

POLITECNICO DI TORINO

Master's Degree in Mechatronics Engineering



**Politecnico
di Torino**

Master's Degree Thesis

Scheduling of Agile Earth Observation Constellation using Quantum Annealing

Supervisors

Prof. Carlo NOVARA

Eng. Mattia BOGGIO

Eng. Luca MASSOTTI

Eng. Deborah VOLPE

Candidate

Vinicius MARCHIOLI

February 2025

Abstract

As Earth Observation (EO) missions advance towards Agile Earth Observation Satellites, the complexity of scheduling problems increases, posing challenges for traditional optimization methods. This thesis investigates the potential of a quantum algorithm to address the scheduling problem in EO constellations. In particular, a novel formulation of the satellite constellation optimization problem is proposed, translating it into a Quadratic Unconstrained Binary Optimization (QUBO) problem, i.e., compliant with quantum solvers. Penalty functions are incorporated to optimize mission energy consumption. The formulated QUBO problem is then implemented and solved on a real quantum computer (a D-Wave Quantum Annealer). The performance provided by the quantum machine is compared with established classical meta-heuristic solvers like Simulated Annealing and Tabu Search. The results show that the proposed quantum optimization process achieves better results in terms of both solution quality and computational efficiency.

Keywords: Quantum optimization, QUBO, Earth Observation mission, Satellite Constellation Scheduling

Table of Contents

List of Tables	IV
List of Figures	V
Acronyms	VIII
1 Introduction	1
1.1 A background on quantum computing for optimization	3
1.2 Problem scope	4
1.3 Thesis outline	4
2 Problem definition	6
2.1 The integer programming model	6
2.1.1 Region F	7
2.1.2 Quality of acquisition	8
2.2 Quadratic Unconstrained Binary Optimization (QUBO)	9
2.3 QUBO model for single satellite scheduling	11
2.4 QUBO model for constellation scheduling	14
3 Overview of meta-heuristic algorithms	15
3.1 Classical algorithms	16
3.1.1 Tabu Search	17
3.1.2 Simulated Annealing	18
3.2 Quantum algorithms	19
3.2.1 Quantum Annealing	19
3.2.2 Hybrid Quantum Annealing	23
4 Design of mission scenarios	25
4.1 Data pre-processing procedure	25
4.2 The satellite constellation	28
4.3 Dataset I: European capitals	29

4.4	Dataset II: UNESCO Natural World Heritage sites	33
5	Results	36
5.1	Dataset I - European capitals	36
5.1.1	Tabu Search	37
5.1.2	Simulated Annealing	38
5.1.3	Hybrid Quantum Annealing	41
5.1.4	Dataset I comparison	42
5.2	Dataset II - UNESCO Natural World Heritage sites	43
5.2.1	Tabu Search	43
5.2.2	Simulated Annealing	48
5.2.3	Quantum Annealing	52
5.2.4	Dataset II results comparison	57
6	Conclusions	58
6.1	Future Works	59
A	Quantum and Hybrid Annealing setup	60
B	QUBO matrix formulation	63
C	Data pre-processing	67
	Bibliography	73

List of Tables

4.1	Orbital elements description	27
4.2	OE for the satellite constellation	28
4.3	High density dataset: European capitals	29
4.4	Quantum Annealer dataset: UNESCO Natural World Heritage sites	33
5.1	Dataset I main properties	36
5.2	Comparison between classic and hybrid algorithms for Dataset I . .	42
5.3	Dataset II main properties	43
5.4	Comparison between classic and quantum algorithms for a subset of Dataset II - satellite 1	57
5.5	Comparison between classic and quantum algorithms for Dataset II	57

List of Figures

1.1	The extra degree of freedom regarding the pitch axis. Even at instant t_b , the S/C can maneuver to acquire the target	2
1.2	Comparison of the observation capability between different satellite types. The AEOS has a larger search space for potential solutions	2
1.3	Propagated orbits for the satellite constellation.	5
2.1	Change of roll and pitch angles during a VTW	9
2.2	Diagram of possible maneuvers from first to second target	12
3.1	A one dimensional example for a generic meta-heuristic approach [18]	16
3.2	Eigenspectrum for a generic quantum system [22]	21
3.3	Monotonic annealing functions A and B [22]. The process begins at $s = 0$ and ends at $s = 1$	22
3.4	A schematic for the D-Wave implementation of the Hybrid Solver [24]	24
4.1	The two-body problem	26
4.2	Propagated orbits for the satellite constellation.	29
4.3	European capitals: visible time window for satellite 1	31
4.4	European capitals: visible time window for satellite 2	31
4.5	European capitals: visible time window for satellite 3	32
4.6	Target disposition and constellation ground tracks over local projection for dataset I.	32
4.7	UNESCO sites: visible time window for satellite 1	33
4.8	UNESCO sites: visible time window for satellite 2	34
4.9	Target disposition and constellation ground tracks over local projection for dataset II.	35
5.1	Captured targets with Tabu Search	37
5.2	Scheduling output for each satellite with Tabu Search	38
5.3	Captured targets with Simulated Annealing	39
5.4	Scheduling output for each satellite with Simulated Annealing	40
5.5	Captured targets with Hybrid Quantum Annealing	41

5.6	Scheduling output for each satellite with Hybrid Quantum Annealing	42
5.7	Captured targets with Tabu Search	44
5.8	Scheduling output for the satellite with Tabu Search	45
5.9	Captured targets with Tabu Search	46
5.10	Scheduling output for each satellite with Tabu Search	47
5.11	Captured targets with Simulated Annealing	48
5.12	Scheduling output for the satellite with Simulated Annealing	49
5.13	Captured targets with Simulated Annealing	50
5.14	Scheduling output for each satellite with Simulated Annealing	51
5.15	Captured targets with Quantum Annealing	53
5.16	Scheduling output for the satellite with Quantum Annealing	54
5.17	Captured targets with Quantum Annealing	55
5.18	Scheduling output for each satellite with Quantum Annealing	56

Acronyms

AEOS

Agile Earth Observation Satellite

AOCS

Attitude and Orbit Control System

ECEF

Earth-Centered Earth-Fixed Frame

ECI

Earth-Centered Inertial Frame

EOS

Non-Agile Earth Observation Satellite

F

Forbidden Observation Region

LEO

Low Earth Orbit

MINLP

Mixed-Integer Nonlinear Programming

OE

Orbital Elements

OTW

Observation Time Window

QA	Quantum Annealing
QPU	Quantum Processing Unit
QUBO	Quadratic Unconstrained Binary Optimization
S	Set of Satellites
S/C	Spacecraft
SA	Simulated Annealing
T	Set of Observation Targets
VTW	Visible Time Window
w	Profit of Acquisition
ξ	Optimization Decision Variable

Chapter 1

Introduction

As space becomes more accessible through private and governmental entities, Earth observation (EO) initiatives start to constitute a critical topic of concern and interest not only regarding national sovereignty and security, but also to the research of natural phenomena and climate change, encompassing challenges like forest fires, hurricanes, and deforestation. More broadly, their significance extends to several applications, playing a pivotal role also in fostering economic development within the mining, energy and agricultural sectors. The last decades showed a major transformation on how these missions are approached. Classical endeavors tended to focus on long-lasting, multi-purpose and non-agile satellites, such as the SPOT, the ENVISAT or the ERS missions. However, even if these missions are still relevant and employed, private companies and national agencies have recently begun to give more attention to smaller Agile Earth Observation Satellites (AEOSs).

AEOSs are smaller and more agile satellites, where an extra degree of freedom around the pitch axis enable the satellites for fast and precise orientation changes in nadir. They operate in Low Earth Orbits (LEO), often in constellations, and can perform specific tasks efficiently [1, 2]. Fig. 1.1 shows the additional manoeuvrability of these satellites, while in Fig. 1.2 the main difference in observation capability between Non-Agile Earth Observation Satellite (EOS) and AEOS is illustrated. For the EOS, the specific Observation Time Window (OTW), which is the time period required for the satellite to observe a specific area on Earth, coincides with its Visible Time Window (VTW), namely the time period when the target is physically visible to the satellite's instruments. In contrast, AEOSs have expanded observation capabilities, offering multiple feasible OTWs for a single target.

While the agile characteristic greatly improves the observation efficiency and flexibility of AEOS, it also significantly increases the complexity of the scheduling problem for EO missions. This problem consists of selecting and scheduling satellite observation tasks to maximize the entire observation profit while satisfying all the

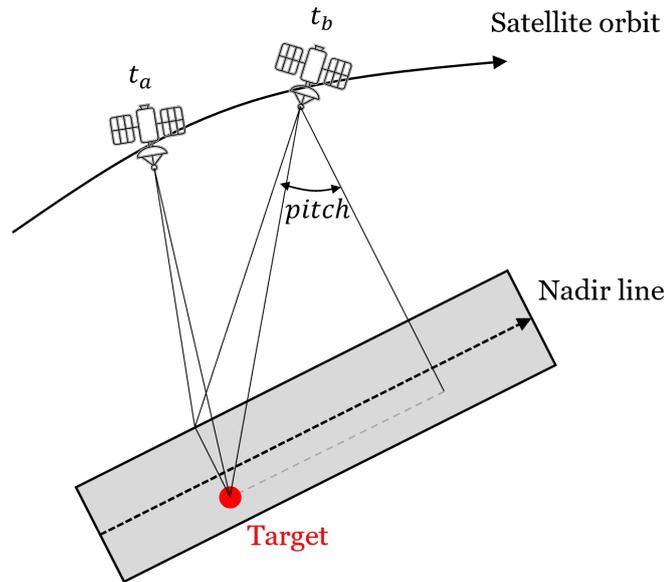


Figure 1.1: The extra degree of freedom regarding the pitch axis. Even at instant t_b , the S/C can maneuver to acquire the target

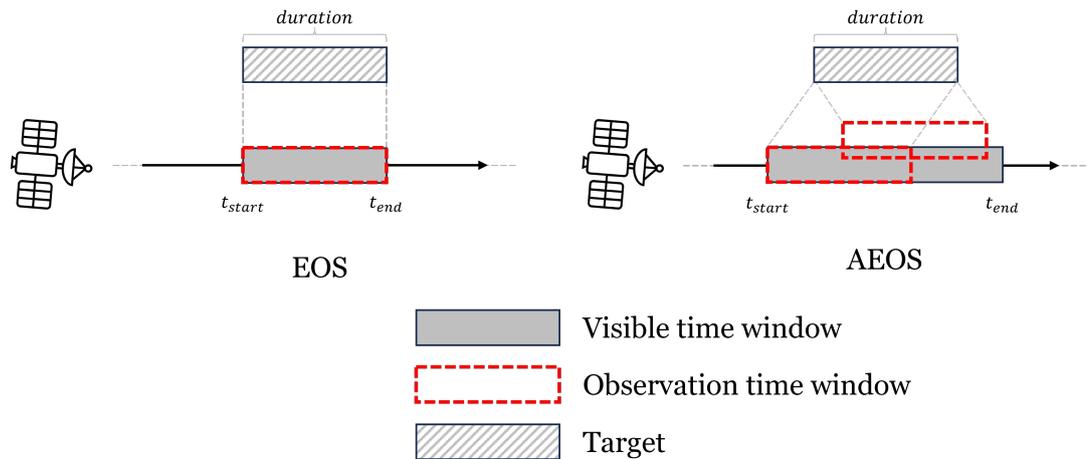


Figure 1.2: Comparison of the observation capability between different satellite types. The AEOS has a larger search space for potential solutions

operational constraints. For traditional EOS, the scheduling problem is relatively straightforward as VTW coincides with OTW. Instead, for AEOS, each VTW contains multiple potential OTWs, resulting in a considerable expansion in the search space for scheduling observations. This makes the scheduling problem NP-hard and thus potentially challenging for traditional optimization methods.

Indeed, these approaches may either struggle to find a global solution or require too long execution times to achieve satisfactory results. The limitations of traditional approaches become even more pronounced when dealing with constellations of AEOS, namely the case study considered in this work, where the complexity of the scheduling problem increases further.

In the literature, the mathematical description of these instances is compared to the traveling salesman problem, given its combinatorial nature and NP-hardness, and may be approached in different manners depending on the chosen solving method. Some divisions can be made regarding the number of satellites, their agility, mission constraints, and if the problem is dealt by a continuous or discrete approach. A pertinent bibliography review is displayed in Chapter 2, where different methods are shown and the discretized formulation to the constellation problem is explained.

1.1 A background on quantum computing for optimization

Optimization problems play a pivotal role in addressing significant challenges within the aerospace domain. Indeed, modern control techniques, resource allocation, design optimization (eg. chips, batteries, rovers ecc), space debris removal, trajectory planning and several other real-world issues are often formulated through a cost function that needs to be either maximized or minimized. Classical approaches may struggle to find a global minimum or demand too much computational power to provide a satisfactory solution. Following some publications on the same field, this thesis aims at some possible applications of Quantum Annealing, a particular type of quantum computing, for an earth observation scheduling problem. For instance, Daisuke and Yoshida [3] successfully explored a discretized model predictive control (MPC) instance utilizing the D-Wave computational architecture, while the authors in [4, 5] were able to implement an efficient mission planning for Earth observation satellites.

Even if quantum computing is in its early stages, it already has shown promising perspectives on solving a certain range of complex combinatorial problems, such as those related to artificial intelligence, cryptography and logistics. Particularly, planning problems such as the AEOSs scheduling are already formulated with binary state variables, have a combinatorial nature and many local minima, resulting in a typical problem to be addressed by the meta-heuristic quantum approach.

1.2 Problem scope

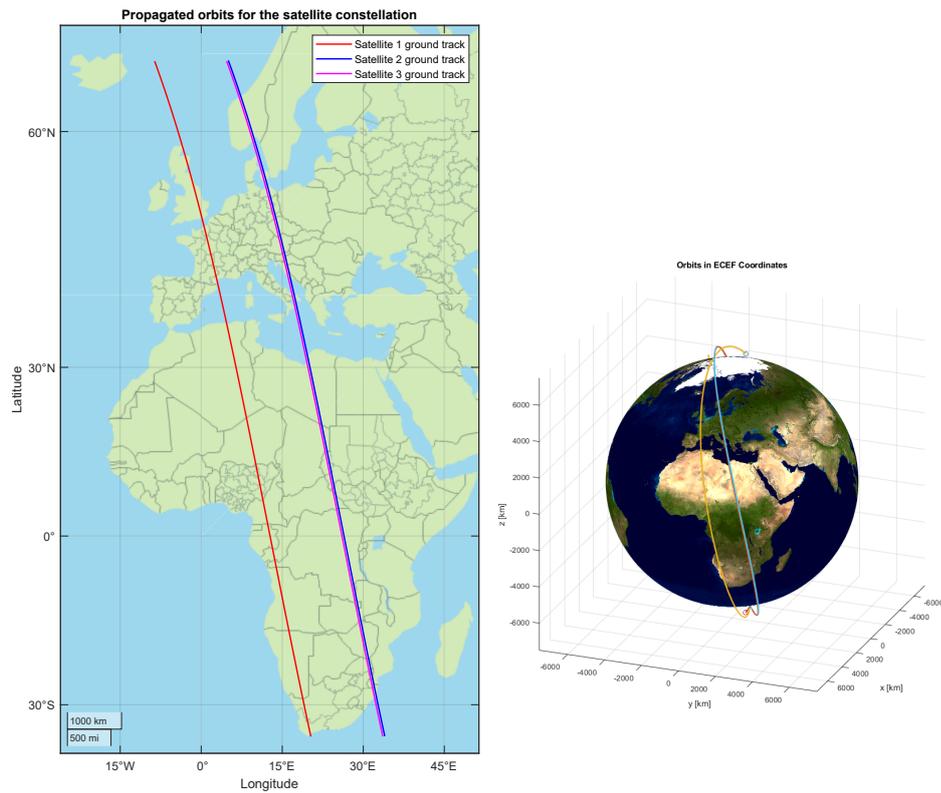
As detailed before, the AEOSs scheduling may be formulated in several manners. In order to convert the description to the QUBO formulation so the embedding onto the QPU is possible, some assumptions need to be taken into consideration. This thesis focuses on a case described as follows:

- A set of Agile satellites (AEOSs) is considered in a constellation formation.
- This constellation has a set of heliosynchronous, highly polar and circular orbits.
- A discretization of the Visible Time Window (VTW) is preferred.
- Down-link opportunity constraints are not implemented as it is assumed that the satellites have necessary range and time to transmit the collected data at every orbital revolution.
- Data storage and energy constraints are not explicitly taken into consideration. It is assumed that batteries are charged to a sufficient constant amount and storage is cleared at a down-link event at every opportunity.
- A soft constraint is implemented for sustainable fuel usage.
- Cloud coverage could be a constraint depending on the sensor type used, but not explored by this work.
- The scheduling horizon is set to a single orbital revolution so a target may be observed by a S/C at most one time.

As a general notion on how the mission is perceived, figure 1.3 shows the propagated one orbital revolution for the S/Cs, with a focus on Europe and Africa on the left. Further details will be given once the mission scenarios and the constellation orbital elements are presented, in Chapter 4.

1.3 Thesis outline

This work is divided as follows. Chapter 2 presents a mathematical description for the Earth observation problem, which is then translated to a Quadratic Unconstrained Binary Optimization (QUBO) model. It includes a motivation behind this choice and a pertinent description about the adopted variables and constraints. Chapter 3 focuses on distinguishing the many meta-heuristic algorithms used to solve the problem, fundamentally exposing their theoretical foundations and the differences between classical and quantum approaches.



(a) Projection over interest area. (b) Constellation trajectory over the globe.

Figure 1.3: Propagated orbits for the satellite constellation.

Chapter 4 explains how the mission-specific scenarios are designed for each type of solver, alongside a pertinent theoretical explanation regarding orbital mechanics. Moreover, the mission outputs for the classical and quantum approaches are shown in the fifth Chapter and, finally, a discussion about the obtained results, their relevancy and suggestions for potential future works related to AEOSs scheduling are presented in Chapter 6.

Chapter 2

Problem definition

The scheduling of Agile Earth Observation Satellites (AEOSs) can be characterized in several manners. Typically, the problem formulations are classified into continuous or discrete models and, depending on the specific constraints and requirements, these problems may exhibit nonlinearities in addition to their inherent combinatorial nature.

An extensive bibliography review is performed by Wang [2], where a Mixed-Integer Nonlinear Programming (MINLP) modelling technique addresses both continuous and discrete problems. While still dealing with a satellite constellation, this study does not consider restrictions for data up or downlink, as it is assumed the transmissions may occur in parallel and at any convenient time window [6]. Moreover, it omits the storage limit and energy constraints to adopt an Integer Programming Model with a discrete time approach, appropriate for a straightforward conversion into the Quadratic Unconstrained Binary Optimization (QUBO) formulation, yet to be presented in subsection 2.2.

In summary, a list of Latitude and Longitude pairs corresponding to point targets locations is submitted, each of these has an associated duration d_i , which is related to their geographical extension, and a profit w_i , indicating the priority of acquisition. Together with the constellation information along a scheduling horizon of one orbital revolution, the data is preprocessed and mission nodes $OM_{ijk} = [tstart_{ijk}, tend_{ijk}, w_{ijk}]$ are found as inputs of the optimization problem.

2.1 The integer programming model

To address the satellite scheduling, the integer model 2.1 is constructed as follows: Consider $i \in T$ the candidate point targets for observation, $j \in S$, a set of all satellites to be scheduled, and finally, let $k \in VTW_{ij}$ be the discretized and finite number of possible imaging attempts for a target i and a satellite j . More

specifically, a Visible Time Window (*VTW*) is built with slots at every $\Delta T = 10\text{s}$, as proposed by [6].

$$\min_{\xi} \sum_{i \in T} \sum_{j \in S} \sum_{k \in VTW_{ij}} \left(-w_{ijk} \cdot \xi_{ijk} + \lambda_{m_{ijk}} \cdot \xi_{ijk} \right) \quad (\text{I})$$

$$\text{s.t.} \quad \sum_{j \in S} \sum_{k \in VTW_{ij}} \xi_{ijk} \leq 1 \quad \forall i \in T \quad (\text{II}) \quad (2.1)$$

$$\xi_{ijk} + \xi_{i'jk} \leq 1 \quad \forall (i, i') \in F, i \neq i' \quad (\text{III})$$

$$\xi_{ijk} \in \{0, 1\} \quad \forall i \in T, j \in S, k \in VTW_{ij} \quad (\text{IV})$$

in which I is the objective function considering all satellites and targets. The total profit should be maximized, so the problem is rewritten in its negative form.

A soft constraint $\lambda_{m_{ijk}} \cdot \xi_{ijk}$ penalizes acquisitions in which the overall attitude adjustment needed for a maneuver is deemed to result in a high fuel consumption. This is achieved by comparing the total roll angle change between a pair of candidate acquisitions: if $\Delta_{roll} > 30^\circ$, $\lambda_{m_{ijk}}$ assumes a positive value, raising the cost function.

The *unicity* constraint II guarantees that a target is, at most, acquired one time, while the *timing* constraint III does not allow two consecutive unfeasible observations belonging to the forbidden area F due to time restrictions (acquisition and maneuver). Lastly, constraint IV imposes ξ as a binary decision variable.

2.1.1 Region F

The forbidden maneuver region F marks the unfeasible combinations of *VTW* slots from the i -th and the i' -th targets. Particularly, the pair of targets (i, i') is said to be inside the aforementioned region when there are timing violations. The start time of the next acquisition $ts_{i'jk}$ shall not be less than the start time of the last valid observation plus the duration of acquisition, attitude maneuver and stabilization, as mathematically described by 2.2.

$$F = \{(i, i') \in VTW_{ij} \times VTW_{i'j} : ts_{i'jk} < ts_{ijk} + d_i + \Delta M_{jk}(i, i')\} \quad (2.2)$$

Still regarding F , and proportional to the geographical size of the target, d_i corresponds to the imaging duration, period when the sensor is active and the

S/C's attitude is fixed. More specifically, the larger the target, the longer it will take for the sensor to capture it.

