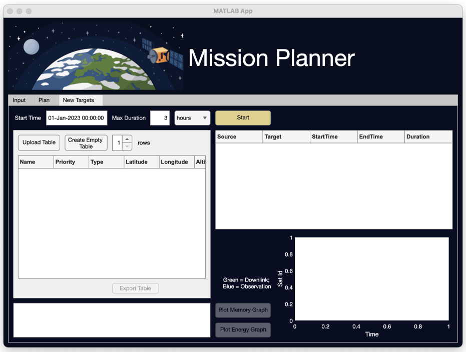


Figure 4.7: Second page of the mission planner, designated to the outputs.


Start Time 01-Jan-2023 00:00:00 Max Duration 3 hours

1 rows

Figure 4.8: Third page of the mission planner, designated to the new targets.

The third page concerns the short notice tasking. The start time of the new plan, the maximum duration and the new set of targets are the requested inputs.

The outputs are the same as the previous page.

4.10 Example

In order to show how the program works, a hypothetical mission involving a constellation of four satellites is here depicted and compared to a manual mission planning process.

The mission starts on January 1st 2023 and is expected to last 12 hours. The considered sample time is 30 seconds.

4.10.1 The Satellites

Two couples of satellites are considered. For each couple, the satellites are identical, but phased of 180° degrees. One couple have infrared sensors, the other one cameras which work in the visible light frequency. Both the payloads require that the target is illuminated by Sun. The inputs are presented in Table 4.5.

Name	SemiMajor Axis [m]	Eccentricity	Inclination [deg]	RAAN [deg]	Argument Of Periapsis [deg]	True Anomaly [deg]	Payload	
Satellite 1	7164000	0.0001	98.56	77	87	0	Visible	
Satellite 2	7164000	0.0001	98.56	77	87	180	Visible	
Satellite 3	6993000	0.0001	97.85	77	85	0	Infrared	
Satellite 4	6993000	0.0001	97.85	77	85	180	Infrared	
Name	Night	Data Storage [Gb]	Payload Power [W]	Payload Data Rate [Mbps]	Sensor Aperture Time [s]	Obs. Time [s]	Max Sensor Time [s]	Base Power [W]
Satellite 1	0	2400	250	490	5	60	300	1100
Satellite 2	0	2400	250	490	5	60	300	1100
Satellite 3	0	256	110	600	5	60	300	350
Satellite 4	0	256	110	600	5	60	300	350
Name	Solar Charging [W]	Energy Storage [Wh]	DOD	Frequency	Link Power [W]	Link Min Duration [s]	Link Max Duration [s]	Link Data Rate [Mbps]
Satellite 1	1700	2856	0.3	X-Band	20	90	1500	560
Satellite 2	1700	2856	0.3	X-Band	20	90	1500	560
Satellite 3	800	2710	0.3	X-Band	20	90	1000	155
Satellite 4	800	2710	0.3	X-Band	20	90	1000	155

Table 4.5: The inputs for the constellation for the mission planning example.

4.10.2 The Ground Stations

The ground segment consists of two ground stations, both in Europe: Matera and Kiruna. They are real ground stations regularly employed by the European Space Agency (ESA), but here they have been randomly chosen and the geographical coordinate are not extremely precise. In fact, the important aspect of the choice is that they are relatively close to each other. That tests the ability of the algorithm of distinguishing and valuating close but different link opportunities. The inputs are shown in Table 4.6.

Name	Latitude [deg]	Longitude [deg]	Altitude [m]	Min Elevation Angle [deg]	Frequency
Kiruna	67.86	20.23	550	15	X-Band
Matera	40.67	16.60	401	15	X-Band

Table 4.6: The inputs for the ground segment for the mission planning example.

4.10.3 The Targets

The observation targets are eight: 4 require a infrared sensor and 4 the visible light sensor. They have been chosen in different parts of the globe in order to test the ability of the program of recognising different light conditions. Each target has a priority factor which goes from 1 to 5, where 5 is the maximum priority. The inputs are presented in Table 4.7.

Name	Priority	Type	Latitude [deg]	Longitude [deg]	Altitude [m]	Min Elevation Angle [deg]	Max Elevation Angle [deg]
Petra	1	Infrared	30.33	35.44	1040	20	90
Taj Mahal	2	Visible	27.17	78.04	170	20	90
Giza Pyramid	3	Infrared	30.01	31.21	19	20	90
Great Wall	4	Infrared	40.43	116.57	1000	20	70
Colosseum	5	Infrared	41.89	12.49	15	20	90
Chichen Itza	1	Visible	20.68	-88.57	39	20	70
Machu Picchu	2	Visible	-13.16	-72.54	2430	20	70
The Redeemer	3	Visible	-22.95	-43.21	710	20	90

Table 4.7: The inputs for the targets for the mission planning example.

4.10.4 Manual Scheduling

The process in manual scheduling can change according to the needs and the preferences of the user, but reasonably the first step is to look at the visibility windows with respect to the targets and select the optimal ones. In this case, the best windows are shown in Table 4.8.

The observation of the targets "Chichen Itza", "Machu Picchu" and "The Redeemer" cannot be scheduled in the given time period because every time the satellites are in line of sight with them, they are in shadow with respect to the sunlight.

The target "Colosseum" shows a interesting behaviour. "Satellite 4" can observe it before "Satellite 3", but the maximum elevation angles they have during those windows are -32.1263° for "Satellite 4" and -74.3315° for "Satellite 3". Since the difference between the two windows is only about 45 minutes, the fitness function has a better value for a observation scheduled during the second visibility window, although the payoff difference is only 1.71. Similar considerations hold for the

Source	Target	Start Time	End Time	Duration
Satellite 1	Taj Mahal	01-Jan-2023 05:17:00	01-Jan-2023 05:23:30	390
Satellite 3	Giza Pyramid	01-Jan-2023 08:20:30	01-Jan-2023 08:25:30	300
Satellite 3	Petra	01-Jan-2023 08:20:00	01-Jan-2023 08:25:30	330
Satellite 3	Colosseum	01-Jan-2023 09:54:00	01-Jan-2023 09:59:30	330
Satellite 4	Great Wall	01-Jan-2023 02:38:00	01-Jan-2023 02:43:00	300
Satellite 4	Petra	01-Jan-2023 07:34:00	01-Jan-2023 07:35:30	90
Satellite 4	Colosseum	01-Jan-2023 09:07:30	01-Jan-2023 09:09:00	90

Table 4.8: Considered visibility windows of observation targets for the mission planning example.

target "Petra", for which the payoff difference between the two windows is 5.07.

The amount of data which has to be transferred to the ground segment after all the observations is 29400 Mb for "Satellite 1", 108000 Mb for "Satellite 3" and 36000 Mb for "Satellite 4". That means "Satellite 1" needs 53 s to transfer the data by means of a downlink, "Satellite 3" requires 697 s and "Satellite 4" needs 233 s. Considering that the link requires 90 s to be set and in that time no data is transferred and considering that it is preferable to transfer the data in as few links as possible and as soon as possible, the visibility windows selected are those shown in Table 4.9.

Source	Target	Start Time	End Time	Duration
Satellite 1	Kiruna	01-Jan-2023 08:27:30	01-Jan-2023 08:33:00	330
Satellite 3	Kiruna	01-Jan-2023 09:46:30	01-Jan-2023 09:53:00	390
Satellite 3	Matera	01-Jan-2023 09:53:30	01-Jan-2023 10:00:30	390
Satellite 3	Kiruna	01-Jan-2023 11:22:30	01-Jan-2023 11:28:30	360
Satellite 4	Kiruna	01-Jan-2023 08:58:30	01-Jan-2023 09:04:00	330

Table 4.9: Considered visibility windows of ground stations for the mission planning example.

"Satellite 1" can transfer the whole amount of data in just one visibility window to "Kiruna", as "Satellite 4". "Satellite 3" needs more links. In detail, it may be interesting to deal with the possibilities offered by the visibility window between "Satellite 3" and "Matera" ground station. Since the optimal schedule for the observation of "Colosseum" target starts at 01-Jan-2023 09:55:30 and ends at 01-Jan-2023 09:56:30, a downlink to "Matera" can be performed both right before and after that payload acquisition.

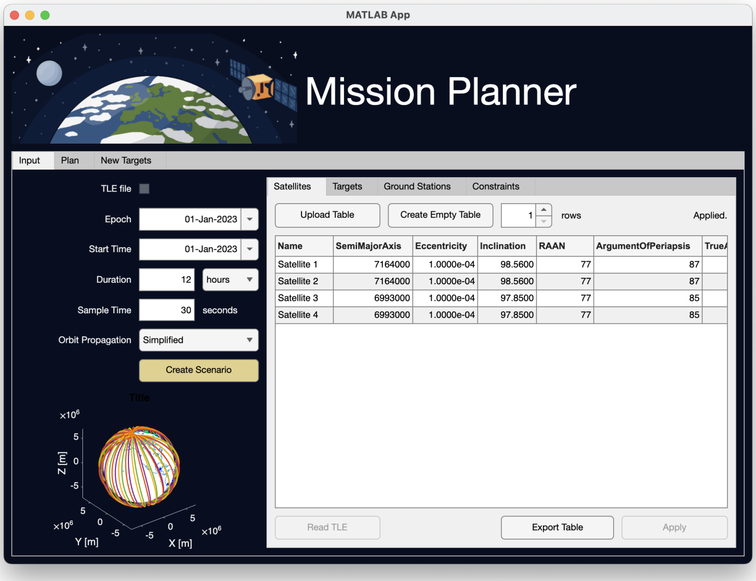
Therefore, the schedule that can be obtained is shown in Table 4.10. The total payoff is equal to 3086.2.

Source	Target	Start Time	End Time	Duration
Satellite 1	Taj Mahal	01-Jan-2023 05:19:30	01-Jan-2023 05:20:30	60
Satellite 1	Kiruna	01-Jan-2023 08:27:30	01-Jan-2023 08:30:00	150
Satellite 3	Petra	01-Jan-2023 08:21:30	01-Jan-2023 08:22:30	60
Satellite 3	Giza Pyramid	01-Jan-2023 08:22:30	01-Jan-2023 08:23:30	60
Satellite 3	Kiruna	01-Jan-2023 09:46:30	01-Jan-2023 09:53:00	390
Satellite 3	Matera	01-Jan-2023 09:53:30	01-Jan-2023 09:55:30	120
Satellite 3	Colosseum	01-Jan-2023 09:55:30	01-Jan-2023 09:56:30	60
Satellite 3	Matera	01-Jan-2023 09:56:30	01-Jan-2023 10:00:00	210
Satellite 3	Kiruna	01-Jan-2023 11:22:30	01-Jan-2023 11:28:30	360
Satellite 4	Great Wall	01-Jan-2023 02:39:30	01-Jan-2023 02:40:30	60
Satellite 4	Kiruna	01-Jan-2023 08:58:30	01-Jan-2023 09:04:00	330

Table 4.10: Manual schedule for the mission planning example.

4.10.5 Mission Planner's Process and Results

The data in the previous sections have been inserted into the mission planner through the GUI. It is important to create a empty table for the constraints in order for the software to allow the start of the planning process.



