

QEOPS – Quantum Earth Observation Planning System

Sven Prüfer, Michael Alexander Anderle

German Aerospace Center (DLR), Institute for Space Operations and Astronaut Training
Münchener Str. 20
82234 Weßling, Germany
Sven.Pruefer@dlr.de, Michael.Anderle@dlr.de

Abstract

Even with recent advances in Quantum Computing the question about potential quantum advantage in optimization problems remains open. While this paper does not aim to prove the existence of a quantum advantage, it describes a particular use-case from spacecraft mission planning which is implemented on various quantum computers within the Quantum Mission Planning Challenges project of the German Quantum Computing Initiative at DLR. Besides the detailed problem definition and a formalization of the problem as an Integer Linear Programming problem (ILP), this paper provides a list of experiences and insights gained when trying to apply quantum computing to a real-world problem. In the end we discuss early results comparing classical solvers with quantum optimization algorithms for our particular sub problem of the general EnMAP mission planning system.

1 Introduction

1.1 The EnMAP Mission

The Environmental Mapping Program (EnMAP, see (DLR 2012)) is a German hyperspectral earth observation satellite launched in 2022. It has two cameras on-board with spectral ranges from 420 nm to 1000 nm as well as 900 nm to 450 nm with a resolution of 30 m. EnMAP is on a polar sun-synchronous orbit with a local time descending node of 11am and a 27-day repeat cycle. It uses S-band to receive commands typically via a ground station in Weilheim and downlinks data via ground stations in Neustrelitz as well as Inuvik. The EnMAP mission has already been presented in various papers such as (Storch et al. 2023; Chabrilat et al. 2024).

1.2 Overview of the EnMAP Mission Planning System

The EnMAP mission planning system (MPS) was already described in quite some detail in (Fruth et al. 2018; Lenzen et al. 2023). To summarize, it is based on the German Space Operations Center (GSOC) Reactive Planning framework (RePL, (Wörle et al. 2014)) and its planning library Plains, hence using a feature-rich planning language allowing to model all relevant constraints for the EnMAP mission. RePL is message-driven, meaning that any new input to the system

triggers an immediate update of the planning state, the applied method is referred to as an *algorithm*. The EnMAP algorithms are conceptually illustrated in (Prüfer et al. 2021), and they are typically heuristic searches for a feasible solution. The maximum planning horizon is 14 days into the future, while the earliest commanding (EC) is usually after the upcoming uplink, see Figure 1 for a PintaOnWeb screenshot of the EnMAP MPS. Once a timeline was commanded, EC is moved to the next uplink and the planning horizon shifted correspondingly. The most important case is the ingestion of a new observation request which shall be added to the plan. The corresponding algorithm will trigger a search for opportunities using SCOTA (see (Gross et al. 2021)), which form the possible discrete choices for planning the request. In case all opportunities are blocked, the algorithm tries to remove other acquisitions in order to find a solution that increases the benefit. In most situations, this is quite fast, however this can break down when replanning e. g. the whole planning horizon. To be able to deal with such large planning cases GSOC is investigating quantum algorithms for optimization problems as they may provide a better asymptotic run-time duration when solving larger instances.

From this description of the common planning problem for EnMAP we derive our sample problem for this paper. Note that this is slightly simplified compared to the problem formulation currently used in the production EnMAP MPS, e. g. we assume that there is no on-board memory compression. To recall the nomenclature, a customer submits a *request* which has multiple *opportunities*. If an opportunity is planned, the MPS “earns” the *benefit* of the request. A request also specifies how large the corresponding acquisition will be and hence how much data it will use on-board. Thus, we can define the *memory consumption* as a function of time, determined by the total amount of data of the planned acquisitions minus the downlinked data until a given time. Additional constraints include *gapless aux telemetry*, which means that every 24 hours there must be at least one downlink in order to have enough ranging data for Flight Dynamics to determine the orbit. The goal of the optimization problem is to maximize the total benefit while respecting all constraints and minimizing memory consumption. A detailed integer program formulation is given in Section 2.

To keep the problem feasible, the considered problem lacks a few more involved aspects of the EnMAP mission