Finally, $\Delta M_{jk}(i, i')$ accounts for both stabilization and attitude maneuver. The first quantity is adopted as a constant, while the last is calculated based on the difference between the current and next pitch and roll angles, as well as the slew rate in which the S/C is able to rotate to its new configuration. More specifically, in the literature, two approaches are often adopted [7, 8, 9] and [10, 11]. Liu and Peng prefer the piecewise and more detailed, but still continuous function 2.3, with constants a_i and v_i that change according to the satellite model and maneuverability techniques.

$$\Delta M_{ij}(i, i') = \begin{cases} a_0, & \Delta g \leq 10^\circ \\ a_1 + \Delta g/v_1, & 10^\circ < \Delta g \leq 30^\circ \\ a_2 + \Delta g/v_2, & 30^\circ < \Delta g \leq 60^\circ \\ a_3 + \Delta g/v_3, & 60^\circ < \Delta g \leq 90^\circ \\ a_4 + \Delta g/v_4, & \Delta g > 90^\circ \end{cases} \quad (2.3)$$

$$\text{with } \Delta g = |\text{roll}_{ijk} - \text{roll}_{i'jk}| + |\text{pitch}_{ijk} - \text{pitch}_{i'jk}|$$

While this approach can accommodate varying fuel consumption based on the total required angle change, it significantly increases the computational workload during dataset preprocessing. For simplicity and speed, this study follows the second approach, with a streamlined equation, shown by 2.4. The constant component accounts for the stabilization time, while the second refers to the attitude change rate.

$$\Delta M_{ij}(i, i') = a + \Delta g/v \quad (2.4)$$

$$\text{with } \Delta g = |\text{roll}_{ijk} - \text{roll}_{i'jk}| + |\text{pitch}_{ijk} - \text{pitch}_{i'jk}|$$

2.1.2 Quality of acquisition

In the context of agile satellites, image quality is related to the distortion caused by highly angular perspectives. Some approaches may consider actively imposing constraints for threshold image quality [7], while others change the method when dealing with area or strip targets [12]. This study, as presented by [6], takes into consideration the roll and pitch at every *VTW* slot and, from the original value \bar{w}_i provided by the mission client, a *true profit* w_{ijk} is calculated. This process is regulated by equation 2.5.

$$w_{ijk} = \bar{w}_i \cdot \cos(roll_{ijk}) \cdot \cos(pitch_{ijk}) \quad (2.5)$$

The chosen method significantly increases the level of complexity of the problem, as every slot $k \in VTW$ will assume a different associated profit value, however the conceived scenario is considerably more realistic.

Figure 2.1 shows the trend for both angles. For a typical observation, there will be a full sweep for the pitch, ranging from the minimum to the maximum angle limits for the S/C and, as expected, the image quality degrades as these angles go away from the origin. Ideally, for maximum profit, a target should be aligned with the S/C's nadir line. Note that the yaw angle is not relevant for this evaluation, since it does not significantly affects image quality [6, 9].

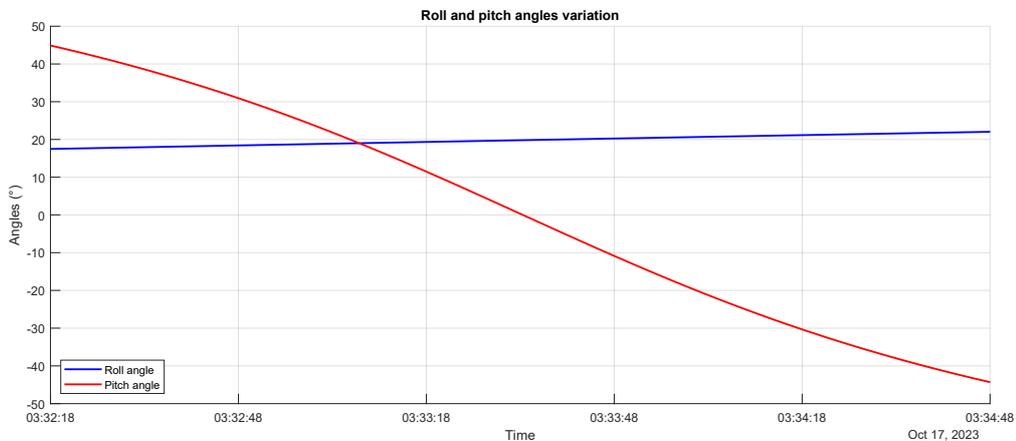


Figure 2.1: Change of roll and pitch angles during a VTW

2.2 Quadratic Unconstrained Binary Optimization (QUBO)

In order to solve the AEOSs scheduling problem using any of the proposed classical techniques or embedding it onto the D-Wave platform, it is necessary to convert the current representation into a QUBO matrix format. Also, depending on the algorithm to be implemented, this representation may further be mapped into an Ising model.

Glover [13] reviews some distinct classes for combinatorial optimization. While some may be naturally mapped into QUBO, such as the Number Partitioning and the Cut-Max problems, the AEOSs case 2.1 shall have its constraints manually converted into quadratic penalties and its cost function represented by the linear term.

2.3 QUBO model for single satellite scheduling

While this study aims to develop a QUBO representation for multiple AEOSs, arranged in a constellation formation, it is essential to firstly provide a description for a single satellite mission. From section 2.1, it is known that there are two constraints to be translated: one for *unicity* and another for *timing*.

Initially, the *unicity* constraint can be translated into penalties by addressing each request individually. The adopted principle is to forbid the presence of product combinations $\xi_{ijk} \cdot \xi_{ijk'}$ with $k \neq k'$. Indeed, if such term is accepted in the final solution, a satellite j would be allowed to capture target i at different instants k and k' , breaking the restriction of uniqueness.

This assignment can be achieved by forcing all upper triangular entries, except the main diagonal of $\mathbb{Q}_u^{(i)}$, to receive a penalty term λ . The size k of each submatrix is the number of slots of their corresponding *VTW*, shown by 2.9. If a request has only one opportunity to be acquired, there will not be a constraint requiring its uniqueness.

$$\mathbb{Q}_u^{(i)} = \begin{bmatrix} 0 & \lambda & \lambda & \dots & \lambda \\ 0 & 0 & \lambda & \dots & \lambda \\ 0 & 0 & 0 & \dots & \lambda \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{k \times k} \quad (2.9)$$

Consequently, for a single satellite, a possible representation for the *unicity* restriction is shown by 2.10. The mapped submatrices for each target are positioned in order, on the main diagonal of \mathbb{Q}_u . Evidently, once all the requests $i \in T$ are computed, the total size of this subproblem n^* will be the sum of all *VTW* for a S/C scheduling horizon.

$$\mathbb{Q}_u = \begin{bmatrix} \mathbb{Q}_u^{(1)} & & & & \\ & \mathbb{Q}_u^{(2)} & & & \\ & & \mathbb{Q}_u^{(3)} & & \\ & & & \ddots & \\ & & & & \mathbb{Q}_u^{(i)} \end{bmatrix}_{n^* \times n^*} \quad (2.10)$$

In contrast, the *timing* requirement is dealt by building the unfeasible domain region F . A case study with three targets and 15 variables is used in this section as an example on how the mapping occurs. After adequate generalization, this method may be applied to the real sized problem.

A structured *for* cycle is build to sequentially address all targets. Within this cycle, an inner loop iterates through each time slot k for current and future requests $i' > i$. In summary, for every candidate maneuver, two conditions must be satisfied:

1. The sensor has sufficient time to start and finish the acquisition respecting the demanded duration d_i ;
2. There is enough time for maneuver and stabilization of the S/C within the considered pair, as described by subsection 2.1.1.

A scheme on how this analysis works is presented by diagram 2.2. In this case, for each imaging attempt in 1, all possible maneuver combinations with target 2 slots are formed and have the aforementioned timing requirements checked. Systematically, the algorithm iterates to target 3 and continues until all non redundant start and end points are examined.

The highlighted slots represent forbidden maneuvers. For instance, from ξ_{114} , all combinations considering this slot as starting point are not allowed, since it does not satisfy the duration requirement.

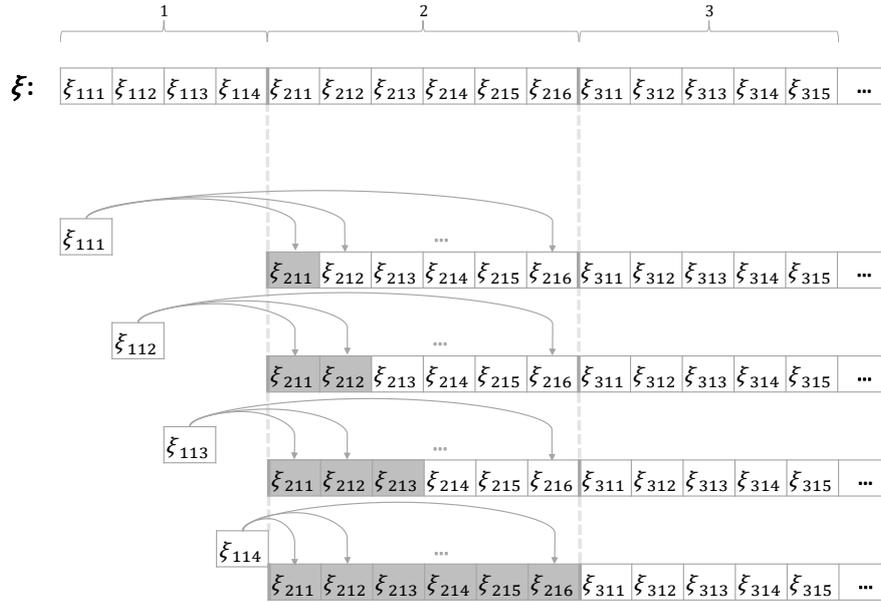


Figure 2.2: Diagram of possible maneuvers from first to second target

Although the presented algorithm outlines the approach, these forbidden pairs must be translated into penalties in matrix form. This mapping yields the upper triangular block matrix for the *timing* constraint \mathbb{Q}_t , shown in 2.11, whereas

2.4 QUBO model for constellation scheduling

A QUBO model for multiple AEOSs is accomplished by integrating individual S/C formulations with the additional and necessary *unicity* constraints between satellites. Indeed, for a constellation, the *timing* requirements are not considered, as they have already been accounted for in the case of a single spacecraft.

From section 2.3, the known \mathbb{Q}_j 's are directly mapped onto the main diagonal of the final \mathbb{Q} , as presented by 2.13, and a new procedure is applied to build the block submatrices λ_{ij} .

As the algorithm iterates through unique pairs of satellites, a list with every recurring target is constructed. Accordingly, all the *VTW* slots that correspond to those repetitions will be filled with penalties λ , forming the structure of a submatrix λ_{ij} . This routine is repeated until all combinations are covered. This is only possible because each target carries a unique ID, allowing for a global view of the problem.

$$\mathbb{Q} = \begin{bmatrix} \begin{bmatrix} \mathbb{Q}_1 \end{bmatrix} & \lambda_{12} & \dots & \lambda_{1j} \\ & \begin{bmatrix} \mathbb{Q}_2 \\ \dots \\ \mathbb{Q}_j \end{bmatrix} & & \lambda_{2j} \\ & & \dots & \vdots \\ & & & \begin{bmatrix} \mathbb{Q}_j \end{bmatrix} \end{bmatrix}_{n \times n} \quad (2.13)$$

The QUBO formulation for a constellation is then finalized, and the resulting upper triangular matrix \mathbb{Q} will serve as input for the proposed algorithms, yet to be described in Chapter 3.

Chapter 3

Overview of meta-heuristic algorithms

The choice of a solving method is directly connected to the structure of the problem it is meant to solve. Moreover, this selection is also based on the number and type of decision variables and optimality requirements.

In general, in the literature [10, 14, 15], the AEOSs scheduling may be addressed using various approaches. Dynamic programming, branch-and-bound and divide and conquer frameworks constitute a major class of exact algorithms and, even if they provide global optimality, their performance is hindered by extensive time and memory consumption, especially in real-world applications involving large-scale scenarios. Indeed, as the size of the NP-Hard instances to be solved increases, an exact algorithm would require a computation time and resources that grow exponentially. Also, their implementation is heavily problem dependent, which may require impractical reformulations of the solver for every specific case.

Furthermore, as the number of variables increases for real-world scale implementations, heuristic approaches, such as greedy algorithms prove useful considering their ability to speed up the process, even if the quality of the solutions and their optimality are not guaranteed. Another commonly applied method, particularly in recent times, involves algorithms based on machine learning. Through quantum-enhanced reinforcement learning, where the classical approach is assisted by the quantum annealer, Rainjonneau [2] performs a training phase for a multilayer perceptron (MLP) in order to address the mission planning problem.

Due to the generic description adopted, with a QUBO formulation, and following the intentions on exploring quantum possibilities, this study aims at two meta-heuristic Quantum and Hybrid Quantum algorithms, comparing their performance to the Tabu Search and Simulated Annealing, as two classical counterparts.

3.1 Classical algorithms

The meta-heuristic algorithms chosen to serve as the classical counterparts are the Tabu Search (TS) [16] and the Simulated Annealing (SA) [17], both implemented by D-Wave in Python. These have been vastly explored by the literature as they can accept generic and combinatorial formulations, like the QUBO, and explore the search space with relatively high speed. Figure 3.1 shows the exploration of a generic one dimensional search space performed by a meta-heuristic algorithm.

Opting for the D-Wave implementations is more advantageous, given their provision of well-documented libraries and the flexibility to modify key parameters such as the number of desired runs and the capacity to retrieve the total run-time of the algorithm, a crucial comparison parameter.

Alternative classical algorithms incorporated in MATLAB, such as the Genetic Algorithm, were also explored and tested, serving as viable initial solutions, but the obtained results were not competitive. Additionally, these alternatives lacked the flexibility to adjust many parameters, so they are not displayed here.

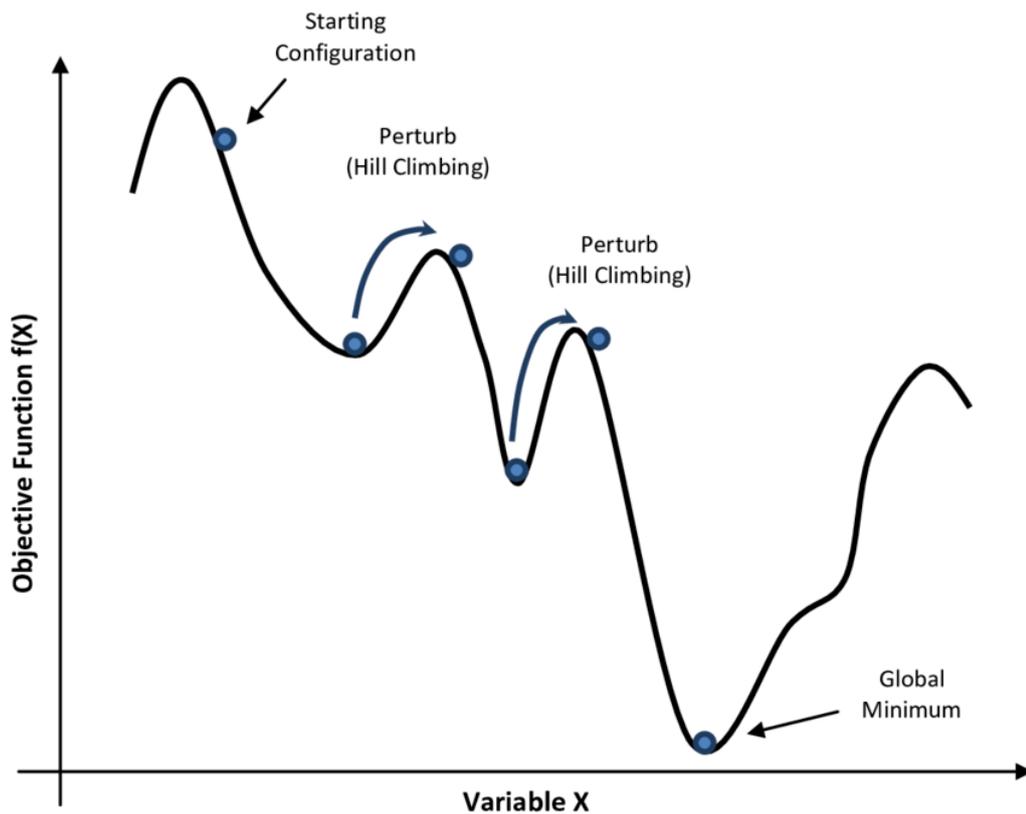


Figure 3.1: A one dimensional example for a generic meta-heuristic approach [18]

3.1.1 Tabu Search

The TS algorithm, which uses the formulation specifically proposed by Palubeckis [19] for QUBO problems, is based on building a short-term memory list, with size defined by the tabu tenure parameter, used to escape sub optimal minima if certain conditions are satisfied. Additionally, it incorporates a local search in order to find competitive close-to-optimal solutions.

A higher tenure encourages some diversification by preventing the search from revisiting the same solutions too quickly, promoting exploration of different regions of the solution space. For this work, the assumed tenure value is 20.

A pseudo-code for the proposed procedure is presented by algorithm 1. The Simple Tabu Search (STS) iterates over the search space until a better candidate optimization variable is found, its neighborhood is analyzed with the *localSearch* method so a local minima is identified. Particularly, the algorithm starts with the assignment of initial states for the counter z , the tabu value T_i , f' , V and ρ . If during the first loop some vector better than the current solution is found, then a *LOCALSEARCH()* method is called, returning a new best solution x , optimal regarding its neighborhood.

This process occurs until it returns the *bestSolution* at the end of the iterations and the stop conditions are met. Furthermore, this implementation is improved by adopting random multistart strategies, so more solutions are explored.

Algorithm 1 A generic Tabu Search implementation

procedure STS

$z \leftarrow 0$

$f' \leftarrow f_x$

$T_i \leftarrow 0$

$\triangleright T_i$: tabu value for x_i

$i \leftarrow 1, \dots, n$

$V \leftarrow -\infty$

$\rho \leftarrow 0$

for $k = 1, \dots, n$ **do**

if $T_k > 0$ **then**

goto line 8

end if

$z \leftarrow z + 1$

if $f' + c'_k > f^*$ **then**

$r \leftarrow k$

$\rho \leftarrow 1$

goto line 23

end if

if $c'_k > V$ **then**

$V \leftarrow c'_k$

```

         $r \leftarrow k$ 
    end if
end for
 $x_r \leftarrow 1 - x_r$ 
 $f' \leftarrow f' + c'_r$ 
UPDATEC()                                ▷ update  $c'_{ij}$  and  $c'_i$  using loop in line 8
if  $\rho = 0$  then
    goto line 30
end if
 $\mathbf{x} \leftarrow \text{LOCALSEARCH}()$           ▷ with  $\mathbf{x}$  being a possible improved solution
 $T_i \leftarrow T_i - 1$                     ▷ when  $T_i > 0$ 
 $T_r \leftarrow T$ 
if  $z < z_{max}$  then
    goto line 6
end if
Return bestSolution
end procedure

```

3.1.2 Simulated Annealing

Originally established in the field of metallurgic sciences, a second classical technique utilized is the SA. In this method, a material undergoes a slow and guided cooling schedule to alter its physical properties. This same notion is then applied to the optimization algorithm as, at each time step, a random solution in the vicinity is grabbed and its quality analyzed. Based on probabilities influenced by the current cooling conditions, this solution is kept or discarded until the temperature of the process reaches zero. In this context, the temperature is intricately linked to the exploration of the solution space — higher values entail more exploration but also elevate the chance of accepting bad solutions at each step. Algorithm 2 presents a generic implementation of the procedure.

More specifically, as the procedure starts, the temperature T , energy function S , cooling schedule $T(i)$ and a starting model are assigned as the initial configuration. While there is no convergence to an acceptable minima, a random vicinity model is assigned to the *currentModel* and its relative energy with respect to the *currentModel* is checked. If the variation is lower than zero, the *currentModel* becomes the *newModel* with a probability based on the current temperature and energy variation. This procedure continues until the temperature reaches zero based on its schedule.

Algorithm 2 A generic Simulated Annealing implementation

procedure SIMULATEDANNEALING

```

currentModel ← initialState
T ← INITIALTEMPERATURE()
T(i) ← COOLINGSCHEDULE()
S ← ENERGYFUNCTION()
while NOT CONVERGED do
    newModel ← random
     $\Delta S \leftarrow S(\text{NEWMODEL}) - S(\text{CURRENTMODEL})$ 
    if  $\Delta S < 0$  then
        currentModel ← newModel
    else
        currentModel ← newModel with probability  $P = e^{-\frac{\Delta S}{T}}$ 
    end if
    T ←  $\alpha T$ 
end while
Return bestSolution
end procedure

```

Moreover, as part of meta-heuristic algorithms, the provided results are approximated to the global minima, but the solution space is vastly explored in a short span of time, depending on imposed initial conditions. The D-Wave implementation utilizes a Boltzmann sampling, well suited to finding acceptable solutions for large scale instances. In terms of optimization algorithms, the SA is in fact a direct classical analog to the quantum annealing, which is one of the reasons it has been selected for this comparison.

3.2 Quantum algorithms

The Quantum Annealing (QA) and the Hybrid Quantum Annealing (HQA) algorithms used in this study and their practical implementations are presented in this section. Firstly, the theoretical fundamentals for the quantum approach are shown, alongside a brief explanation on how adiabatic quantum machines work. Accordingly, the hybrid method is proposed combining these properties with classical algorithms in order to run large scale problems.

3.2.1 Quantum Annealing

Quantum computing, following the brief introduction from Chapter 1, is a field that combines quantum mechanics with classical computer science. Instead of bits, that can take 0 or 1, this devices utilize qubits, which can assume both states simultaneously to perform parallel computations, as explained by the superposition principle, and correlate with each other through the phenomenon of entanglement,

so they are no longer independent [20]. For instance, if entangled states were to be measured, the probability of observing them with the same spin value would be correlated.

