

Quantum Computing for Earth Observation
Machine Definition Report

<i>Written by</i>	<i>Responsibility</i> + handwritten signature if no electronic workflow tool
<i>INFN Team</i>	Prof. Simone Montangero, Dr. Ilaria Siloi, Marco Tesoro
<i>Verified by</i>	
<i>TASF Team</i>	
<i>FZJ Team</i>	
<i>Approved by</i>	

Approval evidence is kept within the document management system

Change Records

ISSUE	DATE	§ CHANGE RECORDS	AUTHOR
1	06/07/2023	First issue	INFN Team
2	21/09/2023	Second issue	INFN Team
3	22/09/2023	Second issue, minor corrections	TASF Team
4	25/09/2023	Third issue	INFN Team
5	18/11/2023	Third issue, minor corrections	FZJ Team

Table of contents

1	Scope of the document and terminology.....	5
2	Bibliography.....	5
2.1	Applicable documents	8
2.2	Acronyms.....	8
3	Perimeter of the analysis	9
4	Machine sizing description	11
5	List of use cases	12
6	Use cases definition and sizing	12
6.1	Use case n°1: Mission planning problem (TASI, TASF, INFN, FZJ)	12
6.1.1	<i>Problem instance</i>	12
6.1.2	<i>Quantum algorithm</i>	16
6.1.3	<i>Quantum platforms</i>	19
6.1.4	<i>Quantum instance</i>	20
6.1.5	<i>Number of qubits</i>	27
6.1.6	<i>KPIs for quantum simulators</i>	29
6.1.7	<i>KPIs for digital quantum computers</i>	30
6.1.8	<i>Tensor Network (TN) approaches</i>	30
6.1.9	<i>Proposal of targeted quantum platforms</i>	32
6.2	Use case n°2: Multiple-view geometry on optical images (TASF).....	33
6.2.1	<i>Problem instance</i>	33
6.2.2	<i>Quantum algorithm</i>	34
6.2.3	<i>Quantum instance</i>	35
6.2.4	<i>Quantum platforms</i>	37
6.2.5	<i>Number of qubits</i>	37
6.2.6	<i>KPIs for digital quantum computers</i>	38
6.2.7	<i>Proposal of targeted quantum platforms</i>	40
6.3	Use case n°3: Optical satellite data analysis (FZJ).....	41
6.3.1	<i>Problem instance</i>	41
6.3.2	<i>Quantum algorithm</i>	41
6.3.3	<i>Quantum instance</i>	42
6.3.4	<i>Quantum platforms</i>	42
6.3.5	<i>Number of qubits</i>	42
6.3.6	<i>KPIs for digital quantum computing</i>	42
6.3.7	<i>Proposal of targeted quantum platforms</i>	43
6.4	Use case n°4: QFT for SAR data processing (FZJ).....	43
6.4.1	<i>Problem instance</i>	43
6.4.2	<i>Quantum algorithm</i>	45
6.4.3	<i>Quantum instance</i>	45
6.4.4	<i>Quantum platforms</i>	46
6.4.5	<i>Number of qubits</i>	46
6.4.6	<i>KPIs for digital quantum computing</i>	46
6.4.7	<i>Proposal of targeted quantum platforms</i>	46

7 Conclusions.....	47
END OF DOCUMENT	48

1 Scope of the document and terminology

Beginning with the compilation of Use Cases (UCs) identified in Work Package (WP) 1, the primary aim of this document is to provide an initial quantitative evaluation of the quantum computers and simulators required for each specific UC. This effort aligns with the overarching objective of the QC4EO project, which seeks to create a roadmap delineating the timeframe during which Quantum Computing (QC) can deliver advantages in addressing critical challenges within the field of Earth Observation (EO).

The consortium methodology involves the formulation of a "template" wherein each partner contributes their expertise to determine the sizing requirements for each UC. The sizing process begins by instantiating the classical problem, considering the number of variables and parameter ranges. Subsequently, the sizing proceeds by formulating an appropriate quantum instance that encompasses the necessary number of qubits and other hardware resources for the efficient execution of quantum algorithms. The assessment of machine sizing involves estimating resources such as the number of qubits, their connectivity, error rates, coherence times, and other relevant key performance indicators. This document presents a selection of targeted platforms, which will undergo further refinement in subsequent WPs.

The structure of this document is as follows:

- Chapter 2 provides a comprehensive bibliography encompassing all the information utilized in evaluating the various scenarios.
- Chapter 3 elucidates the adopted methodology employed to define the quantum instance within the framework of machine sizing.
- Chapter 4 establishes the template employed for the sizing process for each UC.
- Chapter 5 comprises the inventory of UCs alongside references to their corresponding mathematical descriptions as refined in the deliverable of WP1.
- Chapter 6 represents the primary output of this document, presenting the machine sizing particulars for each UC.
- Chapter 7 presents a concise summary table incorporating the sizing outcomes and the targeted platforms.

2 Bibliography

(n.d.). Retrieved from <https://ionq.com/quantum-systems/forte>

(n.d.). Retrieved from <https://quantum-computing.ibm.com/services/resources/docs/resources/manage/systems/processors>

Albash, T., & Lidar, D. (2018, January 29). Adiabatic quantum computation. *Reviews of Modern Physics*, 90, 15002. doi:10.1103/revmodphys.90.015002

Allcock, J., Hsieh, C.-Y., Kerenidis, I., & Zhang, S. (2020, October 2). Quantum Algorithms for Feedforward Neural Networks. *ACM Transactions on Quantum Computing*, 1, 1-24. doi:10.1145/3411466

- Apolloni, B., & Falco, D. (1988). The clock of a quantum computer. *Journal of Physics A: Mathematical and General*, 35, 10033-10051. doi:10.1088/0305-4470/35/47/305
- Barahona, F. (1982, October 1). On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15, 3241-3253. doi:10.1088/0305-4470/15/10/028
- Barry, M., McCoy, T., & Wu. (2013). TWO DIMENSIONAL ISING MODEL. In *Quantitative Theory of Critical Phenomena* (pp. 91-104). Elsevier. doi:10.1016/b978-0-12-075120-4.50010-5
- Bartkiewicz, K., Cernoch, A., & et al. (2020). Experimental kernel-based quantum machine learning in finite feature space. *Sci. Rep.*, 12356.
- Benedetti, M., Lloyd, E., Sack, S., & Fiorentini, M. (2019, November 1). Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4, 043001. doi:10.1088/2058-9565/ab4eb5
- Born, M., & Fock, V. (1928, March). Beweis des Adiabatenatzes. *Zeitschrift f_r Physik*, 51, 165-180. doi:10.1007/bf01343193
- Cavinato, S., & et al. (2021). Optimizing Radiotherapy Plans for Cancer Treatment with Tensor Networks. *Physics in Medicine & Biology*, 125015.
- Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L., & Coles, P. (2022, September 15). Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2, 567-576. doi:10.1038/s43588-022-00311-3
- Cruz, H., Véstias, M., Monteiro, J., Neto, H., & Duarte, R. (2022). A Review of Synthetic-Aperture Radar Image Formation Algorithms and Implementations: A Computational Perspective. *Remote Sensing*.
- Cumming, I., & Wong, F. (2005). Processing of Synthetic Aperture Radar Data: Algorithms and Implementation. *Artech House: Norwood*.
- Das. (n.d.).
- Das, A., & Chakrabarti, B. (2008, September 5). Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80, 1061-1081. doi:10.1103/revmodphys.80.1061
- Dave, K. (2023). Efficient Earth Observation Satellites Mission Planning with Quantum Algorithm. *Efficient Earth Observation Satellites Mission Planning with Quantum Algorithm*, 2023.
- Djidjev, H. N. (n.d.). Quantum annealing with inequality constraints: the set cover problem. *arXiv:2302.11185*.
- Ebadi, S., Wang, T., Levine, H., Keesling, A., Semeghini, G., Omran, A., . . . Lukin, M. (2021, July 7). Quantum phases of matter on a 256-atom programmable quantum simulator. *Nature*, 595, 227-232. doi:10.1038/s41586-021-03582-4
- Eisert, J. (n.d.). Entanglement and tensor network states. *arXiv:1308.3318*, 2013.
- Fahri, E. e. (2014). A quantum approximate optimization algorithm. *arXiv:1411.4028*.
- Farhi, E., Goldstone, J., Gutmann, S., & Zhou, L. (2014). The Quantum Approximate Optimization Algorithm and the Sherrington-Kirkpatrick Model at Infinite Size. *Quantum*, 6, 759. doi:10.22331/q-2022-07-07-759
- Felser, T. (2021). *Tree tensor networks for high-dimensional quantum systems and beyond*. PhD Thesis.
- Gardas, B., & Deffner, S. (2018, November 21). Quantum fluctuation theorem for error diagnostics in quantum annealers. *Scientific Reports*, 8, 17191. doi:10.1038/s41598-018-35264-z
- Giovannetti, V., Lloyd, S., & Maccone, L. (2008, April 21). Quantum Random Access Memory. *Physical Review Letters*, 100, 160501. doi:10.1103/physrevlett.100.160501
- Glover, F., Kochenberger, G., & Du, Y. (2019). Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models. *4OR-Q J Oper Res* , 335-371.
- Gujju, Y., Matsuo, A., & Raymond, R. (n.d.). Quantum Machine Learning on Near-Term Quantum Devices: Current State of Supervised and Unsupervised Techniques for Real-World Applications. *arXiv:2307.00908v1*.

- Hadfield, S., Wang, Z., O'gorman, B., Rieffel, E., Venturelli, D., & Biswas, R. (2019, February 12). From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *Algorithms*, 12, 34. doi:10.3390/a12020034
- Havlíček, V., Córcoles, A., Temme, K., Harrow, A., Kandala, A., Chow, J., & Gambetta, J. (2019, March). Supervised learning with quantum-enhanced feature spaces. *Nature*, 567, 209-212. doi:10.1038/s41586-019-0980-2
- Henderson, M., Shukla, S., Pradhan, S., & Cook, T. (2020, February 27). Quantum neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence*, 2, 2. doi:10.1007/s42484-020-00012-y
- Hubregtse, T., Wierichs, D., Gil-Fuster, E., Derks, P.-J. H., Faehrmann, P., & Meyer, J. (2022, October 20). Training quantum embedding kernels on near-term quantum computers. *Physical Review A*, 106, 42431. doi:10.1103/physreva.106.042431
- Johri, S., Debnath, S., Mocherla, A., Singk, A., Prakash, A., Kim, J., & Kerenidis, I. (2021, August 5). Nearest centroid classification on a trapped ion quantum computer. *npj Quantum Information*, 7, 122. doi:10.1038/s41534-021-00456-5
- Kerenidis, I., Landman, J., & Prakash, A. (2019). *Quantum algorithms for deep convolutional neural networks*. Tech. rep.
- Lanting, T., Przybysz, A., Smirnov, A., Spedalieri, F., Amin, M., Berkley, A., . . . Rose, G. (2014, May 29). Entanglement in a Quantum Annealing Processor. *Physical Review X*, 4, 21041. doi:10.1103/physrevx.4.021041
- Lechner, W., Hauke, P., & Zoller, P. (2015). A quantum annealing architecture with all-to-all connectivity from local interactions. *Science Advances*, 1. doi:10.1126/sciadv.1500838arXiv
- Lucas, A. (2014). Ising formulations of many NP problems. *Frontiers in Physics*, 2. doi:10.3389/fphy.2014.00005
- Miroszewski, A. (2023). *Detecting Clouds in Multispectral Satellite Images Using Quantum-Kernel Support Vector Machines*. Tech. rep.
- Mitarai, K., Negoro, M., Kitagawa, M., & Fujii, K. (2018, September 10). Quantum circuit learning. *Physical Review A*, 98, 32309. doi:10.1103/physreva.98.032309
- Montangero, S. (2018). *Introduction to Tensor Network Methods*. Springer.
- Musk, D., & Sullivan, F. (2020). A Comparison of Quantum and Traditional Fourier Transform Computations. *Comput Sci Eng*, 103-110.
- Nielsen, M., & Chuang, I. (2010). *Quantum Computation and Quantum Information*. Cambridge.
- Orús, R. (2019). Tensor networks for complex quantum systems. *Nature Review Physics*.
- Ozfidan, I., Deng, C., Smirnov, A., Lanting, T., Harris, R., Swenson, L., . . . Amin, M. (2020, March 16). Demonstration of a Nonstoquastic Hamiltonian in Coupled Superconducting Flux Qubits. *Physical Review Applied*, 13, 34037. doi:10.1103/physrevapplied.13.034037
- Piatkowski, N., Gerlach, T., Hugues, R., Sifa, R., Bauckhage, C., & Barbaresco, F. (2022, July 4). Towards Bundle Adjustment for Satellite Imaging via Quantum Machine Learning. In *2022 25th International Conference on Information Fusion (FUSION)*. IEEE. doi:10.23919/fusion49751.2022.9841388
- Rainjonneau, S. (2023). *Quantum algorithms applied to satellite mission planning for Earth observation*. Tech. rep.
- Razavy, M. (2003). Quantum Theory of Tunneling. *Quantum Theory of Tunneling*. WORLD SCIENTIFIC. doi:10.1142/8901
- Scholl, P., Schuler, M., Williams, H., Eberharter, A., Barredo, D., Schymik, K.-N., . . . Browaeys, A. (2021, July 1). Quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms. *Nature*, 595, 233-238. doi:10.1038/s41586-021-03585-1

- Schuld, M., & Petruccione, F. (2021). *Quantum Models as Kernel Methods*. Springer International Publishing. doi:10.1007/978-3-030-83098-4_6
- Schuld, M., Bocharov, A., Svore, K., & Wiebe, N. (2020, March 6). Circuit-centric quantum classifiers. *Physical Review A*, *101*, 32308. doi:10.1103/physreva.101.032308
- Schuld, M., Sinayskiy, I., & Petruccione, F. (2015). An introduction to quantum machine learning. *Contemporary Physics*, *56*, 172-185. doi:10.1080/00107514.2014.964942
- Schuld, M., Sweke, R., & Meyer, J. (2021, March 24). Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, *103*, 32430. doi:10.1103/physreva.103.032430
- Silvi, P., Tschirsich, F., Gerster, M., Jüneman, J., Jaschke, D., Rizzi, M., & Montangero, S. (2019). The Tensor Networks Anthology: Simulation techniques for many-body quantum lattice systems. *SciPost*.
- Sim, S., Johnson, P., & Aspuru-guzik, A. (2019, October 14). Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. *Advanced Quantum Technologies*, *2*, 1900070. doi:10.1002/qute.201900070
- Stollenwerk, T. e. (2021). Agile earth observation satellite scheduling with a quantum annealer. *IEEE Transactions on Aerospace and Electronic Systems*, 3520-3528.
- Vasil, S., Denchev, S., Boixo, S., Isakov, N., Ding, R., Babbush, V., . . . Neven. (2016, August). What is the Computational Value of Finite-Range Tunneling? *Phys. Rev. X*, *6*, 31015. doi:10.1103/PhysRevX.6.031015
- Verma, A., & Lewis, M. (n.d.). Variable Reduction For Quadratic Unconstrained Binary Optimization. *arXiv:2105.07032*.
- Wurtz, J., Lopes, P., Gemelke, N., Keesling, A., & Wang, S. (2022). In *Industry applications of neutral-atom quantum computing solving independent set problems*.
- Zhang, G., Li, X., Hu, G., Zhan, Z., An, J., & Man, W. (2021). Mission Planning Issues of Imaging Satellites: Summary, Discussion, and Prospects. *International Journal of Aerospace Engineering*, 7819105.