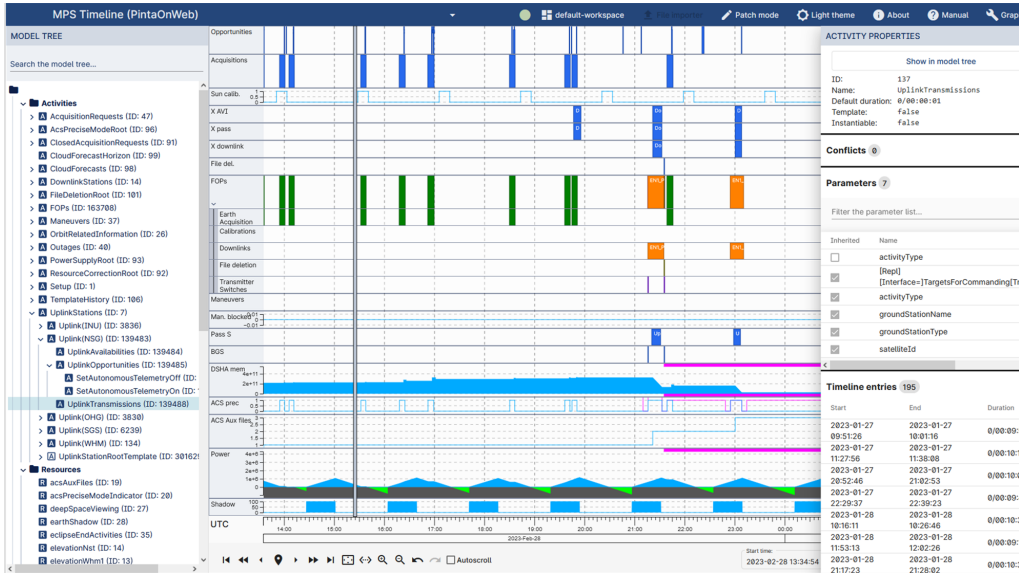


Figure 1: The EnMAP PintaOnWeb view of the planning model.

planning system such as back-to-back imaging, calibrations, area coverage, cloud forecasts or instrument mode switches. Some of these can be easily included during post-processing (and hence are uninteresting for potential quantum algorithms), others, such as cloud forecasts, can be taken into account by modifying the benefit values correspondingly. The model includes slew times for targets by adding configurable but fixed *guard times* to each image opportunity both to the start and the end. As long as the guard times cover the maximum slew time of the satellite it is hence assured that the opportunities do not overlap. To make this more efficient, the actual EnMAP MPS uses so-called *back-to-back imaging* (see (Lenzen et al. 2023)) which is easily integrable into our model by doing pre-computations for possible back-to-back sequences and adding these as separate opportunities with a binary decision variable and corresponding constraints. Since this will not change fundamentally the problem but instead increase the variable and constraint size only, we will not consider this in this paper.

1.3 Quantum Computing and QMPC

In the recent past, quantum computing has gained significant attention in various fields such as machine learning, material science, and optimization. Besides the various applications and sample implementations of algorithms, the quantum computing hardware itself has made a lot of progress in recent years, see e. g. (Arute et al. 2019; Gambetta 2022; Quantum et al. 2025). In addition to demonstrations of quantum supremacy and error correction, there are gate-based quantum computers with 156 qubits generally available, a size that is commonly referred to as *utility scale*, e. g. (Gambetta 2022). In the field of optimization, there are many promising approaches to apply quantum computers in the hopes of finding asymptotically faster algorithms or better solutions within a fixed time window, see (Abbas et al. 2024). Still, no proof of so-called quantum advantage is yet

known for optimization problems, which means that so far no industry-relevant optimization challenge could be solved faster or better on a quantum computer than on a classical one.

Besides gate-based quantum computing there is also the paradigm of quantum annealing which refers to computers that use adiabatic evolution of a Hamiltonian to find solutions to optimization problems. While these tools are very much suited for unconstrained binary optimization problems, it is still somewhat unclear how well they can be applied to realistic mission planning problems due to the difficulty of implementing various complicated constraints. Even though it is possible to reformulate the planning problem in this language there is a considerable overhead associated with this translation.

Notice that most quantum algorithms target combinatorial optimization problems, i. e. finite sets of integer (or binary) variables. When planning an activity A on a timeline running from t_0 to t_1 it is natural to consider t_0, t_1 as continuous variables of the optimization problem. Hence, spacecraft planning problems are rather of a mixed type, involving both discrete and continuous variables. Unfortunately, there are only very few quantum optimization algorithms that can deal with such problems and hence there is a lot less known about performance or possible quantum advantage in this area, see e. g. (Abbas et al. 2024). We will give an overview of some quantum algorithms for optimization problems and then discuss how they can be applied in spacecraft planning in section Section 3.

1.4 Quantum Computing for Spacecraft Mission Planning

(Stollenwerk et al. 2021) use quantum computing, specifically a D-Wave Quantum Annealer, to optimize the schedule for an Earth observation satellite. They focus on an ag-

ile satellite and take slewing times into consideration which exclude combinations of certain image opportunities. However, as they only consider acquisitions, all their variables are naturally binary while for QEOPS, we also take into account downlinks, the beginning and ending of which can be planned continuously within an interval. In terms of constraints, they only consider a subset of those in QEOPS, for example, we have in addition the gapless aux telemetry and memory constraints (type 5 and 6) which tend to be significantly more complicated to model than the common ones. While they perform evaluations on real quantum annealing hardware, we limit our experiments to simulated annealing.

The authors of (Quetschlich et al. 2023) consider a similar planning problem for an earth observation satellite with only acquisition opportunities and constraints for certain combinations of acquisitions due to the required camera rotation time (again similar to our No Overlap Constraint). They formulate this problem as a QUBO and solve it with simulated variational quantum algorithms.

(Rainjonneau et al. 2023) are also concerned with a satellite scheduling problem where acquisition opportunities must be selected to form a valid schedule such that the total benefit is maximized. They first formulate this as an integer linear programming problem and then translate it to QUBO form, similar to our workflow. While they find classical branch-and-cut based solvers to perform well on optimizing the linear problem, they do not achieve satisfactory performance on the QUBO optimization from neither a hybrid classical-quantum annealing nor a purely classical optimizer. In addition, they also employ reinforcement learning with a hybrid quantum neural network as policy model.