These principles are the core of any quantum device, however, prior to delving into more specifics, it is important to establish some significant distinctions. Quantum annealers, the device used for this work, differ from other machines such as gate model or universal quantum computers, as they are specifically posed to solve sampling or optimization problems described by their energy landscape and adiabatic considerations.

Annealers use these principles of quantum physics to find a minimum energy configuration at the end of an adiabatic state evolution. The problem setup is performed only at the beginning, through the Hamiltonian, while other quantum machines aim to interact with quantum states as the evolution occurs, allowing them to manipulate a bigger variety of problems.

This added layer of complexity and generality has implications for the scale of quantum machines. State-of-the-art universal computers feature only dozens of qubits, while the D-Wave annealer machine used in this work has over 5000 qubits, allowing for its use in real world problems [21].

Moreover, even if there are other general adiabatic devices that exhibit certain similarities to annealers, their physical structure and hardware constraints are significantly distinct, as clarified subsequently.

Theoretical background

Fundamentally, QA relies on the adiabatic theorem, which stipulates that a quantum system commencing from its ground state will persist in that state provided the changes in dynamics, given by the Hamiltonian, occur at a sufficiently slow pace over time, so that the minimum gap is not violated. A generic representation of this process can be seen in figure 3.2. At some point during the annealing, the energy difference between ground state and the first excited state is at its lowest. If the dynamics transition sufficient slowly, the minimum gap will not be violated and the system will remain at ground.

The energy representation through the Hamiltonian \mathcal{H} of the system is shown by equation 3.1. This expression has two terms accounting for an initial \mathcal{H}_i and final \mathcal{H}_f (target) configurations.

A smart choice to enforce the aforementioned adiabatic and slow transition would be to select a simple initial Hamiltonian representing the ground state and slowly shift the system to a more complex and realistic one, seen in equation 3.2, that represents the actual problem. Spin variable σ_i and flux bias h_i for qubit i , and the couplings J_{ij} between qubits account for the variables of the energy description.

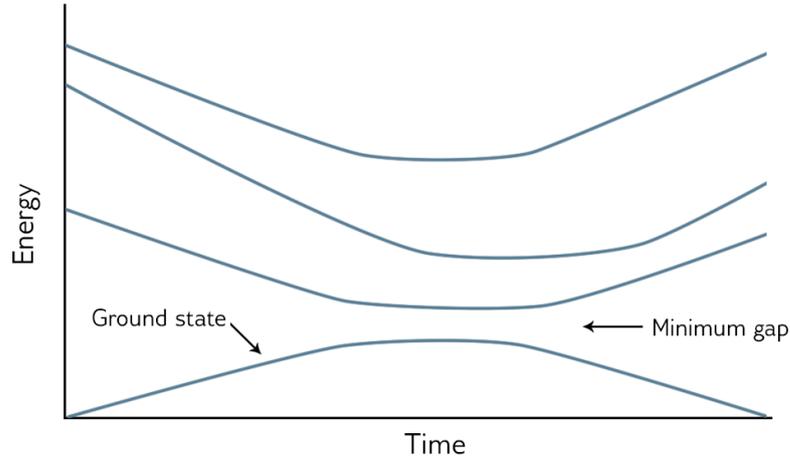


Figure 3.2: Eigenspectrum for a generic quantum system [22]

$$\mathcal{H}(t) = A(t)\mathcal{H}_i + B(t)\mathcal{H}_f \quad (3.1)$$

$$\mathcal{H}_f = \sum_i h_i \sigma_i + \sum_{(i,j)} J_{ij} \sigma_i \sigma_j \quad (3.2)$$

This change over time is commanded by the annealing time and annealing schedule, mapped to the monotonic functions A and B , seen in figure 3.3. As the magnitude of \mathcal{H}_i decreases, the quantum characteristics of the system also slows down, until they become zero after the annealing time is passed. This results in a purely classical system allowing the qubits to be measured.

Ensuring that this transition is adiabatic and the rate of change is slow enough could be, in theory, accomplished by tuning these input parameters so the system would output the lower energy state as the final solution, successfully solving the optimization problem. However, despite the strong implications of the adiabatic theorem, it is not possible to completely avoid thermal fluctuations and background noise in practice, making the slow rate of change requirement hard to fulfill.

In this way, QA can be seen as a relaxation of the purely adiabatic quantum computing, where the annealing schedule and other parameters are determined heuristically and the probability of leaving the ground state bypassing the minimum energy gap is non-zero. Indeed, it can be compared to the SA, in which the thermal fluctuations are replaced by quantum fluctuations.

Solving with D-Wave

From Chapter 2, the system's QUBO \mathbb{Q} matrix was built. This format represents a standard and preferred by the industry as it can describe a large number of

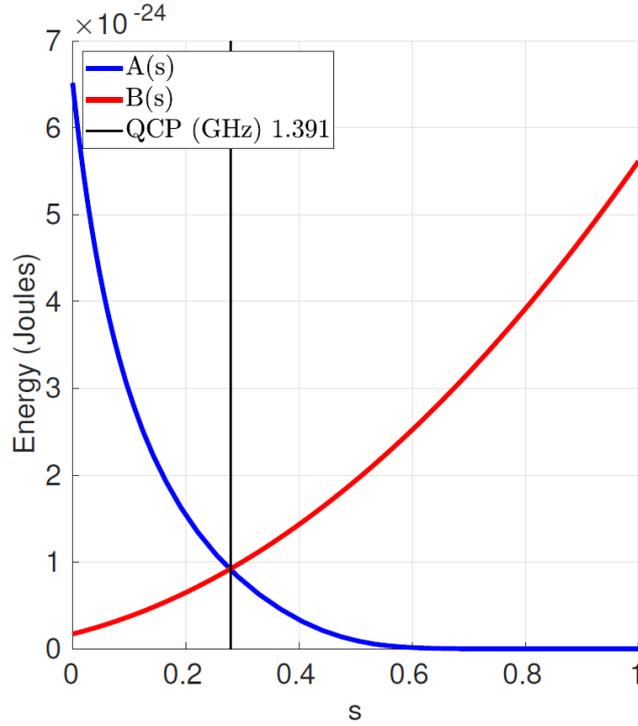


Figure 3.3: Monotonic annealing functions A and B [22]. The process begins at $s = 0$ and ends at $s = 1$

problems and has decent similarities to the Ising representation 3.2 presented before.

To submit it to the D-Wave solver, some steps must be performed. Generated in MATLAB, the \mathbb{Q} matrix for the instance is read and mapped to a dict structure in Python considering only non-null entries - this consideration is important otherwise zeros would also be forced during minor-embedding, taking away important processing capability and can be done since they do not carry information for this problem.

Alongside the prepared dict, that is internally mapped as $\sigma = 2\xi - 1$ and transformed to a Binary Quadratic Model (BQM: a generic larger class that encompasses Ising models), an embedding technique should be selected. This choice is based on the size of the problem and how its variables are connected.

For this study, the method *EmbeddingComposite* is called and automatically minor-embeds the problem into the specific structured sampler, mapping a set of physical qubits that represents a single logical variable to the quantum processing unit (QPU) and imposing the coupling strength between nodes. Then, the purely quantum *DWaveSampler* [23] class is responsible for solving the problem taking into

consideration heuristically chosen parameters such as the annealing time, annealing schedule, chain strength and number of runs. These parameters, however, are very hard to calibrate since not much information about the physical embedding is returned or known a priori. The Python script that implements all the described steps is shown in A.1.

The annealing process begins and the Ising model is solved. According to the set parameters, the system's dynamics transition from purely quantum to classical, and the expected low energy state can be measured. As there is a non-zero probability for the system to leave the ground state, the problem is resampled, based on number of runs parameter, and the most repeated output is selected as the final optimization solution.

3.2.2 Hybrid Quantum Annealing

The HQA is proposed as an alternative method to the purely quantum approach. Many of the difficulties found when trying to work directly with the QPU can be avoided. For instance, correctly choosing the weights for constraints violations (couplings) and profits (bias) while formulating the QUBO problem is not a simple task. If the selected values are too small or indistinguishable, the system will be translated with a small energy gap and surely jump to more energetic states during annealing. On the other hand, if these entries are too big, the search space would not be explored sufficiently.

These weights also influence the chain strength needed to keep coherent a set of physical qubits that represents a logical value. If the frequency of chain breaks is excessively high, a meaningless state will be returned as solution. If we manually pick a high chain strength, the original problem gets mischaracterized. Even with the auto-scale feature from the sampler, calibrating other parameters is still demanding. Moreover, other challenges appear when choosing the embedding technique or the impossibility of avoiding background noise and thermal fluctuations.

Accordingly, the hybrid algorithm offers a combination of heuristics and quantum annealing in order to solve problems that are too large or difficult to embed directly onto the QPU. Figure 3.4 shows a general scheme on how a problem is divided and processed through D-Wave technology. In simpler terms, the hybrid-classical approach can be seen as a splitting tool for real-world problems, where smaller instances are submitted to the QPU and their outputs combined for comprehensive solution to the original problem. Furthermore, scaling implications and back-end configurations for embedding are automatically performed, so no miscalibration of parameters occurs. A simple setup for the solver is required and performed in Python, shown in A.2.

A single parameter to setup, besides the matrix \mathbb{Q} , is a time limit for computation. Within this time limit, the front end (blue) reads the QUBO matrix and calls the

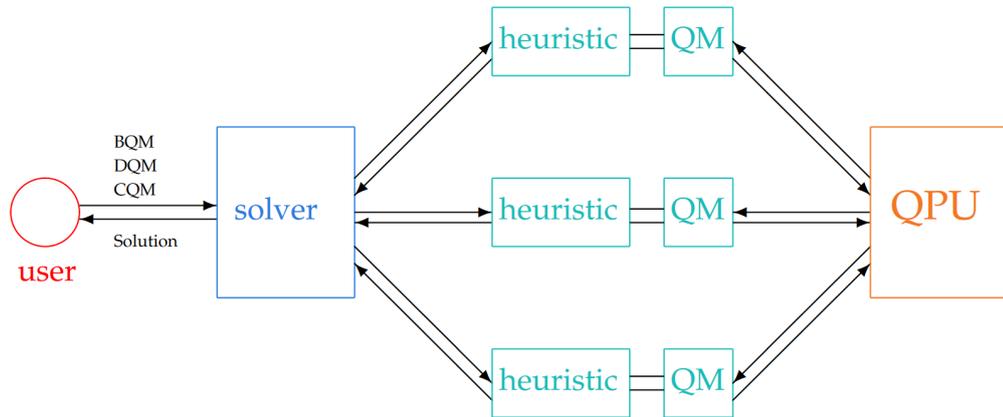


Figure 3.4: A schematic for the D-Wave implementation of the Hybrid Solver [24]

heuristic solvers (teal) that search, in parallel, for acceptable quality solutions. If needed, each of these solvers have a quantum module that maps and sends the problem to the D-Wave QPU (orange). The cycle is completed when the quantum units return queries that guide the heuristic search to new solutions or directly improve the overall global response [24]. A peculiar feature of this implementation is that, occasionally, when the sub-problems responses with the heuristic solvers are judged to have good quality, the QPU is not evoked and the end solution is purely classical.

Chapter 4

Design of mission scenarios

In order to test and validate the proposed algorithms advantages, some scenarios with different characteristics are designed. Dataset I accounts for all European capitals, while Dataset II is built upon a list of UNESCO Natural World Heritage sites.

These two different scenarios are able to explore the capabilities of both Hybrid quantum annealing and Quantum annealing, with different data sparsity and number of variables. However, a pre-treatment phase is judged necessary as this data must be prepared to serve as input for the QUBO problem formulation.

4.1 Data pre-processing procedure

Generally, in a nominal operation scenario, the client provides a list of desired targets, characterized by their latitude, longitude and altitude (LLA) coordinates, duration d_i and priority w_i . This data must be analyzed and subsequently integrated with the propagated orbits in order to generate the input files for the problem formulation.

An algorithm, that can be seen in appendix C, is developed with MATLAB so the Visible Time Windows of each S/C with respect to every target are generated. This procedure starts by firstly importing the target list and the satellite orbital elements (OE), and calculating its propagated orbit in particular frames of reference.

As illustrated in figure 4.1, a general setting for the translational motion in relation to the inertial frame with origin O takes into account the S/C as a rigid body and it is fundamentally based on the two-body problem, described with point masses, m_0 and m_1 , located at P_0 and P_1 , respectively.

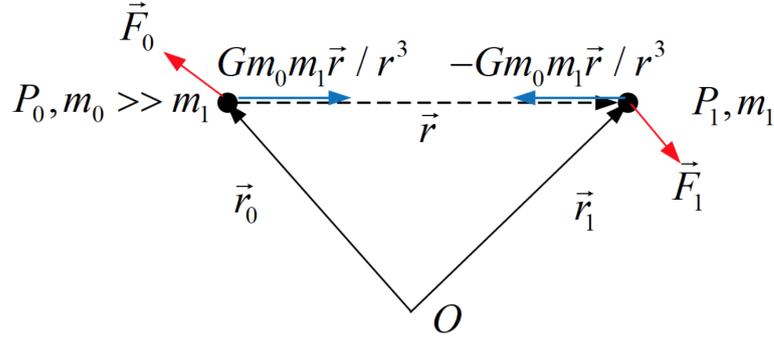


Figure 4.1: The two-body problem

A system of equations 4.1 can be build by mathematically describing this problem with Newton's Second and Gravitational Laws, in which \mathbf{r} is the relative position and $r = |\mathbf{r}|$. v_0 and v_1 are the velocities, and F_0 and F_1 the external non-graviational forces.

$$\begin{aligned}\dot{v}_0 &= \frac{Gm_1}{r^3}\mathbf{r} + \frac{1}{m_0}\mathbf{F}_0 \\ \dot{v}_1 &= -\frac{Gm_0}{r^3}\mathbf{r} + \frac{1}{m_1}\mathbf{F}_1\end{aligned}\quad (4.1)$$

If this system is rewritten in terms of its center of mass, and considering $m_0 \gg m_1$, the restricted two-body equation 4.2 is obtained for the Earth and S/C arrangement, with $\mu = Gm_0$ being the gravitational parameter.

$$\dot{v} + \mu \frac{\mathbf{r}}{r^3} = \frac{1}{m_1}\mathbf{F}_1 \quad (4.2)$$

For the orbital propagation, it is of particular interest the free motion of the satellite, with $\mathbf{F}_1 = 0$. Furthermore, as it is typically adopted by the literature, this satellite translation may also be derived from its six classical OE, presented by table 4.1 and thought the geometric description 4.3, in which $p = a(1 - e^2)$ is the semi-latus rectum, and r the position in polar coordinates.

$$r = \frac{p}{1 + e \cos \nu} \quad (4.3)$$

However, in order to proceed with the algorithm, a more detailed representation is needed. A first approach is to describe the S/C's trajectory in the Earth Centered Inertial (ECI) frame. This can be achieved with the rotation matrix $T_{313}(\Omega, i, \omega)$, as shown by equation 4.4.

Table 4.1: Orbital elements description

Element	Name	Description
a	Semi-major axis	Size
e	Eccentricity	Shape
i	Inclination	Tilt
Ω	Right ascension of the ascending node	Swivel
ω	Argument of perigee	Angle from ascending node to perigee
ν	True anomaly	Angle from perigee to the S/C's position

$$\mathbf{r}_{ECI} = T_{313}(\Omega, i, \omega) \cdot \begin{bmatrix} r \cos \nu \\ r \sin \nu \\ 0 \end{bmatrix} \quad (4.4)$$

On the other hand, when the rotational motion of the Earth must be accounted for, an additional transformation to an Earth-centered Earth-fixed (ECEF) frame is used. This procedure is facilitated by the Aerospace toolbox [25], in MATLAB, where conversion methods are already implemented.

Accordingly, for one orbital evolution, a discretization time of one second is chosen so the generated data is accurate enough. A similar derivation in different frames of reference is also performed for the velocities, so useful resulting data structures are found:

1. S/C's position and velocity in ECI.
2. S/C's position in ECEF.
3. S/C's position in LLA.
4. S/C's ground track position in ECI.
5. Target's position in ECI.

The pre-treatment phase continues with an iteration loop. Within every time slot of the orbital revolution, the function *insideVTW* is called with the S/C position and velocity in ECI, its maximum pitch and roll angles (set at 45°), its ground track and, finally, the target's position. Subsequently, a line of sight vector pointing from the S/C to the target is computed and its angle from the nadir line is examined. If the resulting angle falls within the physical limitations imposed by

pitch and roll, the current time slot is identified as a valid observation opportunity in the target's VTW. As this procedure is done for all targets and satellites, text files with useful information are generated with a unique mission identifier, ending the pre-processing phase.

4.2 The satellite constellation

The three S/Cs employed in this study are presented in table 4.2, their parameters are based on [6], the well-documented SuperView-1 constellation consisting of Chinese commercial and remote sensing satellites, launched since 2018. Their agile nature is accounted for allowing maneuvers in the roll and pitch directions, with a maximum range of $\pm 45^\circ$, for both angles.

Table 4.2: OE for the satellite constellation

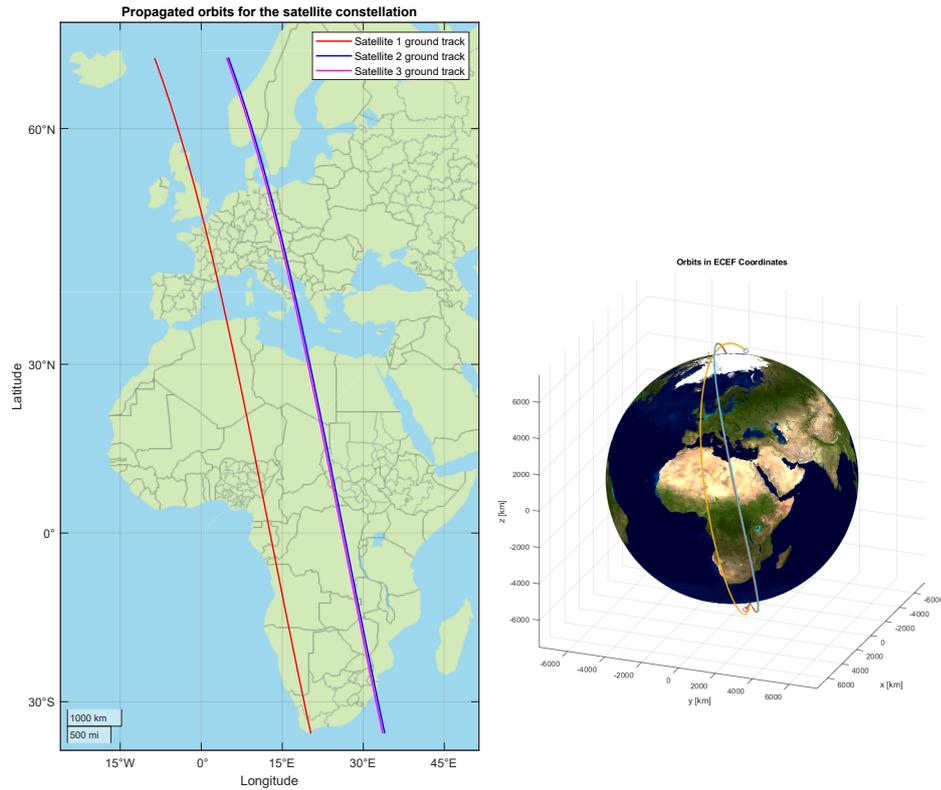
ID	a (km)	i ($^\circ$)	Ω ($^\circ$)	e	ω ($^\circ$)	ν ($^\circ$)
1000	6903.673	97.5839	97.8446	0.0016546	130.9890	2.0288
2000	6909.065	97.5840	93.1999	0.0009966	254.4613	155.2256
3000	6898.602	97.5825	92.3563	0.0014595	276.7332	140.1878

The first column shows the unique ID of the S/Cs, while the OE are presented from columns 2 to 7, respectively: semi-major axis, inclination, right ascension of the ascending node, eccentricity, argument of perigee and mean anomaly. From the original specifications, the OE for satellite 1, with ID 1000, is slight modified to fit the generated data requirements for sparsity.

These OE accounts for a heliosynchronous, highly polar and almost circular orbits. One orbital revolution T for the constellation is around 95 minutes, and can be calculated by the equation 4.5.

$$T = 2\pi \sqrt{\frac{a^3}{\mu}} \tag{4.5}$$

Their propagated orbits are shown by figure 4.2. A special focus is given to a local projection over Africa and Europe, as all targets are allocated in these continents.



(a) Projection over interest area. (b) Constellation trajectory over the globe.

Figure 4.2: Propagated orbits for the satellite constellation.

4.3 Dataset I: European capitals

The initial dataset built encompasses all European capitals listed in table 4.3. The chosen profit w_i , from 1 to 5, and duration d_i are arbitrarily selected, in an attempt to be as generic as possible, but yet preserve the realistic nature of the problem.