2.1 Applicable documents

- [AD-1] QC4EO Study Statement of Work
[AD-2] Proposal submitted for QC4EO
[WP1-del] Deliverable of WP1

2.2 Acronyms

AQC	Adiabatic Quantum Computation
AR	Acquisition Request
BAQ	Block Adaptive Quantization
DFT	Discrete Fourier Transform
DLO	Downlink Opportunity
DTO	Data Take Opportunity
EO	Earth Observation
FFT	Fast Fourier Transform
InSAR	Interferometric SAR

INFN	Istituto Nazionale di Fisica Nucleare
IFFT	Inverse Fast Fourier Transform
FZJ	Forschungszentrum Jülich
LULC	Land Use Land Cover Classification
LHZ	Lechner-Hauke-Zoller
MCF	Minimum Cost Flow
ML	Machine Learning
NISQ	Noisy Intermediate-Scale quantum
PCA	Principal Component Analysis
PPO	Proximal Policy Optimization
PQC	Parameterized Quantum Circuits
PRF	Pulse Repetition Frequency
QA	Quantum Annealing
QAOA	Quantum Approximate Optimization Algorithm
QC4EO	Quantum Computing for Earth Observation
QFT	Quantum Fourier Transform
QKE	Quantum Kernel Estimation
QML	Quantum Machine Learning
QNN	Quantum Neural Network
QPU	Quantum Processing Unit
qRAM	quantum Random Access Memory
QUBO	Quadratic Unconstrained Binary Optimization
RCMC	Range Cell Migration Correction
RDA	Range Doppler Algorithm
RL	Reinforcement Learning
RSF	Range Sample Frequency
SAR	Synthetic Aperture Radar
SIFT	Scale Invariant Feature Transform
SotA	State-of-the-Art
SVM	Support Vector Machine
TASI	Thales Alenia Space Italy
TASF	Thales Alenia Space France
TN	Tensor Network
TXPL	Transmitted Pulse Length
TTN	Tree Tensor Network
TWMP	Tree Weighted Message Passing
UC	Use Case
VHR	Very High Resolution
WP	Work Package

3 Perimeter of the analysis

In order to explore the potential applications of quantum computing in the Earth Observation (EO) field, we have identified four specific cases during the first work package (WP1) that cover all stages of a satellite-based EO mission, including mission analysis, data acquisition, processing, and data analysis. This study focuses on comparing quantum solutions to these identified problems in relation to the current

approaches employed by the consortium's companies or how the researchers involved have benchmarked their algorithms.

In this document, we initially present the mathematical models for each problem and provide the sizing of classical instances in terms of the number of variables and parameter ranges. These instances reflect real-world problems commonly addressed using standard classical approaches. Subsequently, we analyse the viable quantum algorithms, including hybrid and quantum-inspired ones, for each computational instance. We also provide a brief overview of the quantum devices that can currently be employed to tackle these problems, encompassing both quantum simulators and general-purpose quantum computers.

The process of identifying the potential optimal quantum computing architecture(s) within the given reference timeframe (3-5 to 15 years) involves evaluating the sizing of these architectures using Key Performance Indicators (KPIs). These KPIs must effectively differentiate between the chosen quantum platforms while considering the problem level of complexity. For this reason, we define different quantities for quantum simulators as opposed to quantum computers. The outcome of this work will be a summary of the relevant machine sizing information for the selected UCs, accompanied by a proposal of the targeted quantum platform(s) for each use case.

4 Machine sizing description

This section details the information utilized in determining the machine sizing for each of the identified use cases:

Problem instance: This refers to a comprehensive representation of the variables and parameters necessary for the classical implementation of the problem in real-world scenarios, considering its actual size.

Quantum algorithms: A detailed description of the selected quantum algorithm(s) employed for solving the problem.

Quantum instance: In the context of machine sizing for quantum computing, a quantum instance refers to the mathematical formulation of a quantum problem, encompassing its parameters and variables. This quantum instance comprises various elements, including the number of qubits involved, their connectivity or couplings, the specific quantum gates utilized, as well as the initialization procedures and measurement operations employed for executing the quantum algorithm.

Quantum platforms: These denote the actual devices or platforms where the algorithm can currently be implemented.

Number of qubits: The number of qubits required to represent the problem and execute the algorithm at a real-scale level. This involves understanding the problem size and complexity and mapping it to the appropriate qubit representation.

Key Performance Indicators (KPIs) for digital quantum computing: This encompasses metrics such as circuit depth, the order of magnitude for 2-qubit gates, and the order of magnitude for measurements (or Pauli terms).

KPIs for quantum simulators: These include factors like connectivity and the local/global addressing of qubits within the simulator.

Tensor Network (TN) approaches: This entails the contribution of quantum-inspired methods based on tensor network representations.

Proposal of the targeted quantum platform: This outlines the rationale behind the selection of a specific quantum platform for the next 15 years. It is crucial to consider that available platforms may have limitations and constraints, such as restricted qubit connectivity, a limited gate set, or limitations on measurement capabilities. The proposal for the quantum platform must account for these constraints and leverage the choice based on the available roadmaps.

5 List of use cases

The following table lists the use cases which have been already selected and analyzed in the report delivered by WP1. References to the corresponding mathematical formulations are provided.

SCENARIO NUMBER	USE CASE TITLE	MATHEMATICAL DESCRIPTION
1	Mission Planning for EO Acquisitions	See Section 5.1 [WP1-del]
2	Multiple-view Geometry on Optical Images	See Section 5.2 [WP1-del]
3	Optical Satellite Data Analysis	See Section 5.3 [WP1-del]
4	SAR Raw Data Processing	See Section 5.4 [WP1-del]

Table 1 List of selected use cases.

6 Use cases definition and sizing

This chapter contains the machine sizing for each UC according to the criteria previously defined. Starting from the mathematical formulation of the problem, we first provide the range of variables and parameters expected for realistic-size problems solved with classical techniques. To proceed with the sizing first, we describe the selected quantum algorithm and then, define the corresponding quantum instance and the quantum platforms now available to address the specific problem. With this information, we proceed with the sizing following the KPIs selected above. Based on this evaluation, each use case study proposes a target quantum platform aiming to provide an effective solution in the next 15 years. If possible different algorithms will be considered as well as the corresponding sizing. This analysis will be deepened in the next WP3.

6.1 Use case n°1: Mission Planning for EO Acquisitions (TASI, TASF, INFN, FZJ)

6.1.1 Problem instance

Mission planning is a crucial aspect of Earth Observation (EO) as it involves optimizing and scheduling the acquisition requests from end-users. The current methods employed to address this problem utilize deterministic and metaheuristic algorithms, which can generate (optimal) solutions for existing constellations consisting of a few satellites (Zhang , et al., 2021). However, with the growing deployment of large constellations comprising numerous small satellites ($N > 100$), the search for optimal solutions becomes more challenging in terms of both time and quality. Consequently, the concept of quantum speedup takes into account not only the computational time required but also the quality of the resulting solution.

The mission planning problem deals with the optimal scheduling of satellite observations for a given list of user requests. This optimization problem can be written in terms of a knapsack problem. For each satellite and for each “Acquisition Request” (AR) there are several “data take opportunities” (DTOs) and “downlink opportunities” (DLOs): the optimizer must return a time ordered short-list of those

observations and downlinks that are possible when several constraints are considered. We report here the cost function defined in the documents released by WP1:

$$F(\vec{x}, \vec{y}) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^{\theta_{i,j}} \sum_{r=1}^{\sigma_i} \left(\alpha t_{i,j}^k x_{i,j}^k + \beta s_{i,j}^r y_{i,j}^r - \frac{\gamma}{m} x_{i,j}^k \right)$$

where the decision variables $x_{i,j}^k \in \{0, 1\}$ and $y_{i,j}^r \in \{0, 1\}$ are binary variables describing the DTOs and the DLOs respectively: if the AR j is scheduled by satellite i at the opportunity k , then $x_{i,j}^k = 1$; otherwise $x_{i,j}^k = 0$, and similarly for the DLO variables $y_{i,j}^r$. The following parameters appear in the above cost function as inputs to the optimizer:

- $n \in \mathbb{N}$ the number of satellites;
- $m \in \mathbb{N}$ the number of ARs;
- $\theta_{i,j} \in \mathbb{N}$ the number of opportunities of satellite i w.r.t. the AR j ;
- $\sigma_i \in \mathbb{N}$ the number of available downlink opportunities of satellite i ;
- $t_{i,j}^k \in \mathbb{R}$ the acquisition time of the DTO of the AR j from satellite i at the opportunity k ;
- $s_{i,j}^r \in \mathbb{R}$ the acquisition time of the DLO of the AR j from satellite i at the opportunity r ;
- $\alpha, \beta, \gamma \in \mathbb{R}$ weights associated with the problem.

The goal in this scenario is to minimize $F(\vec{x}, \vec{y})$ subject to the following constraints:

- 1) One acquisition attempt per target:

$$\sum_{i=1}^n \sum_{k=1}^{\theta_{i,j}} x_{i,j}^k - 1 = 0 \quad \forall j \in [1, m]$$

- 2) One download attempt per target:

$$\sum_{i=1}^n \sum_{r=1}^{\sigma_i} y_{i,j}^r - 1 = 0 \quad \forall j \in [1, m]$$

- 3) Each acquisition must be downloaded:

$$\sum_{k=1}^{\theta_{i,j}} x_{i,j}^k - \sum_{r=1}^{\sigma_i} y_{i,j}^r = 0 \quad \forall i \in [1, n], \quad \forall j \in [1, m]$$

- 4) Target acquisition precede download:

$$\sum_{k=1}^{\theta_{i,j}} x_{i,j}^k t_{i,j}^k - \sum_{r=1}^{\sigma_i} y_{i,j}^r s_{i,j}^r \leq 0 \quad \forall i \in [1, n], \quad \forall j \in [1, m]$$

- 5) Sufficient preparation time between two acquisitions:

$$x_{i,s}^k p_{i,s}^k - |x_{i,s}^k t_{i,s}^k - x_{i,j}^{k'} t_{i,j}^{k'}| \leq 0 \quad \forall i \in [1, n], \quad \forall j, s \in [1, m] \\ \forall k \in [1, \theta_{i,s}], \quad \forall k' \in [1, \theta_{i,j}]$$

6) Sufficient preparation time between two downlinks:

$$y_{i,g}^r d_{i,g}^r - |y_{i,g}^r s_{i,g}^r - y_{i,j}^{r'} s_{i,j}^{r'}| \leq 0 \quad \forall i \in [1, n], \quad \forall j, g \in [1, m]$$

$$\forall r \in [1, \sigma_i], \quad \forall r' \in [1, \sigma_i]$$

7) Available memory on board:

$$\sum_{j=1}^m (\sum_{k=1}^{k'} x_{i,j}^k - \sum_{r=1}^{r'} y_{i,j}^r) q_{i,j} - q^M \leq 0 \quad \forall i \in [1, n], \quad \forall k' \in [1, \theta_{i,j}]$$

$$\forall r' \in [1, \sigma_i] : t_{i,j}^{k'} \leq s_{i,j}^{r'}$$

where other parameters are introduced:

- $p_{i,j}^k \in \mathbb{R}$ the preparation time to gain the DTO of the AR j from satellite i at the opportunity k ;
- $d_{i,j}^r \in \mathbb{R}$ the preparation time to perform the DLO of the AR j from satellite i at the opportunity r ;
- $q_{i,j} \in \mathbb{R}$ the needed memory to gain the AR j from satellite i ;
- $q^M \in \mathbb{R}$ the total memory available on satellite i .

Information on the realistic range of variables and parameters in the mathematical model are listed in the following tables.

PARAM ID	DESCRIPTION	DOMAIN	RANGE VALUES	NOTES
n	Number of satellites	\mathbb{N}	[1, 1000]	Average ~ 30
m	Number of ARs in the horizon time window	\mathbb{N}	[200, 50k]	- Horizon time window of 24 hrs. - 200 ARs per satellite. - Saturation of 50k ARs is independent of constellation size.
$\theta_{i,j}$	Number of DTOs of satellite i to acquire the AR j in the horizon time window	\mathbb{N}	[2, 50]	Horizon time window of 24 hrs.
σ_i	Number of DLOs of satellite i in the horizon time window	\mathbb{N}	[10, 100]	Horizon time window of 24 hrs.

Table 2 Mission planning problem variables and parameters.

PARAM ID	DESCRIPTION	DOMAIN	NOTES
Objective function variables			
$t_{i,j}^k$	Instant acquisition time of the $j - th$ target from the $i - th$ satellite at the $k - th$ opportunity	\mathbb{R}	This value includes the duration of acquisition: 2-15 min.
$s_{i,j}^r$	Instant downlink time of the $j - th$ target from the $i - th$ satellite at the $r - th$ opportunity	\mathbb{R}	This value includes duration of downlink dependent on the visibility of the GS and on the data-rate of DW.
Constraints variables			
$p_{i,j}^k$	Preparation time of the $i - th$ satellite to gain the $j - th$ target at the $k - th$ opportunity	\mathbb{R}	This parameter could consider reconfiguration and manoeuvre times: 10 sec – 60 sec.
$d_{i,j}^r$	Preparation downlink time of the $i - th$ satellite to gain the $j - th$ target at the $r - th$ opportunity	\mathbb{R}	This parameter could consider reconfiguration and manoeuvre times.
$q_{i,j}$	Required memory to gain the $j - th$ target from the $i - th$ satellite	\mathbb{R}	Memory size for 1 acquisition. SAR Raw product: 5-300 Gbit. On average ~ 40 Gb.
q_i^M	Total memory available on satellite i	\mathbb{R}	[10Gb, 1.5 Tb]. On average ~ 1000Gb.