In (Makarov et al. 2024) a satellite planning problem with similar constraints to our type 2 and 4 constraints is investigated, as well as a simpler version of our memory constraint, by enforcing fixed upper bounds for the memory without considering the planning of downlinks to free memory. Furthermore, the satellite they focus on has two cameras and may take stereo pictures giving rise to a constraint that from certain triples of acquisition requests only two may be planned. They optimize their problems in QUBO form with Quantum Annealing on two different D-Wave QPUs as well as with a D-Wave hybrid optimizer, as well as simulated QAOA.

In summary, our work differs from previous work in that it does not only consider the selection of image acquisition opportunities, but also the flexible planning of downlinks within predefined intervals, which interacts with acquisition planning through the constraint of limited memory and is subject to a limit for the gaps between downlinks.

2 ILP Formulation of the EnMAP Problem for QEOPS

While the QEOPS problem is in principle of a mixed-integer linear programming (MILP) type we present a discrete ILP version here, since the MILP version is significantly more complex and requires dedicated techniques in the context of quantum computing.

2.1 Input

To define the problem in detail, we first give a few definitions which we will use throughout the remaining paper.

- Any point in time t is defined as a real number $t \in \mathbb{R}$.
- Any time interval $w \in W$ (window) is defined as an ordered pair of two points in time:

$$W = \{(t_1, t_2) \in \mathbb{R}^2 | t_1 < t_2\}$$
- In some constraints, we use the well-known “big-M” trick and assume that $M \in \mathbb{R}$ is a sufficiently large number.

The time interval $T = (T_0, T_1) \in W$ for the schedule to be planned, is also referred to as *planning horizon*.

Within this horizon, a set of acquisition requests R , each corresponding to a single request by a researcher for a target and with some parameters, is to be fulfilled if possible.

For every such acquisition $r \in R$ request we have a set of precalculated opportunities O_r corresponding to time intervals, with $O = \cup_r O_r$ being the set of all opportunities. For an acquisition opportunity $i \in O$, $s_i, s'_i \in \mathbb{R}$ are the start and end times. Any request $r \in R$ has an attached memory consumption $m_r \in \mathbb{R}$: The amount of memory the request requires depends mainly on the size of the image which is specified by the original requester and hence constant and known for all opportunities of a request. We can thus also use m_i to denote the memory consumption m_r of the corresponding request r for an opportunity i .

Any opportunity $i \in O$ has an attached benefit $b_i \in \mathbb{R}$: The benefit refers to an abstract measure how important the overall project considers an acquisition opportunity. There are various aspects that factor into this number, e. g. the customer priority, emergencies, cloud prediction or how much time remains to schedule the corresponding request.

In addition, we have a set of downlink visibilities D which correspond to time intervals during which a downlink ground station is visible and available, and hence a downlink can be scheduled. This is described by a set $\{d_1, \dots, d_{|D|} | d_i \in W\}$ of time intervals. A downlink visibility $d_j \in D$ is hence defined by s_j, s'_j as its start and end times. Downlinks also have a download speed $v \in \mathbb{R}$ associated with them, allowing to model different modes or downlink speed. Hence, a downlink d of length $|d|$ will reduce the memory consumption by $v|d|$.

2.2 Discrete Decision Variables

Notice that image opportunities are scheduled in a binary fashion as they were precomputed and are considered an input to the MPS. See Figure 2 for a graphical overview.

Hence, x_i is the binary variable indicating whether acquisition opportunity i is planned and y_j is the binary variable indicating whether downlink visibility j is planned.

We also need a way to model the start and end times of the downlink within visibility j , which would initially be continuous variables, with discrete variables, hence discretizing the downlinks.

A key observation is the following: we can precompute all times when a downlink visibility window intersects with any other downlink visibility or acquisition opportunity and use

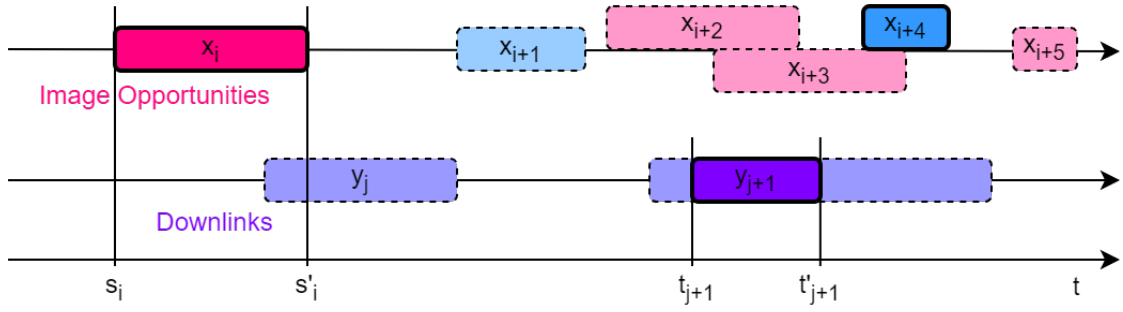


Figure 2: An illustration of the variables of the QEOPS problem. Notice that the image opportunities are split into subsets according to their corresponding request. Also, the start and end times of the image opportunities are denoted by s_i and s'_i , but they are fixed parameters rather than decision variables. Even though the downlink j has start and end time variables associated to it, they should be irrelevant to the problem since in the example $y_j = 0$ and hence the downlink is not scheduled.

those times as the possible start or end times for the downlink, in addition to the start and end of the visibility window. Letting the solver choose from those times is sufficient to enable all possible combinations of downlinks and acquisitions. Of course, the downlinks formed from these start and end times will often be too long, i.e. they would continue to downlink when there is no data left in the memory to transmit. To avoid this we can shorten these downlinks as necessary during post-processing.