Table 4.3: High density dataset: European capitals

ID	Name	Latitude (°)	Longitude (°)	Profit	Duration (s)
101	Tirana	41.3317	19.8172	1	8
102	Andorra la Vella	42.5075	1.52180	4	9
103	Vienna	48.2092	16.3728	3	19
104	Minsk	53.9678	27.5766	4	13
105	Brussels	50.8371	4.3670	2	7
106	Sarajevo	43.8608	18.4214	3	15
107	Sofia	42.7105	23.3238	3	14

108	Zagreb	45.8150	15.9785	4	15
109	Prague	50.0878	14.4205	5	13
110	Copenhagen	55.6763	12.5681	3	4
111	Tallinn	59.4389	24.7545	4	4
112	Helsinki	60.1699	24.9384	5	3
113	Paris	48.8567	2.3510	5	9
114	Berlin	52.5235	13.4115	4	14
115	Athens	37.9792	23.7166	3	13
116	Budapest	47.4984	19.0408	3	11
117	Reykjavik	64.1353	-21.8952	4	10
118	Dublin	53.3441	-6.2675	5	13
119	Rome	41.8955	12.4823	1	15
120	Pristina	42.6740	21.1788	2	2
121	Riga	56.9465	24.1049	1	7
122	Vaduz	47.1411	9.5215	3	12
123	Vilnius	54.6896	25.2799	1	20
124	Luxembourg	49.6100	6.1296	1	22
125	Valletta	35.9042	14.5189	2	15
126	Chisinau	47.0167	28.8497	2	17
127	Monaco	43.7325	7.4189	3	15
128	Podgorica	42.4602	19.2595	1	4
129	Amsterdam	52.3738	4.8910	5	7
130	Skopje	42.0024	21.4361	1	8
131	Oslo	59.9138	10.7387	4	11
132	Warsaw	52.2297	21.0122	4	17
133	Lisbon	38.7072	-9.1355	3	12
134	Bucharest	44.4479	26.0979	4	15
135	Moscow	55.7558	37.6176	5	4
136	San Marino	43.9424	12.4578	1	16
137	Belgrade	44.8048	20.4781	3	15
138	Bratislava	48.2116	17.1547	2	11
139	Ljubljana	46.0514	14.5060	2	10
140	Madrid	40.4167	-3.7033	5	2
141	Stockholm	59.3328	18.0645	5	14
142	Bern	46.9480	7.4481	4	11
143	Kiev	50.4422	30.5367	3	18
144	London	51.5002	-0.1262	5	10

For satellites 1, 2 and 3, the complete VTW can be seen in figures 4.3, 4.4 and 4.5, respectively. At every ten seconds, a vertical dash indicates where the discretization occurred. In practice, the down-sampled data will be used for the

problem formulation, considerably reducing the number of variables, but still maintaining the system's physical characteristics and real world applicability, as adopted by [6]. Some authors prefer a finer resolution [9], while others [7] opt for an even greater time step.

This collection is considered a high density dataset, given the close vicinity and number of targets. Indeed, from these plots, it can be noticed that 26 targets may be observed by satellite 2 in less than 10 minutes, with many overlapping VTWs. Consequently, this populates the generated \mathbb{Q} and results in a harder and slower problem to be solved.

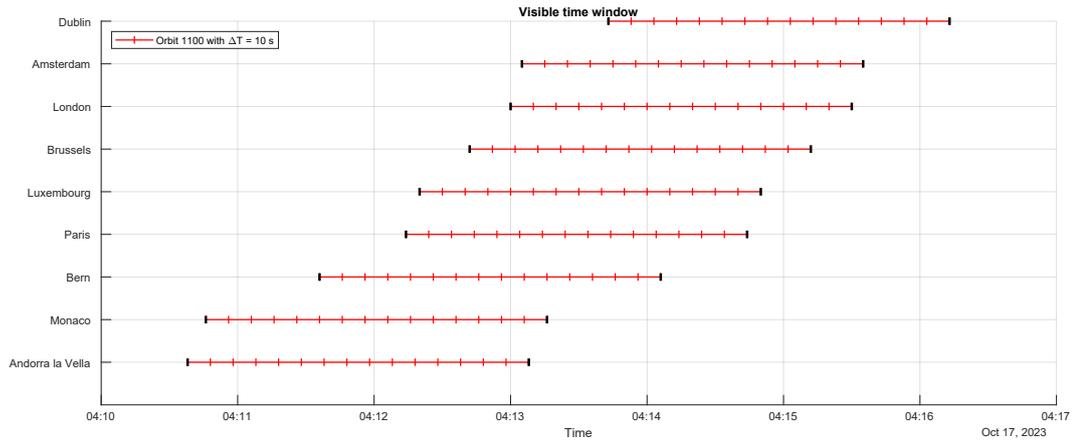


Figure 4.3: European capitals: visible time window for satellite 1

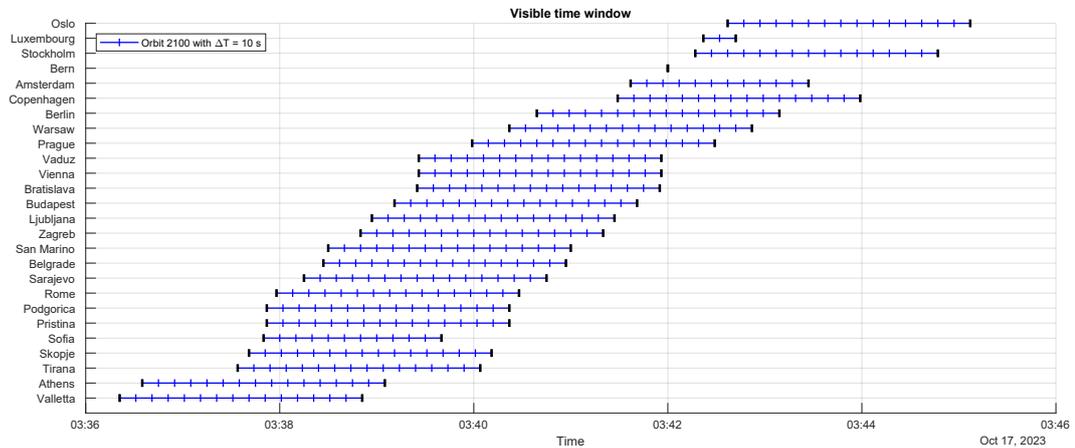


Figure 4.4: European capitals: visible time window for satellite 2

A local projection over Europe is shown in figure 4.6. Alongside the ground tracks, two auxiliary roll lines, computed during the pre-treatment phase, are

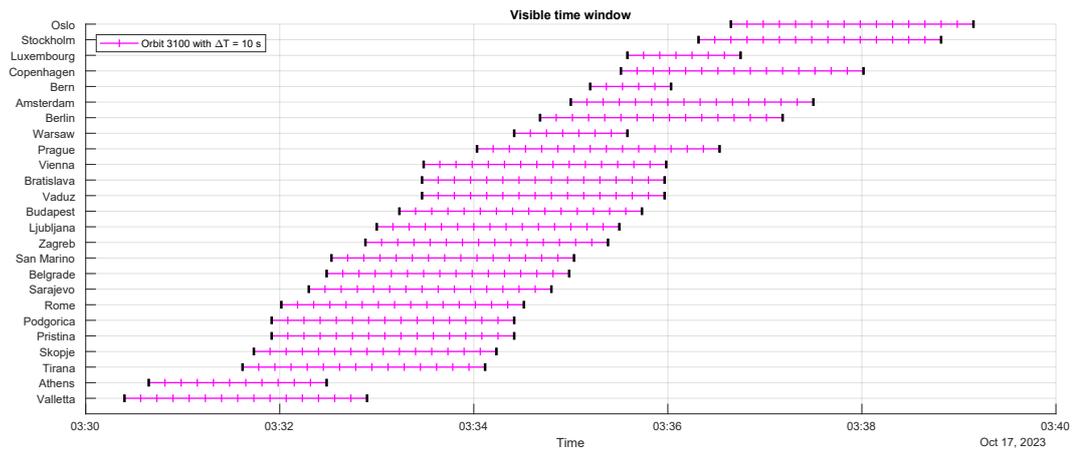


Figure 4.5: European capitals: visible time window for satellite 3

plotted. These are a good visual indication to each satellite’s line of sight reach.

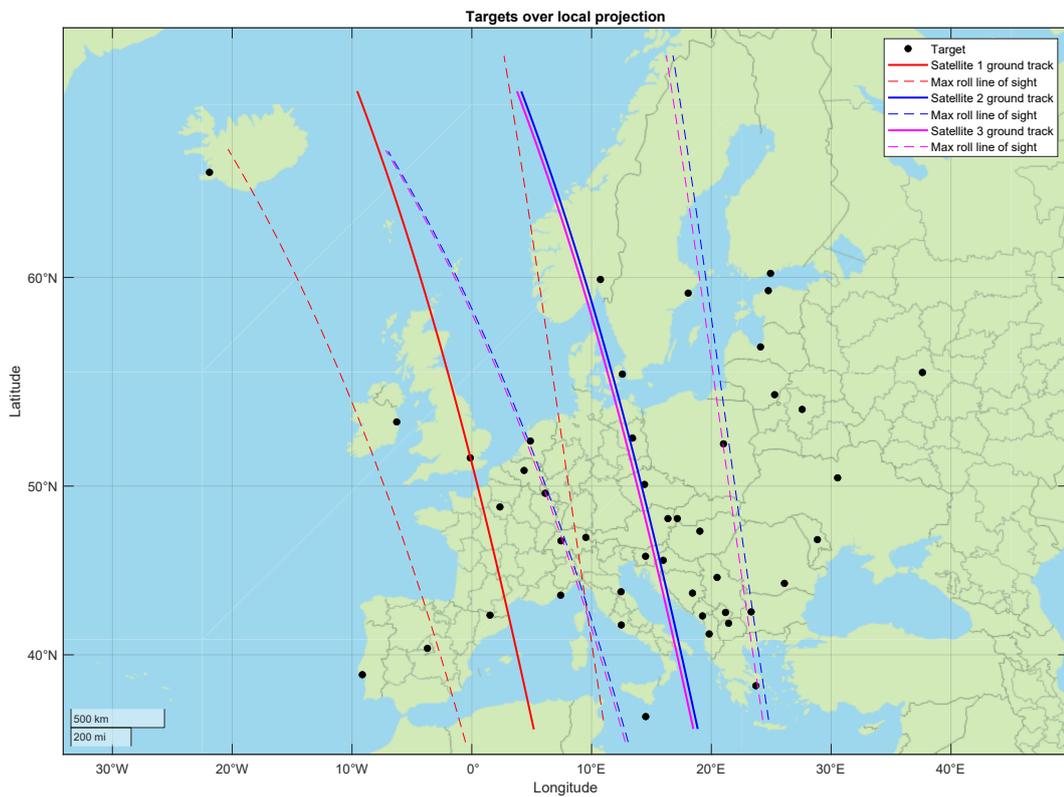


Figure 4.6: Target disposition and constellation ground tracks over local projection for dataset I.

4.4 Dataset II: UNESCO Natural World Heritage sites

Dataset II is designed with UNESCO Natural World Heritage sites, shown by table 4.4, and utilizes a formation with satellites 1 and 2. Endangered locations receive maximum observation priority w_i , while the duration d_i is proportional to their respective area.

Table 4.4: Quantum Annealer dataset: UNESCO Natural World Heritage sites

ID	Name	Lat. (°)	Long. (°)	Profit	Duration (s)
314	Ichkeul Nat. Park	37.1636	9.67472	1	1
317	Lakes of Ounianga	19.0550	20.5056	1	1
318	Air and Ténéré Reserves	18.0000	9.0000	5	25
320	Manovo-Gounda Park	9.0000	21.5000	5	10
323	Dja Faunal Reserve	3.0000	13.0000	2	3
324	Sangha Trinational	2.6094	16.5542	2	4
325	Okapi Wildlife Reserve	2.0000	28.5000	5	7
330	Namib Sand Sea	-24.8853	15.4078	4	16
331	iSimangaliso Wetland	-27.8389	32.5500	3	2
332	Cape Floral	-34.3611	18.4750	2	6

All VTWs for satellites 1 and 2 calculated in the pre-processing phase can be seen in figures 4.7 and 4.8, correspondingly. This data is prepared to have a lower number of variables, so it can be directly embedded onto the D-Wave Quantum annealer.

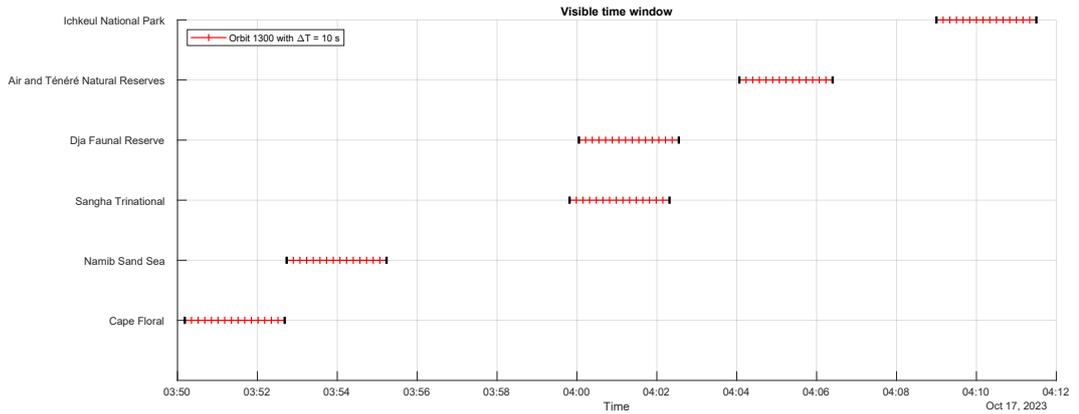


Figure 4.7: UNESCO sites: visible time window for satellite 1

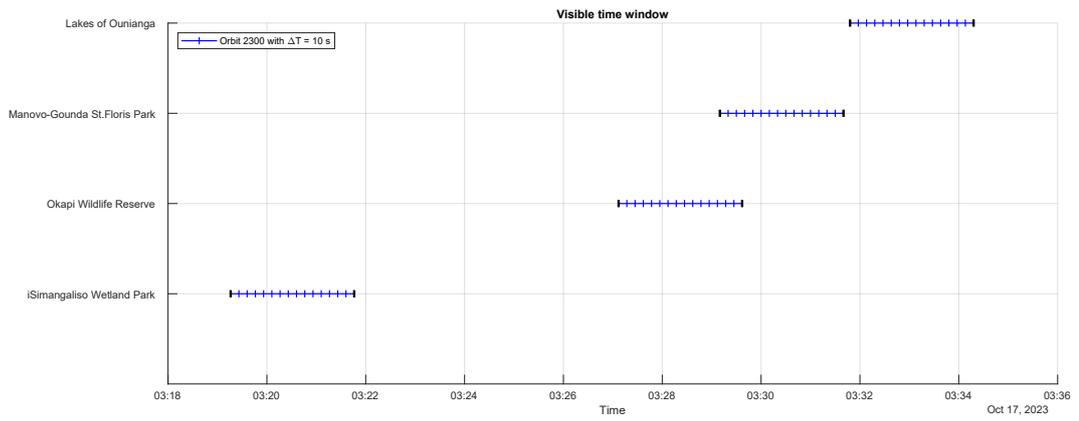


Figure 4.8: UNESCO sites: visible time window for satellite 2

Finally, similarly to the previous dataset, the satellite ground tracks and their associated roll lines are plotted, as seen in figure 4.9.

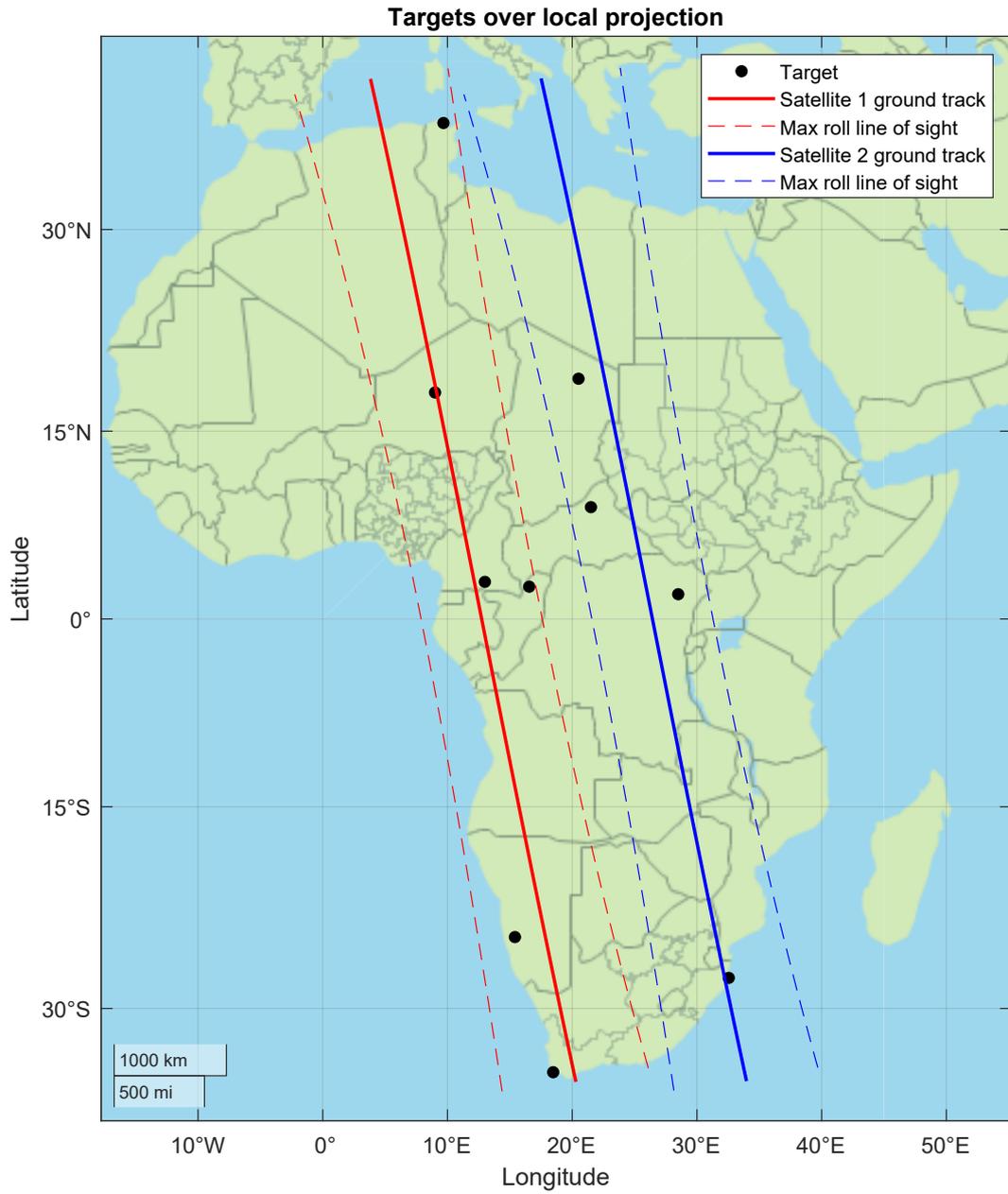


Figure 4.9: Target disposition and constellation ground tracks over local projection for dataset II.

Chapter 5

Results

The scheduling problem related to AEOSs constellation has been addressed and resolved using the methodologies outlined previously, and the results are presented in this chapter. The selected algorithms for the comparative analysis are picked to match the characteristics of each dataset and follows the division:

1. For dataset I, with a large number of decision variables, the optimization problem is addressed with the Tabu Search, the Simulated Annealing and the Hybrid Quantum Annealing.
2. Dataset 2 specifically delves into the implementation of purely Quantum Annealing, in comparison with the Tabu Search and the Simulate Annealing. Furthermore, a subset of this dataset is identified to showcase instances where the purely quantum approach demonstrates competitive performance.

5.1 Dataset I - European capitals

Table 5.1 shows the main characteristics for the European capitals, pertinent to implementing and solving the problem. A general idea of how sparse is the resulting QUBO matrix can be translated by the density parameter, in which non-zero entries representing constraints are accounted in relation to the square of number of variables.

Table 5.1: Dataset I main properties

	Number of decision variables	Number of unique targets	Constraint density (%)
Dataset I: European Capitals	894	32	25.84

Even if the geographical disposition for this dataset can be considered dense, the resulting matrix is quite sparse with about 25% of constraint density.

5.1.1 Tabu Search

The results for the Tabu Search are presented in figure 5.1, in which each arrow indicates the moment - specific discretized VTW slot - of acquisition and the different colors, a distinct satellite. The algorithm ran 100 times with a tenure size of 20.

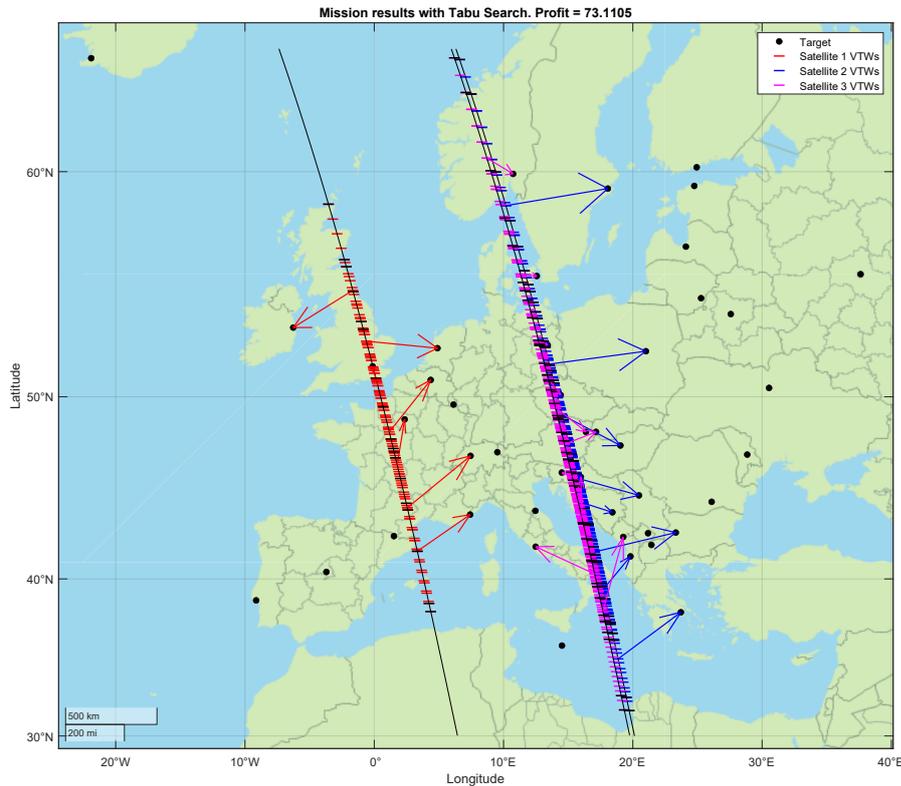


Figure 5.1: Captured targets with Tabu Search

From the 32 unique possible targets considered, the algorithm took 5.634 seconds to run, resulting in 27 being captured and showed a total profit of 73.11. A timeline of events for each S/C is shown by figure 5.2. Every acquisition is represented by a thicker black line, with length varying depending of its specific duration, while dashed lines represent a generic attitude maneuver necessary to perform two consecutive observations.