Table 3 Mission planning problem variables and parameters

Other information from TASI will complete the problem instantiation:

1. Average revisit time (constellation): hourly
2. Average acquisition time (mean access duration) [m]: 2÷15
3. Downlink data rate [Mbit/s]: ~ 500

The problem instance of mission planning is addressed also in the framework of classical Reinforcement Learning (RL). We refer to Section 5.1.3.2 in [WP1-del] for the definition of the problem and its instantiation.

A second approach making use of Machine Learning (ML) to solve the optimization of the scheduling of satellites observations for a given list of requests can be considered. A RL formalism can be followed to associate to each satellite a maximum number of requests that can be acquired. In the following, a Proximal Policy Optimization (PPO) algorithm is used. For a given satellite, the state of the environment

is described by a subset of the total number of requests (N_r requests among the N_R in total) along with variables that define the completion status of each request, the reward (binary) variable and the satellite position at the given time. The policy network consists in a neural network whose initial neurons contain the following features for N_r requests:

- DTO start/end times;
- Current request start coordinates;
- Last request end coordinates;
- Satellite position;
- Satellite ID number.

The next step consists of the choice of the action taken by the agent. For the satellite considered, one of the nearest requests that can be completed is chosen (if possible), otherwise none is completed/chosen. The learning process is done by using gradient descent to calculate the expected value of each action at a certain state space and by selecting the action that has the likelihood of the highest reward. The equation for the PPO algorithm is given by:

$$L^{CLIP}(\theta) = \hat{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)]$$

that computes the current expected value of the policy of parameter θ and hyperparameter ϵ . \hat{A}_t is the estimated advantage proposed by the PPO at time t , and $r_t(\theta)$ the importance sampling ratio.

Finally, the state space is updated in such a way that the new N_r requests studied are ones that are yet to be completed.

The quantum contribution consists in adapting the policy model by integrating a parameterized quantum circuit to further reduce the number of features used and access potentially more interesting outputs before the action selection process.

6.1.2 Quantum algorithm

The mission planning problem is approached using two distinct strategies: one involves transforming the optimization problem into a QUBO problem, denoted as **UC1a**, while the other utilizes a QNN algorithm, denoted as **UC1b**. Both approaches are detailed in the following Sections.

6.1.2.1 UC1a: Quantum Annealing

Quantum annealing (QA) is an approach closely related to adiabatic quantum computation (AQC) (Das & Chakrabarti, 2008) (Albash & Lidar, 2018) proposed in 1988 by Apolloni et al. (Apolloni & Falco, 1988). It relaxes the requirement of adiabaticity, resulting in a heuristic variational quantum algorithm. For this reason, QA can be used, for instance, to find the ground state of Ising models (Barry, McCoy, & Wu, 2013), a known NP-hard task (Barahona, 1982), even though not necessarily in polynomial time. It is well-established in literature how to transform canonical NP-hard and NP-complete combinatorial optimization problems into forms suitable for quantum annealers (Lucas, 2014). Currently, state-of-the-art QA hardware does not allow for universal quantum computation. Nonetheless, it has been shown that increased control, e.g., non-stoquastic Hamiltonian terms (Ozfidan, et al., 2020) of the annealing path can significantly improve the quality of results obtained while still in the absence of universality.

The quantum algorithm first prepares the system in the ground state of the initial Hamiltonian, H_i , which is known and easy to prepare. Then, the system evolves according to a time-dependent Hamiltonian that slowly changes in time such that the contribution of H_i is slowly reduced while the magnitude of the target Hamiltonian, H_f , encoding the optimization problem, is increased. The so-called annealing schedule is defined by the monotonic functions $A(t)$ and $B(t)$, with $A(0) = 1, B(0) = 0$, and $A(T) = 0, B(T) = 1$, and $t \in [0, T]$. Typically, a transverse field in x -direction is used as the initial Hamiltonian $H_i = \sum_i \sigma_{i,x}$, where $\sigma_{i,x}$ is the x -Pauli matrix acting on the i -th qubit. Initially, the system is prepared in the ground state $|\Psi(0)\rangle = |+\dots+\rangle$ with $|+\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$. Then, the system evolves towards the ground state $|\Psi(T)\rangle$ of the final Hamiltonian, $H_f = \sum_i h_i \sigma_{i,z} + \sum_{\langle i,j \rangle} J_{ij} \sigma_{i,z} \sigma_{j,z}$, that encodes the solution of the optimization problem. As the magnitude of H_i decreases, the quantum dynamics of the qubits slow down until at $t = T$, when the system reaches a purely classical state. Finally, qubits are measured.

Maintaining the condition of adiabaticity in practical settings poses challenges due to the presence of background noise and thermal fluctuations. Consequently, the quantum system may be perturbed by the environment, causing it to deviate from its ground state. Additionally, achieving the condition of slowly varying the Hamiltonian evolution is demanding, as the rate of change relies on the low-energy states and spectral gaps of $H(t)$, which are not known beforehand. For these reasons, QA can be regarded as a relaxation of AQC wherein the annealing schedule, determining the transition time from H_i to H_f , is heuristically determined, then relaxing adherence to adiabatic conditions. As a result, instead of a deterministic quantum algorithm, QA yields a heuristic optimization algorithm in which a non-zero probability of remaining in the ground state is maintained throughout the annealing process.

QA can be viewed to implement a very similar logic to simulated annealing, but where the thermal fluctuations are replaced by quantum fluctuations. In this setting, the search algorithm gains the ability to escape local minima with superposition and quantum tunneling thus allowing for a direct transition between states even if there is a high energy barrier between them (Razavy, 2003). These effects have been successfully demonstrated in simple experiments (Lanting, et al., 2014), while it is not clear how these properties translate to an important role in the optimization process in large ensembles of qubits (Vasil, et al., 2016).

QA methods have been tested for numerous real-world applications, from traffic flow optimization and job-shop scheduling to machine learning, portfolio optimization, protein folding, molecular similarity, and computational biology. The solution strategy involves formulating QUBO problems, which usually imply fully connected interactions between Ising variables. The mapping from these dense logical graphs to the sparse connectivity in QPUs prevents efficient computation in the current generation of quantum annealers. This is probably the most crucial and common bottleneck of QA for real-world applications.

Quantum annealing (QA) has also been examined in the context of Earth Observation (EO) applications. Specifically, the focus has primarily been on the mission planning problem, with a particular emphasis on scenarios involving a single satellite. Stollenwerk et al. (Stollenwerk, 2021) explored the challenge of acquiring high-value images while adhering to the satellite maneuvering constraints. In this case, the solution by the quantum annealer is comparable to the operational heuristic method for small-scale problem instances. However, this quality deteriorates rapidly due to the limited precision of the quantum annealer. On the other hand, Dave et al. (Dave, 2023) optimized

the mission schedule with QA and obtained enhanced performance compared to simulated annealing and the Gurobi optimizer in selecting high-priority targets while satisfying all imposed constraints.

These studies illustrate the exploration of quantum algorithms in EO mission planning, showcasing their potential benefits and highlighting areas for improvement in terms of solution quality and scalability.

6.1.2.2 UC1a: gate-based quantum computing algorithm for optimization

In addition to quantum annealing the mission planning problem can be solved in principle with gate-based quantum computers. For fault tolerant quantum computers, Grover's algorithm can be used, leading to a quadratic speedup in the best-case scenario. For NISQ devices, the quantum approximate optimization algorithm (QAOA) (Fahri, 2014) can be used to find approximate solutions. The algorithm can be derived from QA by discretizing the time evolution which yield an alternating pattern of operations derived from the cost function and the initial (or driver) Hamiltonian. A potential quantum advantage of this algorithm is still subject to active research but can only be quadratic at most. Similar to quantum annealing a QUBO formulation is needed in the default version of QAOA. Therefore, sophisticated constraints as they appear in real-world problems like mission planning, lead to significant demands for the quantum resources, with potential negative impact onto the performance. An alternative approach is the Quantum Alternating Operator Ansatz (Hadfield, et al., 2019) which incorporates the constraints directly into the algorithm, by using tailored driver Hamiltonians that are symmetric with respect to the constraints.

6.1.2.3 UC1b: Quantum Machine Learning

Quantum Machine Learning (QML) is an approach in which expensive subroutines of classical machine learning algorithms are adapted to run on a quantum computer (Schuld, Sinayskiy, & Petruccione, 2015). Typically, a QML algorithm establishes an input-output relation from a given set of data (classical or quantum) in order to predict the output from the initial data of interest. The learning process can be done in a supervised manner, in which case the training phase of the algorithm is executed with a set of input data whose outputs are known a priori; or in an unsupervised manner in which case the training phase consists of parameters optimization problem.

In practice, QML exists in a variety of forms and techniques such as clustering methods, quantum neural networks (QNN), optimization of parameterized quantum circuits (PQC) (Benedetti, Lloyd, Sack, & Fiorentini, 2019), or evaluation of kernel maps by estimating the distance between two feature vectors (Cerezo, Verdon, Huang, Cincio, & Coles, 2022).

The use of variational algorithms for QML has been considered in view of potential quantum advantages accessible in the near future (Mitarai, Negoro, Kitagawa, & Fujii, 2018) (Schuld, Bocharov, Svore, & Wiebe, 2020). This hybrid quantum-classical approach makes use of parameterized quantum circuits to approximate nonlinear functions, classify and fit relatively large quantum systems. Typically, the parameters are tuned iteratively, using a gradient-based optimization method such as the parameter-shift rule to reduce the depth of the circuit otherwise very large. The potential advantage relies in the exponentially large feature space the quantum circuit provides as well as the circuit expressivity (Sim, Johnson, & Aspuru-guzik, 2019). The latter is determined by the frequency spectrum of the quantum model (Schuld, Sweke, & Meyer, 2021), it strongly influences the circuit ability to represent the Hilbert space, and thus the functions the circuit can model. In their

paper, Sim et al. (Sim, Johnson, & Aspuru-guzik, 2019) have done a deep study of this characteristic for multiple quantum circuits (ansatzes).

In general, the ansatz contains a first encoding part to map the classical data into an initial quantum state, followed by one or multiple layers of encoding and entangling operations (also called encoding and variational layers). Encoding operations are usually performed via R_X , R_Y or R_Z gates where the rotation angles are given by the pre-processed classical data or parameters to be optimized. Entangling gates (CZ or CNOT) are necessary to take advantage of the high dimensional Hilbert space. A potential quantum advantage may result from these operations, but further studies are required to better understand potential correlations. Having many layers usually improves the expressivity of the circuit but it requires expensive resources in terms of circuit depth and number of two-qubit gates.

Another technique for QML is to implement quantum neural networks (QNNs) from existing classical neural networks (feed-forward, convolutional) (Allcock, Hsieh, Kerenidis, & Zhang, 2020) (Kerenidis, Landman, & Prakash, 2019) (Henderson, Shakya, Pradhan, & Cook, 2020). This method is particularly suited for classification and image recognition applications. Theoretical works have shown that such quantum algorithms can provide computational speedups when compared to their classical counterpart (Allcock, Hsieh, Kerenidis, & Zhang, 2020). However, they rely on quantum Random Access Memories (qRAMs) (Giovannetti, Lloyd, & Maccone, 2008) and state preparation methods that are either not accessible yet, or very demanding in terms of resources. Alternative methods for efficiently encoding classical data into quantum states, such as a data loader, have been demonstrated experimentally (Johri, et al., 2021). The hybrid classical-quantum solution proposed by (Rainjonneau, 2023) makes use of a Reinforcement Learning (RL) training algorithm where an agent interacts with an environment. It uses a reward function to assign values to the actions of the agent. The RL agent uses a policy model to decide which action to make given a state in the environment, thus transforming the state. The policy model is trained to select the action that maximizes the probability of achieving a positive reward. The policy model considered is a parameterized circuit to establish the input-output relation of the model and apply it to the context of mission planning. This quantum subroutine is performed after a classical pre-processing step based on classical neural network to reduce the number of meaningful features. Two approaches have shown promising results that outperform classical greedy methods: the hybrid-quantum Proximal Policy Optimization (PPO) and the hybrid AlphaZero algorithms.

6.1.3 Quantum platforms

6.1.3.1 Use case 1a: optimization problem

The quantum platforms for solving the mission planning as an optimization problem are quantum annealers. Presently, the most widely employed QA devices are the D-Wave Systems annealers, which employ superconducting qubits. Promising alternative approaches, such as the Lechner-Hauke-Zoller (LHZ) (Lechner, Hauke, & Zoller, 2015) scheme or platforms employing cold atoms developed by Pasqal (Scholl, et al., 2021) or QuEra (Ebadi, et al., 2021) are also currently advancing significantly in terms of hardware development. However, these atomic-based QPUs can implement adiabatic protocols based on Rydberg atom Hamiltonians, which operate in an analog mode as opposed to the digital mode of quantum circuits (Wurtz, Lopes, Gemelke, Keesling, & Wang, 2022).

On the other hand, gate-based approaches, such as QAOA, have been successfully implemented both on IBM superconducting QPU ([IBM Quantum Documentation](#)) and IonQ device ([IonQ Forte](#)).

6.1.3.2 Use case 1b: quantum machine learning

The quantum platforms of interest for implementing quantum machine learning algorithm must be compatible with gate-based approaches. The focus is set on digital quantum computers.

Due to the small size of the quantum circuit, both simulators and real quantum computing hardware could be used. However, accumulation of errors from decoherence in real devices strongly limit the performance of the circuit on a real device in the near future. The training of this QNN has been successfully conducted on QMware Quantum Cloud in (Giovannetti, Lloyd, & Maccone, 2008).

The ansatz proposed in hybrid PPO and hybrid AlphaZero algorithms assumes a ring-shaped layout of qubits. The chosen quantum platform must allow for such interactions between qubits in order to avoid using additional gates and operations that would induce noise and errors in the computation.