TLE file □

Epoch 01-Jan-2023

Start Time 01-Jan-2023

Duration 12 hours

Sample Time 30 seconds

Orbit Propagation Simplified

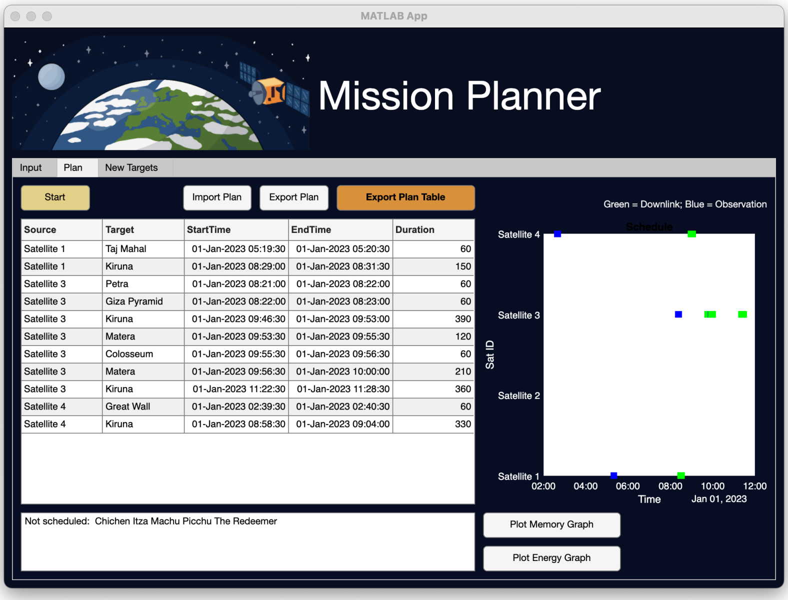
1 rows

Applied.

Name	SemiMajorAxis	Eccentricity	Inclination	RAAN	ArgumentOfPeriapsis	True
Satellite 1	7164000	1.0000e-04	98.5600	77	87	
Satellite 2	7164000	1.0000e-04	98.5600	77	87	
Satellite 3	6993000	1.0000e-04	97.8500	77	85	
Satellite 4	6993000	1.0000e-04	97.8500	77	85	

Figure 4.9: Example of the inputs page of the mission planner's GUI.

The heuristic nature of the optimisation algorithm implies different possible solutions. For the example, multiple simulations have run. In Figure 4.10 there is a example of output in the GUI. Figure 4.11 and Figure 4.12 show the available on-board memory and the electrical energy during one of the mission plans.



Source	Target	StartTime	EndTime	Duration
Satellite 1	Taj Mahal	01-Jan-2023 05:19:30	01-Jan-2023 05:20:30	60
Satellite 1	Kiruna	01-Jan-2023 08:29:00	01-Jan-2023 08:31:30	150
Satellite 3	Petra	01-Jan-2023 08:21:00	01-Jan-2023 08:22:00	60
Satellite 3	Giza Pyramid	01-Jan-2023 08:22:00	01-Jan-2023 08:23:00	60
Satellite 3	Kiruna	01-Jan-2023 09:46:30	01-Jan-2023 09:53:00	390
Satellite 3	Matera	01-Jan-2023 09:53:30	01-Jan-2023 09:55:30	120
Satellite 3	Colosseum	01-Jan-2023 09:55:30	01-Jan-2023 09:56:30	60
Satellite 3	Matera	01-Jan-2023 09:56:30	01-Jan-2023 10:00:00	210
Satellite 3	Kiruna	01-Jan-2023 11:22:30	01-Jan-2023 11:28:30	360
Satellite 4	Great Wall	01-Jan-2023 02:39:30	01-Jan-2023 02:40:30	60
Satellite 4	Kiruna	01-Jan-2023 08:58:30	01-Jan-2023 09:04:00	330

Not scheduled: Chichen Itza Machu Picchu The Redeemer

Figure 4.10: Example of the outputs page of the mission planner's GUI.

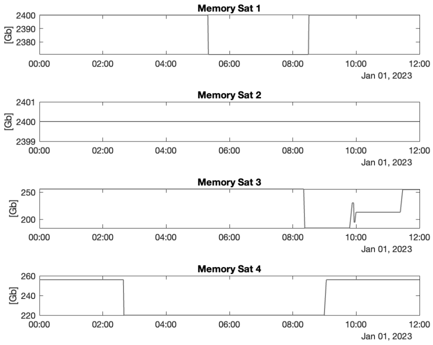


Figure 4.11: The available on-board memory during a example of mission plan.

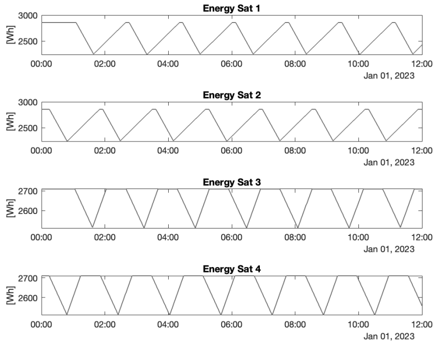


Figure 4.12: The electrical energy during a example of mission plan.

Obviously the constellation and the targets definition are not the optimal ones for that scenario, since "Satellite 1" only has one payload acquisition, "Satellite 2" does not perform any activities and the tasks are not equally shared between

"Satellite 3" and "Satellite 4", but this example has been thought only to address some characteristics of the planning algorithm and it does not want to depict a plausible space mission. Due to the sunlight condition on the target observations, the payload acquisitions happen when the satellite is not in eclipse and therefore the electrical power consumption does not represent a important constraint in this case. Also the on-board memory capacity is more than sufficient for the amount of data produced by the acquisitions. The interesting aspect of the example relies on the differences between the manual schedule and the Mission Planner's ones. Figure 4.13 shows the payoff of the Mission Planner's plans and the one of the manual schedule. Figure 4.14 highlights the precision of the Planner, since the relative error is under 0.35%. The algorithm is also accurate, since the values deviates from the mean value less than 0.2%. In the run simulations the highest fitness value was 3083.3; the lowest 3076.0. Table 4.11 and Table 4.12 show the corresponding plans. The difference with the manual plan of the first one is only in an advance of 30 seconds of the payload acquisitions of the targets "Petra" and "Giza Pyramid". The latter has a delay of 1 minute and 30 seconds in the link between "Satellite 1" and "Kiruna", the same advance of the targets "Petra" and "Giza Pyramid" and another ground station for the downlink of "Satellite 4" (i.e. "Matera" instead of "Kiruna"), with consequent delay. Therefore, even the worst case of the ones obtained does not deviate too much from the optimal solution.

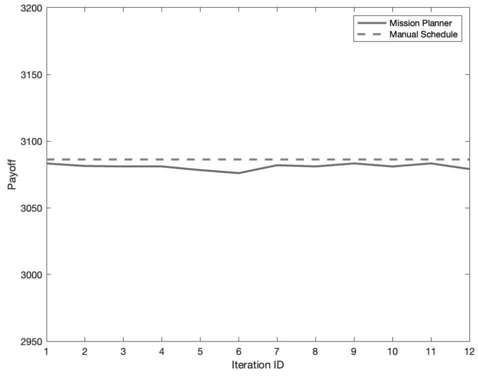


Figure 4.13: Comparison between total payoff values of the manual schedule and Mission Planner's results.

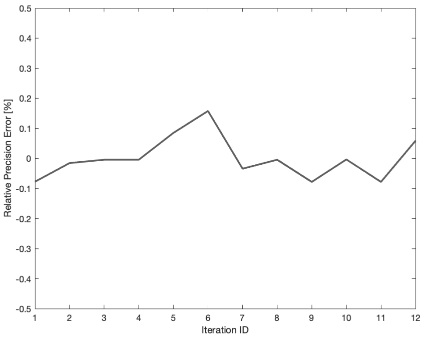


Figure 4.14: Relative error on the payoff of the Mission Planner's results with respect to the payoff of the manual schedule.

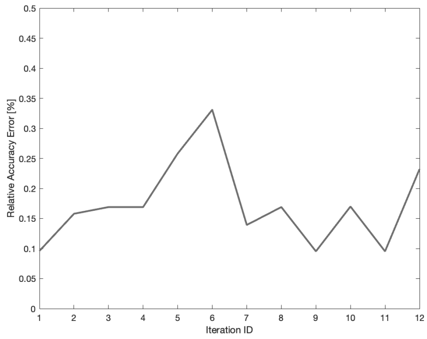


Figure 4.15: Relative error on the payoff of the Mission Planner's results with respect to their mean value.

Source	Target	Start Time	End Time	Duration
Satellite 1	Taj Mahal	01-Jan-2023 05:19:30	01-Jan-2023 05:20:30	60
Satellite 1	Kiruna	01-Jan-2023 08:27:30	01-Jan-2023 08:30:00	150
Satellite 3	Petra	01-Jan-2023 08:21:00	01-Jan-2023 08:22:00	60
Satellite 3	Giza Pyramid	01-Jan-2023 08:22:00	01-Jan-2023 08:23:00	60
Satellite 3	Kiruna	01-Jan-2023 09:46:30	01-Jan-2023 09:53:00	390
Satellite 3	Matera	01-Jan-2023 09:53:30	01-Jan-2023 09:55:30	120
Satellite 3	Colosseum	01-Jan-2023 09:55:30	01-Jan-2023 09:56:30	60
Satellite 3	Matera	01-Jan-2023 09:56:30	01-Jan-2023 10:00:00	210
Satellite 3	Kiruna	01-Jan-2023 11:22:30	01-Jan-2023 11:28:30	360
Satellite 4	Great Wall	01-Jan-2023 02:39:30	01-Jan-2023 02:40:30	60
Satellite 4	Kiruna	01-Jan-2023 08:58:30	01-Jan-2023 09:04:00	330

Table 4.11: Computed schedule with highest payoff for the mission planning example.

Source	Target	Start Time	End Time	Duration
Satellite 1	Taj Mahal	01-Jan-2023 05:19:30	01-Jan-2023 05:20:30	60
Satellite 1	Kiruna	01-Jan-2023 08:29:00	01-Jan-2023 08:31:30	150
Satellite 3	Petra	01-Jan-2023 08:21:00	01-Jan-2023 08:22:00	60
Satellite 3	Giza Pyramid	01-Jan-2023 08:22:00	01-Jan-2023 08:23:00	60
Satellite 3	Kiruna	01-Jan-2023 09:46:30	01-Jan-2023 09:53:00	390
Satellite 3	Matera	01-Jan-2023 09:53:30	01-Jan-2023 09:55:30	120
Satellite 3	Colosseum	01-Jan-2023 09:55:30	01-Jan-2023 09:56:30	60
Satellite 3	Matera	01-Jan-2023 09:56:30	01-Jan-2023 10:00:00	210
Satellite 3	Kiruna	01-Jan-2023 11:22:30	01-Jan-2023 11:28:30	360
Satellite 4	Great Wall	01-Jan-2023 02:39:30	01-Jan-2023 02:40:30	60
Satellite 4	Matera	01-Jan-2023 09:06:00	01-Jan-2023 09:11:30	330

Table 4.12: Computed schedule with lowest payoff for the mission planning example.

Chapter 5

Case Study

In order to show the potential of the Mission Planner previously depicted, a realistic mission planning problem is presented and solved in this Chapter.