The logic is illustrated in Figure 3. We refer to the found sets of possible start and end times of downlink j as t_j or t'_j , respectively. We then add corresponding binary variables $v_{j,t}$, $t \in t_j$ and $v'_{j,t'} \in t'_j$ indicating for each possible downlink start and end time whether it was selected.

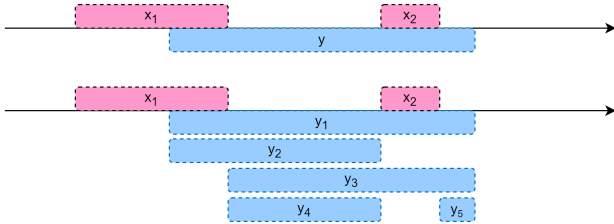


Figure 3: Given a downlink shown as on top with two overlapping image opportunities, we can find the five possible maximal intervals $y_1, \dots, y_5$. Notice that if y_5 is shorter than allowed, we can immediately drop this case during pre-processing.

We also add the constraint that exactly one start and end time must be selected per downlink if the downlink is scheduled, and none otherwise:

$$\forall j \in D : \sum_{t \in t_j} v_{j,t} = y_j \quad (1)$$

and

$$\forall j \in D : \sum_{t \in t'_j} v'_{j,t} = y_j. \quad (2)$$

2.3 Discrete Constraints

Let us describe the applied constraints in a bit more detail.

Constraint Type 1: Downlink Minimum Length If a downlink is scheduled, its length must at least be the minimum length $L_{\min} \in \mathbb{R}$. Since Equation (1) and Equation (2) require that always either exactly one start and end time, or neither any start nor any end time must be selected, we can formulate this constraint as:

$$\forall j \in D : \left(\sum_{t' \in t'_j} v_{j,t'} \cdot t' - \sum_{t \in t_j} v_{j,t} \cdot t \right) \geq y_j \cdot L_{\min} \quad (3)$$

Constraint Type 2: At Most One Acquisition Opportunity Planned Per Request

$$\forall r \in R : \sum_{i \in r} x_i \leq 1 \quad (4)$$

Constraint Type 3: No Unplanning We may sometimes be given sets $\bar{O}$ and $\bar{D}$ of acquisition opportunities and downlinks which were planned in a previous iteration and must still be planned in this iteration. We can easily model this with the constraints

$$\forall i \in \bar{O} : x_i = 1 \text{ and } \forall j \in \bar{D} : y_j = 1, \quad (5)$$

or by leaving out these variables and adapting all constraints correspondingly.

Constraint Type 4: No Overlap We must prevent scheduling overlapping acquisitions and downlinks because the satellite execute both procedures at the same time. For the purpose of this constraint, additional guard times g_o, g'_o apply at the start and end of each opportunity and g_d, g'_d at the start and end of each downlink, respectively. These constant time values account for e.g. pre-heating or slewing between acquisitions. We use $\bar{i}$ and $\bar{j}$ to refer to the intervals of an acquisition opportunity i and downlink visibility j extended by the applicable guard times.

Since acquisitions are only scheduled in a binary fashion, and there is only a finite number of possible start/end times for the downlinks, we can efficiently check all combinations and prohibit those that are invalid because they overlap when extended by their guard times. This yields constraints like

$$x_i + x_j \leq 1. \quad (6)$$

Constraint Type 5: Gapless Aux Telemetry The maximum gap between two consecutive downlinks must not exceed the length L_{aux} , which can be expressed as a so-called sliding window constraint, see Figure 4 for an illustration. This feature is called gapless aux telemetry and is necessary to ensure sufficient ranging data for Flight Dynamics orbit determination.

We verify sliding windows of size L_{aux} and ensure that at least one downlink is active during any such window. However, it is clearly enough to check this only for finitely many windows, namely at the planning horizon start, at any downlink visibility start time and L_{aux} before the planning horizon end. Let us denote the set of these sliding window starting times by C , noticing that this is determined during preprocessing.

We then add the constraint that

$$\forall c \in C : \sum_{j \in D} f(j, c) \geq 1$$

where $f(j, c)$ indicates whether downlink j satisfies the gapless aux telemetry constraint by starting within L_{aux} after c if it is scheduled. We refrain from writing down an explicit formula for f , but this is straight-forward given the description above.

Constraint Type 6: Memory Below Upper Bound EnMAP has a maximum memory size for images which needs to be obeyed, hence we have a maximum allowed memory usage $M_{\text{max}} \in \mathbb{R}$ and an initial memory consumption $m_{\text{init}} \in \mathbb{R}$ at the beginning of the planning horizon.