6.1.4 Quantum instance

6.1.4.1 Use case 1a: optimization problem

As we aim to harness quantum algorithms for finding the optimal solution to the mission planning optimization problem, a powerful method entails transforming the preceding mathematical formulation into a so-called *Quadratic Unconstrained Binary Optimization (QUBO) problem*. QUBOs represent one of the most widely used methods for solving optimization problems on quantum computers (Glover, Kochenberger, & Du, 2019) and consists in converting the linear-binary-cost function $F(\vec{x}, \vec{y})$ defined above from a constrained optimization problem to a corresponding unconstrained optimization problem, where the original constraints now appear as penalty terms. Constraints in the form of linear equalities (1, 2, 3)

$$A \vec{x} = b$$

where A represents the coefficient matrix defining the constraint, while \vec{x} denotes the vector of decision variables pertaining to the problem, are squared and added to the linear cost function as penalty terms

$$P(\vec{x}) = \lambda(A \vec{x} - b)^2$$

where $\lambda > 0$ is a penalty constant. If \vec{x} satisfies the constraint, then $P(\vec{x}) = 0$ and the penalty term doesn't change the value of the cost function. Conversely, if x is such that $P(\vec{x}) \neq 0$ then the penalty term will prevent this solution vector to be a minimum of the cost function, provided λ is appropriately chosen. Consequently, we can enforce constraints 1), 2) and 3) within the QUBO framework by adding the following penalty terms to the cost function:

- One acquisition attempt per target:

$$P_1(\vec{x}) = \lambda_1 \sum_{j=1}^m \left(\sum_{i=1}^n \sum_{k=1}^{\theta_{i,j}} x_{i,j}^k - 1 \right)^2$$

- One download attempt per target:

$$P_2(\vec{y}) = \lambda_2 \sum_{j=1}^m \left(\sum_{i=1}^n \sum_{r=1}^{\sigma_i} y_{i,j}^r - 1 \right)^2$$

- Each acquisition must be downloaded:

$$P_3(\vec{x}, \vec{y}) = \lambda_3 \sum_{i=1}^n \sum_{j=1}^m \left(\sum_{k=1}^{\theta_{i,j}} x_{i,j}^k - \sum_{r=1}^{\sigma_i} y_{i,j}^r \right)^2$$

This procedure introduces two-site terms while leaving unmodified the number of binary variables:

$$N_{var} = \sum_{i=1}^n \sum_{j=1}^m (\theta_{i,j} + \sigma_i).$$

When dealing with constraints in the form of inequalities (4, 5, 6, 7), the conventional approach involves converting them into equations and then proceeding as described above. This requires the introduction of auxiliary variables known as *slack variables* (Djidjev). For instance, a constraint

$$A \vec{x} \leq b$$

can be represented as

$$A \vec{x} + s = b$$

by introducing the new integer slack variable $s \geq 0$. In the absence of further information regarding the binary variables of the problem, the slack variable can potentially be as large as b , i.e., $0 \leq s \leq b$. Since the QUBO reformulation accept only binary variables, this entails introducing $\mathcal{O}(\log_2 b)$ auxiliary binary variables to encode the value of the integer slack variable s . In situations where the optimization problem has multiple inequality constraints, as for the mission planning problem, this approach can potentially result in a substantial increase in the number of variables. Therefore, if feasible, it is preferable to find alternatives that avoid the use of slack variables. Specifically, constraints 4), 5) and 6) can be reformulated as quadratic penalty terms and incorporated into the cost function without the need for slack variables. This can be achieved by carrying out a preprocessing step for each constraint. To elaborate further, let us consider constraint 4): for each satellite i and target j , we have that the DLO variable $y_{i,j}^r$ can only be present in conjunction with $x_{i,j}^k$ if $t_{i,j}^k \leq s_{i,j}^r$. For this reason, we can identify the set of forbidden pairs of indices in a preprocessing step as

$$\mathcal{D}_{i,j} = \{(k, r) \mid t_{i,j}^k > s_{i,j}^r\}$$

and we can enforce constraint 4) with the following penalty term:

$$P_4(\vec{x}, \vec{y}) = \lambda_4 \sum_{i=1}^n \sum_{j=1}^m \sum_{(k,r) \in \mathcal{D}_{i,j}} x_{i,j}^k y_{i,j}^r$$

This preprocessing will cost $\sum_{i=1}^n \sum_{j=1}^m \theta_{i,j} \sigma_i$ steps. By applying the same reasoning to constraints 5) and 6) we derive the following penalties:

$$P_5(\vec{x}) = \lambda_5 \sum_{i=1}^n \sum_{(s,j,k,k') \in \mathcal{G}_i} x_{i,s}^k x_{i,j}^{k'}$$

$$P_6(\vec{y}) = \lambda_6 \sum_{i=1}^n \sum_{(g,j,r,r') \in \mathcal{J}_i} y_{i,g}^r y_{i,j}^{r'}$$

by introducing the two sets of indices

$$\mathcal{G}_i = \{(s, j, k, k') \mid p_{i,s}^k > |t_{i,s}^k - t_{i,j}^{k'}|\}$$

$$\mathcal{J}_i = \{(g, j, r, r') \mid d_{i,g}^r > |s_{i,g}^r - s_{i,j}^{r'}|\}$$

where the preprocessing for constraint 5) takes $\sum_{i=1}^n \sum_{j,s=1}^m \theta_{i,j} \theta_{i,s}$ steps and the one for constraint 6) takes $m^2 \sum_{i=1}^n (\sigma_i)^2$ steps. Hence, also constraints 4), 5) and 6) introduce two-site interactions without altering the total number of binary variables.

Concerning constraint 7), it is not feasible to employ the preprocessing step, and thus the introduction of slack variables becomes necessary to transform the inequality into an equality. This constraint requires the introduction of n slack variables $\{s_i\}_{i=1}^n$ such that

$$s_i \leq q_i^{tot} + \sum_{j=1}^m |\mathcal{L}_{i,j}| q_{i,j}$$

where $|\mathcal{L}_{i,j}|$ is the number of index pairs in the set

$$\mathcal{L}_{i,j} = \{(k, r) \mid t_{i,j}^k \leq s_{i,j}^r\}$$

Each of these slacks will then require the introduction of

$$\mathcal{N}_i^{slack} = \left\lceil \log_2 \left(q_i^{tot} + \sum_{j=1}^m |\mathcal{L}_{i,j}| q_{i,j} \right) \right\rceil + 1 \quad \forall i \in [1, n]$$

auxiliary qubits, which will increase the number of logical qubits relative to the binary variables in the original optimization problem. Different methods exist to reduce the required number of auxiliary qubits for representing slack variables. For instance, the scalar transformation of the inequality defining the constraint (Verma & Lewis) or the so-called augmented Lagrangian method (Djidjev) can be considered. However, the effectiveness of these approaches depends on the specific optimization problem. Exploring one of these methods to decrease the number of auxiliary qubits introduced by constraint 7) could present an intriguing advancement for implementing it on the existing D-Wave quantum computing platforms.

Therefore, by solving the defined QUBO problem, it becomes feasible to acquire an (optimal) solution for the original constrained optimization problem. Explicitly, the QUBO solution corresponds to the optimized vector of decision variables, represented by the ordered bitstring:

$$\mathbf{x} = \left[\dots, x_{1,j}^1, \dots, x_{1,j}^{\theta_{1,j}}, y_{1,j}^1, \dots, y_{1,j}^{\sigma_1}, \dots, x_{n,j}^1, \dots, y_{n,j}^{\sigma_n}, x_{1,j+1}^1, \dots, y_{n,j+1}^{\sigma_n}, \dots \right],$$

where the binary variable $x_{i,j}^k \in \{0,1\}$ refers to satellite i aiming at the target j for the k -th DTO, while $y_{i,j}^r \in \{0,1\}$ refers to satellite i downloading the image of target j for the r -th DLO. This way the slowly varying index is the target one. A motivation for this arrangement can be found by looking at the matrix representation of the QUBO problem, see **Figure 1**. Further, the matrix representation allows the classification of the two-site terms based on the constraints. Here we consider a general case with $n = 3$ satellites, $m = 3$ targets, $(\theta_{1,1}, \theta_{2,1}, \theta_{3,1}, \theta_{1,2}, \theta_{2,2}, \theta_{3,2}, \theta_{1,3}, \theta_{2,3}, \theta_{3,3}) = (2, 3, 2, 3, 2, 2, 2, 2, 3)$, and $(\sigma_1, \sigma_2, \sigma_3) = (2, 2, 2)$.

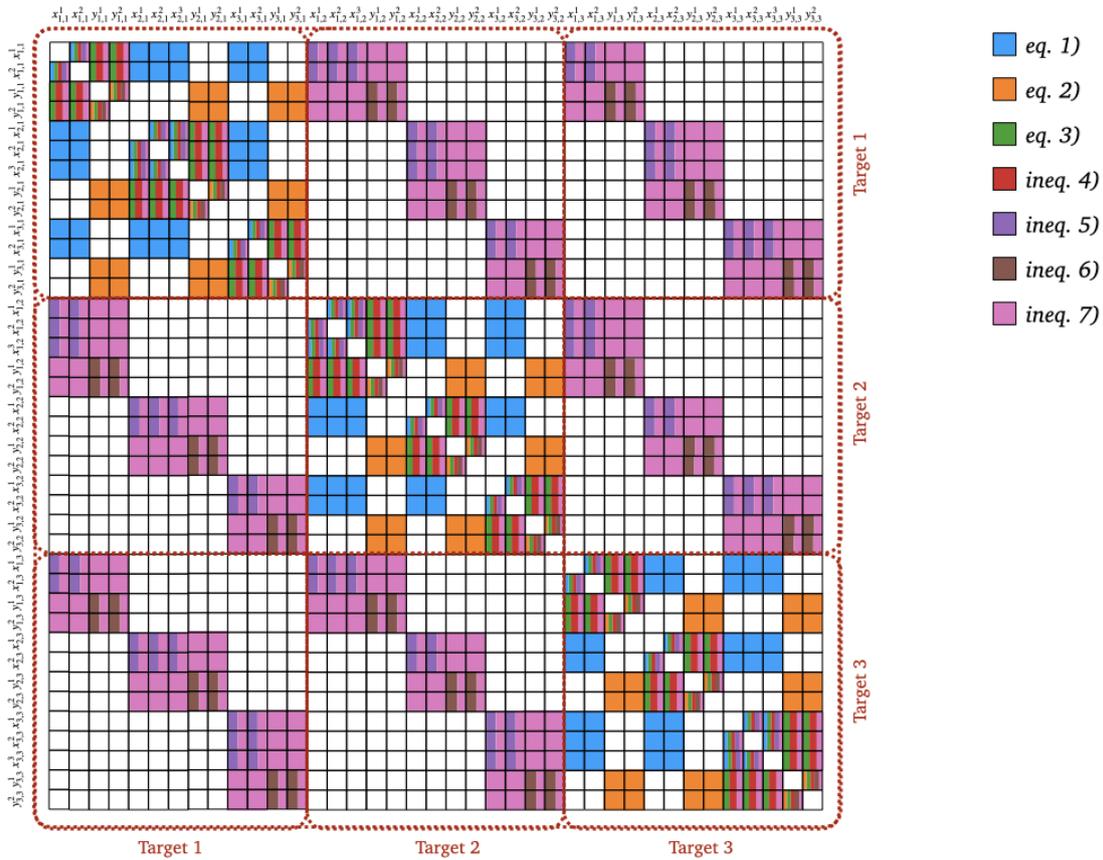


Figure 1: Matrix representation of the QUBO formulation for the mission planning problem. Different colors correspond to different constrains. Bitstrings entries are arranged such that the target index is the slowest one.

One can recognize that equation constraints (1,2,3) and inequality (4) do not couple variables corresponding to different targets. On the contrary, inequalities (5,6,7) introduce all-to-all terms across all the images. These considerations will help in designing an ansatz for solving this problem with a quantum (inspired) method, and the partitioning of the problem. Overall, the matrix is sparse. The initialization of parameters using real data will determine also the more relevant couplings.

The typical problem size to be managed considering all the constraints above is $N_{var} \sim 10^6 - 10^9$ as current mission planning problems involve constellations with $n \sim 10^2$, $m \sim 10^3$ and $\theta_{i,j} \sim \sigma_i \sim 1$.

Current mission planning problems involves constellations with $N \sim 100$, $M \sim 1000$ while $\theta_{i,j}, \sigma_i$ depend on the specific problem, see **Table 2**.

To turn the search of the minimum into a ground-state search of a quantum many-body problem, the QUBO problem is mapped into an Ising-like Hamiltonian, that contains only diagonal elements. Binary decision variables are promoted to spin operators:

$$x_{i,j}^k \in \{0, 1\} \rightarrow \tau_{i,j}^k = 2x_{i,j}^k - 1 \in \{-1, 1\},$$

$$y_{i,j}^r \in \{0, 1\} \rightarrow v_{i,j}^r = 2y_{i,j}^r - 1 \in \{-1, 1\}.$$

The ground state of this Hamiltonian encodes the solution of the initial problem. This additional mapping allows us to address the original optimization problem using quantum-inspired method, such as Tensor Network (TN) approaches (see paragraph 6.1.8).

6.1.4.2 Use case 1b: quantum machine learning

The reinforcement learning (RL) environment is a function of three variables: the state space S , the action space A and a transition function P . The reward r is attributed following a Markov decision process. Typically, the agent starts from a state s_0 , takes action a_0 and gets the reward r_0 in each step of the training. Eventually, the agent produces a trajectory $T = (s_0, a_0, r_0, s_1, a_1, r_1, \dots)$. Two environments are considered in (Rainjonneau, 2023).

In the satellite-centred environment, the agent considers the 100 closest data points (in terms of DTO times) for each satellite. Each of these data points utilizes 10 parameters. In addition, three features are made available to the agent: the current time and the latitude/longitude coordinates of the starting point of the last completed request. Overall, the observation space has a dimension of 1003.

In the request-centred environment, the agent decides which satellite is best suited to complete a given request. For a selected satellite, the 5 closest requests are considered and the one that requires the minimum execution time is chosen. The execution time is computed as the sum of the satellite timestamp, the manoeuvring duration, and the acquisition duration.

In order to avoid redundancy, each data point is flagged with a boolean value to indicate whether it has been completed or not.

In the hybrid PPO algorithm, a hybrid QNN is used as a policy model. A classical neural network first reduces the number of features to a small enough number of outputs that serve as inputs to the parametrized quantum circuit. The figures in **Figure 2** (Rainjonneau, 2023) describe the structure of the algorithm, where the quantum circuit is added to the end of the classical network, just before the agent takes an action.

The quantum part of the algorithm consists in a 4-qubit circuit composed of a first variational layer with parameterized R_X rotations and ring-shaped CNOT gates. The same structure is repeated for 8 additional layers in which features from the classical network are encoded as parameters of R_Z rotations. This structure enables the encoding of 32 features. It becomes obvious that a trade-off is made between the number of features to be encoded in the quantum circuit and the resources needed to implement it in terms of 2-qubit gates and circuit depth. Finally, a measurement layer acting on the first two qubits returns two real values as outputs.

The ansatz (highlighted in green in **Figure 2**) has been studied in more detail in the work proposed by Sim et al. (Sim, Johnson, & Aspuru-guzik, 2019). It is shown that its entangling capability is very high no matter the number of layers considered and that its “expressibility” is low when a single layer is considered. However, when multiple layers are added (4 or more), its “expressibility” converges to a high value. This indicates that this ansatz is expected to be efficient in the exploration of the entire Hilbert space.