Due to the adequate implementation of the soft constraint regarding sustainable fuel consumption, the total roll angle change to complete the mission was 433.79

degrees. Without considerate loss of profit, this quantity is about 30% lower than with no soft restrictions. Indeed, the observed solution has a good balance, skipping feasible low profit targets that would also require considerable fuel usage.

This pattern is consistently observed in all subsequent tested mission scenarios, confirming the validity of the approach for fuel-saving.

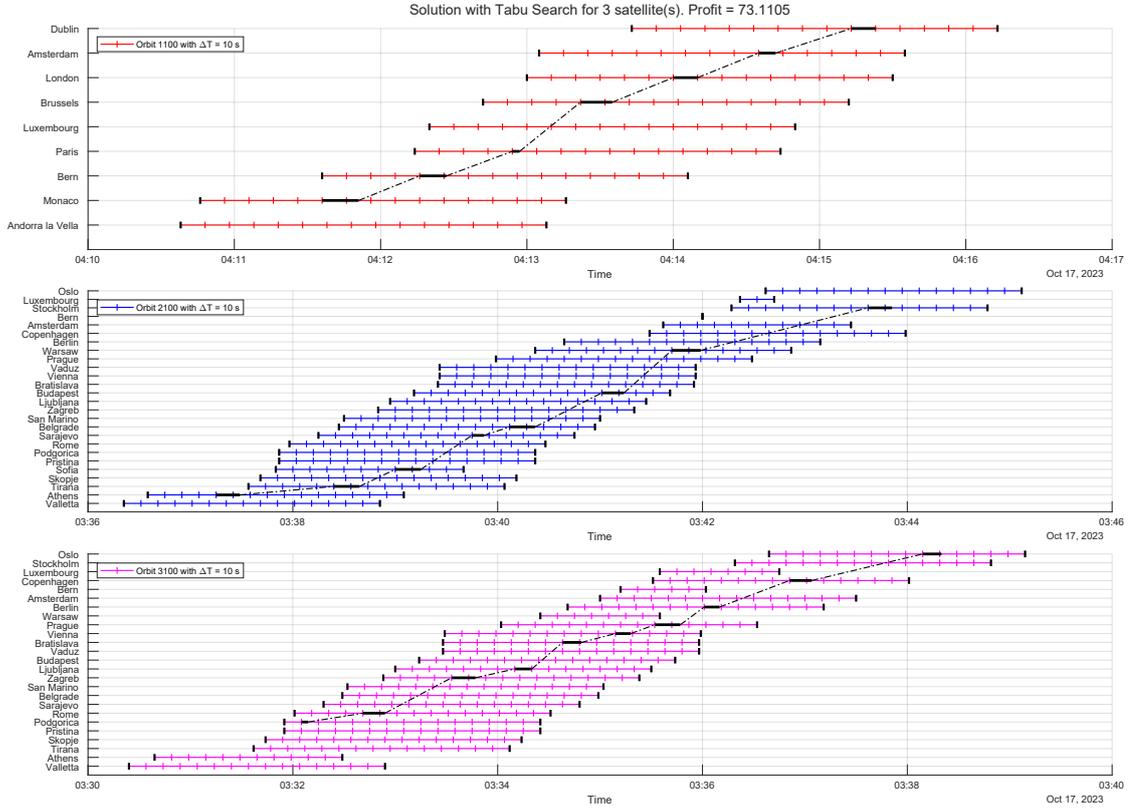


Figure 5.2: Scheduling output for each satellite with Tabu Search

5.1.2 Simulated Annealing

Similar to the Tabu Search, the SA was set for 100 reads with no further change in parameter from the original DWave implementation. The obtained results are shown by figure 5.3, with 26 targets captured, 72.70 profit with a total runtime of 8.783 seconds. The total angle change was 428.19 degrees.

The output sequence for each S/C can be seen in figure 5.4.

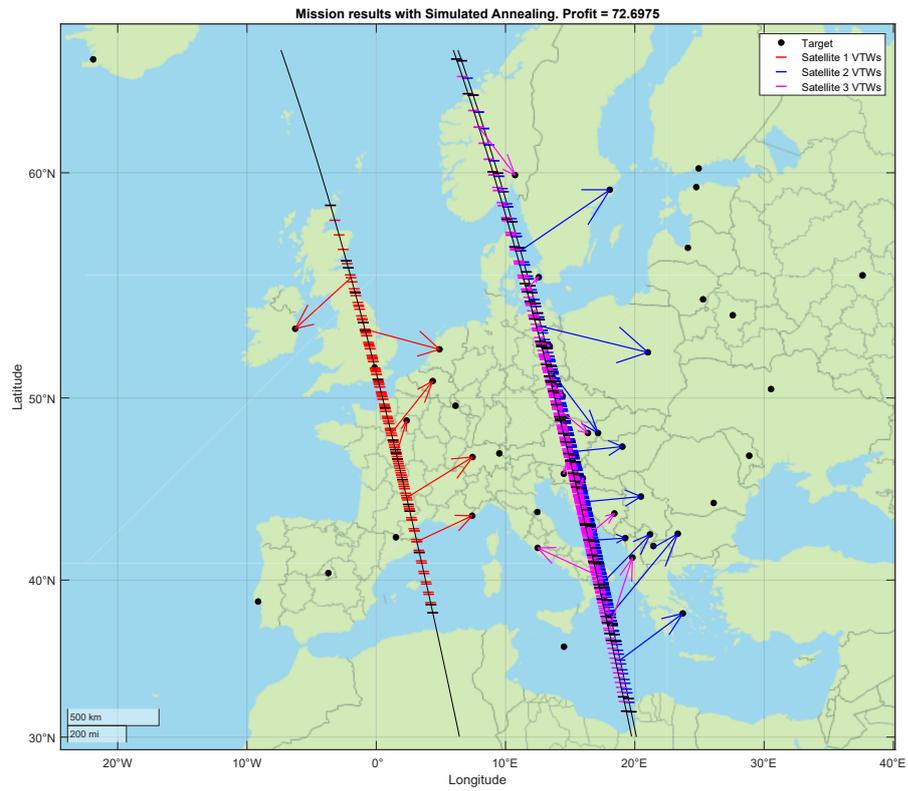


Figure 5.3: Captured targets with Simulated Annealing

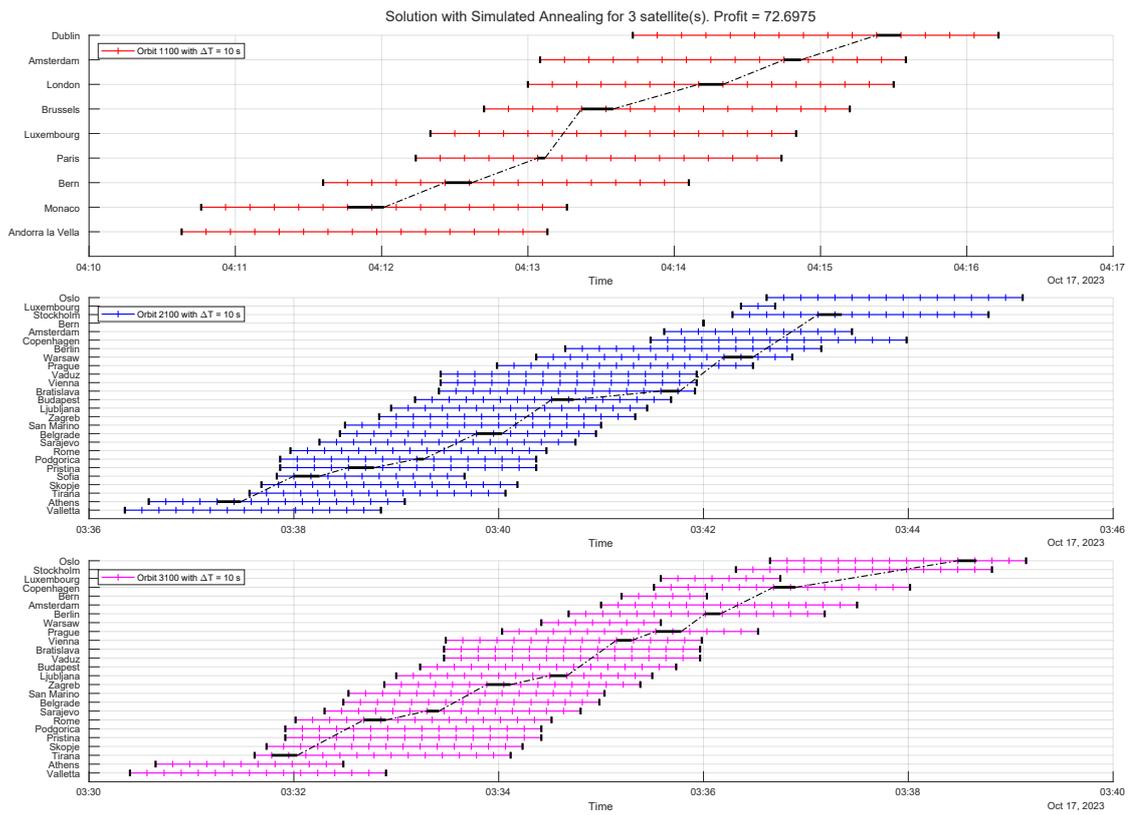


Figure 5.4: Scheduling output for each satellite with Simulated Annealing

5.1.3 Hybrid Quantum Annealing

The hybrid method was imposed to run with a three seconds limit time - the minimum time allowed for problems this size - and output a profit of 75.17 with 27 targets captured. The total roll variation was 422.37 degrees. The results can be seen in figure 5.5 and 5.6.

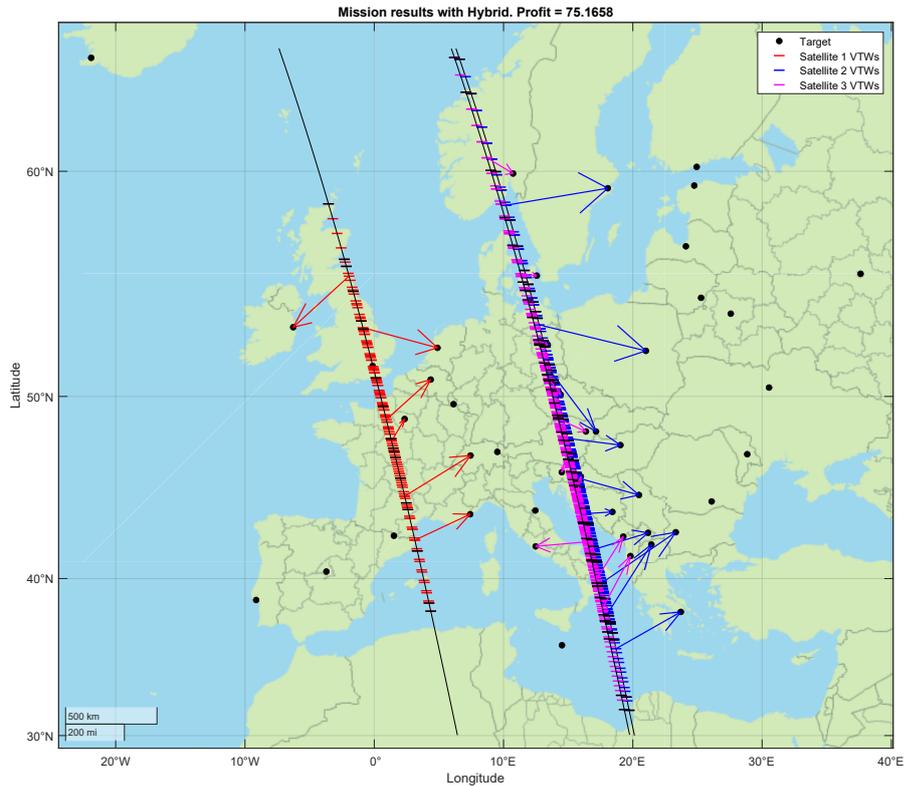


Figure 5.5: Captured targets with Hybrid Quantum Annealing

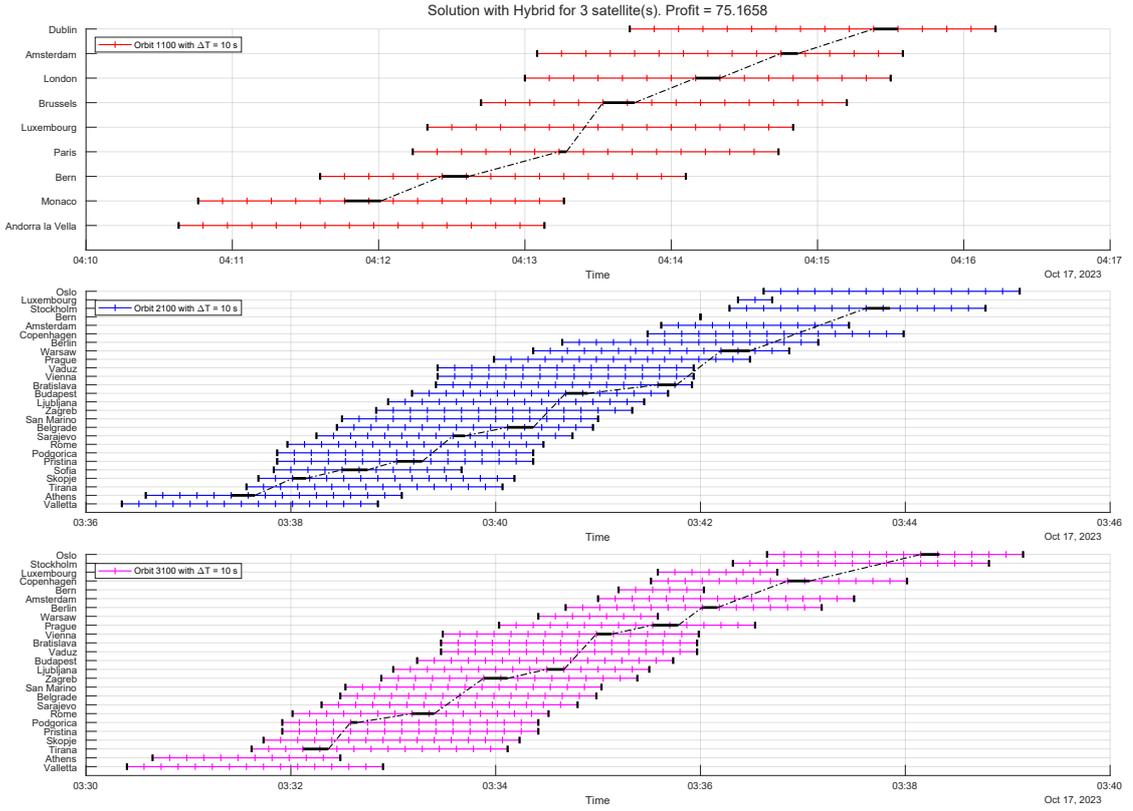


Figure 5.6: Scheduling output for each satellite with Hybrid Quantum Annealing

5.1.4 Dataset I comparison

A collection of all the results for the European capitals is shown in table 5.2. For all algorithms, there can be observed a good balance between the division of tasks, especially for S/Cs 2 and 3 working in tandem, and the compromise with sustainable fuel usage while capturing high profit targets, confirming the quality of the formulated QUBO model.

Table 5.2: Comparison between classic and hybrid algorithms for Dataset I

Dataset I: European capitals	Profit	Total runtime(s)	Total roll change(°)
Tabu Search	73.11	5.634	433.79
Simulated Annealing	72.70	8.783	428.19
Hybrid Quantum Annealing	75.17	2.993	422.37

Furthermore, all the solutions are feasible and of acceptable performance, but it can be noted that the hybrid algorithm output is 46.88% faster and has a slight improvement in profit and fuel usage. This satisfactory result was obtained

utilizing only 42.742 ms of QPU access time in respect to 2.993 seconds of total runtime, showing the effectiveness of combining the classical to quantum methods as described by Chapter 3.

5.2 Dataset II - UNESCO Natural World Heritage sites

The second dataset representing different UNESCO Natural World Heritage sites is split in two cases. Firstly, a smaller subset with only satellite 1 is considered, so the purely quantum approach gives actual competitive solutions. Secondly, the the entire dataset is contemplated. Their characteristics are shown by table 5.3.

Table 5.3: Dataset II main properties

	Number of decision variables	Number of unique targets	Constraint density (%)
Dataset II: UNESCO Sites Only Satellite 1	95	6	63.38
Dataset II: UNESCO Sites	159	10	32.93

Even if the constraint density is higher, the number of variables - most important parameter when embedding onto the QPU - significantly decreases, facilitating the problem resolution.

5.2.1 Tabu Search

Analog to the previous implementation, the Tabu Search was set to run 100 times, with 20 as tenure. All the other parameters are fixed as default.

The results for the smaller subset can be seen in figures 5.7 and 5.8, while those for the complete group of targets are shown by figures 5.9 and 5.10. The quality of these solutions can be seen on tables 5.4 and 5.5.