We assume that the memory usage increases at the end of an acquisition by the total amount of memory required for the acquisition and decreases at the end of a downlink by the amount of memory transmitted by the downlink, see Figure 5.

Hence, the points in time when we need to verify the memory use are at the end times of all downlinks just before the memory is released and at the planning horizon end, in particular we have only finitely many values to verify.

An obvious way to model this constraint is to track the amount of memory at these times by adding the memory of the scheduled acquisitions to the prior value after the last downlink, resulting in linear inequality constraints for every downlink end. Unfortunately, this *naive* implementation may cause negative values and hence model an invalid memory resource behavior, see Figure 6.

While it is possible to model the cutoff using continuous variables for the memory value or many binary ones for a binary count at these times and some auxiliary variables we want to model this in a simpler way.

The key idea is as follows: For any downlink d , and given any schedule, there is either no earlier downlink that was too long or there is a latest one, denote it by d_0 . In the first case, the naive calculation is actually correct and hence the upper bound is enforced correctly. In the second case, we can add a naive memory constraint which assumes that after this downlink d_0 the memory is zero, and we start counting naively from that point on. We can now add these constraints for any downlink d and any prior downlink d_0 , hence the

number of constraints scales quadratically with the number of downlinks. This is sufficient as the individual constraints are either weaker (when considering a downlink later than d_0) or do not matter as a later downlink sets the memory resource to zero (when considering a downlink earlier than d_0).

This way we can model the resource constraint by a set of linear inequalities.

2.4 Goal

We consider two goals:

1. Maximize the total benefit from acquisitions: $\sum_{i \in O} b_i \cdot x_i$
2. Minimize downlinks: $-\sum_{j \in D} y_j$

These can be weighted with factors $w_1, w_2 \in \mathbb{R}$, resulting in the objective function:

$$w_1 \cdot \sum_{i \in O} b_i \cdot x_i - w_2 \cdot \sum_{j \in D} y_j \quad (7)$$

2.5 Classical Implementation

We implemented the linear integer models with the *Google OR-Tools* library (Perron and Furnon 2024), using the open-source solver SCIP (Achterberg et al. 2008; Achterberg 2009). This was used as a baseline for comparison to any quantum algorithms.

3 Overview of Quantum Algorithms for Scheduling Problems

As described roughly in the introduction, most quantum algorithms target discrete optimization problems, where binary variables map e. g. to single qubits and measurements of basis elements $|0\rangle$ and $|1\rangle$ are associated with their respective classical values. For integer based variables, multiple encodings may be chosen such as one-hot, binary or domain-wall encoding, hence reducing the problem to a binary one.

Once reformulated as a binary problem (see Section 2.3 for our use-case), most algorithms require a certain treatment of the constraints, typically by implementing them as soft constraints, i. e. a penalty for violating the constraint is added to the target function. By rescaling this contribution one can make sure that optimal solutions must satisfy the constraints and are hence feasible. Nevertheless, as quantum algorithms are probabilistic in nature one typically observes a certain amount of non-optimal, and hence potentially unfeasible, solutions. The goal of the quantum algorithm is thus to reduce the observed amount of unfeasible or non-optimal solutions. Since many applications in spacecraft planning require strict adherence to the constraints (as the typically model certain safety-relevant aspects of the mission), we want to generally include feasibility rates besides observed best values or time-to-solution as metrics for comparisons.

Since our problem typically contains inequality constraints, let us briefly sketch how to include them into the objective function. Assume that we have four binary variables $x_1, \dots, x_4$ denoting whether certain activities are planned or

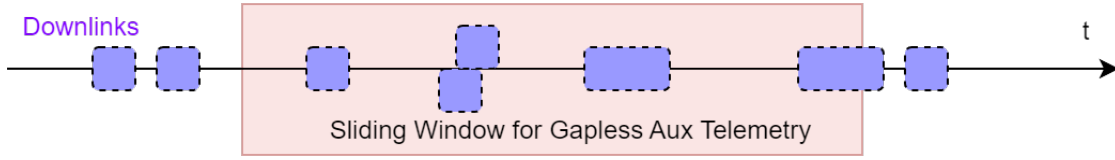


Figure 4: A gapless aux telemetry constraint is modelled as inequality constraints over all downlink decision variables corresponding to a sliding window.