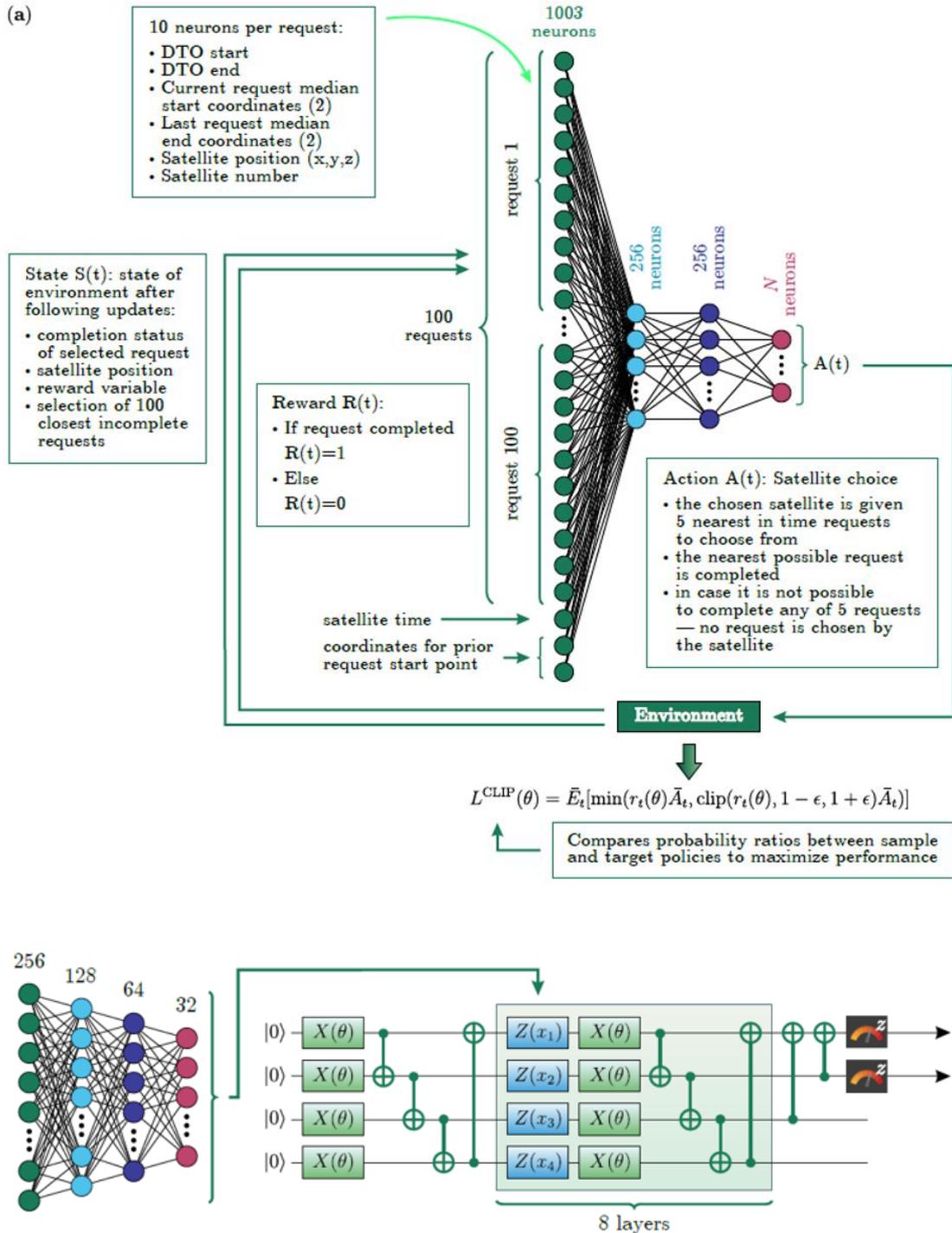


Figure 2 (top) The RL model used to solve the mission planning problem (Rainjonneau, 2023). The state of the model consists of data from 100 requests which are fed into the agent to output an action, which then feeds into the environment (clip equation), generates an appropriate reward, and updates the state. **(bottom)** Quantum-hybrid RL model. A quantum circuit (left) is added to the beginning of the agent in the RL model (right) to incorporate quantum computation into the neural network.

In the hybrid AlphaZero algorithm, which outperforms both the hybrid PPO and the greedy algorithms, the same 4-qubit PQC is used. As for its classical part, this algorithm uses a computational tree of environment states whose values and probabilities result from the output of a value network (classical neural network composed of a single neuron in a fully connected layer) and a policy network (the previously described PQC). A descriptive schematic of the algorithm is given in **Figure 3**. The algorithm relies on the use of a Monte Carlo tree search combined with the policy network while the value network is processed in parallel. At the beginning of the tree search, the model accesses the optimal leaf by starting from the root and choosing the optimal child node. New child nodes are created from this leaf and the model completes the full path-finding process from this point. A backpropagation occurs to update the information of the nodes of the path as well as hyperparameters from the result obtained.

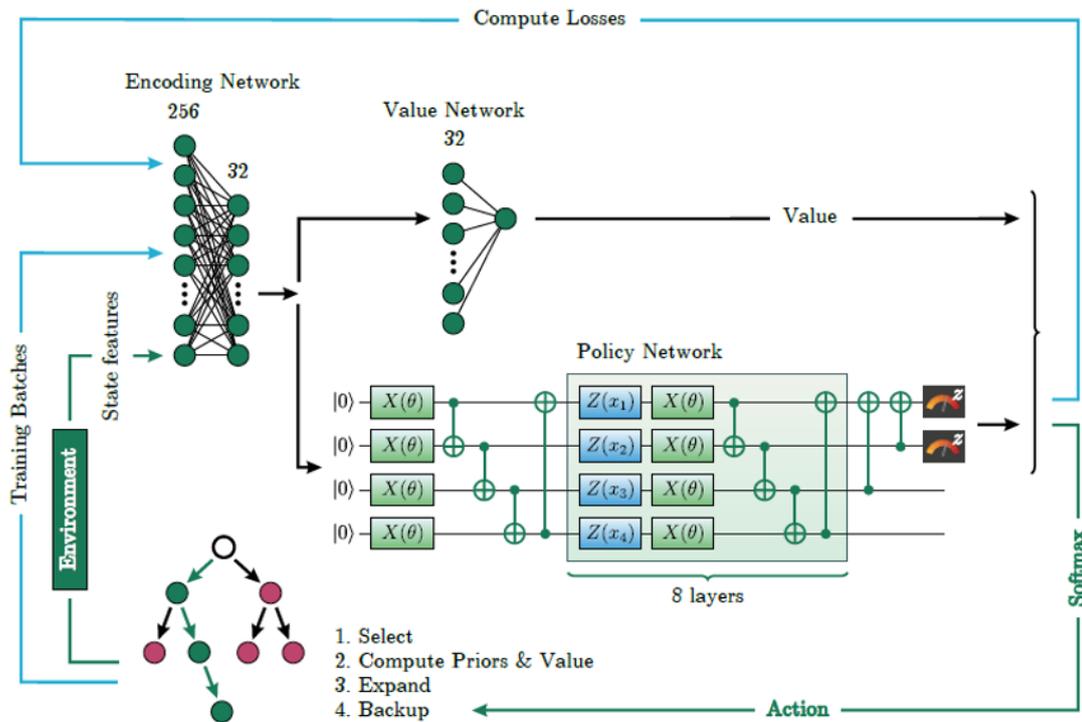


Figure 3 The hybrid AlphaZero architecture used to achieve a near-optimal solution to the satellite mission planning.

6.1.5 Number of qubits

6.1.5.1 Use case 1a: optimization problem

The number of qubits corresponds to the number of binary variables, including slack ones. As detailed in the previous section we compare the case with and without preprocessing of the data. To estimate the scaling with respect to the number of variables, we are considering 6 different constellations, each containing from 1 to 6 satellites. In this simplified model, number of targets m is equal to 2 and satellites are indistinguishable each having 2 DTOs and 2 DLOs. Other parameters in the model, such as the acquisition and download times and the available on-board memory are chosen randomly in a

certain range. In **Figure 4** we observe that both connections (couplings between pairs of qubits) and logical qubits scale linearly with the number of binaries.

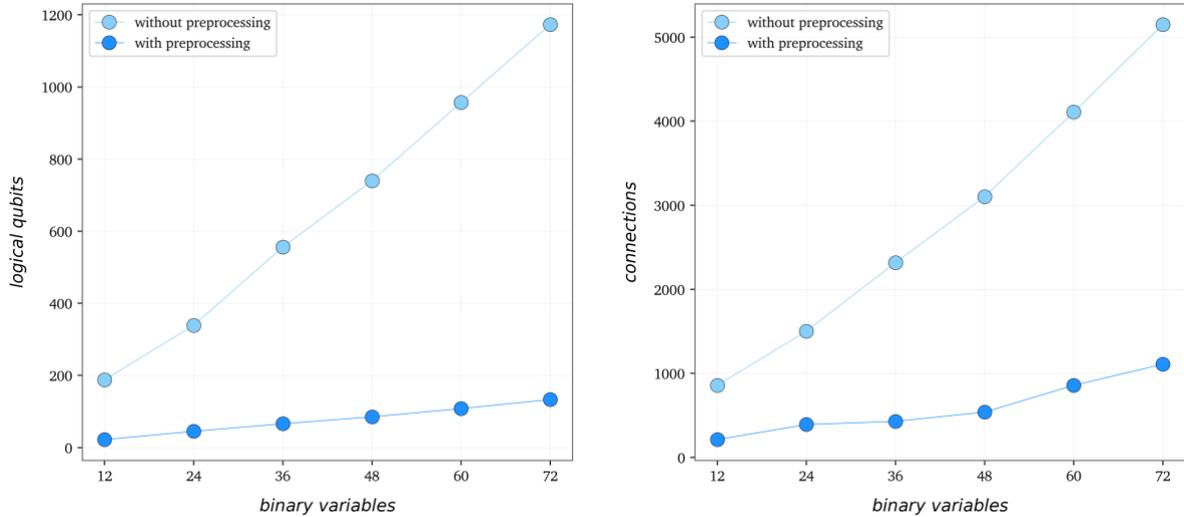


Figure 4 Scaling of logical qubits and connections with increasing constellation size. The effect of preprocessing is considered in the counting. In this simplified example number of targets, DTOs and DLOs are all fixed at 2.

6.1.5.2 Use case 1b: quantum machine learning

The number of qubits corresponds to the size of the inputs of the QNN. Having access to many qubits enables to reduce the classical part of the hybrid algorithm and increase its quantum part since less layers of the classical neural network are required.

NUMBER OF QUBITS	NUMBER OF LAYERS	NUMBER OF ENCODED FEATURES
4	8	32
8	8	64
16	8	128
...
2^N	8	2^{N+3}

Table 4 Number of qubits, layers and encoded features for the QML approach to the mission planning problem.

Following the same structure for the quantum circuit, doubling the number of qubits and keeping the 8-layer structure would enable to remove one layer of the classical network, as shown in **Table 4**. Indeed, the number of features encoded into the quantum circuit corresponds to the number of

parameterized Z-gates in this circuit. The number of parameterized Z-gates is exactly the number of qubits times the number of ansatz layers in the circuit.

This structure may not be optimal. More generally, 2^f features can be encoded using 2^q qubits and 2^l layers with $q + l = f$.

6.1.6 KPIs for quantum simulators

6.1.6.1 Use case 1a: optimization problem

In this study, we investigate the implementation of the Quadratic Unconstrained Binary Optimization (QUBO) formulation on the D-Wave hardware. To achieve this, we employ the ocean-sdk tool to perform minor-embedding on Pegasus topology ([D-Wave Ocean](#)). This embedding process enables us to map the QUBO problem onto the hardware topology.

Figure 5 illustrates the scaling behavior in terms of the number of physical qubits required to embed different instances, where each instance only considers a single constraint. As anticipated, the primary contribution to the scaling arises from inequalities: both the original formulation and the one with preprocessing are considered. In the original formulation that incorporates all constraints, the resource requirements exhibit exponential growth. Preprocessing consumes at least five times less the number of qubits required by the original QUBO formulation. The Pegasus connectivity can accommodate problems with up to 72 variables to be embedded successfully.

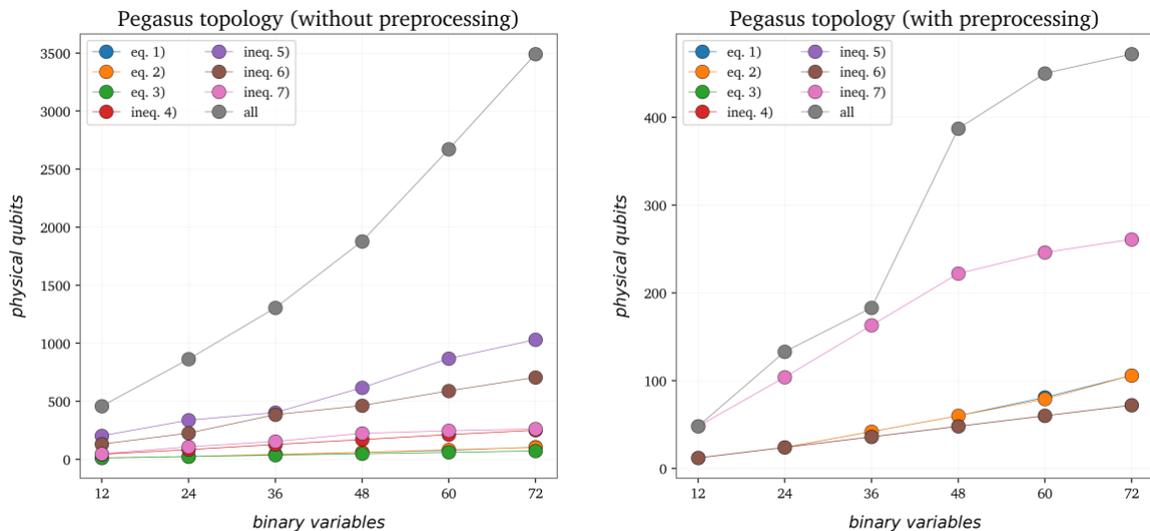


Figure 5 Scaling of the number of physical qubits required with the DW_Advantage ([D-Wave Advantage](#)) taking into account the data preprocessing.

6.1.7 KPIs for digital quantum computers

6.1.7.1 Use case 1b: quantum machine learning

As shown in the paper by S. Rainjonneau et al. (Rainjonneau, 2023), large scale problems (2 satellites and thousands of requests) can already be tackled using a narrow quantum circuit. This is done thanks to the classical neural network that significantly reduces the number of features to encode into the quantum neural network. Increasing the size of the quantum circuit does not necessarily enable solving problems larger than what can be done classically. Nevertheless, it is a direction to look into in order to observe whether a better solution (in term of quality of the solution) can be achieved or not.