This fake mission will take inspiration from IRIDE, a LEO space program of the Italian Space Agency (ASI) which will implement a system that will provide geospatial services at the national and European levels, both to public administration and private customers. It will consist of a “constellation of constellations”, with satellites of various types and sizes combining optical, panchromatic, hyperspectral, SAR and infrared sensors. The constellation will be built in Italy and completed by 2026 with support from ESA and ASI for a total value of 1.3 billion euros [31]. IRIDE will help the Civil Defense Department and other administrations deal with hydrogeological instability and fires, protect coastlines, and monitor critical infrastructure, air quality and weather conditions. The new constellations will include 22 satellites by the end of 2025, with the possibility of producing 27 more. Moreover, IRIDE will be a whole system which will embrace both space and ground segments, integrating preexisting spacecrafts, like the constellation COSMO-SkyMed or the Sentinels, part of Copernicus space program.

5.1 The Satellites

For the simulation, the involved spacecrafts are both existing and fake satellites. The constellation consists of some of the Sentinel satellites [32, 33, 34], COSMO-SkyMed [35], COSMO-SkyMed Second Generation [36] and PRISMA [37]. Additionally, 20 more CubeSat satellites are considered. They are identical, inspired by the 6U-CubeSat designed by Tsitas and Kingston [38], but equipped with different sensors. Their reference orbit has an altitude of 600 km and an inclination of 98°. From that, the 20 satellites were located according to a Walker constellation in 2 orbital planes (Delta 98:20/2/1). Walker constellations are a common solution for maximising geometric coverage over Earth while minimising the number of satellites required to perform the mission. They can be expressed by the notation i:t/p/f, where i is the inclination, t is the total number of satellites, p is the number of equally spaced orbital planes and f is the relative spacing between satellites in adjacent planes, such that the change in true anomaly for equivalent satellites in neighbouring planes is equal to $(f \times 360)/t$, in degrees [2]. There are two types of Walker constellations: *delta*, where orbit planes are evenly distributed over the full 360 degree range of RAAN, and *star*, where orbit planes are evenly distributed over 180 degree range. The required inputs are shown in Table 5.1.

Name	Payload	Night	Data Storage [Gb]	Payload Power [W]	Payload Data Rate [Mbps]	Sensor Aperture Time [s]	Obs. Time [s]	Max Sensor Time [s]	Base Power [W]	Solar Charging [W]	Energy Storage [Wh]	DOD	Frequency	Link Power [W]	Link Min Duration [s]	Link Max Duration [s]	Link Data Rate [Mbps]
Sentinel-1A	C-Band	1	1410	4368	430	10	60	1800	1300	4400	9072	0.3	X-Band	450	90	1800	520
Sentinel-2A	Visible, Near infrared, Short-wave infrared	0	2400	266	160	5	60	2880	1080	1730	2856	0.3	X-Band	450	90	1500	520
Sentinel-2B	Visible, Near infrared, Short-wave infrared	0	2400	266	160	5	60	2880	1080	1730	2856	0.3	X-Band	450	90	1500	520
PRISMA	Visible, Near infrared, Short-wave infrared	0	256	110	600	5	10	200	350	800	2710	0.3	X-Band	20	90	900	155
CSK-1	X-Band	1	300	13000	600	5	10	200	1000	3700	8736	0.35	X-Band	200	90	1000	300
CSK-2	X-Band	1	300	13000	600	5	10	200	1000	3700	8736	0.35	X-Band	200	90	1000	300
CSK-4	X-Band	1	300	13000	600	5	10	200	1000	3700	8736	0.35	X-Band	200	90	1000	300
CSG-1	X-Band	1	1530	17000	2400	5	10	200	1600	4000	8736	0.35	X-Band	450	90	1000	520
CubeSat-1	Visible	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-2	Infrared	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-3	X-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-4	C-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-5	Visible	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-6	Infrared	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-7	X-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-8	C-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-9	Visible	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-10	Infrared	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-11	X-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-12	C-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-13	Visible	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-14	Infrared	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-15	X-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-16	C-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-17	Visible	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-18	Infrared	0	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-19	X-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14
CubeSat-20	C-Band	1	8	19.6	127.5	5	10	200	5.9	17.7	20	0.3	X-Band	7.6	90	575	14

Table 5.1: The inputs for the constellation for the Case Study.

The TLE files of existing satellites were taken from Space-Track.Org, while the TLE files of the fake constellation were created with ANSYS Systems Tool Kit (STK). They are shown in Table 5.2.

5.2 The Ground Stations

The ground stations were chosen from those already involved in European Earth-observation missions. The station in Matera (Italy) is involved, among other missions, in both the Copernicus and COSMO-SkyMed programmes. Another ground station is the one in Kiruna (Sweden), equipped with two antennas involved in supporting several missions in LEO for ESA. Other ground stations collaborating in the Copernicus programme and other missions were also considered, such as Maspalomas (Spain), Svalbard (Norway) and Cordoba (Argentina). All stations can operate in the X-Band, which is the band used by all the considered satellites

Sentinel-1A 1 39634U 14016A 23036.18481414 .00000021 00000-0 14234-4 0 9993 2 39634 98.1819 45.4070 0001293 87.8092 272.3257 14.59200079470963	Sentinel-2A 1 40697U 15028A 23036.34036850 .00000142 00000-0 70998-4 0 9999 2 40697 98.5694 112.7357 0001218 90.4399 269.6929 14.30816603398152
Sentinel-2B 1 42063U 17013A 23036.30537261 -.00000225 00000-0 -69044-4 0 9996 2 42063 98.5662 112.7076 0000850 80.3318 279.7658 14.30817086309063	PRISMA 1 44072U 19015A 23036.22699253 .00000654 00000-0 85660-4 0 9999 2 44072 97.8638 112.5869 0001584 98.2691 261.8714 14.83648189209973
CSK-1 1 31598U 07023A 23036.04328288 .00000484 00000-0 67396-4 0 9998 2 31598 97.8882 221.4065 0001322 90.1387 269.9969 14.82157818847415	CSK-2 1 32376U 07059A 23036.07709463 .00001026 00000-0 13548-3 0 9995 2 32376 97.8880 221.4393 0001465 89.4257 270.7169 14.82150379820167
CSK-4 1 37216U 10060A 23036.08972757 .00000633 00000-0 86044-4 0 9993 2 37216 97.8881 221.4465 0001327 74.0697 286.0662 14.82155035662718	CSG-1 1 44873U 19092A 23036.10233879 .00000149 00000-0 25265-4 0 9992 2 44873 97.8888 221.4667 0001288 77.1233 283.0040 14.82158060169565
CubeSat-1 1 00001U 00000A 23036.33333333 .00001638 00000-0 17998-3 0 00007 2 00001 098.0659 215.2818 0006700 106.7235 348.1793 14.87737357000012	CubeSat-2 1 00002U 00001A 23036.33333333 .00001638 00000-0 17998-3 0 00009 2 00002 098.0659 215.2818 0006700 106.7235 024.1793 14.87737357000014
CubeSat-3 1 00003U 00002A 23036.33333333 .00001638 00000-0 17998-3 0 00001 2 00003 098.0659 215.2818 0006700 106.7235 060.1793 14.87737357000015	CubeSat-4 1 00004U 00003A 23036.33333333 .00001638 00000-0 17998-3 0 00003 2 00004 098.0659 215.2818 0006700 106.7235 096.1793 14.87737357000015
CubeSat-5 1 00005U 00004A 23036.33333333 .00001638 00000-0 17998-3 0 00005 2 00005 098.0659 215.2818 0006700 106.7235 132.1793 14.87737357000017	CubeSat-6 1 00006U 00005A 23036.33333333 .00001638 00000-0 17998-3 0 00007 2 00006 098.0659 215.2818 0006700 106.7235 168.1793 14.87737357000017
CubeSat-7 1 00007U 00006A 23036.33333333 .00001638 00000-0 17998-3 0 00009 2 00007 098.0659 215.2818 0006700 106.7235 204.1793 14.87737357000019	CubeSat-8 1 00008U 00007A 23036.33333333 .00001638 00000-0 17998-3 0 00001 2 00008 098.0659 215.2818 0006700 106.7235 240.1793 14.87737357000010
CubeSat-9 1 00009U 00008A 23036.33333333 .00001638 00000-0 17998-3 0 00003 2 00009 098.0659 215.2818 0006700 106.7235 276.1793 14.87737357000010	CubeSat-10 1 00010U 00009A 23036.33333333 .00001638 00000-0 17998-3 0 00006 2 00010 098.0659 215.2818 0006700 106.7235 312.1793 14.87737357000013
CubeSat-11 1 00011U 00010A 23036.33333333 .00001638 00000-0 17998-3 0 00009 2 00011 098.0659 035.2818 0006700 106.7235 006.1793 14.87737357000014	CubeSat-12 1 00012U 00011A 23036.33333333 .00001638 00000-0 17998-3 0 00001 2 00012 098.0659 035.2818 0006700 106.7235 042.1793 14.87737357000015
CubeSat-13 1 00013U 00012A 23036.33333333 .00001638 00000-0 17998-3 0 00003 2 00013 098.0659 035.2818 0006700 106.7235 078.1793 14.87737357000015	CubeSat-14 1 00014U 00013A 23036.33333333 .00001638 00000-0 17998-3 0 00005 2 00014 098.0659 035.2818 0006700 106.7235 114.1793 14.87737357000017
CubeSat-15 1 00015U 00014A 23036.33333333 .00001638 00000-0 17998-3 0 00007 2 00015 098.0659 035.2818 0006700 106.7235 150.1793 14.87737357000018	CubeSat-16 1 00016U 00015A 23036.33333333 .00001638 00000-0 17998-3 0 00009 2 00016 098.0659 035.2818 0006700 106.7235 186.1793 14.87737357000018
CubeSat-17 1 00017U 00016A 23036.33333333 .00001638 00000-0 17998-3 0 00001 2 00017 098.0659 035.2818 0006700 106.7235 222.1793 14.87737357000010	CubeSat-18 1 00018U 00017A 23036.33333333 .00001638 00000-0 17998-3 0 00003 2 00018 098.0659 035.2818 0006700 106.7235 258.1793 14.87737357000010
CubeSat-19 1 00019U 00018A 23036.33333333 .00001638 00000-0 17998-3 0 00005 2 00019 098.0659 035.2818 0006700 106.7235 294.1793 14.87737357000011	CubeSat-20 1 00020U 00019A 23036.33333333 .00001638 00000-0 17998-3 0 00008 2 00020 098.0659 035.2818 0006700 106.7235 330.1793 14.87737357000014

Table 5.2: TLE files for the Case Study.

to transmit observation data. Very often, in a space mission, TT&C data are handled by other stations, with other frequency bands used for communication, e.g. S-Band. That actually represents one of the limitations of the software that may be overcome in future developments. The required inputs are shown in Table 5.3.

Name	Latitude [deg]	Longitude [deg]	Altitude [m]	Min Elevation Angle [deg]	Frequency
Kiruna 1	67.8571	20.9643	402.2	10	X-Band
Kiruna 2	67.8584	20.9669	385.8	10	X-Band
Matera	40.6486	16.7046	536.9	10	X-Band
Maspalomas	27.7627	-14.3662	205.1	10	X-Band
Cordoba	-31.5242	-64.4636	730.0	10	X-Band
Svalbard	78.2306	15.3894	458.0	10	X-Band

Table 5.3: The inputs for the ground segment for the Case Study.