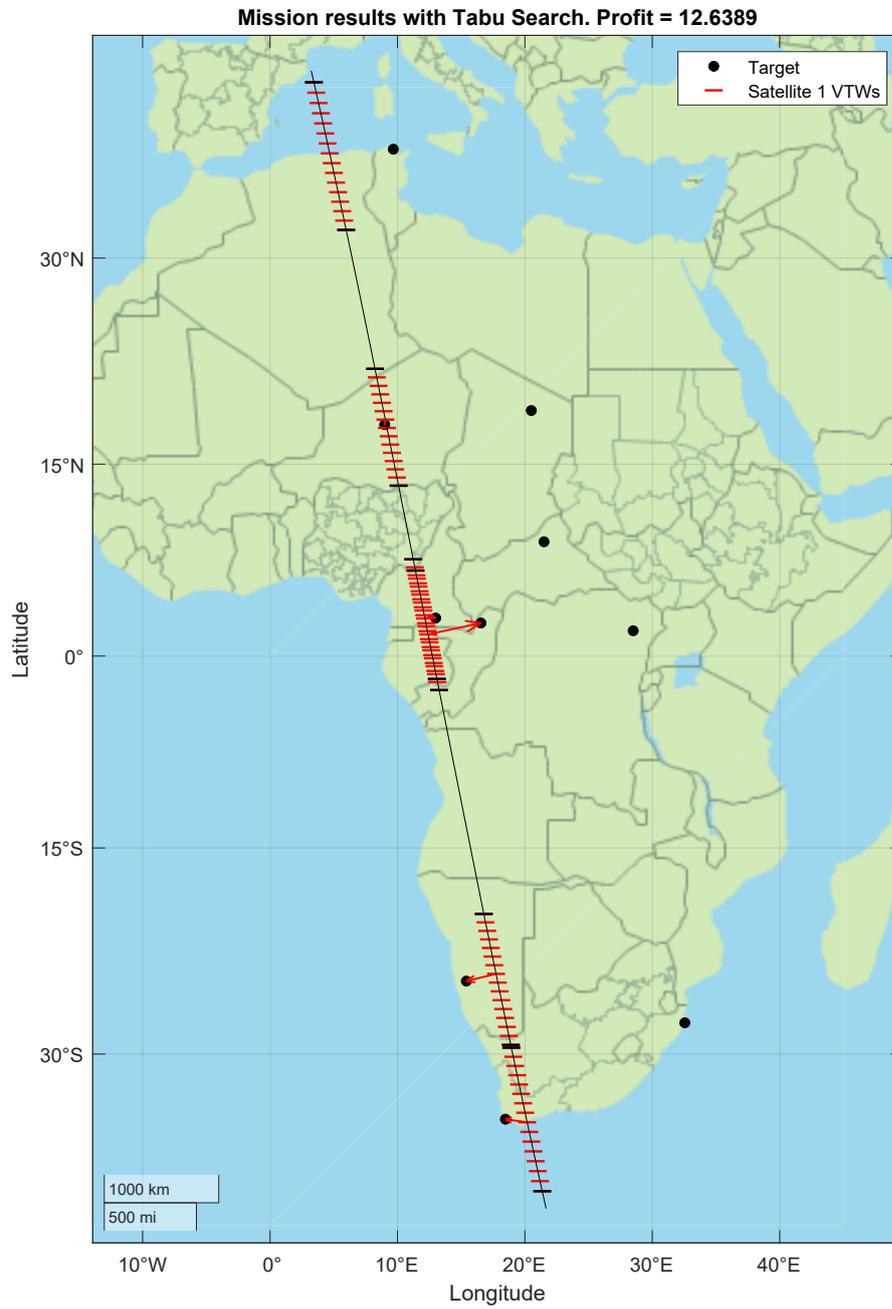


Figure 5.7: Captured targets with Tabu Search

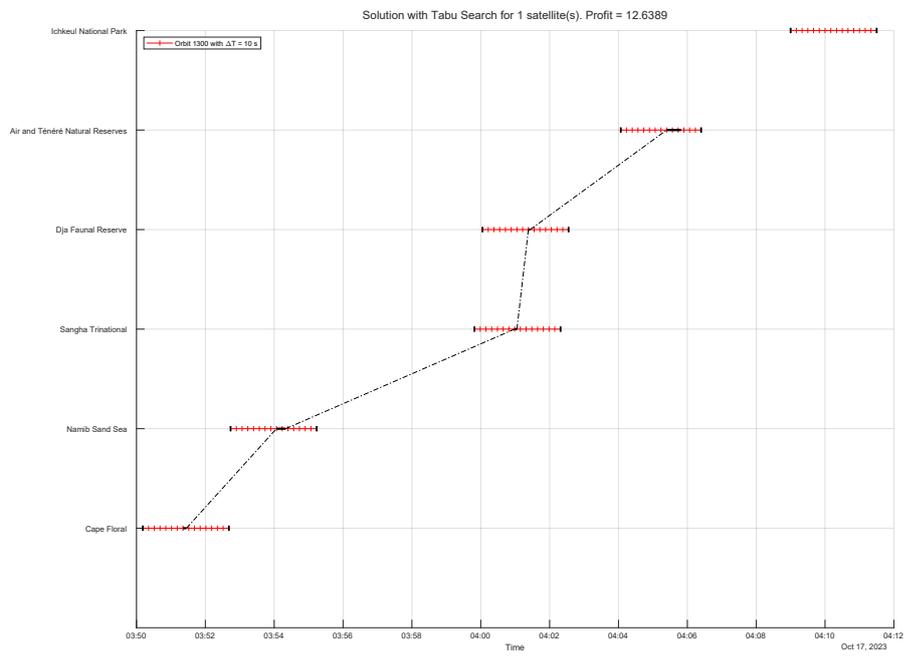


Figure 5.8: Scheduling output for the satellite with Tabu Search

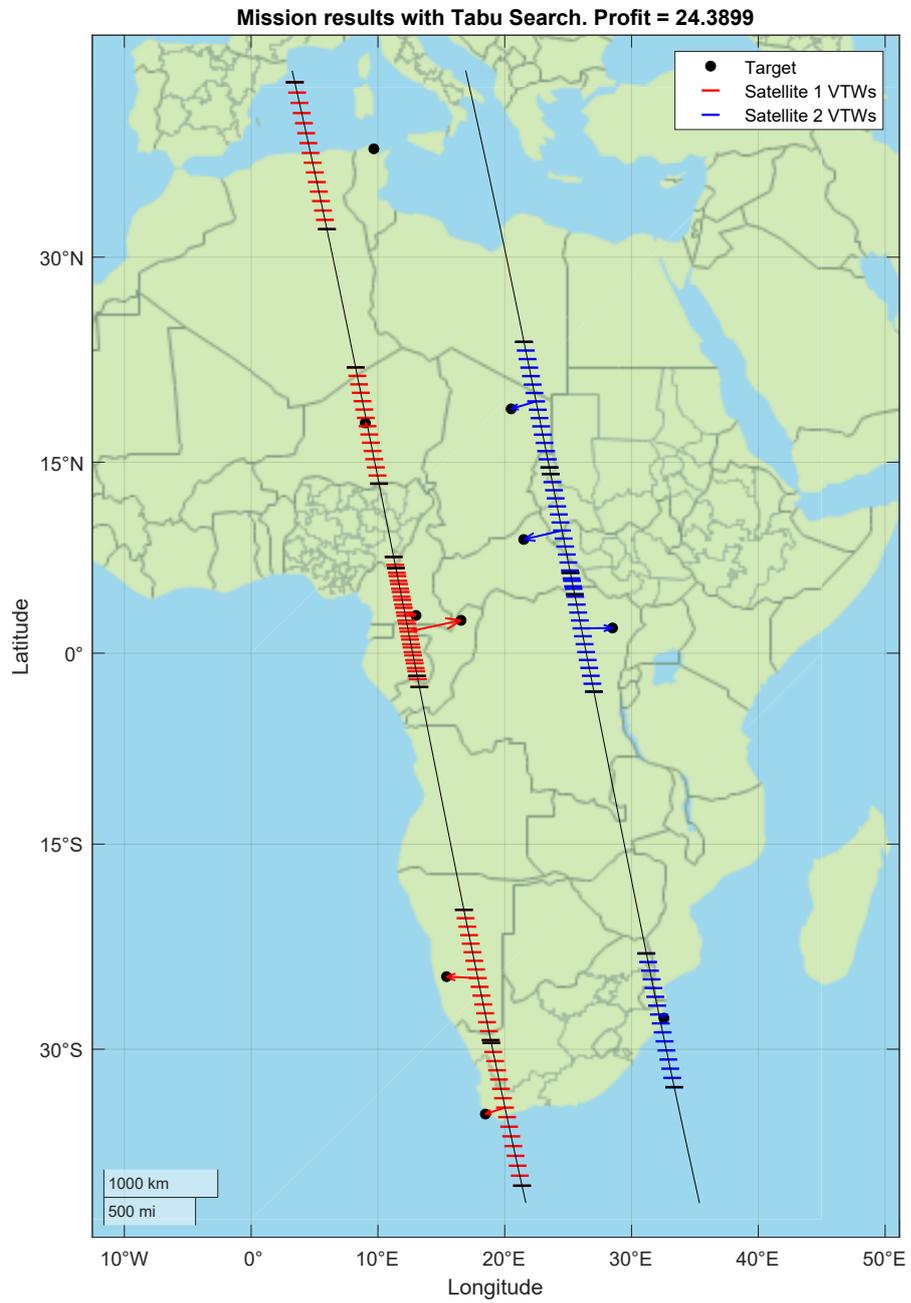


Figure 5.9: Captured targets with Tabu Search

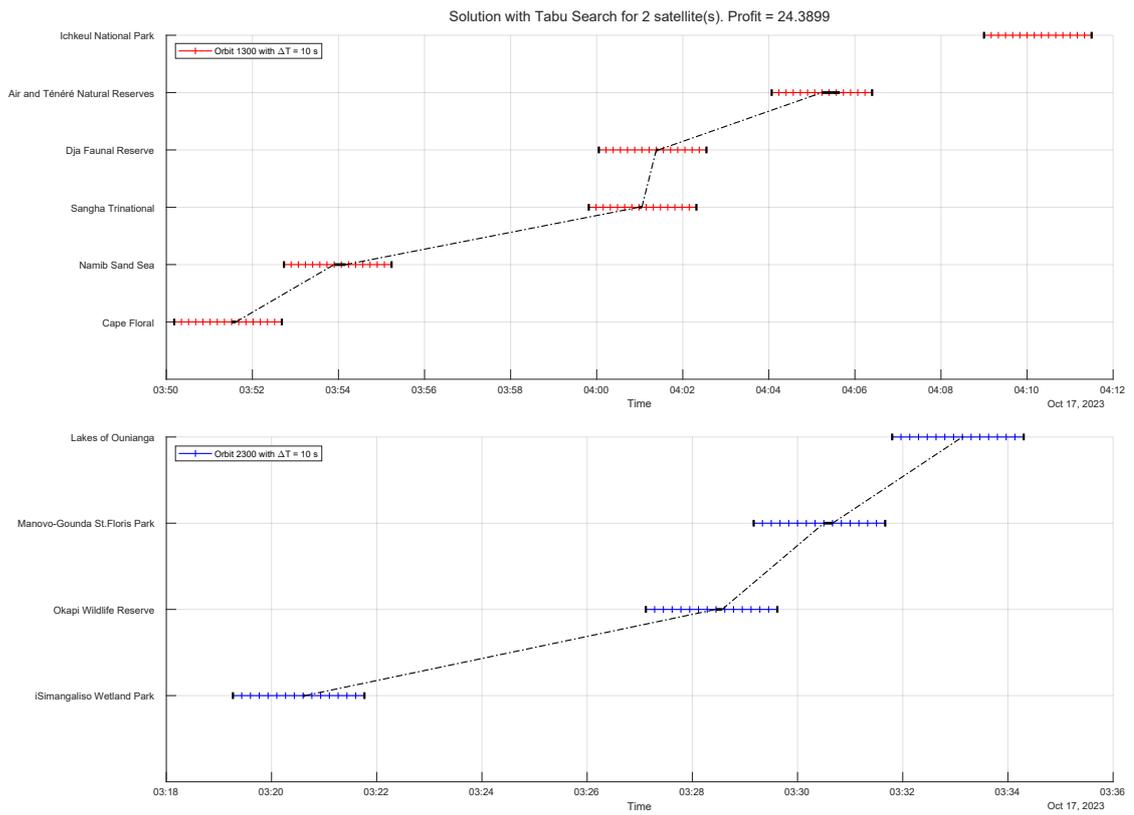


Figure 5.10: Scheduling output for each satellite with Tabu Search

5.2.2 Simulated Annealing

The SA was setup for 100 runs considering all extra parameters as default. As results, the algorithm outputs the responses shown by figures 5.11 and 5.12 for the smaller task group, and figures 5.13 and 5.14 for the complete one.

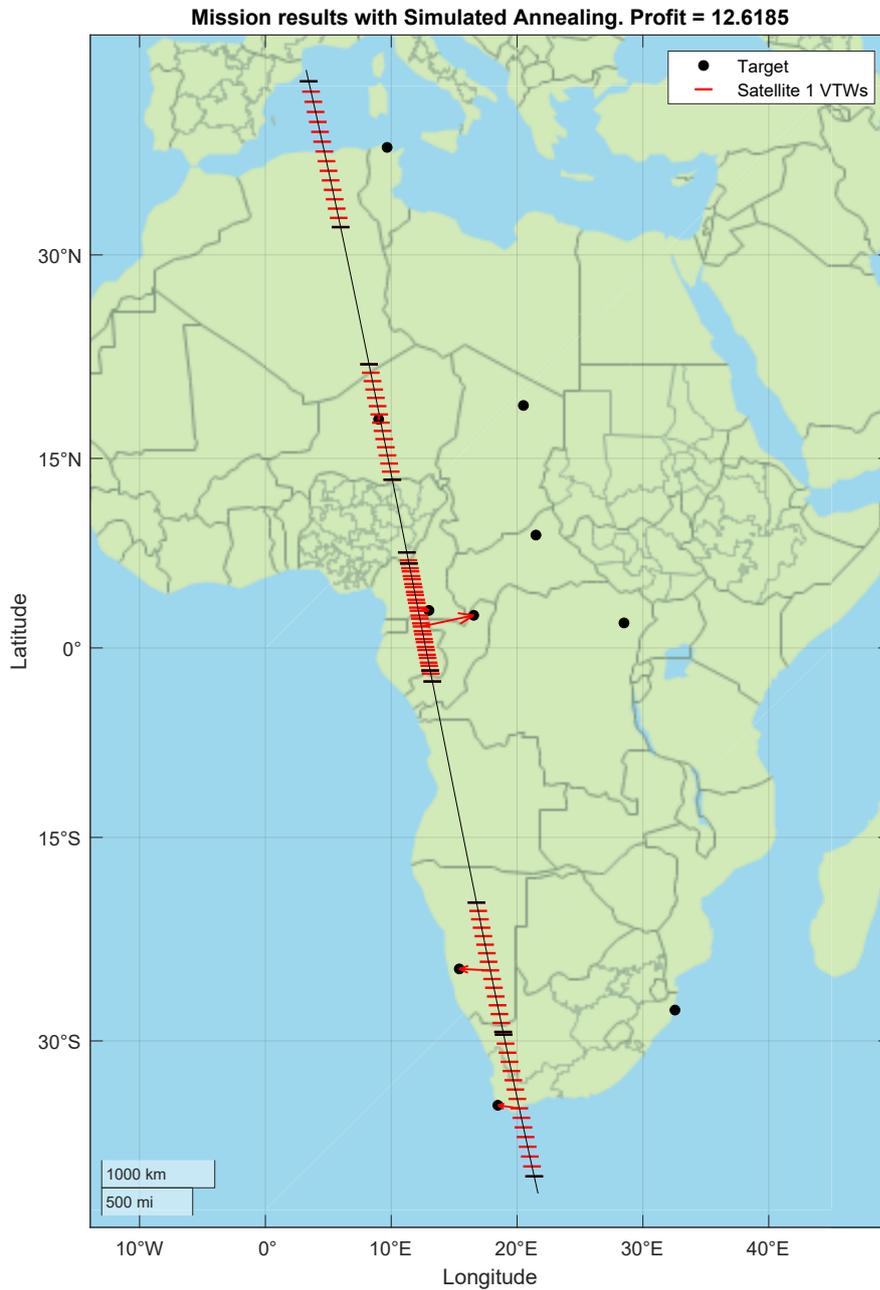


Figure 5.11: Captured targets with Simulated Annealing

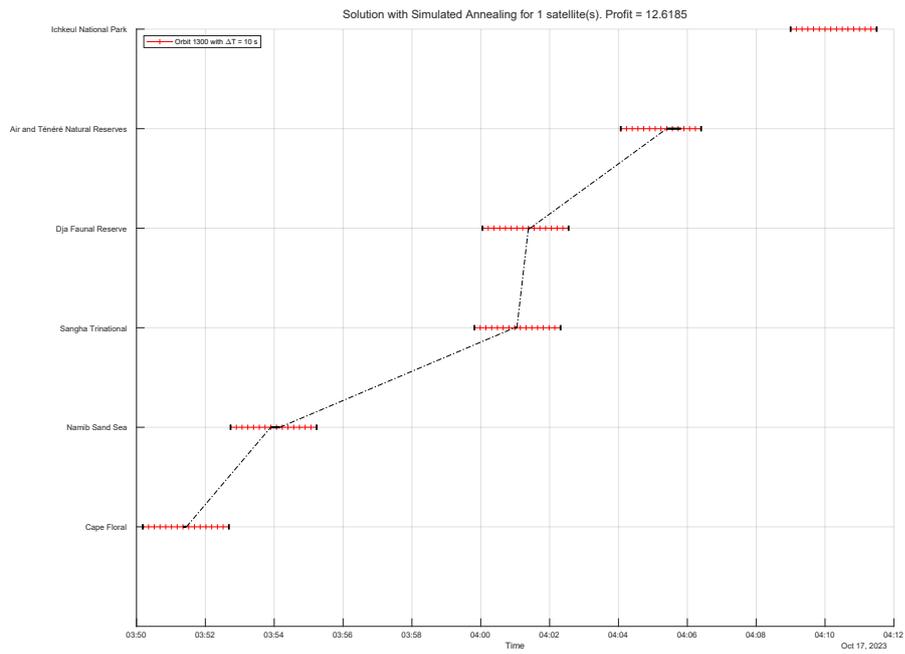


Figure 5.12: Scheduling output for the satellite with Simulated Annealing

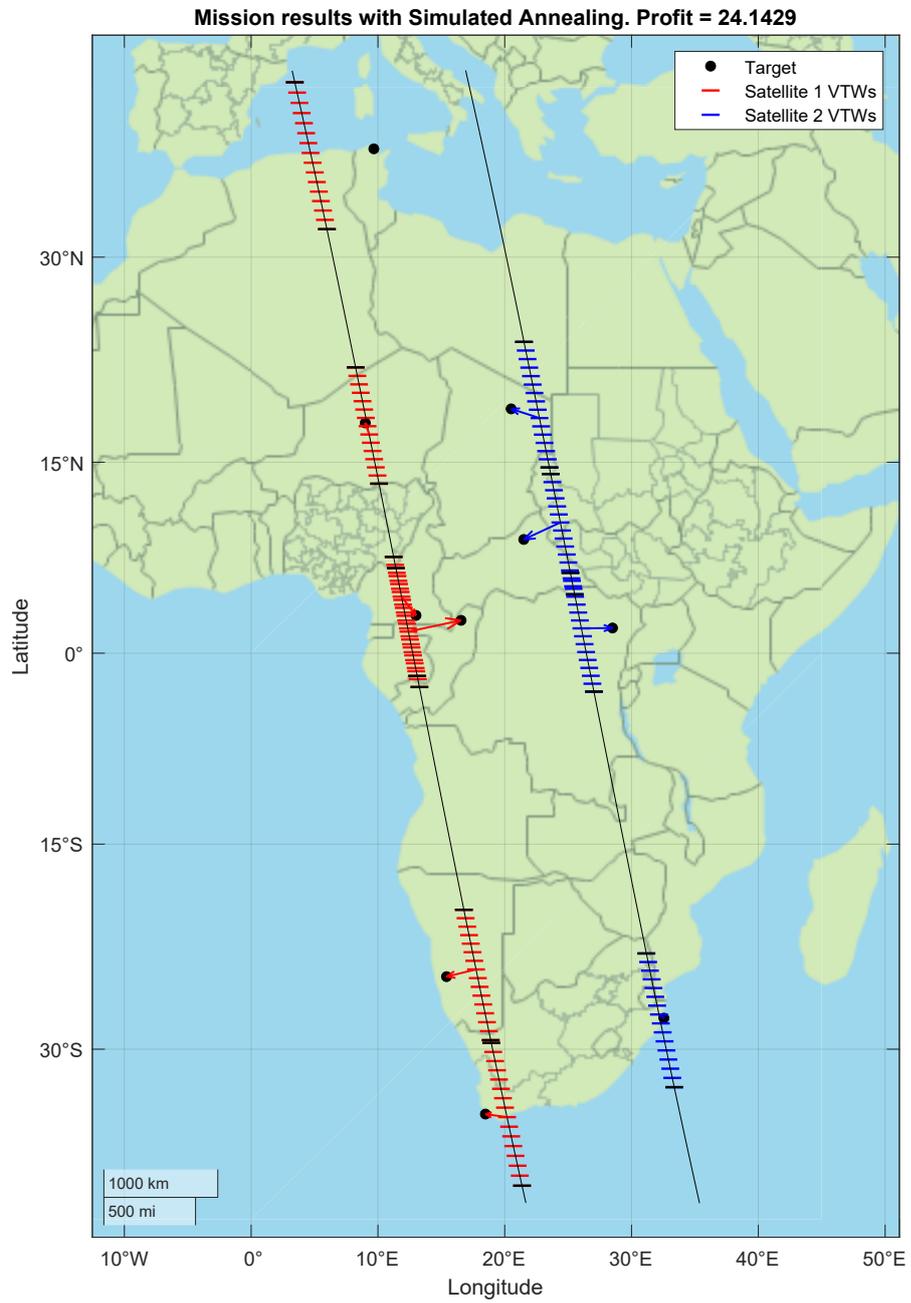


Figure 5.13: Captured targets with Simulated Annealing

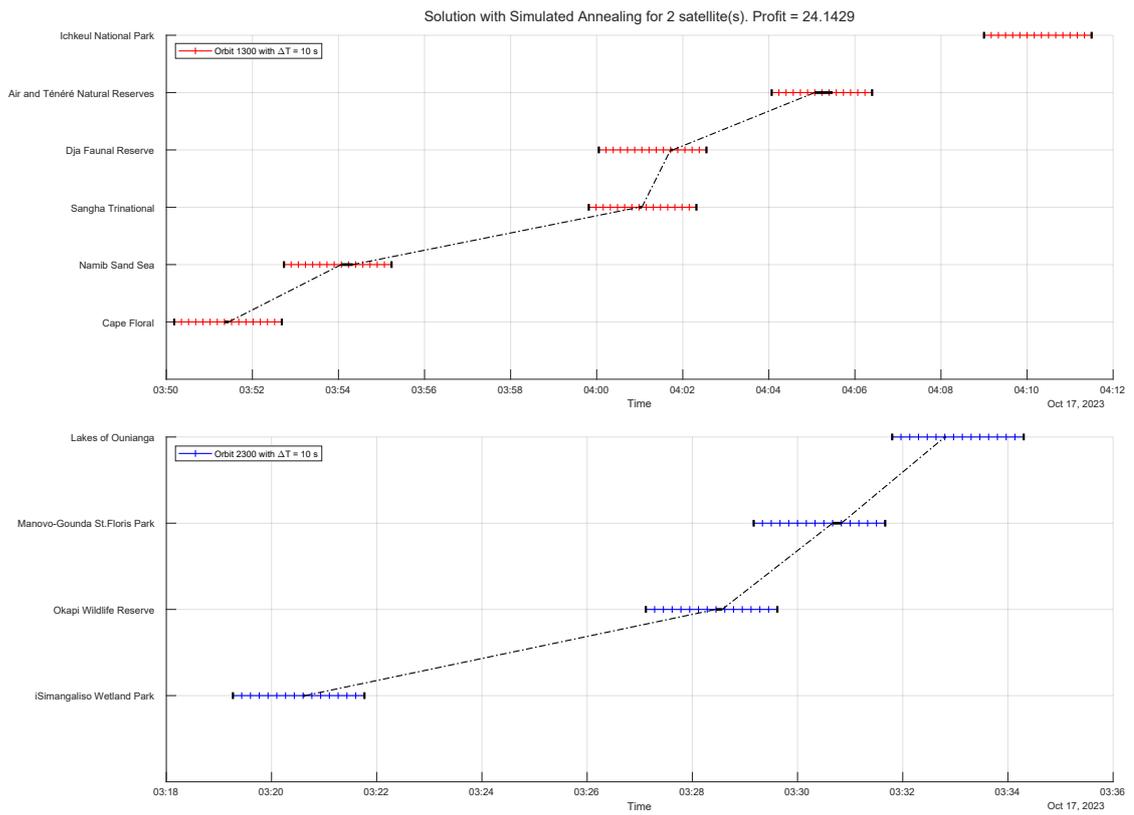


Figure 5.14: Scheduling output for each satellite with Simulated Annealing

5.2.3 Quantum Annealing

Finally, the purely Quantum Annealing approach is shown in this subsection. A pertinent point are the limitations of the trial and free version of the D-Wave platform. Besides the monthly total time for submitting problems, there is also an upper limit for each instance that should not be exceeded, restricting the maximum number of reads and the annealing schedule to be used.

Ideally, given the intrinsic heuristic properties of this method, a high number of runs would be preferred, so outliers solutions are discarded and the most repeating ones are selected, in a majority voting scheme. The number of reads is set to 2000 for both cases and a fixed automatic embedding is preferred.