Multiple KPIs must be taken into account when addressing the implementation of an algorithm on a digital quantum computer. Let us denote as N the number of qubits and L the number of layers of the ansatz. The scaling of the main characteristics of the quantum circuit are described in **Table 5**. The scaling as a function of the number of qubits and ansatz layers, and thus the complexity, is directly deduced from the given structure of the algorithm as presented in the quantum circuit studied. It is assumed that the overall structure of the quantum circuit and its ansatz are kept the same despite increasing the size of the circuit.

KPI	SCALING W.R.T. THE NUMBER OF QUBITS	COMPLEXITY
Number of parameters to optimize (including features)	$N(2L + 1)$	$\mathcal{O}(NL)$
Number of features encoded	NL	$\mathcal{O}(NL)$
Number of 2-qubit gates (CNOTs)	$N(L + 1) + 2$	$\mathcal{O}(NL)$
Circuit depth	$N(L + 1) + 2L + 3$	$\mathcal{O}(NL)$

Table 5 KPIs for QML approach to the mission planning problem.

For a fixed number of layers, all key physical characteristics of the circuit scale linearly with the number of qubits. Note that features from the classical neural network are all encoded via R_Z gates while the parameters of all R_X gates are to be optimized during training. Error rates of the quantum gates (rotations and CNOTs) are device-dependent and they strongly affect the quality of the solution obtained.

6.1.8 Tensor Network (TN) approaches

The matrix representation of QUBO formulation reported in **Figure 1** highlights the long-range nature of the problem. TN methods and specifically Tree Tensor Network (TTN) ansatz provide an efficient tool to address these problems.

TN methods have proven to be remarkably successful in treating many-body quantum problems using classical computers (Montangero, 2018) and they have quickly emerged as a leading numerical technique for studying low-dimensional systems (Eisert). TNs serve as a versatile information processing tool that

leverages correlations and entanglement, making them applicable to any problem involving correlation structures.

TTNs exploit the entanglement structure inherent in the quantum systems to provide an efficient representation of the quantum state. In quantum many-body physics, TTNs are a specific class of TN ansatz devised for representing the ground states of quantum systems (Felser, 2021). The underlying idea of TTNs involves decomposing the exponentially complex many-body wave function, which lives in the tensor product

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \dots \otimes \mathcal{H}_N$$

of N local Hilbert spaces \mathcal{H}_k , one for each quantum degree of freedom in the system under investigation, into a hierarchical arrangement of tensors, forming a tree-like structure wherein each tensor corresponds to a localized region in the system (see **Figure 6**). Entanglement between different regions of the system is captured through the auxiliary indices connecting the tensors within the tree structure. By suitably controlling the dimension of these auxiliary indices with the so-called bond dimension χ , one can regulate the amount of captured information. In this manner, by adjusting the parameter χ , TTNs enable efficient representation of quantum states, interpolating between a product state with $\chi = 1$, where quantum correlations are neglected, and an exact but inefficient representation of the many-body wavefunction, which requires d^N complex numbers, where d is the dimension of each local Hilbert space \mathcal{H}_k .



Figure 6 (Left) The TTN for one-dimensional systems. (Right) The TTN for two-dimensional systems.

TTNs with their hierarchical structure can in principle be defined in any dimension, and provide enhanced connectivity for long-range interacting systems, with a logarithmic distance through the network structure, whereas simpler Matrix Product States (MPS) (Orús, 2019) have a linear distance between connected tensors. Additionally, TTNs can be optimized with lower computational complexity, scaling as $\mathcal{O}(\chi^4)$, compared to other TN geometries such as PEPS (Projected Entangled Pair States) (Silvi, et al., 2019)

Hence, an alternative approach to the mission planning problem involves transforming the minimization of the QUBO cost function into a search for the ground state of an Ising-like Hamiltonian. This transformation involves converting binary variables into spin- $\frac{1}{2}$ variables, which are then promoted to quantum operators, i.e., qubits. The long-range nature of the interactions between binaries of the QUBO

manifests as all-to-all couplings between qubits. Given that TTNs are effectively many-body ansatzes for high-dimensional systems, they can potentially facilitate the efficient exploration of the ground-state search.

By selecting a bond dimension $\chi > 1$, the quantum search for the ground state is performed within the subspace bounded by χ . Consequently, unlike classical optimization methods, the quantum representation of the TTN allows for the exploration of multiple overlapping classical solutions within a single optimization step. This capability enables the tunneling through higher, but reasonably subtle, potential energy barrier within the optimization landscape (Cavinato & et al., 2021).

6.1.9 Proposal of targeted quantum platforms

6.1.9.1 Use case 1a: optimization problem

D-Wave hardware currently provides a substantial number of qubits and a satisfactory level of connectivity, which are already sufficient for solving the problem on a small scale with quantum annealing approaches. However, it is anticipated that alternative platforms such as those based on neutral atoms (such as PasQal and QuEra), may offer a strategic route to surpass the connectivity limitations by enabling easier manipulation of qubits.

6.1.9.2 Use case 1b: quantum machine learning

The targeted platforms are gate-based quantum computing platforms. IBM hardware seems promising as they expect a very fast pace of development for their technology. Superconducting qubits have the benefit of performing operations in a very fast manner, thus allowing for the execution of larger circuits given a certain decoherence time. CNOT gates are not native gates for all technologies. It is important to favour a technology that allows for an efficient implementation of such entangling gates in order to limit the overhead in terms of additional operations that would appear during the compilation of the circuit. While certain IBM processors (Falcon and Hummingbird) directly implement the CNOT operation, it is not the case of Eagle or Osprey processors. Yet, they allow for the realization of an entangling gate that corresponds to a CNOT gate up to single-qubit pre-rotations.

Ion-based quantum computers such as IonQ Forte processor ([IonQ Forte](#)) also seems very promising. Such technology allows for very large decoherence times (10 to 100 seconds for T1, approximately 1 second for T2) and particularly small gate error rates (0.4% for 2-qubit gates and 0.02% for single-qubit gates) that are orders of magnitude lower than that of IBM processors for instance. The number of qubits is also reasonable (32 available now). This technology offers flexibility in the design of algorithms as all-to-all connectivity can be achieved as opposed to superconducting architectures.

6.2 Use case n°2: Multiple-view Geometry on Optical Images (TASF)

6.2.1 Problem instance

Multiple images of a given area of interest can be retrieved as satellites orbit around the planet. These images may be obtained from different satellites or from a single satellite during a long enough DTO window or multiple passes. An important task is to analyse the changes that have occurred on the area of interest as time has passed and perform terrain reconstruction. To do so, these images are compared with each other. However, the agility of the satellites and their different orbits result in the acquisition of different views of the area of interest: images may be rotated or translated, the illumination or scale may differ from one image to another.

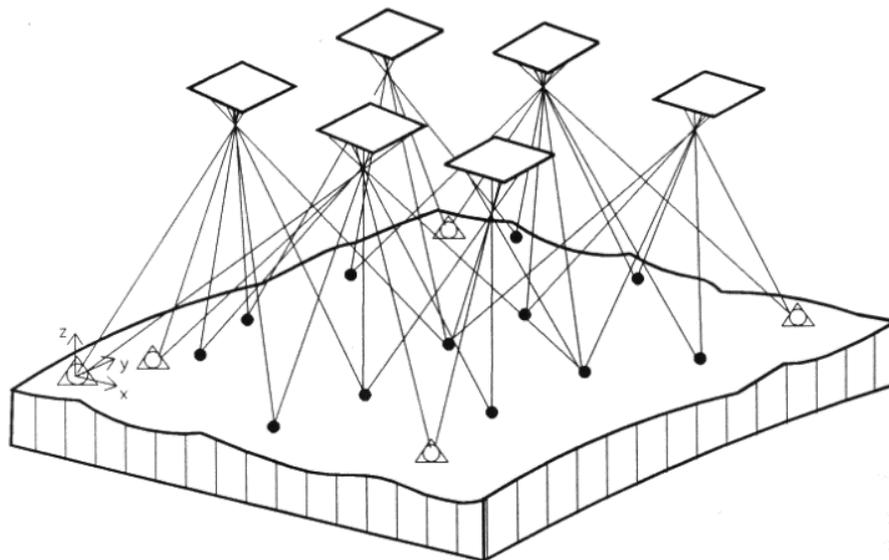


Figure 7 Example of multiple (6) views from a scene.

This problem can be tackled with *bundle adjustment*, which consists in estimating the different changes by minimizing the re-projection error, a single functional with a high number of parameters due to the high number of degrees of freedom. These calculations must be executed in a limited time to allow for more accurate approaches for terrain reconstruction and reduce constraints on the satellite platform localization.

This method is conducted by first extracting keypoints that characterize well the different images (typically described by SIFT descriptors), then by matching those that are common to multiple images **Figure 7**. Finally, a projection that aligns the coordinate systems of all images must be identified, followed by a transformation to align all images in a single plane (typically Direct Linear Transformation or eight-point algorithm).

From there, image classification methods or other techniques allow for further study of the area of interest such as the identification of moving objects.

Mathematically, the problem can be formulated as follows: assume that n 3-dimensional points are visible through m different views (e.g., $m = 6$ in **Figure 7**) and let p_{ij} be the projection of the i -th point to the plane containing the j -th image. It could be possible that the p_{ij} does not lie in the image itself, so we define a binary variable b_{ij} which is 1 if and only if point i is visible in image j . Furthermore, assume that the camera that created the j -th image can be characterized by a vector c_j , and every 3-dimensional point i by a vector x_i . The objective is then to minimize the total re-projection error:

$$f_{BA}(c_j, x_i) = \sum_{i=1}^n \sum_{j=1}^m b_{ij} \|\pi(c_j, x_i) - p_{ij}\|_2^2$$

where π corresponds to the predicted projection. Here, we use the camera pinhole model where camera aperture is described as a point and no lenses are used to focus light. That is, parameters c subsume the camera position, orientation in space, and focal length f . The projection of a 3-dimensional point $x = (x_1, x_2, x_3)$ is:

$$\pi(c, x) = \frac{f}{x_3} \begin{pmatrix} x_1' \\ x_2' \end{pmatrix}$$

where $x' = (x_1', x_2', x_3')$ are the 3-dimensional coordinates of x relative to the camera centered coordinate system.

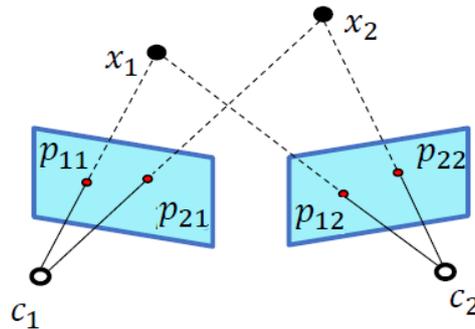


Figure 8: Two 3-dimensional points (characterized by vectors x_i), with their projections p_{ij} on images $j = \{1, 2\}$. c_j is the vector characterizing the camera that created the j -th image.

6.2.2 Quantum algorithm

Quantum annealing and adiabatic quantum computing are two closely related approaches of the class of analog quantum computing. Both methods proceed in the same way by mapping an optimization problem into a target Hamiltonian. The solution to the problem is obtained by finding the ground state of this target Hamiltonian. Typically, the quantum system is initialized to be in the ground state of a first Hamiltonian that is easily implementable. This Hamiltonian is slowly evolved over a period of time in order to eventually match the target Hamiltonian. The adiabatic theorem (Born & Fock, 1928) states that, following these hypotheses, the system ends up in the ground state of the target Hamiltonian, i.e., the

lowest energy state of the system. The adiabatic approach is more restrictive in the sense that it requires the transformation of the Hamiltonian to be adiabatic.

In practice, quantum annealers are particularly suited to solve problem formulated as quadratic unconstrained binary optimization (QUBO) problems, which are equivalent to Ising Hamiltonians:

$$z^* = \operatorname{argmin}_{z \in \{0,1\}^N} z^T Q z + q^T z.$$

where N is the number of qubits. It is known that combinatorial problems of complexity NP can be formulated as QUBOs (Lucas, 2014). Even though no theoretical quantum advantage has been demonstrated on such devices so far, they are of great interest for relatively large-scale problems in the near future.

The quantum algorithm studied for keypoints extraction from optical images and feature matching heavily relies on the quantum annealing approach (Piatkowski, et al., 2022). The keypoints extraction procedure is conducted using two different clustering approaches: quantum k-medoids clustering and quantum kernel density clustering. In the first case, Thales Alenia Space and Fraunhofer Institute formulate the problem as a QUBO to select k distinct objects located in the centre of the image. In the second case, they also formulate the problem as a QUBO but compute the kernel matrix by evaluating a quantum circuit on a gate-based quantum computer. Feature descriptors such as SIFT are added to gain more information about the different scaling and rotations between images before performing the feature matching operation that uses quantum annealing as well as the kernel matrix computed a priori.

6.2.3 Quantum instance

The formulation of a QUBO problem is obtained from the initial formulation of the optimization problem that consists in an objective function to minimize and additional constraints written as equalities or inequalities, all of which contain binary variables. The objective function as well as the initial constraints are squared and added together to form the QUBO. The terms that result from the transformation of original constraints are called penalties.

For this problem, an image is encoded as a series of pixels $I = \{p_1, p_2, \dots, p_M\}$ where each pixel is represented by its location and colour channels $p_i = (x_i, y_i, r_i, g_i, b_i)$. For the quantum k-medoids approach to keypoints extraction, the idea is to minimize the distance between the keypoints and the rest of the pixels of the image while maximizing the distance between two distinct keypoints. In general, this leads to a selection of keypoints equally distributed within the image. The corresponding QUBO formulation is given by (Piatkowski, et al., 2022)

$$Q = \gamma \mathbf{1}_M \mathbf{1}_M^T - \alpha D,$$

$$q = \beta D \mathbf{1}_M - 2\gamma k \mathbf{1}_M$$

where α, β, γ are Lagrange multipliers, $\mathbf{1}_M$ is the vector of size M that consists of ones only, and D is the matrix of distances such that $D_{ij} = \|p_i - p_j\|_2$.

For the quantum kernel density clustering approach, the idea is to minimize the discrepancy between two feature map distributions by finding optimal cluster centroids. The probability densities of pixels are obtained from kernel density estimates:

$$\rho_I(p) = \frac{1}{M} \sum_{i=1}^M K(p, p_i),$$

$$\rho_C(p) = \frac{1}{k} \sum_{j=1}^k K(p, c_j)$$

where the set of k cluster centroids is $C = \{c_1, \dots, c_k\} \subset I$ and K is a kernel function. It is assumed that there is a feature map such that $K(\cdot, \cdot) = \langle \phi(\cdot), \phi(\cdot) \rangle$.