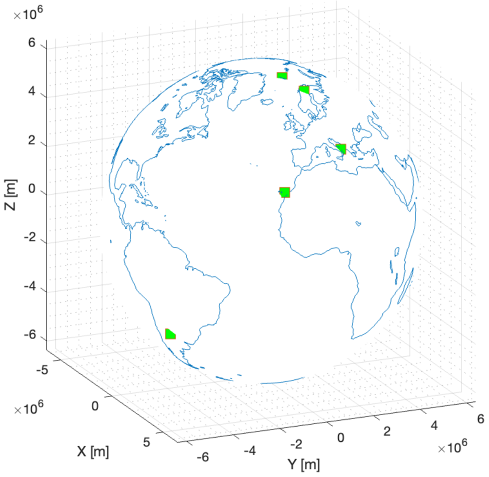


Figure 5.1: Geographical location of the selected ground stations for the Case Study.

5.3 The Targets

IRIDE will provide useful data both for science and emergency management. While some emergencies are unexpected and fast, like fires or earthquakes, other ones can be foreseen and therefore the targets scheduled 24 hours before, like when dealing with floods. A important role is played by the priority value of each target.

A first set of targets represents a list of randomly selected points of interest on Earth's surface. As the programme is Italian and European, targets on the Italian peninsula and in Europe have higher priorities. The list is shown in Table 5.4.

A second set of targets represents a natural emergency, which happens while the previous plan is being executed. The plan has to change in order to schedule the observations of those new high priority targets, without excessively alter the mission. The new targets are shown in Table 5.5. Because it simulates an emergency, the priority factor is set at 6.

5.4 The Scenario

During the morning of February 5th 2023, a set of 100 observation targets shall be scheduled for the day after, February 6th 2023. The total duration of the plan is thus 24 hours. The time resolution (i.e. the sample time) is 15 seconds. The satellites are inserted by TLE files and their orbit is propagated by the SGP4 propagator. COSMO-SkyMed and PRISMA can receive on-demand tasks both

Name	Priority	Type	Latitude [deg]	Longitude [deg]	Altitude [m]	Min Elevation Angle [deg]	Max Elevation Angle [deg]	Name	Priority	Type	Latitude [deg]	Longitude [deg]	Altitude [m]	Min Elevation Angle [deg]	Max Elevation Angle [deg]
Target 1	5	X-Band	46.3397	11.2793	40.0685	17	89	Target 51	3	X-Band	53.1476	24.7683	26.2271	17	89
Target 2	5	Visible	41.1195	13.8859	546.7217	11	79	Target 52	3	C-Band	60.2098	28.8089	142.7424	19	88
Target 3	5	C-Band	46.4446	9.9218	962.8091	20	84	Target 53	3	X-Band	66.8988	-19.3385	196.398	13	71
Target 4	5	C-Band	40.6389	13.8548	184.5927	18	80	Target 54	3	X-Band	46.3333	-6.2628	285.9486	13	71
Target 5	5	Infrared	46.1948	10.3337	42.023	11	77	Target 55	3	C-Band	42.8761	-17.0614	229.2791	13	75
Target 6	5	Visible	44.5006	15.5283	445.642	14	82	Target 56	3	Visible	39.0643	-0.651	275.7909	10	77
Target 7	5	X-Band	40.9313	12.2698	395.7415	20	83	Target 57	3	Visible	46.3458	0.5582	25.7902	13	87
Target 8	5	Infrared	35.1641	10.5255	970.1762	12	74	Target 58	3	Infrared	36.4222	18.509	194.158	19	70
Target 9	5	C-Band	37.8826	14.3265	669.3511	13	81	Target 59	3	Infrared	46.542	4.2805	36.0079	11	75
Target 10	5	X-Band	40.2546	9.4385	730.1733	10	90	Target 60	3	X-Band	38.8033	28.3486	243.2664	16	85
Target 11	5	Visible	45.6557	11.1334	109.6543	11	88	Target 61	2	Visible	-32.8293	167.958	787.4949	10	71
Target 12	5	C-Band	45.9586	8.8404	494.9666	10	90	Target 62	2	Visible	33.9455	5.3425	224.172	19	88
Target 13	5	X-Band	42.4839	12.2828	925.428	17	83	Target 63	2	X-Band	67.7804	158.7995	182.0063	20	73
Target 14	5	Visible	40.6306	17.5753	672.1372	10	78	Target 64	2	X-Band	-89.4227	-96.8574	108.7342	13	79
Target 15	5	Visible	44.8898	14.729	423.3624	15	79	Target 65	2	C-Band	-12.7013	21.2825	721.3545	12	75
Target 16	5	C-Band	39.3517	12.0343	318.8289	18	70	Target 66	2	C-Band	-28.9916	-149.7592	344.7053	20	81
Target 17	5	Visible	43.0011	15.0966	991.7825	10	77	Target 67	2	X-Band	51.4289	116.8247	448.8803	19	90
Target 18	5	Visible	39.1795	14.7877	754.4067	17	86	Target 68	2	C-Band	48.8017	-116.8429	11.5143	18	73
Target 19	5	X-Band	36.4852	11.7146	325.5581	14	83	Target 69	2	Infrared	28.5209	22.8692	381.9578	15	88
Target 20	5	Visible	40.5971	9.821	378.1918	14	79	Target 70	2	Visible	-19.6859	112.8594	667.2341	16	80
Target 21	4	X-Band	35.2962	14.2526	695.7951	16	89	Target 71	2	C-Band	-47.1694	161.6488	46.2871	16	82
Target 22	4	Infrared	38.7401	16.0559	296.0217	20	90	Target 72	2	X-Band	-67.9264	53.849	256.3689	14	70
Target 23	4	C-Band	46.4505	12.5095	191.7865	11	80	Target 73	2	C-Band	-52.3032	-108.4914	308.6509	19	80
Target 24	4	X-Band	45.8636	14.7502	323.1647	12	79	Target 74	2	C-Band	22.0005	-150.3049	581.4274	11	85
Target 25	4	Visible	43.7408	18.6921	933.3635	19	86	Target 75	2	C-Band	16.6259	-48.0185	773.8031	11	81
Target 26	4	X-Band	42.8686	17.4177	667.6186	10	72	Target 76	1	Visible	84.6427	39.9409	555.7541	15	90
Target 27	4	Visible	42.3946	12.2338	751.4	10	87	Target 77	1	X-Band	22.7692	110.2042	32.1148	16	80
Target 28	4	C-Band	37.5634	11.0044	972.7958	16	85	Target 78	1	Infrared	63.2862	-167.7555	587.3355	11	71
Target 29	4	Visible	43.4929	17.1791	427.91	10	82	Target 79	1	X-Band	-38.0204	42.6724	649.0736	11	72
Target 30	4	X-Band	37.2499	13.4147	302.6336	11	76	Target 80	1	Infrared	-22.0662	-49.9042	51.8225	10	79
Target 31	4	C-Band	45.7256	13.1782	854.6226	14	82	Target 81	1	X-Band	-48.5396	-129.4318	389.2864	14	73
Target 32	4	X-Band	35.6973	9.5976	397.0647	15	82	Target 82	1	C-Band	87.0942	-133.7847	794.8463	15	79
Target 33	4	Infrared	38.8764	9.1032	734.3698	12	74	Target 83	1	Infrared	-88.6898	-167.0352	311.7567	10	82
Target 34	4	C-Band	45.242	11.3326	641.0915	14	71	Target 84	1	Visible	34.82	-35.3283	561.3771	12	72
Target 35	4	Infrared	36.4588	18.818	865.2365	20	74	Target 85	1	C-Band	-21.7385	170.3588	171.8275	13	71
Target 36	4	X-Band	44.0005	8.8546	390.7024	13	90	Target 86	1	Infrared	38.501	140.4385	399.5587	10	70
Target 37	4	Visible	40.672	10.8877	993.6668	15	86	Target 87	1	Infrared	-67.0148	35.9886	44.5044	17	71
Target 38	4	C-Band	38.6305	11.9912	314.8082	12	70	Target 88	1	Infrared	-76.039	-36.991	28.3906	19	78
Target 39	4	Infrared	42.2279	15.0768	40.3652	13	81	Target 89	1	X-Band	12.2803	-173.2499	391.4439	14	79
Target 40	4	C-Band	37.5487	9.0598	345.8276	19	75	Target 90	1	C-Band	71.1454	98.0015	235.8748	18	87
Target 41	3	C-Band	42.1202	25.6574	198.9068	20	80	Target 91	3	X-Band	-78.3328	-171.2294	298.2473	20	75
Target 42	3	Visible	38.7119	25.2052	301.2681	15	82	Target 92	3	Infrared	36.0315	99.5173	672.4135	14	87
Target 43	3	Visible	43.2347	-1.9763	305.5114	19	70	Target 93	3	C-Band	-33.1886	-31.3483	704.344	20	77
Target 44	3	Visible	56.367	30.4166	182.0419	18	88	Target 94	3	Infrared	50.9288	-6.976	193.6759	14	70
Target 45	3	Visible	55.8534	14.6086	256.7567	11	80	Target 95	3	Visible	-57.4666	-91.1447	557.4587	12	85
Target 46	3	Visible	53.7257	-16.0496	141.6855	10	74	Target 96	4	Visible	-41.6791	-29.9921	305.364	12	73
Target 47	3	Visible	57.8204	17.3289	8.2034	18	90	Target 97	4	C-Band	-10.6574	151.9849	363.0167	18	70
Target 48	3	Infrared	61.5537	-0.23361	68.0581	15	88	Target 98	4	C-Band	-72.6749	45.5528	725.845	20	82
Target 49	3	Infrared	45.7055	-11.5576	77.0428	12	83	Target 99	4	X-Band	-36.1616	72.8511	543.1648	19	86
Target 50	3	C-Band	45.2966	-17.2706	189.777	17	77	Target 100	4	Infrared	83.525	19.638	93.8383	14	84

Table 5.4: The inputs for the randomly selected observation targets for the Case Study.

Name	Priority	Type	Latitude [deg]	Longitude [deg]	Altitude [m]	Min Elevation Angle [deg]	Max Elevation Angle [deg]
Emergency 1	6	Infrared	36.9914	35.3308	27.9150	30	90
Emergency 2	6	X-Band	36.9914	35.3308	27.9150	20	90
Emergency 3	6	Visible	37.7837	37.6413	678.2790	30	90
Emergency 4	6	X-Band	37.7837	37.6413	678.2790	30	90
Emergency 5	6	C-Band	36.2021	37.1342	382.8860	20	90

Table 5.5: The inputs for the targets for a hypothetical emergency situation during the Case Study.

from institutional and private entities and therefore they can integrate well in the algorithm. On the other hand, Sentinel satellites are independent, they perform continuous imaging, regularly downlink the data and make them publicly available after a while. The results obtained for those satellites are thus to be considered only as a yardstick to organise the mission and not as part of the actual plan execution.

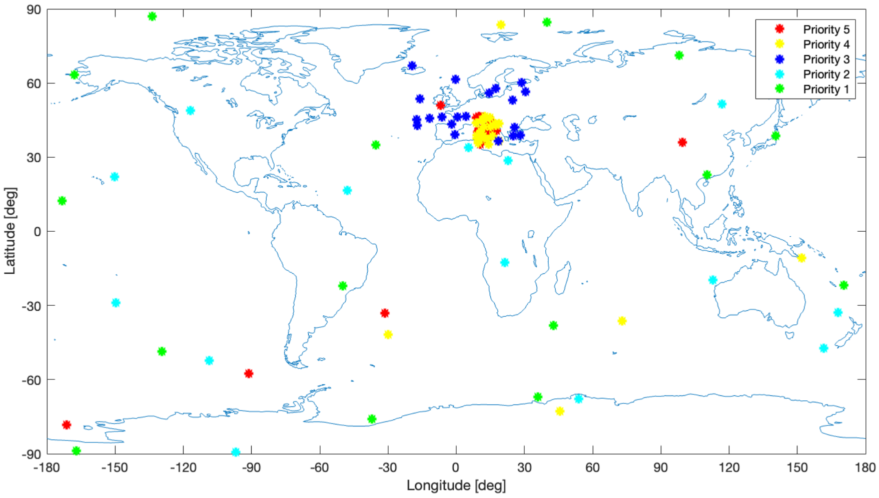


Figure 5.2: Geographical location of the randomly selected targets for the Case Study with their priorities.