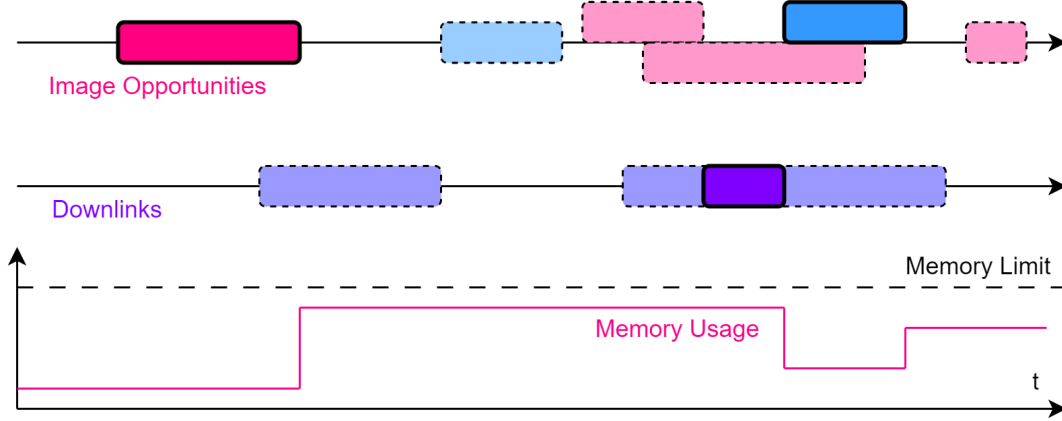


Figure 5: QEOPS timeline with discrete opportunities in the first row and downlink visibilities in the second row. At the bottom you can see the corresponding memory usage resource as a function of time

not. In case we need to implement a constraint that at most two out of these four can be one, we first transform the inequality

$$x_1 + x_2 + x_3 + x_4 \leq 2$$

into an equality by introducing so-called slack variables a_1 and a_2 and requiring

$$x_1 + x_2 + x_3 + x_4 + a_1 + a_2 = 2.$$

Now we can replace the target function $f(x)$ by

$$F(x, a) = f(x) + \lambda(x_1 + x_2 + x_3 + x_4 + a_1 + a_2 - 2)^2$$

for a suitable parameter λ . This allows us to include both the objective function and constraints (equalities and inequalities) into a single *quadratic* binary polynomial which is without any further constraints, called a QUBO. While this formulation is a common entry point for most quantum algorithms, it is somewhat unnatural for classical optimizers which should be kept in mind when doing a benchmark comparison.

This QUBO formulation can either be used by a Quantum Annealing algorithm, such as provided as a service by D-Wave, or be equivalently reformulated as a *Hamiltonian* H for gate-based quantum computers. The notion of a Hamiltonian comes from physics where it denotes a quantum observable measuring the energy of a system. This means that by encoding the binary variables on qubits and expressing the QUBO as a Hamiltonian operator we can reformulate our optimization problem as an energy minimization problem.

The goal is thus to find a ground state $|\Psi\rangle$ of the Hamiltonian H . One type of algorithms operating on such Hamiltonians are so-called variational quantum algorithms, e. g. Variational Quantum Eigensolver (VQE) and Quantum Approximate Optimization Algorithm (QAOA). Testing such variational quantum algorithms and various variants for the EnMAP problem in QEOPS is still work in progress.

Another relevant type of problems for practical applications is Mixed Integer Problems (MIP), i. e. a subset of the variables are continuous. As it is non-trivial to map such variables to qubits and henceforth solve the optimization problem using a quantum circuit, there are hybrid algorithms that split the problem into a continuous and a discrete part. If done efficiently, one can then apply a quantum algorithm to the discrete part and look for a possible quantum advantage. Notice that such splitting algorithms are well-known in classical optimization, such as Alternating Direction Method of Multipliers (ADMM, see (Glowinski and Marroco 1975)), or Benders Decomposition. For QEOPS, we have already formulated a mixed-integer version of the problem and have successfully applied Benders decomposition with a suitable discrete quantum optimization algorithm.

Benders decomposition was introduced in e. g. (Benders 1962). Roughly, it works by splitting the problem into an integer master and a continuous sub problem, which are iteratively solved and updated. The solution of the continuous part provides bounds for the original optimization problem which are then used to iteratively solve the discrete part (in our case using a quantum algorithm) until convergence. This is illustrated in Figure 7.

4 Results and Discussion

In this section we compare the classical solutions using SCIP and CBC of the discrete QEOPS problem with a (simulated) Quantum Annealing solution using the D-Wave Ocean library.

4.1 Evaluation and Methodology

Test data was generated by randomly slicing intervals of different length from real EnMAP data. The randomness was seeded for reproducibility, and all data was persisted.

The classical solver was run with a time limit of 10 minutes. It turned out that the difficulty of the problem instances depends more on the particular structure of the constraints (or input data) than on the pure size. This causes slightly strange results as long optimization runs get aborted and hence decrease the average runtime. Furthermore, CBC had some trouble parsing a part of the sample inputs due to the size of some constraints. Since these instances are expected to be more difficult, this also filters out more difficult examples. All tests were executed on a 24-core i9-13590HX CPU with 128 GB memory.

4.2 Results

Some convergence results for the CBC optimizer are shown in Figure 8. We can see that in these samples even the first found feasible solution is rather close to the optimum (typically less than 10% off), so these cases are easy. It is expected that the graphs for more days (and hence more variables and more constraints) tend to start more to the right, corresponding to longer computation times.

In Figure 9 we plot the average runtime of the CBC optimization runs binned with respect to the number of days used for defining the test cases which correlates with problem size. The decrease in average run time for 10 days is likely due to the accidental removal of larger and hence longer optimization runs. Apart from this phenomenon one sees the very large variance of the run times due to the apparent dependence of the hardness of the problem instance on the concrete constraints which in turn depend on the concrete input data.

The simulated annealing for each QUBO was performed with 5000 reads and remaining parameters unchanged from their default values.