The corresponding QUBO formulation is given by

$$Q = \frac{1}{k^2} \boldsymbol{\kappa} + \lambda \mathbf{1}_M \mathbf{1}_M^T,$$

$$q = -2 \left(\frac{1}{kN} \boldsymbol{\kappa} \mathbf{1}_N + \lambda k \mathbf{1}_N \right)$$

where $\boldsymbol{\kappa}$ is the kernel matrix with $\kappa_{ij} = K(p_i, p_j)$ and λ is a Lagrange multiplier. Instead of a classical Gaussian kernel (that has been mostly studied in the literature), a quantum kernel is considered. The quantum kernel studied is a problem-agnostic 4-qubit quantum circuit.

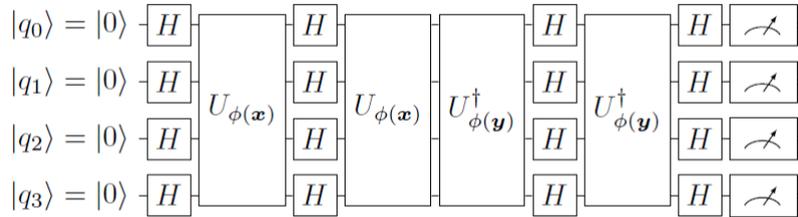


Figure 9 Possible circuit to compute the fidelity between two feature vectors, retrieved from (Havlíček, et al., 2019)

A uniform superposition is prepared and transformed by a N -qubit unitary operator. The data are not used to encode the qubits, but they act as parameters for the unitaries representing the feature map. The kernel value $K(x, y)$ results from the estimation of the transition amplitude

$$|\langle \phi(x), \phi(y) \rangle|^2 = \left| \langle 0^N | V_{\phi(x)}^\dagger V_{\phi(y)} | 0^N \rangle \right|^2$$

with

$$V_{\phi(\cdot)} = U_{\phi(\cdot)} H^{\otimes N} U_{\phi(\cdot)} H^{\otimes N}$$

The choice of the N -qubit operator is not fixed. Nevertheless, a potential quantum speedup can only be obtained if the feature map cannot be efficiently simulated classically but no proof of quantum speedup

has been shown so far. Typically, it is recommended to use an operator of the form given in (Havlíček, et al., 2019)

$$U_{\phi(x)} = \exp(-i \sum_{S \subset [M]} \phi_S(x) \prod_{v \in S} Z_v)$$

where S is a subset of $[M]$ and Z_v is the Z -gate. In order to obtain the kernel matrix for M data points, this circuit must be executed $M(M + 1)/2$ times.

The feature matching part consists in identifying matches between keypoints of two distinct images $p_i^{(1)}$ and $p_i^{(2)}$. To do so, feature descriptors (SIFT) enable to account for variations in scaling, rotation or illumination between the images. The kernel function can also be used to access a high dimensional feature space. The optimization problem is once again formulated as a QUBO problem. The two constraints in the initial formulation correspond to the fact that two keypoints from a given image cannot be matched to the same keypoint in the second image, and every keypoint can only be matched with k points in the second image at most. This last constraint being modelled by an inequality, binary slack variables are introduced when one considers the QUBO formulation.

6.2.4 Quantum platforms

The quantum algorithm for bundle adjustment using quantum clustering methods has been successfully implemented on D-Wave Advantage System 5.1 (5619 qubits) in 2022 (Piatkowski, et al., 2022) for small instances of the problem. The quantum kernel matrix has been computed from a state-vector simulation on IBM Falcon QPU.

These platforms are well suited for experiments in the near future since D-Wave quantum annealers and IBM QPUs are currently the largest available even though qubit connectivity remains a major bottleneck.

6.2.5 Number of qubits

The full-size problem consists in performing bundle adjustment of large satellite images of 3099×2329 pixels. However, such images would require solving a QUBO of dimension 7217571, which cannot be done on current hardware. In view of NISQ compatibility, keypoint extraction would be performed iteratively, from smaller scale sub-images up to the original image. Typically, the idea is to reduce the redundant information within the image so that the initial image becomes an image of 928×704 pixels. This reduced image is then split into 32×32 non-overlapping patches of 29×22 pixels. In a first step, 10 keypoints are extracted from each of these individual patches. Then these patches are considered in 8×8 small groups of 16 patches and, for each group, 20 keypoints are extracted from the previous 160 existing keypoints. This gives 20 keypoints for each of the 64 small groups that compose the initial reduced image. The procedure is repeated by gathering 16 of these previous small groups in 2×2 sets. Thus, each of these four sets initially contains 320 keypoints. From these keypoints, 45 are selected to be the “final” keypoints that represent the initial image. Therefore, the final image is described by 180 keypoints in total.

The real hardware allows for the extraction of 4 keypoints on patches of 8×8 pixels corresponding to a QUBO of dimension 64 but applications on a larger image can only be done using digital annealer for now.

The quantum circuit considered in the computation of the kernel matrix acts on 4 qubits. Larger circuits can be implemented on IBM real hardware but the need to measure each qubit may quickly destroy any hope of obtaining a quantum speedup. Increasing the number of qubits would certainly increase the dimension of the feature space with no guarantee that better results could be obtained. This is particularly true if one considers only pairwise features in order to reduce the circuit depth.

6.2.6 KPIs for digital quantum computers

When it comes to the quantum circuit used to estimate the kernel matrix, the KPIs to take into account are the number of qubits, the circuit depth, the number of 2-qubit gates, error rates and decoherence times. For NISQ applications it has been shown that only pairwise interactions must be considered in order to significantly reduce the overhead in terms of swap operations and 2-qubit gates at the time of transpilation (rewriting of the circuit to match the connectivity structure of a specific QPU). In the formulation of the operator $U_{\phi(x)}$, this means that S contains only 2 elements.

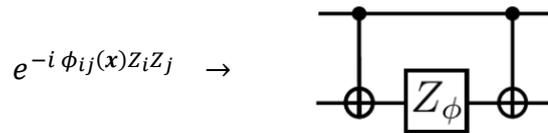


Figure 10 Implementation of the 2-qubit operation using CNOTs and Z-gates, retrieved from (Havlíček, et al., 2019)

Denote as M the number of data points in the image studied and N the number of qubits for the implementation of the quantum circuit. Assuming that only pairwise features are considered, the scaling of the characteristics of the circuit is given in **Table 6**.

KPI	SCALING W.R.T. THE NUMBER OF QUBITS	$N = 4$ CASE
Number of pairs of data points	$C_N^2 = N!/(2(N - 2)!)$	6
Number of 2-qubit gates (CNOTs)	$8C_N^2$	48
Circuit depth d (order of magnitude)	$4 + 6C_N^2 \leq d \leq 4 + 12C_N^2$	40

Table 6 Scaling of circuit characteristics.

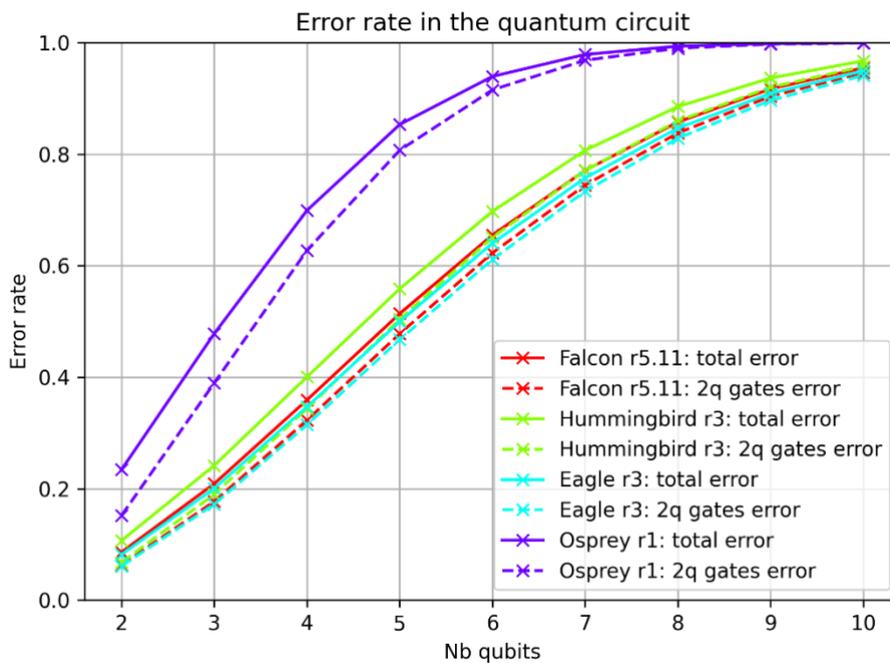
A description of the physical performances of several types of IBM processors ([IBM Quantum Documentation](#)) is given in **Table 7**. The T1 (T2) decoherence time corresponds to the decay constant time after which an initial state $|0\rangle$ ($|+\rangle$) will evolve into an equal probabilistic mixture of the states $|0\rangle$ and $|1\rangle$ ($|+\rangle$ and $|-\rangle$), such that one cannot confidently predict the state. The T1 time is often called the

relaxation time and T2 the dephasing time. The values presented are up to date as of June 15th 2023 and correspond to median times and error rates.

PROCESSOR	NUMBER OF QUBITS	2-QUBIT GATE ERROR	SINGLE-QUBIT S_x GATE ERROR	READOUT ERROR	T1(μ s)	T2(μ s)	QUANTUM VOLUME
Falcon (r5.11)	27	$8.091e^{-3}$	$2.960e^{-4}$	$1.07e^{-2}$	130.06	119.71	128
Hummingbird (r3)	65	$8.730e^{-3}$	$3.064e^{-4}$	$1.94e^{-2}$	179.91	171.24	N. A.
Eagle (r3)	127	$7.843e^{-3}$	$2.289e^{-4}$	$9.70e^{-3}$	278.51	161.87	32
Osprey (r3)	433	$2.036e^{-2}$	$5.927e^{-4}$	$4.64e^{-2}$	89.54	58.54	N. A.

Table 7 Performances of several IBM processors.

An estimation of the scaling of these errors with respect to the size of the quantum circuit chosen previously is showcased in the following graph for each of IBM four processors. One can observe that errors due to the accumulation of 2-qubit gates contribute the most to the total error. To give an order of magnitude, on IBM Falcon device, the 4-qubit circuit offers less than a 65% chance of returning a good result.



In addition, while the adiabatic transformation ensures the good convergence to the solution of the problem, it is important to be able to perform the computation quickly enough not to lose the encoded information due to decoherence. There is a trade-off between performing a slow evolution to obtain a good solution and decoherence time. Improving decoherence times and limiting errors due to thermal noise while keeping a good level of control of the Hamiltonian is a major challenge to be met by companies working with quantum annealing technologies (Gardas & Deffner, 2018).

6.2.7 Proposal of targeted quantum platforms

D-Wave hardware offers a large number of qubits and a reasonable connectivity, which are already satisfying to solve the bundle adjustment problem on a small scale. However, it is expected that other platforms such as quantum annealers based on neutral atoms (PasQal and QuEra) may overcome the limitations in connectivity due to easier manipulation of qubits.

For the computation of the kernel matrix on a digital quantum computer, IBM superconducting QPU, IonQ and AQT ion trapped technologies seem very promising. Their quantum volume are currently among the best and they keep achieving better performances on a fast pace. Currently, among IBM processors, the Eagle r3 seems to offer the best performances in terms of coherence times and error rates, with a rather large number of qubits compared to the Falcon devices.

6.3 Use case n°3: Optical Satellite Data Analysis (FZJ)

6.3.1 Problem instance

The analysis of satellite images plays an important role in EO. Among the most important applications is Land Use Land Cover Classification (LULC), which aims at providing classification maps of the investigated area of observation. In a pixel-based classification scheme the problem consists in assigning to a set of pixels within an image a label from a finite set of classes by analyzing a set of features related to the pixel itself. Such features are usually spectral, i.e., consist of optical measurement acquisitions carried out at different wavelengths. In some instances, also measurements carried out at different times are considered as features (multi-temporal approach).

From a theoretical point of view, therefore, pixel-based LULC amounts to a supervised classification problem: given a training set of N data points $(\mathbf{x}_n, \mathbf{y}_n)$, where \mathbf{x}_n is the vector containing the features related to a given pixel and \mathbf{y}_n its corresponding target label, the objective is to design a learning algorithm that given a new unseen feature data vector can infer the corresponding class label.

6.3.2 Quantum algorithm

6.3.2.1 Quantum Kernel Estimation

The objective of Quantum Kernel Estimation (QKE) is to construct a kernel using Quantum Computing methodologies. Usually this is done by using a Parametrized Quantum Circuit (PQC). The Kernel function is then used to compute the Gram matrix, i.e., the matrix containing the kernel function evaluations between all training data points pairs. The Gram matrix can then be used in conjunction with ML kernel methods such as Gaussian Processes or Support Vector Machines (SVM). To implement the algorithm data is “encoded” in a quantum state using a data-dependent PQC that is applied to an initial reference state, usually $|0\rangle^{\otimes n}$ where n is the number of qubits. The encoding procedure can be loosely interpreted as a mapping of a feature vector to a corresponding quantum state, in a similar way to what is done for classical kernels, in which through the kernel function feature vectors are implicitly mapped to a transformed feature space. The encoding procedure in QKE can be done with many different strategies and is currently an active area of research. In general, the values of the entries of the feature vector are used to determine the parameters of the quantum gates in the circuit. From a mathematical point of view, the encoded quantum state $|\phi(\mathbf{x})\rangle$, associated to the feature datapoint \mathbf{x} , is obtained by applying the data-dependent circuit unitary $U_{\mathbf{x}}$, to the initial state $|0\rangle^{\otimes n}$. The quantum circuit performing the encoding procedure is also referred to as “quantum feature map”.

The values of the feature vector might also undergo a preprocessing step before being used for the encoding: for instance, a feature reduction procedure implemented by a Principal Component Analysis (PCA) or an autoencoder might be applied.

To calculate the kernel function evaluation between two feature vectors \mathbf{x} and \mathbf{y} the fidelity between the corresponding quantum states $|\langle\phi(\mathbf{x})|\phi(\mathbf{y})\rangle|^2$ is considered. The choice of the fidelity between the quantum states as a way to calculate the kernel function ensures that the quantum kernel defined in this way is a symmetric positive semi-definite function, as required by kernel theory (Schuld & Petruccione, 2021). The fidelity between quantum states, however, cannot be directly accessed and thus must be estimated through a sampling procedure.