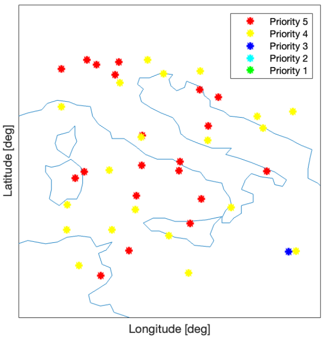


Figure 5.3: Detail of the Italian Peninsula with the geographical location of the randomly selected targets for the Case Study.

5.5 Static Scheduling

The plan obtained by means of the Mission Planner is displayed in Figure 5.4 and Table 5.6. Targets are quite evenly distributed among the constellation and only 2 of them are left unscheduled ("Target 76" and "Target 100"). Most of the payload acquisitions are scheduled in the first half of the time scenario and downlinks are executed soon after.

In the simulation here presented there are some drawbacks of the obtained plan and they can be easily spotted in Figure 5.5. The satellites "Sentinel-2A", "CubeSat-3", "CubeSat-7" and "CubeSat-8" do not have any scheduled tasks. Moreover, the data acquired by "Sentinel-2B" , COSMO-SkyMed-4 ("CSK-4") and partially "CubeSat-17" are not transferred. While the unused satellites do not create any problems, since that is only an indicator that the constellation is able to bear more observation targets, the missed downlinks translate into an unavailability of the acquired data. In detail, the acquisition data for "Target 62" (priority 2), "Target 30" (priority 4) and "Target 47" (priority 3) are not transferred during the plan although they are scheduled. That could be solved with more ground stations or with a extended time slot for the plan execution. However, the remaining 95 targets are observed and their data transferred according to a successful plan execution.

In a real situation, the plan would be ready to be communicated to the constellation. For the purpose of this thesis, multiple simulations have been run, with different sample times, in order to compare the results and computation time. Although it may seem counter-intuitive, bigger sample time does not always translate into a faster analysis. The aforementioned 15 s time resolution took approximately 12 hours. Two 10 s resolution simulations took almost 14 hours each. Two 30 s simulations produced a schedule in less than 10 hours. The 60 s simulation lasted more than 15 hours. The explanation is that less resolution means less accuracy and flexibility in determining and assigning the visibility windows, having more conflicting choices made by the satellites that the algorithm has to coordinate. Nevertheless, the number of unscheduled targets increased with less time resolution with a mean value of 2 unscheduled targets for the 10 s, 2.5 for 30 s and 4 for 60 s. The payoff difference was anyway less than 2.7% in comparison to the one of the plan with the 15 s time resolution. It is interesting to notice that while for 10 s and 15 s simulations the target acquisitions were concentrated in the first half of the plan, for the other simulations the observations were more spread apart. That is due to the limited number of suitable windows. That also translated into more scheduled downlinks, because data were more distant in terms of time but their transfer shall be executed as close as possible to the acquisition.

5.6 Dynamic Scheduling

5.6.1 Short Notice Tasking

In this hypothetical mission, during the night between February 5th and 6th, a emergency occurs and the second set of targets has to be scheduled with high priority. The old plan has already been communicated to the constellation, therefore a new plan should be transferred and be operating starting from February 6th at 5 am. The maximum duration of the planning process shall be 3 hours, after which the plan would be outdated.

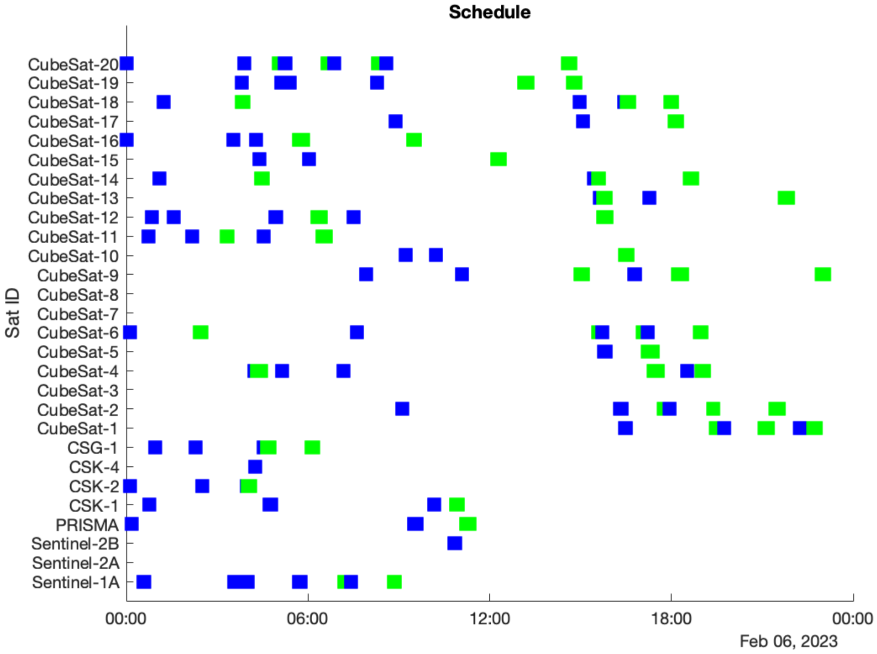


Figure 5.4: The plot of the schedule of the Case Study with 15 s sample time (blue squares are observations; green squares are downlinks).

In the new obtained plan, two of the emergency target observations are left unscheduled. The types of the observations are "Visible" and "Infrared". Looking at the previous plan, it is possible to notice that all the satellites equipped with suitable sensors either do not have a scheduled downlink before the visibility windows with the targets or they have it before the start of the new plan. There are also two more cases. "CubeSat-6" has a scheduled link to "Kiruna 2" that ends at 15:36:00 and has a visibility window to "Emergency 1" starting at 15:39:30. The problem is that the sensor is passive and needs the sunlight, but in the area of the target the sunset is at 15:08:15. The same happens for "CubeSat-9" and "Emergency 3": there is a link to "Svalbard" ending at 15:04:15, a window starting at 15:10:45, but shadow starting at 15:01:45. The algorithm correctly managed to exclude those windows from the feasible choices.

The changes of the plan are displayed in Table 5.7. For "CubeSat-11" a new observation and a new downlink are scheduled after the preexisting tasks. For "CubeSat-19" and "CubeSat-20" a new downlink each is scheduled before the new observations. The reason is that there are already scheduled links after the new acquisitions and therefore the new downlinks are meant to transfer the data coming from previous observations.

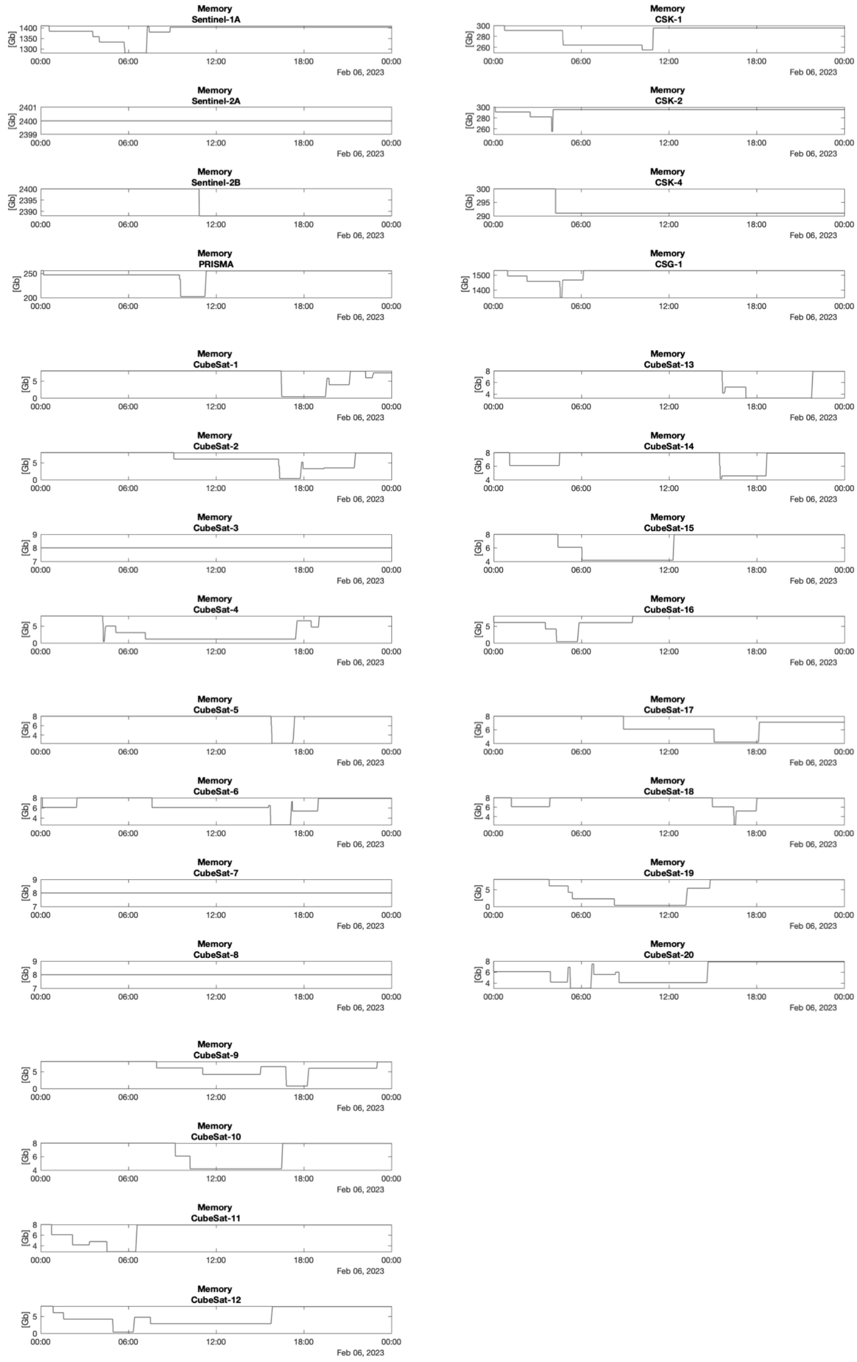


Figure 5.5: The available on-board memory during the Case Study.