On the other hand, the two different sets of targets required distinct parameters for embedding and running, especially due to the number of variables involved. For the optimization problem with the complete set of UNESCO sites, a custom anneal schedule was selected. The routine anneals for 10 microseconds, pauses for 20, then finishes the process within another 10 microseconds. Longer times with interruptions are considered a good practice for larger instances. For the partial dataset, however, this parameter is not changed. Lastly, the rest of the parameters are left as default for both cases.

The selected results are shown bellow, on figures 5.15 and 5.16 for the smaller dataset, and on figures 5.17 and 5.18 for the complete one.

An interesting aspect of these solutions is the high number of chain breaks, mischaracterizing the original problem for the larger instance. Consequently, the soft constraints are not dealt adequately, so high fuel consumption maneuvers are considered.

This behavior, however, could be avoided for the smaller instance, where solutions with more than 10% of chain brakes happened less than a third of the runs - rate considered enough for an acceptable output. Indeed, figure 5.15 shows how satellite 1 skips the last low priority target, since it would require a high fuel usage.

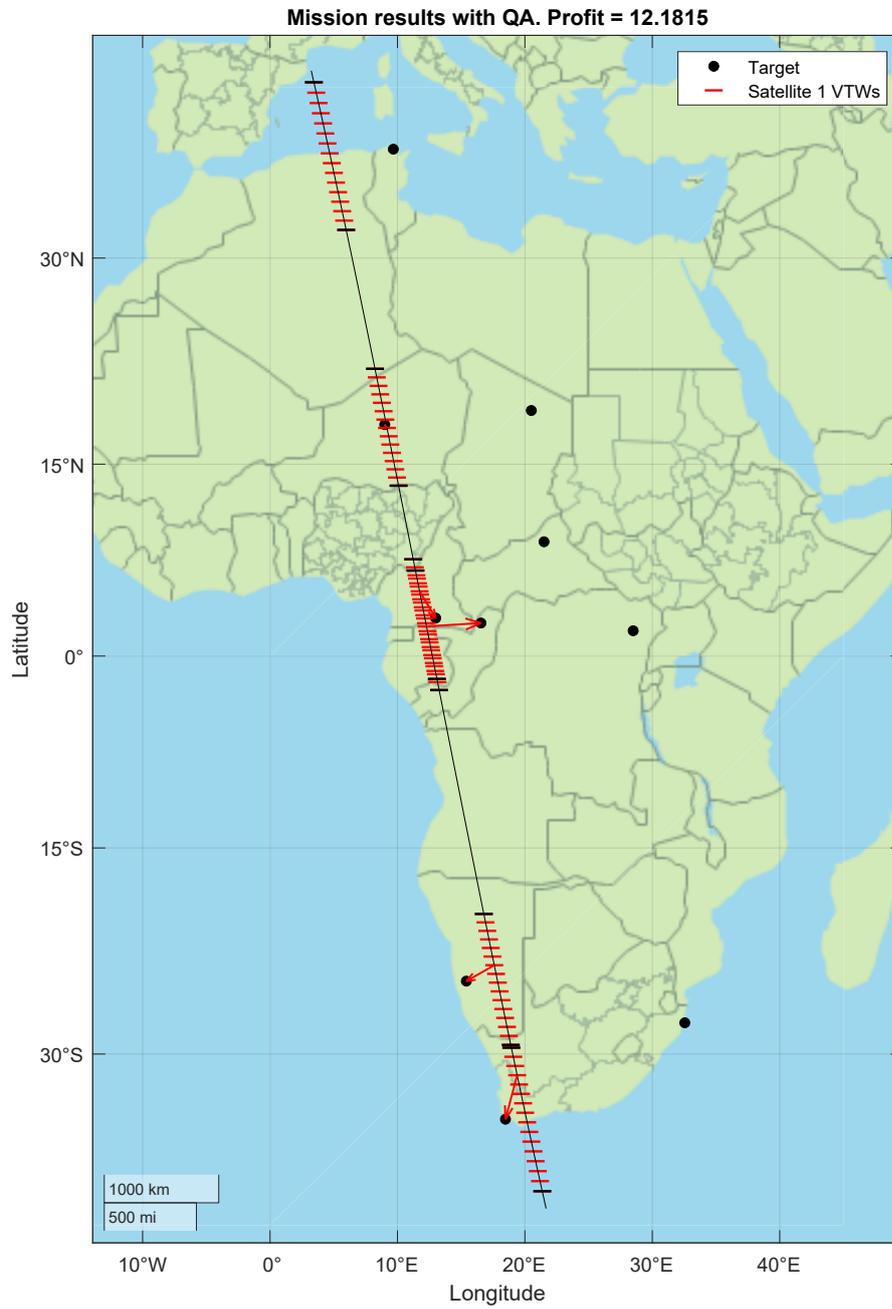


Figure 5.15: Captured targets with Quantum Annealing

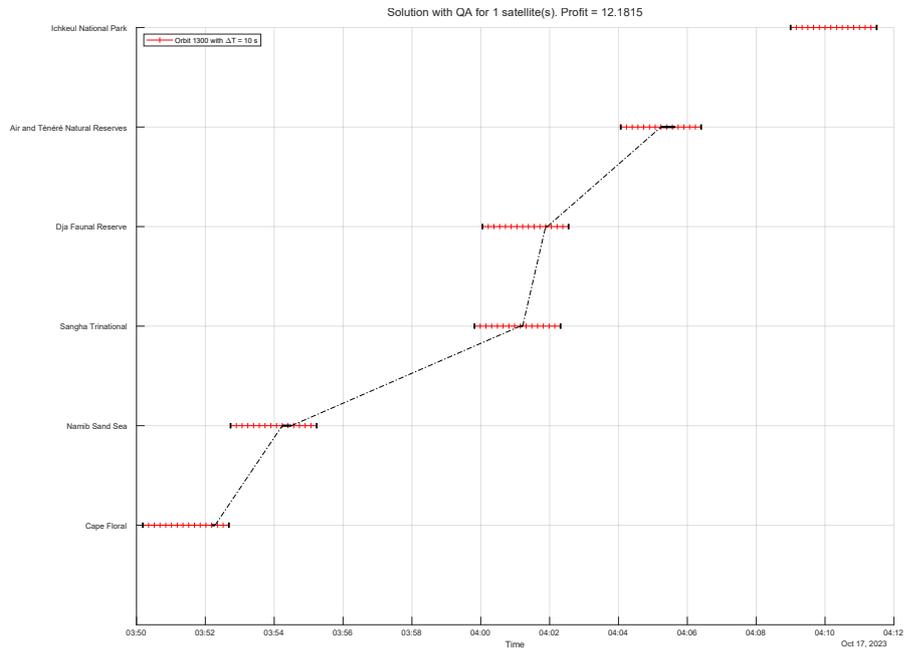


Figure 5.16: Scheduling output for the satellite with Quantum Annealing

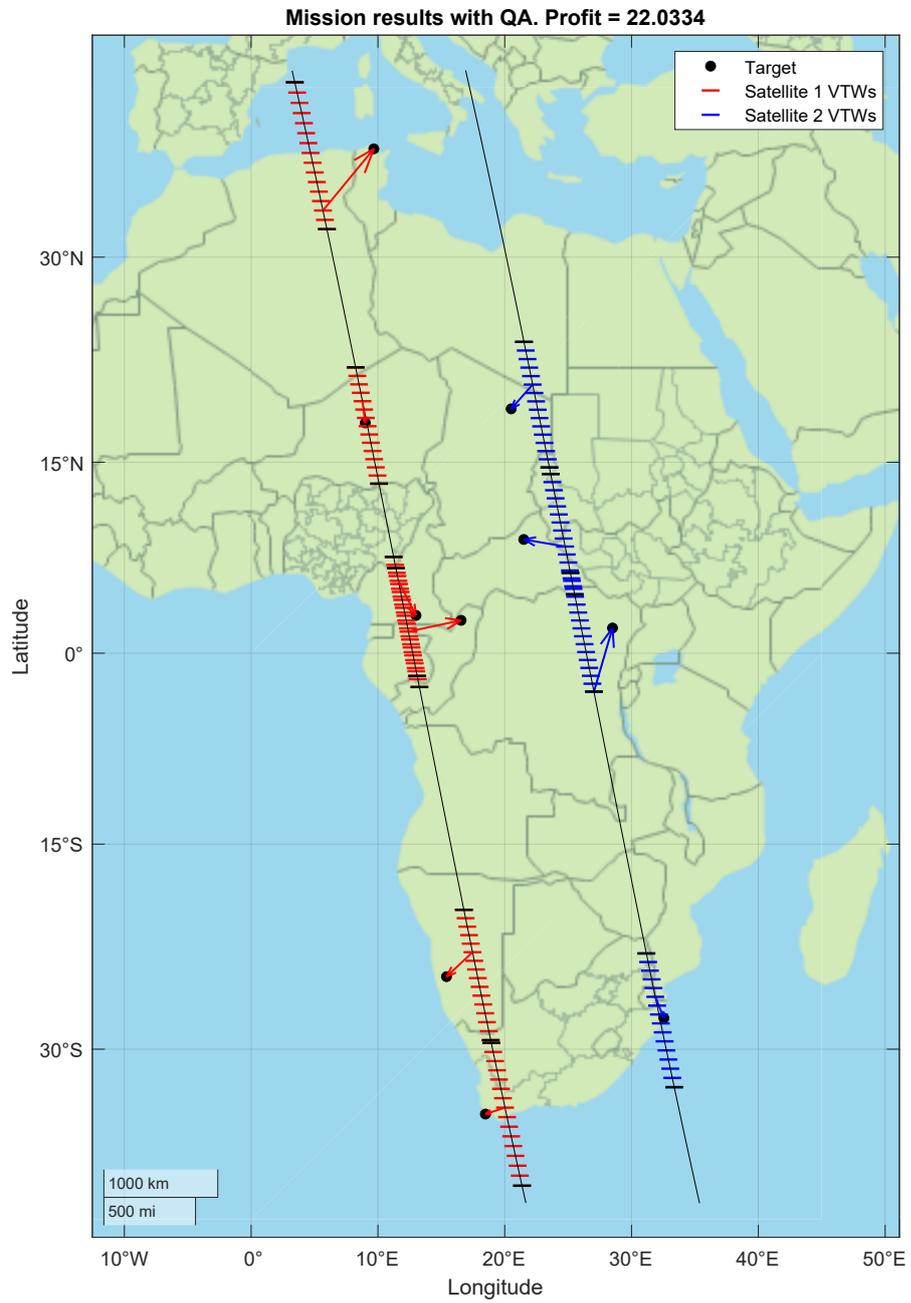


Figure 5.17: Captured targets with Quantum Annealing

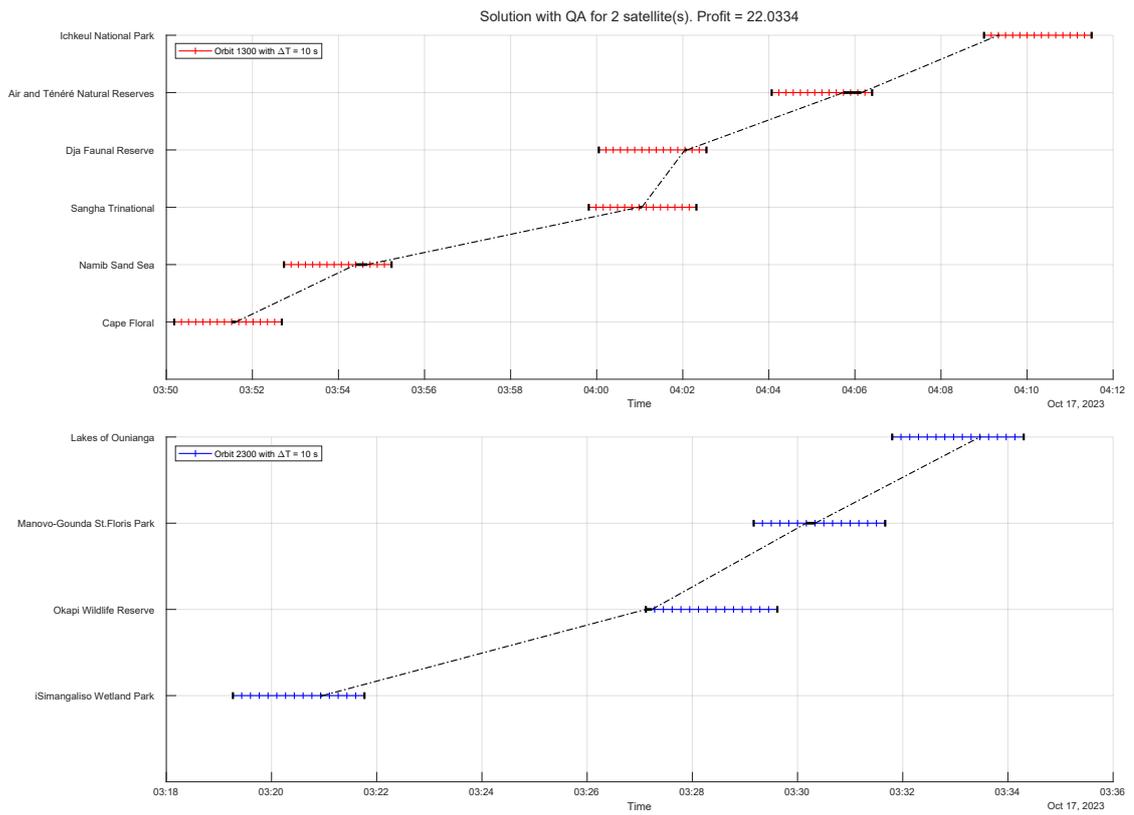


Figure 5.18: Scheduling output for each satellite with Quantum Annealing

5.2.4 Dataset II results comparison

The following section compares the results between the two classical approaches to the purely quantum one. Tables 5.4 and 5.5 collect all of the solutions. In this case, all the methods introduce possible viable mission results with comparable profit and total roll change.

However, it is clear the advantage from the point of view of runtime for both SA and QA. Ideally, better quality quantum solutions could be obtained by raising the number of reads, but the committed time would not be competitive.

Accordingly, even if the purely quantum method outputs viable responses, there are no apparent advantages over the classical counterparts.

Table 5.4: Comparison between classic and quantum algorithms for a subset of Dataset II - satellite 1

Dataset II: Only satellite 1	Profit	Total runtime(ms)	Total roll change(°)
Tabu Search	12.64	2131	116.58
Simulated Annealing	12.62	409.4	117.30
Quantum Annealing	12.18	454.2	118.73

Table 5.5: Comparison between classic and quantum algorithms for Dataset II

Dataset II: UNESCO sites	Profit	Total runtime(ms)	Total roll change(°)
Tabu Search	24.39	2136	211.50
Simulated Annealing	24.14	669.72	209.36
Quantum Annealing	22.03	552.56	241.33

Chapter 6

Conclusions

Foremost, this work was able to model and translate the AEOSs scheduling problem into a flexible QUBO formulation that can be implemented not only onto the QPU for the specific case of Quantum Annealing, but also be treated as a generic input for classical optimization solvers, such as the Simulated Annealing and the Tabu Search. This modeling approach allowed the imposition of different mission scenarios, combining soft and hard constraints in a straightforward manner and simulating realistic client requests after simply changing certain input files representing a list of targets.

Moreover, the carried work points that the purely quantum solutions quality closely aligns with classical heuristic methods for small-scale - yet realistic - problem instances. Accordingly, although these results are initially acceptable, their performance and precision degrade as the number of variables and constraint density increase, due to the heuristic nature of the solver, limited number of physical qubits, environmental noise, and complex embedding process.

The obtained results suggest that hybrid techniques such the ones utilized in this thesis, most specific on Chapter 3, are already ready to be used in real and scalable missions scenarios. We could observe a well coordinated workflow that improved solutions not only regarding suboptimality, but mainly about their computational time through a phenomenon called hybrid acceleration. However, this does not necessarily imply that this improvement always happen. As explained on 3.2.2, if the submitted scenario structure is too complex or even too simple, the solver will output a purely classical heuristic solution without the quantum boost.

Even if the technology looks promising, rapidly advancing and the hybrid method delivers competitive solutions, this thesis hints that fully utilizing a pure quantum method on real-world problems is reserved for the next years as the new generation of QPUs are built, upon significant enhancements in hardware capabilities and the development of more uncomplicated problem-embedding strategies.

6.1 Future Works

Regarding the current developed model for the scheduling problem, some further improvements and modifications may be carried out. Firstly, relaxing the constraints of the model would enable it to encompass a wider array of problems. For instance, transforming the unicity binary constraint into an integer form would allow multiple acquisitions of the same target.

Furthermore, for a more realistic planning scenario, a scheduling horizon of one day should be imposed instead of only a single orbital revolution, mimicking the majority of real earth observation missions. While this would result on a more complex problem instance with an even larger number of decision variables, it could certainly be submitted to the hybrid solver.

Finally, a more sophisticated reconfigurability model for orbital adjustments could be designed in order to track movable or area targets, such that the constellation could change its orbital parameters to track and monitor rapidly moving hurricanes or wild fires. In this case, the algorithm would not only define a timeline of targets to be captured, but also an ideal orbit for maximum profit and low fuel consumption.

Appendix A

Quantum and Hybrid Annealing setup

```
import scipy.io as sio
from dwave.system import DWaveSampler, composites
import numpy as np

5 # Dwave account token
# token = 'insert token here'

# Import Q matrix - Change this matrix
filename = 'Q13002300'
10 importedQ = sio.loadmat('data/'+filename+'.mat')
Q_matlab = importedQ['Q']

n,m = Q_matlab.shape
qubo = {}

15 ##### Construct qubo Dict ##### Considers only entries with values
    so we dont embed 0's
for i in range(n):
    for j in range(i, n):
        if Q_matlab[i,j] != 0.0:
20             qubo[(i, j)] = Q_matlab[i, j]

##### D-WAVE QUANTUM ANNEALER
#####
num_reads=2000
chain_strength=9
25 anneal_schedule=( [0.0,0.0] , [10.0,0.5] , [30.0,0.5] , [40.0,1.0] )

# We can also change the embedding type and parameteres
solver = composites.EmbeddingComposite(DWaveSampler(token=token))
```

```

response = solver.sample_qubo(qubo,
30         num_reads=num_reads,
         anneal_schedule=anneal_schedule,
         label='Quantum_Annealing'
         )

35
bestSolution = response.first.sample
result = {"xopt": list(bestSolution.values())}
print(result)

40
savefile = 'results/QA_'+filename+'_results.mat'
sio.savemat(savefile, {'sol':result, 'full_data': response.record,
        'info':response.info})
print(response.record)
print(response.info["embedding_context"]["chain_strength"])
print("Percentage of samples with >10 percent chain breaks is {}".format(
45     np.count_nonzero(response.record.chain_break_fraction >
        0.10)/num_reads*100))

```

Script A.1: Setup and Embedding for the Quantum Annealing

```

import scipy.io as sio
from dwave.system import LeapHybridSampler

5
# Dwave account token
# token = 'insert token here'

# Import Q matrix - Change this matrix
10 filename = 'Q13002300'
importedQ = sio.loadmat('data/'+filename+'.mat')
Q_matlab = importedQ['Q']

n,m = Q_matlab.shape
15 qubo = {}

#### Construct qubo Dict ####
for i in range(n):
    for j in range(i, n):
20         if Q_matlab[i,j] != 0.0:
            qubo[(i, j)] = Q_matlab[i, j]

##### D-WAVE Hybrid Solver
#####
25

```

```
time_limit=3

# We can call sample_qubo because it will transform it into a BQM
# anyways
# No need to embed the problem
30 solver = LeapHybridSampler(token=token)
response = solver.sample_qubo(qubo,
                               # time_limit=time_limit,
                               label='UNESCO_Reduced_Hybrid'
                               )
35

bestSolution = response.first.sample
result = {"xopt": list(bestSolution.values())}
40 print(result)

savefile = 'results/Hybrid_'+filename+'_results.mat'
sio.savemat(savefile, {'sol':result, 'info':response.info})
print(response.info)
```

Script A.2: Setup for the Hybrid Quantum Annealing


```

    pitch{i} = importedAngles{i}{4};
    w{i} = wbar(i)*cosd(roll{i}).*cosd(pitch{i});
end
35
% Downsampling
Dt = 10;
for i = 1:length(id)
    VTW_ds{i} = downsample(VTW{i},Dt);
    roll_ds{i} = downsample(roll{i},Dt);
    pitch_ds{i} = downsample(pitch{i},Dt);
    w_ds{i} = wbar(i)*cosd(roll_ds{i}).*cosd(pitch_ds{i});
end
40
VTW_targets{1} = VTW_ds';
wij = w_ds{1}';
timeLine = VTW_ds{1}';
fullRoll = roll_ds{1}';
fullPitch = pitch_ds{1}';
for i = 2:length(id)
    timeLine = [timeLine VTW_ds{i}'];
    fullRoll = [fullRoll roll_ds{i}'];
    fullPitch = [fullPitch pitch_ds{i}'];
    wij = [wij w_ds{i}'];
end
45
50
% Grab the profits in minimizing form
c{1} = -wij;

% Number of Variables
N(1) = length(timeLine);

% Build unicity penalty
Qtemp = {1,length(id)};
for j = 1:length(id)
    for k = 1:length(VTW_ds{j})
        Qtemp{j} = triu(ones(k),1);
    end
end
55
Qc = lambda_m*blkdiag(Qtemp{1:end});

% Roll Constraint
checked = 0;
for i = 1:length(id)
    for j = i+1:length(id)
        if abs(mean(roll_ds{i})-mean(roll_ds{j})) > 30
            Qc(checked+1:length(vertcat(VTW_ds{1:i})),
length(vertcat(VTW_ds{1:j-1}))+1:length(vertcat(VTW_ds{1:j})))
= lambda_soft;
        end
    end
end
60
65
70
75
end
end

```

```

        checked = checked + length(VTW_ds{i});
80    end

    % Generate single Q
    % Forbidden maneuver penalty + timing constraints
    checked = 0;
85    for j = 1:length(id)
        for k = 1:length(VTW_ds{j})
            for i = checked+1:N(1)
                if timeLine(checked+k) + seconds(dur(j)) + manTime
                    (fullRoll(checked+k),fullPitch(checked+k), fullRoll(i),
                    fullPitch(i)) > timeLine(i) && i > checked + length(VTW_ds{j})
                    || timeLine(checked+k) + seconds(dur(j)) > VTW_ds{j}(end)
90                    Qc(checked+k,i) = lambda_m;
                end
            end
        end
        checked = checked + length(VTW_ds{j});
    end
95    % Make it upper triangular
    Qc(find(tril(ones(N(1)),-1))) = 0;
    Qsingle{1} = Qc;
    VTW_ds = {};
end
100 % Place Q onto the diagonal
Q = blkdiag(Qsingle{1:end});

%% Build unicity constraint for multiple satellites
105 Qaux = zeros(size(Q));
s_column = 1;
s_row = 1;
s_row_aux = 1;
110 s_column_aux = N(1) + 1;
idx = [];
k = 1;

% All possible pair of sats
115 for j = 1:length(N)
    for k = j+1:length(N)
        Nc = [N(j) N(k)];
        % Building repeated targets list
        for i = 1:length(target_id{k})
120            idx_temp = target_id{j}(target_id{k}(i)==target_id{j})
;
                if ~isempty(idx_temp)
                    idx = [idx idx_temp];
                end
            end
        end
    end
end

```

```

125     end
        % Penalize their VTWs
        for i = 1:length(target_id{j})
            if ismember(target_id{j}(i), idx)
                idx_r = i;
                idx_c = find(target_id{k}==target_id{j}(i));
130                s_row = s_row_aux + length(vertcat(VTW_targets{j}
}{1:idx_r-1}));
                s_column = s_column_aux + length(vertcat(
VTW_targets{k}{1:idx_c-1}));
                Qaux(s_row:s_row+length(VTW_targets{j}{idx_r})-1,
s_column:s_column+length(VTW_targets{k}{idx_c})-1) = lambda_m;
                s_row = s_row_aux;
                s_column = s_column_aux;
135            end
        end
        idx = [];
        s_column_aux = s_column_aux + Nc(2);
    end
    if j < length(N)
        s_column_aux = sum(N(1:j+1)) + 1;
    end
    s_row_aux = s_row_aux + N(j);
end
145 % Final Q
Q = Qaux + Q;
Q = max(abs([c{1:end}]))*Q;
NC = 2*length(find(Q));
150 % Insert profit onto main diag
wfull = [c{1:end}];
wfull(find(diag(Q))) = 0;
Q = Q + diag(wfull);
155 % Calculate constraint
density = 100*NC/(sum(N)^2);

filepath = strjoin(["./data/Q/Q",strjoin(string(orbits'),'')], '');
160 filepath = strjoin([filepath, ".mat"], '');
save(filepath, "Q", "N")

filepath = strjoin(["C:\Users\Vinicius\OneDrive - Politecnico di
Torino\PoliTO\Tesi\Tools\Python\Hybrid_QA_tabu_SA\data\Q",
strjoin(string(orbits'),'')], '');
filepath = strjoin([filepath, ".mat"], '');
165 save(filepath, "Q", "N")

```

Script B.1: Generates QUBO matrix for a constellation from mission files.