A widely used strategy is to employ the so-called inversion test: by applying the circuit $U_x^\dagger U_y$ to the initial state $|0\rangle^{\otimes n}$ and then performing a measurement in the computational basis it is possible to obtain an estimation of the quantity $|\langle 0^{\otimes n} | U_x^\dagger U_y | 0^{\otimes n} \rangle|^2 = |\langle \phi(\mathbf{x}) | \phi(\mathbf{y}) \rangle|^2$, which corresponds to the probability of obtaining the state $|0\rangle^{\otimes n}$ when performing a measurement on the state $U_x^\dagger U_y | 0^{\otimes n} \rangle$ on the computational basis. Other estimation methods can be used, such as the Swap test. The kernel matrix obtained in this way, however, might not be positive-definite because of the sampling error and therefore some extra processing on the matrix is needed to obtain a well-defined Gram matrix.

Some implementations of Quantum Kernel algorithms also include parameters of the feature map that can be variationally optimized, such as the works presented in (Hubregtsen, et al., 2022) and (Miroszewski, 2023), in which the feature map was optimized with respect to a Kernel Alignment loss function. The optimized kernel is then used in the training phase of a ML kernel algorithm.

6.3.3 Quantum instance

The choice of the feature map used to carry out the encoding is an object of research in the QML community. In QML the feature map is commonly composed of alternating rotation gates and entanglement-inducing gates. Among the most popular feature maps used in QML research are the Pauli Feature Map (Havlíček, et al., 2019) and the real amplitudes feature map.

6.3.4 Quantum platforms

Superconducting, ion trap and photonic hardware are the most widespread QPU technologies. In the available literature, kernel methods have been developed on different platforms, mostly superconducting QPUs and simulators (Gujju, Matsuo, & Raymond). A 2-qubit quantum kernel circuit has also been tested on photonic hardware (Bartkiewicz, Cernoch, & et al., 2020).

6.3.5 Number of qubits

The number of qubits needed to implement the QKE algorithm depends on several factors such as the dimensionality of the feature vectors and their processing, if applied.

Another factor that plays a role in determining the number of qubits is the technique used to estimate the fidelity between quantum states.

6.3.6 KPIs for digital quantum computing

The depth of the circuit depends on the structure of the employed feature map. Usually, when implementing a feature map consisting of alternating rotation and entanglement-inducing gates the basic circuit scheme is repeated for a specified number of times and therefore the choice of such number of repetitions will affect the circuit depth. The encoding scheme also affects the number of entangling gates and thus the total depth of the circuit. Moreover, the required depth depends on the technique used to estimate the fidelity between the quantum states: for instance, the routine estimating the fidelity by applying measurements in the state $U_x^\dagger U_y | 0^{\otimes n} \rangle$ requires a circuit depth that is double the depth of the feature map circuit, whereas the Swap test requires in general a less depth overhead.

6.3.7 Proposal of targeted quantum platforms

The main company providing commercial availability of quantum hardware have published some roadmap regarding the planned future development of their technology. For instance, IBM has recently announced its intention to build a 100000 qubits quantum computer by 2033, while PsiQuantum expects to launch its commercially available quantum computer by 2025.

6.4 Use case n°4: SAR Raw Data Processing (FZJ)

6.4.1 Problem instance

Synthetic Aperture Radar (SAR) is an active imaging technique that has had a significant impact on remote sensing. In SAR imaging, microwave signal pulses are sent to the analyzed area by an airborne or spaceborne radar system. Then, the backscattered echo signals are collected and sampled by the radar. The result of the acquisition phase is a 2-dimensional raw digital signal $s(\tau, \eta)$. For Sentinel-1, this signal is part of the Level-0 products. SAR image formation consists in generating an intensity image that gives a visual description of the physical properties of the analyzed area, starting from the acquired raw signal.

A number of compression and correction steps related to the physical setting of the imaging system are applied to the raw signal. Multiple SAR image formation algorithms are available in the literature, which differ in time performance and image quality (Cruz, Véstias, Monteiro, Neto, & Duarte, 2022). The first algorithm designed for spaceborne image processing is the Range Doppler Algorithm (RDA). The current algorithm used for generating Sentinel-1 higher level products is based on RDA. The basic implementation consists in the following steps:

- Range compression: a range Fast Fourier Transform (FFT) is performed on the raw signal $s(\tau, \eta)$, a frequency domain matched filter $G(f_\tau)$ is multiplied and the range inverse FFT is applied

$$s_{rc}(\tau, \eta) = IFFT_\tau[FFT_\tau[s(\tau, \eta)]G(f_\tau)]$$

- Azimuth FFT: an azimuth FFT is applied to the obtained signal

$$s_1(\tau, f_\eta) = FFT_\eta[s_{rc}(\tau, \eta)]$$

- Range Cell Migration Correction (RCMC): a compensation added to signal, due to the fact that the distance between points on the ground and the receiving antenna (e.g., the slant range) is not constant, due to the azimuth movement of the platform. For small slant angles, RCMC can be implemented as an FFT, linear phase multiplier $G_{RCMC}(f_\tau)$, and IFFT (Cumming & Wong, 2005)

$$G_{RCMC}(f_\tau) = \exp\left\{j \frac{4\pi f_\tau \Delta R\{f_\tau\}}{c}\right\}$$

- Azimuth compression: a frequency domain matched filter $H_{az}(f_\eta)$ is multiplied and the azimuth inverse FFT is applied:

$$s_{ac}(\tau, \eta) = IFFT_{\eta}[S_2(\tau, f_{\eta})H_{az}(f_{\eta})]$$

The problem dimension is related to the dimension (N_{τ}, N_{η}) of the raw signal $s(\tau, \eta)$ that is used to form the final image. The signal dimension depends on multiple factors, i.e., extension of the analyzed area, transmitted signal, pulse repetition frequency (PRF), spacecraft configuration (velocity, angle, height), range sample frequency (RSF). The relevant parameters are summarized in **Table 8**. The value ranges are estimated if not otherwise specified.

PARAM ID	DESCRIPTION	DOMAIN	RANGE VALUES	NOTES
N_{τ}	Number of samples of $s(\tau, \eta)$ in range domain	\mathbb{N}	[5k, 50k]	Related to TXPL and RSF and generally larger than $TXPL \times RSF$. Predefined parameter for the specific considered data slice.
N_{η}	Number of samples of $s(\tau, \eta)$ in the azimuth domain	\mathbb{N}	[1, 100k]	Related to PRF and spacecraft configuration. Not analyzed further, as it is complex to estimate, and it can be adapted by selecting a smaller or larger area of interest for each image formation run.
TXPL	Transmitted Pulse Length	\mathbb{R}	[0.00001 s, 0.0001 s]	Time duration of the sent pulses and lower bound for the time duration of the received echo.
RSF	Range Sample Frequency	\mathbb{R}	[10 MHz, 100 MHz]	Sampling frequency of the received echo. Generally higher than the signal maximum frequency (Nyquist condition).

Table 8 Relevant parameters for multiple SAR data.

6.4.2 Quantum algorithm

6.4.2.1 Quantum Fourier Transform

The growing acquisition rate of SAR data would benefit from a computational speedup in data processing. Quantum computing has shown a significant potential in providing a speedup over classical computation in specific cases. In particular, QFT has shown a computational speedup and has been widely employed in the literature, mostly as a component of bigger quantum circuits.

A FFT algorithm is a classical algorithm that performs the Discrete Fourier Transform (DFT) of a signal x_0, \dots, x_{N-1} with lower complexity than the standard algorithm that follows its definition:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k / N}$$

The complexity of a FFT algorithm is $\mathcal{O}(N \log N)$ compared to the $\mathcal{O}(N^2)$ complexity of the DFT. Multiple FFT subroutines are employed in the above presented RDA.

The QFT algorithm is the quantum counterpart of the FFT, where the coefficients of the basis states $|j\rangle$ are the values on which the Fourier transform is applied. An efficient implementation of the QFT is shown in **Figure 11**.

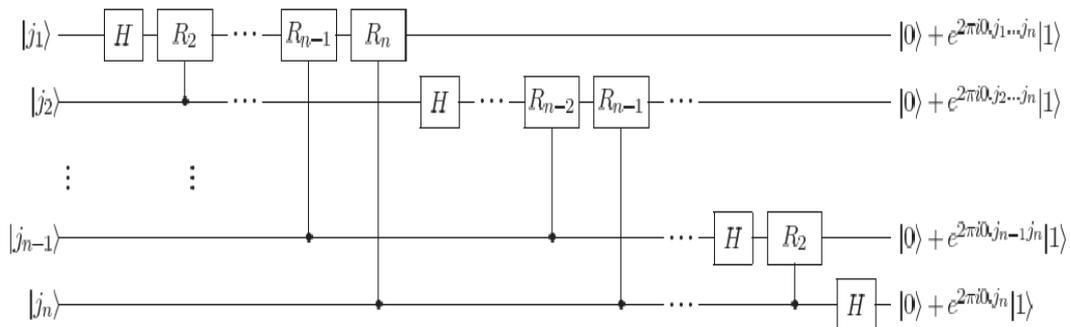


Figure 11 An implementation of the QFT (Nielsen & Chuang, 2010)

The complexity of the QFT is shown to be $\mathcal{O}(\log^2 N)$ (Musk & Sullivan, 2020).

QFT has found its success as part of larger circuits, such as the circuit for quantum phase estimation. It is not meant as a way to compute Fourier transforms, as converting data between classical and quantum domain cancels the speedup.

6.4.3 Quantum instance

The chosen approach makes use of QFT by defining a larger circuit that performs the whole image formation pipeline in the quantum domain. After a preliminary analysis, the version of RDA described in

6.4.1 can be interpreted as a combination of unitary operators. The circuit scheme of a quantum RDA is shown in **Figure 12**.

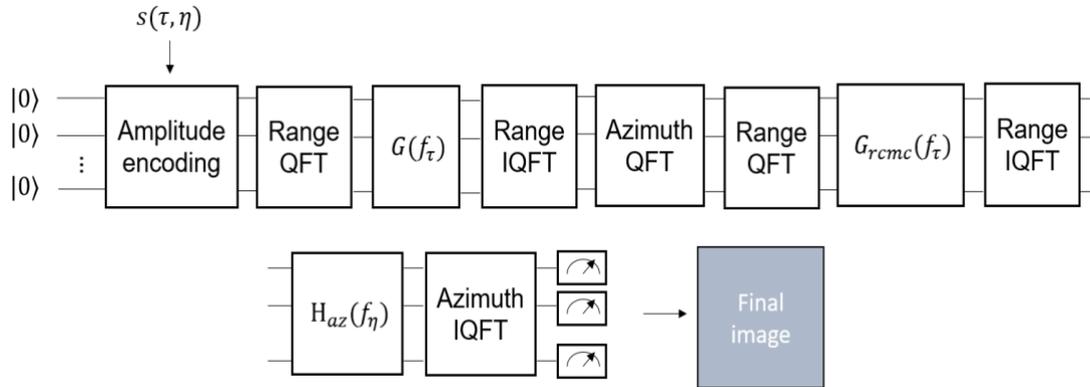


Figure 12 Quantum RDA: proposed quantum circuit approach

6.4.4 Quantum platforms

Being a new research direction, no existing implementations can be found. Superconducting, ion-trap and photonic hardware are potential quantum platforms for this purpose.

6.4.5 Number of qubits

The number of qubits required to execute the algorithm is related to the dimension of the raw signal. The total number of values is $N_\tau \times N_\eta$. Considering the ranges included in **Table 8**, this number is bound to range $[5e^3, 5e^9]$, with a higher likelihood of intermediate values in practical cases. Amplitude encoding requires a number of pixels that is the logarithm of the dimension of the classical data vector. Thus, the number of required qubits can be estimated in the range of $[13, 32]$. This estimation holds true when the whole image is encoded as a single vector, without separating the values in the range and azimuth domain. It is unclear whether the quantum gates, which operate differently on the two domains, can be easily implemented with this assumption.

6.4.6 KPIs for digital quantum computing

The specific implementation of the gates defined in the quantum circuit is ongoing research. This will give more insights on the circuit depth, which is expected to be the main bottleneck for this algorithm.

6.4.7 Proposal of targeted quantum platforms

In theory, the proposed circuit requires a relatively limited number of qubits to be executed. However, the depth and complexity of the circuit poses a requirement on error correction and qubit fidelity. Therefore, trapped ions or photonic qubits are expected to be the preferred choice.

7 Conclusions

The following table summarizes the output of WP2 listing the quantum instance, the KPIs and the targeted quantum platforms for each UC.

USE CASE	QUANTUM ALGORITHM	NUMBER OF QUBITS	KPI	TARGETED QUANTUM PLATFORMS
UC1 Mission Planning for EO Acquisitions	<ul style="list-style-type: none"> QA QAOA QNN <i>Tensor Network</i> 	<ul style="list-style-type: none"> Given by the number of binaries in the cost function (including slack). The connectivity of the problem scale linearly with the system size (better with preprocessing of problem constraints) Size of the input in QNN reduced with a classical NN 	<ul style="list-style-type: none"> Physical qubits scale polynomially with preprocessing. Number of gates per layer is polynomial in the number of qubits. Layers scale logarithmically in the number of qubits. All characteristics of the circuit scale linearly in the number of qubits. 	<ul style="list-style-type: none"> Superconducting Neutral atoms Trapped ions
UC2 Multiple-view Geometry on Optical Images	<ul style="list-style-type: none"> Quantum k-medoids Quantum kernel density clustering 	<ul style="list-style-type: none"> Smaller subimage: 4 keypoints on 8×8 pixels corresponding to 64 binaries in the QUBO. Kernel matrix: 4 qubits circuit. 	<ul style="list-style-type: none"> QUBO size depends on keypoints. All characteristics of the circuit (number of 2-qubit gates, depth) are quadratic in the number of qubits. 	<ul style="list-style-type: none"> Superconducting Neutral atoms Superconducting Trapped ions
UC3 Optical Satellite Data Analysis	<ul style="list-style-type: none"> Quantum Kernel Estimation 	<ul style="list-style-type: none"> Depends on the dimensionality of feature vector 	<ul style="list-style-type: none"> Depends on the feature maps 	<ul style="list-style-type: none"> Superconducting
UC4 SAR Raw Data Processing	<ul style="list-style-type: none"> QFT 	<ul style="list-style-type: none"> Related to raw signal dimension. Amplitude encoding scales logarithmically. Number of qubits in [12, 32]. 	<ul style="list-style-type: none"> Circuit depth is expected to be a bottleneck, still under investigation. 	<ul style="list-style-type: none"> Superconducting Trapped ions

Table 9 WP2 output summary table

END OF DOCUMENT