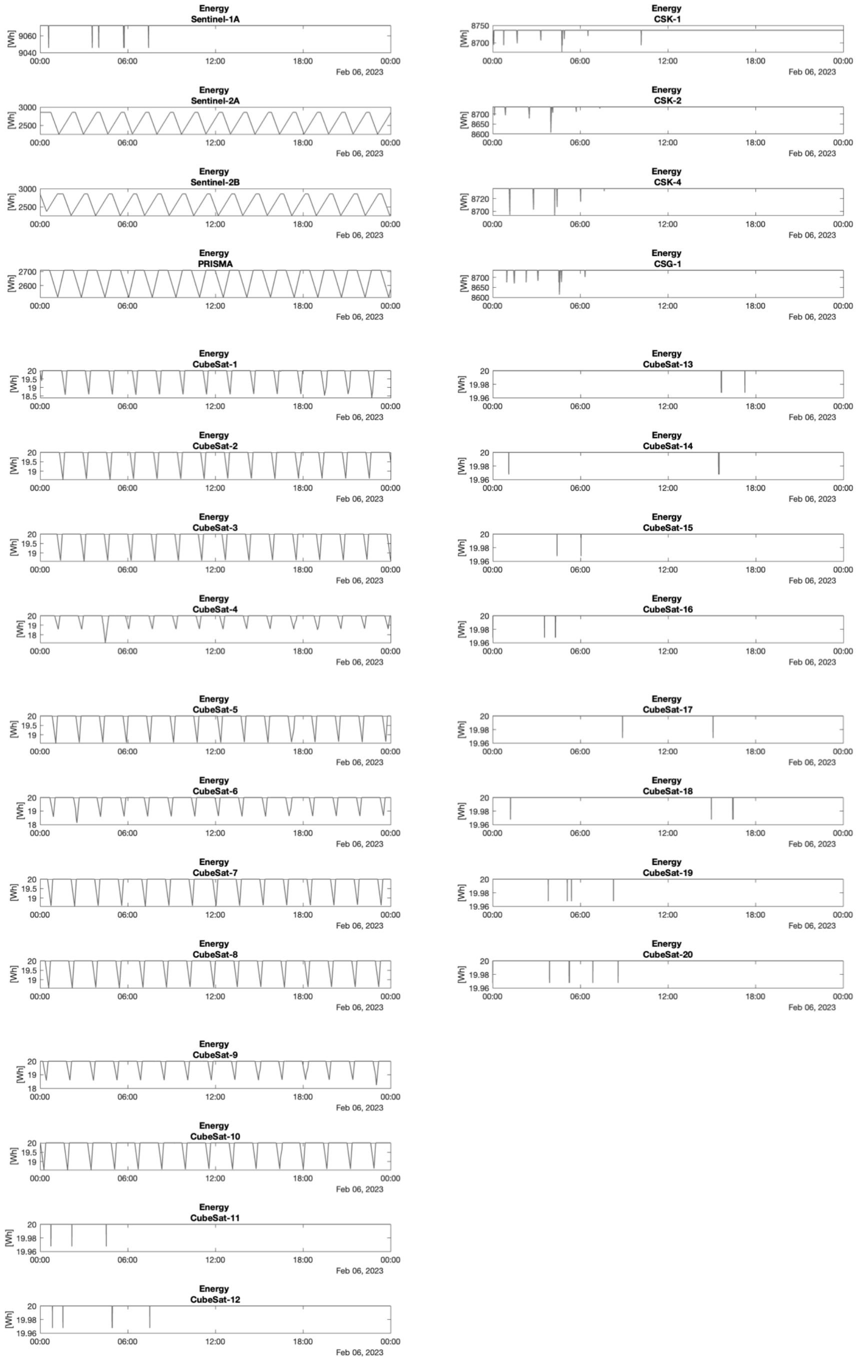


Figure 5.6: The electrical energy during the Case Study.

5.6.2 Unexpected Failures

To further show a possible utilisation of the software, a randomly selected satellite ("CubeSat-20") is exposed to three randomly selected failed activities: the observations of "Target 50" and "Target 75" and the downlink to "Kiruna 1". The spacecraft has thus to re-plan the tasks three times. The algorithm made the new plan start 15 minutes after the failed activity, although the re-planning process took less then 10 minutes for the three cases. As described in Section 4.8, the satellite can keep the tasks or assign them to other spacecrafts. In all the cases, the activities were kept by "CubeSat-20", because it was alternately the optimal or the only feasible solution.

After the first failure, the algorithm deleted the link to "Kiruna 2", because the observation data from that acquisition are no longer present, scheduled a downlink after the observation of "Emergency 5" and a link after the acquisition of "Target 50", which is starting at 17:46:00. The second failure was fixed by planning the observation of the target at 20:50:45. The first link to "Cordoba" was replaced by a extension of the later link to the same ground station. The downlink to "Kiruna 1" was substituted with a link to "Svalbard". All the new scheduled activities are shown in Table 5.8.

The example proved that the algorithm could deal with unexpected events if running onboard the satellite in reasonable times. Indeed it was able to continuously adapt the schedule even after consecutive failures.