When optimizing the QUBO, we had to restrict the problem sizes to intervals of 1 to 23 hours, since large inputs lead

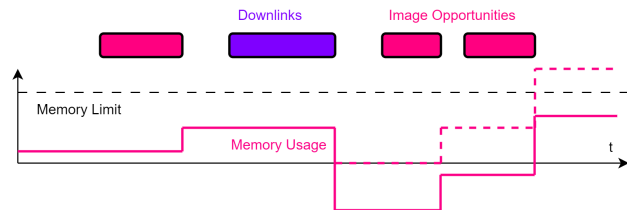


Figure 6: The naive memory resource becomes negative after the downlink that was too long, while it should have been cut off at zero. The real value, shown with the dashed line, is higher and hence may violate the upper bound

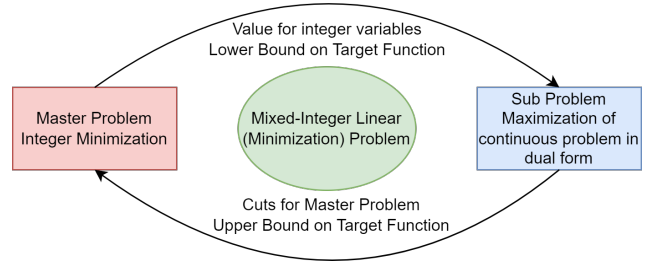


Figure 7: Overview of Benders decomposition. The discrete master problem can be solved using a quantum computer

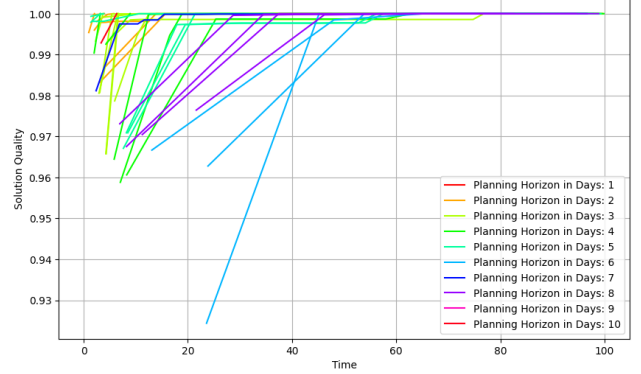


Figure 8: Relative objective function values over time for individual problem cases for the CBC solver, but only those for which the solver could prove optimality within 10 minutes

to unfeasible solutions with very high numbers of violations, hence the annealing does not seem to find the ground state. As can be seen in Figures 10 and 11, the number of constraint violations found in the solution depends on the problem size as expected, where the problem size correlates with the planning horizon length. Interestingly, inputs up to 15 hours mostly result in solutions without violations, which is long enough for the most common scenarios.

When we solved the same input scenarios with the linear model, the classical SCIP solver found the solution within a maximum of 7 seconds in every case, usually even within 1 second. To get a better understanding of the limits of SCIP, we ran inputs with a length of 1 to 10 days. Figure 12 shows that the classical solver has a boundary at about 300 variables where it starts to always use the maximum time and is not able to prove optimality and thus exit earlier. It did, however, find a valid solution in every instance, which may be sufficient for operations.

When comparing Figures 10 and 12 notice that the variable count in a QUBO is typically much higher than that for a linear model due to required slack variables for converting the constraints to penalty terms.

4.3 Discussion and Outlook

When benchmarking classical and quantum algorithms there is always the problem of finding common test cases as quan-

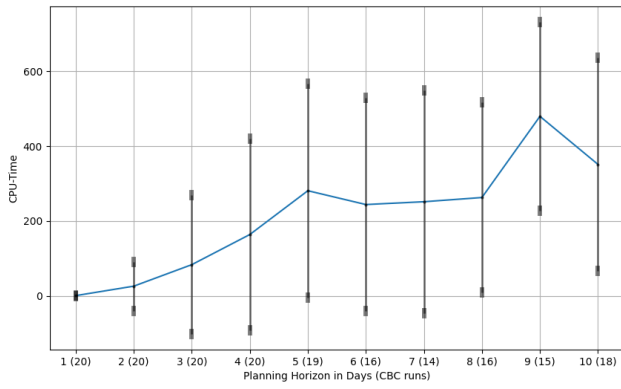


Figure 9: Run time averages in seconds for CBC optimizer over different number of days, which loosely relates to the number of variables

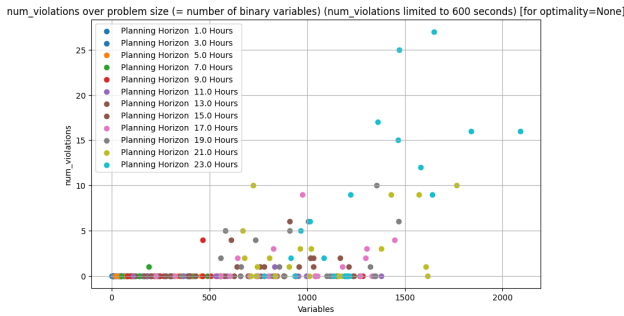


Figure 10: Number of constraint violations as a function of QUBO variables for the Simulated Annealing using the D-Wave simulator

tum computers are typically still rather limited. We can see this in this paper when the quantum annealing does not find feasible solutions anymore for larger problem sizes. However, there is still quite a bit of improvements possible by e. g. trying to rescale the penalty terms in the QUBO formulation.