Appendix C

Data pre-processing

```
clear
close all
clc

5 % Change target list
load data/target_QUANTUM.mat

% Mission/Sat spec
mu = 3.986004418e14;
10 rE = 6.3781370e6;

% Change sat date
load data/sat3.mat

15 % Time horizon - Date for epoch
startDate = datetime(2023, 10, 17, 3, 0, 0);
[Year, Month, Day, H, M, S] = datevec(startDate);

% TimeStep for orbit propagation (may slow the script)
20 timeStep = seconds(1);

% Number of revolutions/orbits
n = 1;

25 % Mission time
revTime = seconds(2*pi*sqrt(a^3/mu));
endDate = startDate + n*revTime;
timeHorizon = startDate:timeStep:endDate;

30 % Orbital evolution - Initial Position
p = a*(1-e^2);
r = p/(1+e*cosd(nu));
p_0 = r*cosd(nu);
```

```

q_0 = r*sind(nu);
35 r_pq = [p_0 q_0 0]';

% Change of frame from PF to ECI and LLA
T313 = rot_mat([3 1 3],[W i w]);
r_ECI = T313*r_pq;
40 r_LLA = eci2lla(r_ECI',[Year,Month,Day,H,M,S'],'IAU-2000/2006');

% Speed of satellite
n_rad_per_s = sqrt(mu/a^3);
n_deg_per_s = rad2deg(n_rad_per_s);
45

% [deg] angular position at every second for n revolution(s)
Evals = 0:n_deg_per_s*seconds(timeStep):n*360;
r = p./(1+e*cosd(Evals));
% [m] orbit positions
50 Orbit_p = r.*cosd(Evals);
% [m] orbit positions
Orbit_q = r.*sind(Evals);
Vel_p = -sqrt(mu/(a*(1-e^2))).*sind(Evals);
Vel_q = sqrt(mu/(a*(1-e^2))).*(e + cosd(Evals));
55 % [s] time since epoch along orbit
deltaT_s = ((Evals-rad2deg(nu)) - e*sind(Evals-rad2deg(nu)))/
n_deg_per_s;

% Initialize matrices
Orbit_ECI = zeros(length(deltaT_s),3);
60 Vel_ECI = zeros(length(deltaT_s),3);
Orbit_LLA = zeros(length(deltaT_s),3);
Orbit_ECEF = zeros(length(deltaT_s),3);
R_Orbit_LLA = zeros(length(deltaT_s),3);
L_Orbit_LLA = zeros(length(deltaT_s),3);
65 groundTrack_ECI = zeros(length(deltaT_s),3);
targets_ECI = {};

for j = 1:length(timeHorizon)
r_pq = [Orbit_p(j) Orbit_q(j) 0]';
70 v_pq = [Vel_p(j) Vel_q(j) 0]';
Orbit_ECI(j,:) = (T313*r_pq)';
Vel_ECI(j,:) = (T313*v_pq)';
Orbit_ECEF(j,:) = eci2ecef(datevec(datetime(Year, Month, Day,
H, M, S+deltaT_s(j))), Orbit_ECI(j,:));
Orbit_LLA(j,:) = eci2lla(Orbit_ECI(j,:),datevec(datetime(
Year, Month, Day, H, M, S+deltaT_s(j))), 'IAU-2000/2006');
75 groundTrack_ECI(j,:) = lla2eci([Orbit_LLA(j,1), Orbit_LLA(j,2)
, 0], datevec(datetime(Year, Month, Day, H, M, S+deltaT_s(j))),
'IAU-2000/2006');
end

```

```

% MAX Roll lines for plotting
normalDir = cross(Orbit_ECI(1,:),Orbit_ECI(ceil(end/2),:));
80 normalDir = normalDir/norm(normalDir);

for j = 1:length(timeHorizon)
    dateVector = datevec(datetime(Year, Month, Day, H, M, S+
        deltaT_s(j)));
    [L_Orbit_LLA(j,:), R_Orbit_LLA(j,:)] = rollLines(maxRoll,
        normalDir, Orbit_ECI(j,:), groundTrack_ECI(j,:), dateVector);
85 end

% Targets to ECI and ECEF
targets_ECEF = lla2ecef(targets(:,1:3),'WGS84');
for i = 1:size(targets,1)
90     for t = 1:length(timeHorizon)
        targets_ECI{i}(t,:) = lla2eci(targets(i,1:3),datevec(
            datetime(Year, Month, Day, H, M, S+deltaT_s(t))),'IAU-2000/2006
        ');
        end
    end

95 % Loop through the targets to find VTW
VTW_raw = {};
rollAngles = {};
pitchAngles = {};

100 for i = 1:size(targets,1)
    idx = []; k = 1;
    for t = 1:length(timeHorizon)
        % Check if the target is within the sensor's field of view
        ,
        % considering pitch and roll limits
105     [isInside, pitch, roll] = insideVTW(targets_ECI{i}(t,:),
        Orbit_ECI(t,:), Vel_ECI(t,:), groundTrack_ECI(t,:), maxPitch,
        maxRoll);
        if isInside
            idx = [idx t];
            pitchAngles{i}(k) = pitch;
            rollAngles{i}(k) = roll;
110             k = k + 1;
        end
        VTW_raw{i} = timeHorizon(idx);
    end
end

115 % Reorder targets and write Files
target_ID = [];
for i = 1:size(targets,1)
    if ~isempty(VTW_raw{i})

```

```

120     target_ID = [target_ID i];
        end
    end

    [tstart, arranged] = sort(cellfun(@(v)v(1), VTW_raw(target_ID)));
125 target_ID = target_ID(arranged);
    tend = cellfun(@(v)v(end),VTW_raw(target_ID));
    tstart.Format = 'yyyy/M/dd H:mm:ss';
    tend.Format = 'yyyy/M/dd H:mm:ss';

130 header = {'Orbit Europe, number 1';
            ['Task ID, Target ID, profit, duration, Window start time,
            Window end Time']};

    profit = targets(target_ID,4);
    duration = targets(target_ID,5);
135 task_ID = sat_baseID + targets_baseID +target_ID;

    data = table(task_ID', target_ID', profit, duration, tstart', tend
                ');

    filename = strcat('./GeneratedData/Orbit', num2str(sat_baseID +
                targets_baseID),'.txt');
140 writelines(header, filename)
    writetable(data, filename, 'WriteMode', 'append');

    VTW = VTW_raw(target_ID);

145 % Building task files with angles
    header2 = {'MetaTask ID, Moment, Roll, Pitch'};
    for i = target_ID
        metaTask_ID = sat_baseID + targets_baseID + i;
        Moment = VTW_raw{i};
150 Moment.Format = 'yyyy/M/dd H:mm:ss';
        data2 = table(metaTask_ID*ones(size(Moment))', Moment',
            rollAngles{i}',pitchAngles{i}');
        writelines(header2, './GeneratedData/LookAngle/'+ metaTask_ID
            +".txt")
        writetable(data2, './GeneratedData/LookAngle/'+ metaTask_ID
            +".txt", 'WriteMode', 'append');
    end
155 filename = strcat('./data/mission', num2str(sat_baseID +
        targets_baseID),'.mat');
    save(filename)

```

Script C.1: Transforms LLA to VTW and generate mission files.

```

% Function to check the validity of a line of sight vector with
pitch and roll angle limits
function [inside, pitchAngle, rollAngle] = insideVTW(target, Orbit
, velocity, groundTrack, pitchLimit, rollLimit)

nadir = groundTrack - Orbit;
5 LOS = target - Orbit;
movDir = velocity/norm(velocity);

% Pitch
projPitch = dot(LOS,movDir);
10 pitchAngle = atand(projPitch/norm(nadir));

% Roll
dirRoll = cross(nadir,movDir);
dirRoll = dirRoll/norm(dirRoll);
15 projRoll = dot(LOS,dirRoll);
rollAngle = atand(projRoll/norm(nadir));

if abs(pitchAngle) <= pitchLimit && abs(rollAngle) <=
rollLimit && norm(LOS) < 2*norm(nadir)
inside = true;
20 else
inside = false;
end
end

```

Script C.2: Checks when the target is inside the S/C's Visible Time Window.

```

% Find roll maximum lines for a satellite orbit
function [leftLine_LLA, rightLine_LLA] = rollLines(maxRoll,
normalDir, Orbit_ECI, groundTrack_ECI, dateVector)
rE = 6.3781370e6;
5 shiftMag = tand(maxRoll)*(groundTrack_ECI - Orbit_ECI);
leftLine_ECI = groundTrack_ECI - norm(shiftMag)*normalDir;
leftDir = (-leftLine_ECI - Orbit_ECI + groundTrack_ECI)/norm(
leftLine_ECI - Orbit_ECI + groundTrack_ECI);
rightLine_ECI = groundTrack_ECI + norm(shiftMag)*normalDir;
rightDir = (-rightLine_ECI - Orbit_ECI + groundTrack_ECI)/norm
(rightLine_ECI - Orbit_ECI + groundTrack_ECI);

10 % Left
a = leftDir*leftDir';
b = 2*leftDir*Orbit_ECI';
c = Orbit_ECI*Orbit_ECI' - rE*rE;
d = b*b - 4*a*c;

15 if d >= 0
t = roots([a b c]);

```

```

    if abs(t(1)) <= abs(t(2)) && t(1) >= 0
        leftLine_ECI = Orbit_ECI + t(1)*leftDir;
    elseif abs(t(2)) <= abs(t(1)) && t(2) <= 0
        leftLine_ECI = Orbit_ECI + t(2)*leftDir;
    end
end

% Right
a = rightDir*rightDir';
b = 2*rightDir*Orbit_ECI';
c = Orbit_ECI*Orbit_ECI' - rE*rE;
d = b*b - 4*a*c;

if d >= 0
    t = roots([a b c]);
    if abs(t(1)) <= abs(t(2)) && t(1) >= 0
        rightLine_ECI = Orbit_ECI + t(1)*rightDir;
    elseif abs(t(2)) <= abs(t(1)) && t(2) <= 0
        rightLine_ECI = Orbit_ECI + t(2)*rightDir;
    end
end

leftLine_LLA = eci2lla(leftLine_ECI, dateVector, 'IAU
-2000/2006');
rightLine_LLA = eci2lla(rightLine_ECI, dateVector, 'IAU
-2000/2006');
end

```

Script C.3: Calculates max roll lines for a given ground track.

Bibliography

- [1] Gérard Verfaillie and Michel Lemaître. «Selecting and Scheduling Observations for Agile Satellites: Some Lessons from the Constraint Reasoning Community Point of View». In: (Nov. 2001), pp. 670–684. DOI: 10.1007/3-540-45578-7_55 (cit. on p. 1).
- [2] Xinwei Wang, Guohua Wu, Lining Xing, and Witold Pedrycz. «Agile Earth Observation Satellite Scheduling Over 20 Years: Formulations, Methods, and Future Directions». In: *IEEE Systems Journal* 15.3 (Sept. 2021), pp. 3881–3892. DOI: 10.1109/jsyst.2020.2997050. URL: <https://doi.org/10.1109/jsyst.2020.2997050> (cit. on pp. 1, 6, 15).
- [3] Daisuke Inoue and Hiroaki Yoshida. «Model Predictive Control for Finite Input Systems using the D-Wave Quantum Annealer». In: *Scientific Reports* 10 (2020). URL: <https://api.semanticscholar.org/CorpusID:209862375> (cit. on p. 3).
- [4] Kshitij Dave, B R Nikilesh, Amit Patel, Jitesh Lalwani, Babita Jajodia, and Mandaar Pande. «Efficient Earth Observation Satellites Mission Planning with Quantum Algorithm». In: (Apr. 2023). DOI: 10.13140/RG.2.2.33966.38729/1 (cit. on p. 3).
- [5] Vinicius Marchioli, Mattia Boggio, Deborah Volpe, Luca Massotti, and Carlo Novara. «Scheduling of Satellite Constellation Operations in EO Missions Using Quantum Optimization». In: *International Conference on Optimization, Learning Algorithms and Applications*. Springer. 2024, pp. 227–242 (cit. on p. 3).
- [6] Xinwei Wang, Chao Han, Rui Zhang, and Yi Gu. «Scheduling Multiple Agile Earth Observation Satellites for Oversubscribed Targets Using Complex Networks Theory». In: *IEEE Access* 7 (2019), pp. 110605–110615. DOI: 10.1109/ACCESS.2019.2925704 (cit. on pp. 6–9, 28, 31).
- [7] Xiaolu Liu, Gilbert Laporte, Yingwu Chen, and Renjie He. «An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time». In: *Comput. Oper. Res.* 86 (2017), pp. 41–

53. URL: <https://api.semanticscholar.org/CorpusID:13056845> (cit. on pp. 8, 31).
- [8] Guansheng Peng, Guopeng Song, Yongming He, Jing Yu, Shang Xiang, Lining Xing, and Pieter Vansteenwegen. «Solving the Agile Earth Observation Satellite Scheduling Problem With Time-Dependent Transition Times». In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.3 (2022), pp. 1614–1625. DOI: 10.1109/TSMC.2020.3031738 (cit. on p. 8).
- [9] Guansheng Peng, Guopeng Song, Lining Xing, Aldy Gunawan, and Pieter Vansteenwegen. «An Exact Algorithm for Agile Earth Observation Satellite Scheduling with Time-Dependent Profits». In: *Computers & Operations Research* 120 (Mar. 2020), p. 104946. DOI: 10.1016/j.cor.2020.104946 (cit. on pp. 8, 9, 31).
- [10] Jianjiang Wang, Erik Demeulemeester, Xuejun Hu, Dishan Qiu, and Jin Liu. «Exact and Heuristic Scheduling Algorithms for Multiple Earth Observation Satellites Under Uncertainties of Clouds». In: *IEEE Systems Journal* 13.3 (2019), pp. 3556–3567. DOI: 10.1109/JSYST.2018.2874223 (cit. on pp. 8, 15).
- [11] Doo-Hyun Cho, Jun-Hong Kim, Han-Lim Choi, and Jaemyung Ahn. «Optimization-Based Scheduling Method for Agile Earth-Observing Satellite Constellation». In: *Journal of Aerospace Information Systems* 15 (Sept. 2018), pp. 1–16. DOI: 10.2514/1.I010620 (cit. on p. 8).
- [12] Yi Gu, Chao Han, Yuhan Chen, Shenggang Liu, and Xinwei Wang 0006. «Large Region Targets Observation Scheduling by Multiple Satellites Using Resampling Particle Swarm Optimization». In: *IEEE Trans. Aerospace and Electronic Systems* 59.2 (Apr. 2023), pp. 1800–1815. DOI: 10.1109/TAES.2022.3205565. URL: <https://doi.org/10.1109/TAES.2022.3205565> (cit. on p. 8).
- [13] Fred Glover, Gary Kochenberger, and Yu Du. «Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models». In: *4OR* 17.4 (Dec. 2019), pp. 335–371. DOI: 10.1007/s10288-019-00424-. URL: https://ideas.repec.org/a/spr/aqjoor/v17y2019i4d10.1007_s10288-019-00424-y.html (cit. on p. 9).
- [14] Serge Rainjonneau et al. «Quantum Algorithms Applied to Satellite Mission Planning for Earth Observation». In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 16 (2023), pp. 7062–7075. DOI: 10.1109/JSTARS.2023.3287154 (cit. on p. 15).

-
- [15] Guohua Wu, Qizhang Luo, Xiao Du, Yingguo Chen, Ponnuthurai Nagarathnam Suganthan, and Xinwei Wang. «Ensemble of Metaheuristic and Exact Algorithm Based on the Divide-and-Conquer Framework for Multisatellite Observation Scheduling». In: *IEEE Transactions on Aerospace and Electronic Systems* 58.5 (2022), pp. 4396–4408. DOI: 10.1109/TAES.2022.3160993 (cit. on p. 15).
- [16] D-Wave Systems. *D-Wave Ocean Documentation - Tabu Sampler*. Accessed on: 2023-12-5. 2023. URL: <https://docs.ocean.dwavesys.com/projects/tabu/en/latest> (cit. on p. 16).
- [17] D-Wave Systems. *D-Wave Ocean Documentation - Simulated Annealing Sampler*. Accessed on: 2023-12-5. 2023. URL: https://docs.ocean.dwavesys.com/en/stable/docs_samplers/#simulated-annealing (cit. on p. 16).
- [18] Omid Ghasemalizadeh, Seyedmeysam Khaleghian, and Saied Taheri. «A Review of Optimization Techniques in Artificial Networks». In: *International Journal of Advanced Research* 4.9 (Sept. 2016), pp. 1668–1686. ISSN: 2320-5407. DOI: 10.21474/IJAR01/1627. URL: <http://www.journalijar.com> (cit. on p. 16).
- [19] Gintaras Palubeckis. «Multistart Tabu Search Strategies for the Unconstrained Binary Quadratic Optimization Problem». In: *Annals of Operations Research* 131 (2004), pp. 259–282. URL: <https://api.semanticscholar.org/CorpusID:207727800> (cit. on p. 17).
- [20] Catherine C. McGeoch. «Quantum Annealing». In: *Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice*. Cham: Springer International Publishing, 2014, pp. 29–42. ISBN: 978-3-031-02518-1. DOI: 10.1007/978-3-031-02518-1_3. URL: https://doi.org/10.1007/978-3-031-02518-1_3 (cit. on p. 20).
- [21] Sheir Yarkoni, Elena Raponi, Thomas Bäck, and Sebastian Schmitt. «Quantum annealing for industry applications: introduction and review». In: *Reports on Progress in Physics* 85.10 (Sept. 2022), p. 104001. ISSN: 1361-6633. DOI: 10.1088/1361-6633/ac8c54. URL: <http://dx.doi.org/10.1088/1361-6633/ac8c54> (cit. on p. 20).
- [22] D-Wave Systems. *D-Wave Ocean Documentation - What is Quantum Annealing?* Accessed on: 2023-12-10. 2023. URL: https://docs.dwavesys.com/docs/latest/c_gs_2.html (cit. on pp. 21, 22).
- [23] D-Wave Systems. *D-Wave Ocean Documentation - Samplers*. Accessed on: 2023-11-30. 2020. URL: <https://docs.ocean.dwavesys.com/projects/system/en/stable/reference/samplers.html> (cit. on p. 22).

BIBLIOGRAPHY

- [24] *Hybrid Solver for Constrained Quadratic Models*. Tech. rep. Whitepaper. D-Wave Systems, 2021. URL: https://www.dwavesys.com/media/rldh2ghw/14-1055a-a_hybrid_solver_for_constrained_quadratic_models.pdf (cit. on p. 24).
- [25] The MathWorks Inc. *Aerospace Toolbox version 23.2 (R2023b)*. Natick, Massachusetts, United States, 2023. URL: <https://www.mathworks.com> (cit. on p. 27).