Source	Target	Start Time	End Time	Duration	Source	Target	Start Time	End Time	Duration
Sentinel-1A	Target 82	06-Feb-2023 00:34:15	06-Feb-2023 00:35:30	75	CubeSat-6	Target 69	06-Feb-2023 15:43:45	06-Feb-2023 15:44:00	15
Sentinel-1A	Target 74	06-Feb-2023 03:33:30	06-Feb-2023 03:34:45	75	CubeSat-6	Svalbard	06-Feb-2023 17:03:45	06-Feb-2023 17:10:45	420
Sentinel-1A	Target 52	06-Feb-2023 04:00:15	06-Feb-2023 04:01:30	75	CubeSat-6	Target 94	06-Feb-2023 17:13:30	06-Feb-2023 17:13:45	15
Sentinel-1A	Target 31	06-Feb-2023 05:42:30	06-Feb-2023 05:43:45	75	CubeSat-6	Maspalomas	06-Feb-2023 18:55:30	06-Feb-2023 19:00:00	270
Sentinel-1A	Target 40	06-Feb-2023 05:44:45	06-Feb-2023 05:46:00	75	CubeSat-9	Target 96	06-Feb-2023 07:55:45	06-Feb-2023 07:56:00	15
Sentinel-1A	Kiruna 2	06-Feb-2023 07:12:30	06-Feb-2023 07:18:00	330	CubeSat-9	Target 95	06-Feb-2023 11:05:00	06-Feb-2023 11:05:15	15
Sentinel-1A	Target 55	06-Feb-2023 07:24:45	06-Feb-2023 07:26:00	75	CubeSat-9	Svalbard	06-Feb-2023 15:00:00	06-Feb-2023 15:04:15	255
Sentinel-1A	Kiruna 2	06-Feb-2023 08:49:30	06-Feb-2023 08:51:45	135	CubeSat-9	Target 20	06-Feb-2023 16:46:30	06-Feb-2023 16:46:45	15
Sentinel-2B	Target 62	06-Feb-2023 10:50:00	06-Feb-2023 10:51:15	75	CubeSat-9	Target 57	06-Feb-2023 16:47:00	06-Feb-2023 16:47:15	15
PRISMA	Target 61	06-Feb-2023 00:11:15	06-Feb-2023 00:11:30	15	CubeSat-9	Target 56	06-Feb-2023 16:48:30	06-Feb-2023 16:48:45	15
PRISMA	Target 44	06-Feb-2023 09:30:00	06-Feb-2023 09:30:15	15	CubeSat-9	Kiruna 1	06-Feb-2023 18:13:00	06-Feb-2023 18:20:45	465
PRISMA	Target 15	06-Feb-2023 09:33:45	06-Feb-2023 09:34:00	15	CubeSat-9	Svalbard	06-Feb-2023 22:58:15	06-Feb-2023 23:02:00	225
PRISMA	Target 17	06-Feb-2023 09:34:00	06-Feb-2023 09:34:15	15	CubeSat-10	Target 88	06-Feb-2023 09:12:45	06-Feb-2023 09:13:00	15
PRISMA	Target 25	06-Feb-2023 09:34:15	06-Feb-2023 09:34:30	15	CubeSat-10	Target 92	06-Feb-2023 10:13:15	06-Feb-2023 10:13:30	15
PRISMA	Target 29	06-Feb-2023 09:34:30	06-Feb-2023 09:34:45	15	CubeSat-10	Kiruna 1	06-Feb-2023 16:27:30	06-Feb-2023 16:33:30	360
PRISMA	Maspalomas	06-Feb-2023 11:12:45	06-Feb-2023 11:20:00	435	CubeSat-11	Target 72	06-Feb-2023 00:44:30	06-Feb-2023 00:44:45	15
CSK-1	Target 81	06-Feb-2023 00:45:45	06-Feb-2023 00:46:00	15	CubeSat-11	Target 79	06-Feb-2023 02:10:45	06-Feb-2023 02:11:00	15
CSK-1	Target 32	06-Feb-2023 04:44:00	06-Feb-2023 04:44:15	15	CubeSat-11	Svalbard	06-Feb-2023 03:18:15	06-Feb-2023 03:20:30	135
CSK-1	Target 19	06-Feb-2023 04:44:45	06-Feb-2023 04:45:00	15	CubeSat-11	Target 89	06-Feb-2023 04:32:00	06-Feb-2023 04:32:15	15
CSK-1	Target 36	06-Feb-2023 04:46:30	06-Feb-2023 04:46:45	15	CubeSat-11	Svalbard	06-Feb-2023 06:28:45	06-Feb-2023 06:36:15	450
CSK-1	Target 77	06-Feb-2023 10:10:30	06-Feb-2023 10:10:45	15	CubeSat-12	Target 73	06-Feb-2023 00:51:00	06-Feb-2023 00:51:15	15
CSK-1	Cordoba	06-Feb-2023 10:52:45	06-Feb-2023 10:56:30	225	CubeSat-12	Target 90	06-Feb-2023 01:34:00	06-Feb-2023 01:34:15	15
CSK-2	Target 64	06-Feb-2023 00:07:15	06-Feb-2023 00:07:30	15	CubeSat-12	Target 4	06-Feb-2023 04:56:00	06-Feb-2023 04:56:15	15
CSK-2	Target 53	06-Feb-2023 02:30:45	06-Feb-2023 02:31:00	15	CubeSat-12	Target 28	06-Feb-2023 04:57:00	06-Feb-2023 04:57:15	15
CSK-2	Target 13	06-Feb-2023 03:58:30	06-Feb-2023 03:58:45	15	CubeSat-12	Svalbard	06-Feb-2023 06:19:00	06-Feb-2023 06:25:45	405
CSK-2	Target 26	06-Feb-2023 03:58:45	06-Feb-2023 03:59:00	15	CubeSat-12	Target 97	06-Feb-2023 07:30:45	06-Feb-2023 07:31:00	15
CSK-2	Target 24	06-Feb-2023 03:59:00	06-Feb-2023 03:59:15	15	CubeSat-12	Matera	06-Feb-2023 15:44:00	06-Feb-2023 15:51:30	450
CSK-2	Kiruna 2	06-Feb-2023 04:01:15	06-Feb-2023 04:05:00	225	CubeSat-13	Target 14	06-Feb-2023 15:38:00	06-Feb-2023 15:38:15	15
CSK-4	Target 30	06-Feb-2023 04:14:45	06-Feb-2023 04:15:00	15	CubeSat-13	Target 11	06-Feb-2023 15:39:45	06-Feb-2023 15:40:00	15
CSG-1	Target 99	06-Feb-2023 00:58:15	06-Feb-2023 00:58:30	15	CubeSat-13	Svalbard	06-Feb-2023 15:45:30	06-Feb-2023 15:47:15	105
CSG-1	Target 91	06-Feb-2023 02:17:45	06-Feb-2023 02:18:00	15	CubeSat-13	Svalbard	06-Feb-2023 15:48:00	06-Feb-2023 15:50:30	150
CSG-1	Target 21	06-Feb-2023 04:32:00	06-Feb-2023 04:32:15	15	CubeSat-13	Target 43	06-Feb-2023 17:15:30	06-Feb-2023 17:15:45	15
CSG-1	Target 10	06-Feb-2023 04:33:45	06-Feb-2023 04:34:00	15	CubeSat-13	Cordoba	06-Feb-2023 21:43:30	06-Feb-2023 21:50:30	420
CSG-1	Target 7	06-Feb-2023 04:34:00	06-Feb-2023 04:34:15	15	CubeSat-14	Target 78	06-Feb-2023 01:05:45	06-Feb-2023 01:06:00	15
CSG-1	Kiruna 1	06-Feb-2023 04:39:15	06-Feb-2023 04:44:30	315	CubeSat-14	Kiruna 1	06-Feb-2023 04:27:45	06-Feb-2023 04:31:30	225
CSG-1	Maspalomas	06-Feb-2023 06:06:30	06-Feb-2023 06:10:00	210	CubeSat-14	Target 58	06-Feb-2023 15:27:15	06-Feb-2023 15:27:30	15
CubeSat-1	Target 27	06-Feb-2023 16:27:45	06-Feb-2023 16:28:00	15	CubeSat-14	Target 39	06-Feb-2023 15:29:45	06-Feb-2023 15:30:00	15
CubeSat-1	Target 37	06-Feb-2023 16:28:00	06-Feb-2023 16:28:15	15	CubeSat-14	Svalbard	06-Feb-2023 15:35:00	06-Feb-2023 15:37:00	120
CubeSat-1	Target 2	06-Feb-2023 16:28:45	06-Feb-2023 16:29:00	15	CubeSat-14	Maspalomas	06-Feb-2023 18:36:30	06-Feb-2023 18:42:00	330
CubeSat-1	Target 18	06-Feb-2023 16:29:30	06-Feb-2023 16:29:45	15	CubeSat-15	Target 51	06-Feb-2023 04:24:00	06-Feb-2023 04:24:15	15
CubeSat-1	Svalbard	06-Feb-2023 19:27:45	06-Feb-2023 19:35:45	480	CubeSat-15	Target 54	06-Feb-2023 06:02:30	06-Feb-2023 06:02:45	15
CubeSat-1	Target 84	06-Feb-2023 19:44:00	06-Feb-2023 19:44:15	15	CubeSat-15	Svalbard	06-Feb-2023 12:15:00	06-Feb-2023 12:21:00	360
CubeSat-1	Svalbard	06-Feb-2023 21:05:00	06-Feb-2023 21:11:15	375	CubeSat-16	Target 98	06-Feb-2023 00:00:30	06-Feb-2023 00:00:45	15
CubeSat-1	Target 70	06-Feb-2023 22:13:45	06-Feb-2023 22:14:00	15	CubeSat-16	Target 66	06-Feb-2023 03:32:15	06-Feb-2023 03:32:30	15
CubeSat-1	Svalbard	06-Feb-2023 22:40:30	06-Feb-2023 22:43:00	150	CubeSat-16	Target 41	06-Feb-2023 04:16:45	06-Feb-2023 04:17:00	15
CubeSat-1	Svalbard	06-Feb-2023 22:43:30	06-Feb-2023 22:45:45	135	CubeSat-16	Target 9	06-Feb-2023 04:18:00	06-Feb-2023 04:18:15	15
CubeSat-2	Target 80	06-Feb-2023 09:06:30	06-Feb-2023 09:06:45	15	CubeSat-16	Kiruna 1	06-Feb-2023 05:42:30	06-Feb-2023 05:50:45	495
CubeSat-2	Target 59	06-Feb-2023 16:17:15	06-Feb-2023 16:17:30	15	CubeSat-16	Cordoba	06-Feb-2023 09:27:45	06-Feb-2023 09:31:30	225
CubeSat-2	Target 22	06-Feb-2023 16:20:00	06-Feb-2023 16:20:15	15	CubeSat-17	Target 46	06-Feb-2023 08:53:30	06-Feb-2023 08:53:45	15
CubeSat-2	Target 8	06-Feb-2023 16:21:00	06-Feb-2023 16:21:15	15	CubeSat-17	Target 47	06-Feb-2023 15:05:15	06-Feb-2023 15:05:30	15
CubeSat-2	Kiruna 1	06-Feb-2023 17:44:30	06-Feb-2023 17:51:45	435	CubeSat-17	Maspalomas	06-Feb-2023 18:07:00	06-Feb-2023 18:12:00	300
CubeSat-2	Target 49	06-Feb-2023 17:57:00	06-Feb-2023 17:57:15	15	CubeSat-18	Target 87	06-Feb-2023 01:13:30	06-Feb-2023 01:13:45	15
CubeSat-2	Kiruna 1	06-Feb-2023 19:22:00	06-Feb-2023 19:23:45	105	CubeSat-18	Kiruna 2	06-Feb-2023 03:48:30	06-Feb-2023 03:52:15	225
CubeSat-2	Cordoba	06-Feb-2023 21:26:00	06-Feb-2023 21:32:45	405	CubeSat-18	Target 48	06-Feb-2023 14:58:15	06-Feb-2023 14:58:30	15
CubeSat-4	Target 38	06-Feb-2023 04:14:30	06-Feb-2023 04:14:45	15	CubeSat-18	Target 33	06-Feb-2023 16:25:45	06-Feb-2023 16:26:00	15
CubeSat-4	Target 3	06-Feb-2023 04:16:15	06-Feb-2023 04:16:30	15	CubeSat-18	Target 5	06-Feb-2023 16:27:45	06-Feb-2023 16:28:00	15
CubeSat-4	Target 12	06-Feb-2023 04:16:45	06-Feb-2023 04:17:00	15	CubeSat-18	Kiruna 2	06-Feb-2023 16:31:45	06-Feb-2023 16:36:45	300
CubeSat-4	Target 34	06-Feb-2023 04:17:00	06-Feb-2023 04:17:15	15	CubeSat-18	Maspalomas	06-Feb-2023 17:57:15	06-Feb-2023 18:02:00	285
CubeSat-4	Kiruna 2	06-Feb-2023 04:19:15	06-Feb-2023 04:26:00	405	CubeSat-19	Target 60	06-Feb-2023 03:48:30	06-Feb-2023 03:48:45	15
CubeSat-4	Svalbard	06-Feb-2023 04:26:15	06-Feb-2023 04:28:00	105	CubeSat-19	Target 63	06-Feb-2023 05:06:30	06-Feb-2023 05:06:45	15
CubeSat-4	Target 71	06-Feb-2023 05:08:00	06-Feb-2023 05:08:15	15	CubeSat-19	Target 1	06-Feb-2023 05:23:45	06-Feb-2023 05:24:00	15
CubeSat-4	Target 93	06-Feb-2023 07:10:00	06-Feb-2023 07:10:15	15	CubeSat-19	Target 67	06-Feb-2023 08:16:15	06-Feb-2023 08:16:30	15
CubeSat-4	Kiruna 1	06-Feb-2023 17:25:00	06-Feb-2023 17:33:00	480	CubeSat-19	Kiruna 2	06-Feb-2023 13:08:15	06-Feb-2023 13:15:45	450
CubeSat-4	Target 85	06-Feb-2023 18:30:30	06-Feb-2023 18:30:45	15	CubeSat-19	Kiruna 2	06-Feb-2023 14:45:30	06-Feb-2023 14:50:00	270
CubeSat-4	Svalbard	06-Feb-2023 18:58:45	06-Feb-2023 19:04:00	315	CubeSat-20	Target 68	06-Feb-2023 00:00:30	06-Feb-2023 00:00:45	15
CubeSat-5	Target 45	06-Feb-2023 15:46:00	06-Feb-2023 15:46:15	15	CubeSat-20	Target 65	06-Feb-2023 03:53:45	06-Feb-2023 03:54:00	15
CubeSat-5	Target 42	06-Feb-2023 15:48:00	06-Feb-2023 15:48:15	15	CubeSat-20	Svalbard	06-Feb-2023 05:02:00	06-Feb-2023 05:06:45	285
CubeSat-5	Target 6	06-Feb-2023 15:49:00	06-Feb-2023 15:49:15	15	CubeSat-20	Target 23	06-Feb-2023 05:14:15	06-Feb-2023 05:14:30	15
CubeSat-5	Svalbard	06-Feb-2023 17:13:00	06-Feb-2023 17:15:15	135	CubeSat-20	Target 16	06-Feb-2023 05:15:30	06-Feb-2023 05:15:45	15
CubeSat-5	Kiruna 2	06-Feb-2023 17:15:15	06-Feb-2023 17:22:45	450	CubeSat-20	Svalbard	06-Feb-2023 06:38:00	06-Feb-2023 06:44:45	405
CubeSat-6	Target 83	06-Feb-2023 00:06:45	06-Feb-2023 00:07:00	15	CubeSat-20	Target 50	06-Feb-2023 06:51:30	06-Feb-2023 06:51:45	15
CubeSat-6	Kiruna 2	06-Feb-2023 02:25:45	06-Feb-2023 02:29:30	225	CubeSat-20	Kiruna 1	06-Feb-2023 08:19:00	06-Feb-2023 08:21:00	120
CubeSat-6	Target 86	06-Feb-2023 07:37:15	06-Feb-2023 07:37:30	15	CubeSat-20	Target 75	06-Feb-2023 08:35:15	06-Feb-2023 08:35:30	15
CubeSat-6	Kiruna 2	06-Feb-2023 15:34:00	06-Feb-2023 15:36:00	120	CubeSat-20	Kiruna 2	06-Feb-2023 14:34:45	06-Feb-2023 14:40:45	360
CubeSat-6	Target 35	06-Feb-2023 15:42:45	06-Feb-2023 15:43:00	15					

Table 5.6: Static schedule for the Case Study.

Source	Target	Start Time	End Time	Duration
CubeSat-11	Emergency 4	06-Feb-2023 14:19:45	06-Feb-2023 14:20:00	15
CubeSat-11	Matera	06-Feb-2023 15:56:30	06-Feb-2023 16:00:30	240
CubeSat-19	Kiruna 2	06-Feb-2023 05:16:30	06-Feb-2023 05:20:30	240
CubeSat-19	Emergency 2	06-Feb-2023 14:37:00	06-Feb-2023 14:37:15	15
CubeSat-20	Cordoba	06-Feb-2023 08:47:45	06-Feb-2023 08:51:45	240
CubeSat-20	Emergency 5	06-Feb-2023 14:28:30	06-Feb-2023 14:28:45	15

Table 5.7: The new scheduled activities for the emergency plan of the Case Study.

Source	Target	Start Time	End Time	Duration
CubeSat-20	Maspalomas	06-Feb-2023 17:39:15	06-Feb-2023 17:43:00	225
CubeSat-20	Target 50	06-Feb-2023 17:46:00	06-Feb-2023 17:46:15	15
CubeSat-20	Cordoba	06-Feb-2023 22:14:30	06-Feb-2023 22:17:45	195
CubeSat-20	Target 75	06-Feb-2023 20:50:45	06-Feb-2023 20:51:00	15
CubeSat-20	Cordoba	06-Feb-2023 22:13:45	06-Feb-2023 22:17:45	240
CubeSat-20	Svalbard	06-Feb-2023 17:52:30	06-Feb-2023 17:56:45	255

Table 5.8: The new scheduled activities of the Case Study after the failures.

Chapter 6

Conclusions

As the State-of-the-Art has shown, the mission planning is still a current and lively research topic. New techniques that make planning for satellites more intelligent and autonomous can solve the issues that future space missions will face, such as the overloading of ground stations or the need to schedule a copious amount of tasks distributed in constellations composed of a large number of spacecrafts. However the solving algorithms explored so far still have the disadvantage of high computational cost, which leads to significant delays between task request, planning and actual execution. That is an aspect that becomes especially critical when dealing with dynamic planning. As seen, many researchers have proposed various ways to solve this problem, but the greatest hope for future developments is currently artificial intelligence. To date, that technology is still in its infancy, but in a few years it could speed up the meta-heuristic approaches currently employed.

The Mission Planner presented in this thesis shows many of the limitations mentioned above. Its greatest weakness is certainly the computation time required for large constellations and large sets of static planning tasks. Nevertheless, the software has innumerable strengths. First and foremost, it is simple to use and suitable for all kinds of users. It is easily adaptable to different types of architectures, without requiring specific hard-coding work. The results it offers are also optimised and take into account the various constraints that space systems encounter in remote sensing missions. The Case Study has shown how it can already be used for the planning of Earth-observation constellations, which represent a strategic development for national space agencies, with great contribution for scientific research and especially for emergency management.

6.1 Future Work

Future plans are to make the software faster and even more adaptable, without sacrificing accuracy. Many features can still be optimised in the code and any improvements in the algorithm duration will allow new aspects of a space mission to be taken into account.

In particular, one of the advisable developments concerns flight mechanics and thus the use of fuel, both for manoeuvres required for payload acquisitions and for orbit and formation keeping.

By increasing the computation speed, more attention could also be directed to the TT&C and dedicated ground stations could be considered.

Furthermore, targets are currently only represented as points. A strip can be entered as a set of targets, but in the future, the software may treat it in a more conscious manner, without the use of artefactual solutions.

Another level of accuracy could be also reached by treating in a more detailed manner the thermal control system.

Additionally, a new figure of merit concerning the resolution of sensors could be added to the fitness function.

The list of improvements could continue to infinity, since every small detail would make the model more accurate and fitting with reality. But, as often mentioned in this thesis while presenting the trade-offs the development of the Mission Planner has faced, the real bottleneck is the planning time. In fact, in such operational problem, the duration of each scheduling activity is crucial. Therefore, any new features of the software will firstly need to be optimised to maintain (if not even improve) the planning duration currently achieved.

Bibliography

[1] D. Modenini G. Curzi and P. Tortora. «Large Constellations of Small Satellites: A Survey of Near Future Challenges and Missions». In: *Aerospace* Vol. 7 (2020), p. 133 (cit. on pp. 1, 2, 27).

[2] J. R. Wertz and W. J. Larson (editors). *Space Mission Analysis and Design.* Third Edition. Space Technology Library. El Segundo, CA, USA: Microcosm Press and Kluwer Academic Publishers, 1999, pp. 135–136, 194–195, 428–434, 587–620 (cit. on pp. 1, 17, 45).

[3] Consulative Committee for Space Data Systems (CCSDS). *Mission Planning and Scheduling.* Green Book 529.0-G-1. Washington, DC, USA: CCSDS, June 2018. URL: `https://public.ccsds.org/Pubs/529x0g1.pdf` (cit. on p. 2).

[4] R.A. Richards et al. «Distributed Satellite Constellation Planning and Scheduling». In: *FLAIRS Conference AAAI.* Key West, FL, USA, May 2001, pp. 68–72 (cit. on p. 2).

[5] G. Wu et al. «Multi-satellite observation integrated scheduling method oriented to emergency tasks and common tasks». In: *Journal of Systems Engineering and Electronics* Vol. 23 (2012), pp. 723–733 (cit. on pp. 4, 6, 7, 11, 12, 14, 25).

[6] G. Zhang et al. «Mission Planning Issues of Imaging Satellites: Summary, Discussion, and Prospects». In: *International Journal of Aerospace Engineering* Vol. 2021 (2021) (cit. on pp. 4–7, 10, 14).

[7] C. G. Valicka et al. «Mixed-Integer Programming Models for Optimal Constellation Scheduling Given Cloud Cover Uncertainty». In: *European Journal of Operational Research* Vol. 275 (2019), pp. 431–445 (cit. on pp. 4, 5, 10).

[8] M. Vasquez and J. Hao. «A “Logic-Constrained” Knapsack Formulation and a Tabu Algorithm for the Daily Photograph Scheduling of an Earth Observation Satellite». In: *Computational Optimization and Applications* Vol. 20 (2001), pp. 137–157 (cit. on pp. 4, 6, 13).

[9] M. Lemaître et al. «Selecting and scheduling observations of agile satellites». In: *Aerospace Science and Technology* Vol. 6 (2002), pp. 367–381 (cit. on pp. 4, 12).

[10] J. C. Pemberton and F. Galiber. «A constraint-based approach to satellite scheduling». In: *Constraint Programming and Large Scale Discrete Optimization.* 1998 (cit. on pp. 4, 25).

[11] M. Abramson et al. «Real-Time Optimized Earth Observation Autonomous Planning». In: *NASA Earth Science Technology Conference.* Houston, TX, USA, Oct. 2002, pp. 68–72 (cit. on pp. 5, 20, 27).

[12] L. Liu et al. «A Study of Distributed Earth Observation Satellites Mission Scheduling Method Based on Game-Negotiation Mechanism». In: *Sensors* Vol. 21, 6660 (2021) (cit. on pp. 8, 11, 25, 28, 34).

[13] P. Skobelev et al. «Multi-Agent Planning of Spacecraft Group for Earth Remote Sensing». In: *Service Orientation in Holonic and Multi-Agent Manufacturing* (2016), pp. 309–317 (cit. on pp. 8, 25).

[14] J. Van Der Horst and J. Noble. «Task allocation in dynamic networks of satellites». In: *IJCAI Workshop on Artificial Intelligence in Space.* Barcelona, Spain, July 2011 (cit. on p. 8).

[15] M.C. Brito D.M. Surka and C.G Harvey. «The real-time ObjectAgent software architecture for distributed satellite systems». In: *2001 IEEE Aerospace Conference Proceedings.* Vol. Vol. 6. 2001, pp. 2731–2741 (cit. on p. 8).

[16] M. Campbell T. Schetter and D. Surka. «Multiple agent-based autonomy for satellite constellations». In: *Artificial Intelligence* Vol. 145 (2003), pp. 147–180 (cit. on p. 8).

[17] I. H. Osman and G. Laporte. «Metaheuristics: A bibliography». In: *Annals of Operational Research* Vol. 63 (1996), pp. 513–628 (cit. on p. 10).

[18] C. Iacopino et al. «How ants can manage your satellites». In: *IJCAI 2013 workshop AI in Space: Intelligence beyond planet Earth.* Bejing, China, Aug. 2013 (cit. on pp. 11, 12).

[19] C. Li et al. «Data-driven Onboard Scheduling for an Autonomous Observation Satellite». In: *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18).* 2018, pp. 5773–5774 (cit. on pp. 11, 14).

[20] Y. Chen et al. «Task Planning for Multiple-Satellite Space-Situational-Awareness Systems». In: *Aerospace* Vol. 8 (2021), p. 73 (cit. on p. 12).

[21] C. Blum and A. Roli. «Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison». In: *ACM Computing Surveys* Vol. 35 (2001), pp. 268–308 (cit. on pp. 13, 27).

[22] N. Bianchessi et al. «A Heuristic for the Multi-Satellite, Multi-Orbit and Multi-User Management of Earth Observation Satellites». In: *European Journal of Operational Research* Vol. 177 (2007), pp. 750–762 (cit. on p. 13).

[23] MathWorks. *Aerospace Toolbox Documentation*. URL: `https://mathworks.com/help/aerotbx/` (visited on 10/18/2022) (cit. on p. 16).

[24] Space-Track.Org. *Basic Description of the Two Line Element (TLE) Format*. URL: `https://www.space-track.org/documentation#tle-basic` (visited on 10/18/2022) (cit. on pp. 18, 19).

[25] T. S. Kelso. «Validation of SGP4 and IS-GPS-200D Against GPS Precision Ephemerides». In: *17th AAS/AIAA Space Flight Mechanics Conference*. Sedona, AZ, USA, 2007 (cit. on p. 20).

[26] MathWorks. *Orbit Propagator Documentation*. URL: `https://www.mathworks.com/help/aeroblks/orbitpropagator.html` (visited on 10/28/2022) (cit. on p. 20).

[27] I. Reda and A. Andreas. *Solar Position Algorithm for Solar Radiation Applications (Revised)*. USA, 2008. DOI: `10.2172/15003974`. URL: `https://www.osti.gov/biblio/15003974` (cit. on p. 23).

[28] C. R. Ortiz Longo and S. L. Rickman. *Method for the Calculation of Spacecraft Umbra and Penumbra Shadow Terminator Points*. NASA Technical Paper 3547. Houston, TX, USA: NASA, Apr. 1995 (cit. on p. 23).

[29] A. Globus et al. «A Comparison of Techniques for Scheduling Earth Observing Satellites». In: *Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence*. San Jose, CA, USA, July 2004, pp. 836–843 (cit. on p. 25).

[30] J. C. Pemberton and L. G. Greenwald. «On the Need for Dynamic Scheduling of Imaging Satellites». In: *Pecora 15/Land Satellite Information IV/ISPRS Commission I/FIEOS 2002 Conference Proceedings*. 2002 (cit. on p. 34).

[31] Planetek Italia. *OSIRIDE: Operational Solutions for the Italian IRIDE Constellation*. URL: `https://www.planetek.it/eng/news_events/news_archive/2022/11/osiride_operational_solutions_for_the_italian_iride_constellation` (visited on 01/10/2023) (cit. on p. 44).

[32] eoPortal. *Copernicus: Sentinel-1*. URL: `https://www.eoportal.org/satellite-missions/copernicus-sentinel-1` (visited on 01/18/2023) (cit. on p. 44).

[33] eoPortal. *Copernicus: Sentinel-2*. URL: `https://www.eoportal.org/satellite-missions/copernicus-sentinel-2` (visited on 01/18/2023) (cit. on p. 44).

[34] Copernicus Space Component Mission Management Team. *Sentinel High Level Operations Plan (HLOP)*. European Space Agency (ESA), Dec. 2021 (cit. on p. 44).

[35] eoPortal. *COSMO-SkyMed*. URL: `https://www.eoportal.org/satellite-missions/cosmo-skymed` (visited on 01/18/2023) (cit. on p. 44).

[36] eoPortal. *COSMO-SkyMed - Second Generation*. URL: `https://www.eoportal.org/satellite-missions/cosmo-skymed-second-generation` (visited on 01/18/2023) (cit. on p. 44).

[37] eoPortal. *PRISMA (Hyperspectral)*. URL: `https://www.eoportal.org/satellite-missions/prisma-hyperspectral` (visited on 01/18/2023) (cit. on p. 44).

[38] S. R. Tsitas and J. Kingston. «6U CubeSat design for Earth observation with 6.5m GSD, five spectral bands and 14Mbps downlink». In: *The Aeronautical Journal* Vol. 114 (2010), pp. 689–697 (cit. on p. 44).