Both CBC and SCIP perform well in smaller problem sizes but fail to run to optimum for a full week planning within 10 minutes. One of the main takeaways is the dependence of the solution quality and speed on the concrete modelling for all algorithms. In particular the inclusion of the memory constraint and the sliding windows for gapless aux telemetry offer a lot of potential for improvement, although one has to make sure that the memory constraints remain satisfied as they are crucial to EnMAP.

As goals for the near future we intend to finalize the implementation of various other variational quantum optimization algorithms to run a larger benchmark. Furthermore, we can include more aspects of the EnMAP MPS to bring these simulations closer to the operational situations. One of the final goals of the QMPC project is to demonstrate the usage of these algorithms within the actual EnMAP MPS.

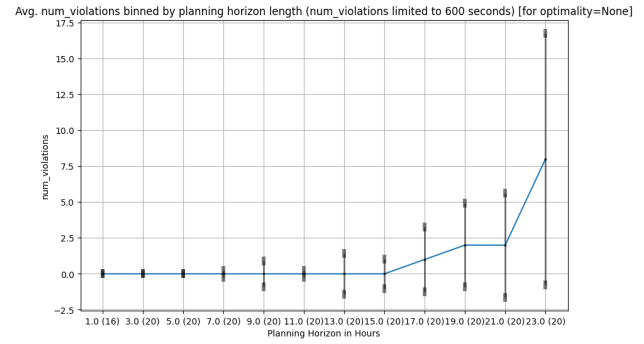


Figure 11: Number of constraint violations as a function of planning horizon length in hours. Simulations were done using D-Wave Simulated Annealing



Figure 12: CPU-Time required by SCIP as a function of the number of variables, cut off at 600s

Acknowledgements

This project was made possible by the DLR Quantum Computing Initiative and the Federal Ministry for Economic Affairs and Climate Action; <https://qci.dlr.de/en/qmpc/>

References

- Abbas, A.; Ambainis, A.; Augustino, B.; Bärtschi, A.; Buhrman, H.; Coffrin, C.; Cortiana, G.; Dunjko, V.; Egger, D. J.; Elmegreen, B. G.; et al. 2024. Challenges and opportunities in quantum optimization. *Nature Reviews Physics*, 1–18.
- Achterberg, T. 2009. SCIP: solving constraint integer programs. *Mathematical Programming Computation*, 1(1): 1–41.
- Achterberg, T.; Berthold, T.; Koch, T.; and Wolter, K. 2008. Constraint Integer Programming: A New Approach to Integrate CP and MIP. In Perron, L.; and Trick, M. A., eds., *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 6–20. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-68155-7.
- Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J. C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F. G.; Buell, D. A.; et al. 2019. Quantum supremacy using a pro-

grammable superconducting processor. *Nature*, 574(7779): 505–510.

Benders, J. F. 1962. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.* 4(1): 238–252.

Chabrilat, S.; Foerster, S.; Segl, K.; Beamish, A.; Brell, M.; Asadzadeh, S.; Milewski, R.; Ward, K. J.; Brosinsky, A.; Koch, K.; Scheffler, D.; Guillaso, S.; Kokhanovsky, A.; Roessner, S.; Guanter, L.; Kaufmann, H.; Pinnel, N.; Carmona, E.; Storch, T.; Hank, T.; Berger, K.; Wocher, M.; Hostert, P.; van der Linden, S.; Okujeni, A.; Janz, A.; Jakimow, B.; Bracher, A.; Soppa, M. A.; Alvarado, L. M.; Buddenbaum, H.; Heim, B.; Heiden, U.; Moreno, J.; Ong, C.; Bohn, N.; Green, R. O.; Bachmann, M.; Kokaly, R.; Schodlok, M.; Painter, T. H.; Gascon, F.; Buongiorno, F.; Mottus, M.; Brando, V. E.; Feilhauer, H.; Betz, M.; Baur, S.; Feckl, R.; Schickling, A.; Krieger, V.; Bock, M.; La Porta, L.; and Fischer, S. 2024. The EnMAP spaceborne imaging spectroscopy mission: Initial scientific results two years after launch. *Remote Sensing of Environment*, 315: 114379.

DLR. 2012. EnMAP project website. <https://www.enmap.org/>. Accessed: 2025-02-13.

Fruth, T.; Lenzen, C.; Gross, E.; and Mrowka, F. 2018. The EnMAP Mission Planning System. In *15th International Conference on Space Operations, SpaceOps 2018*.

Gambetta, J. 2022. Expanding the IBM Quantum roadmap to anticipate the future of quantum-centric supercomputing. *IBM Research Blog*.

Glowinski, R.; and Marroco, A. 1975. Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2): 41–76.

Gross, E. M.-L.; Fruth, T.; Dauth, M.; Petrak, A.; and Mrowka, F. 2021. SCOTA: The Mission Planning Orbit Analysis Tool at GSOC. In *IAF Space Operations Symposium 2021 at the 72nd International Astronautical Congress, IAC 2021*.

Lenzen, C.; Wörle, M. T.; Prüfer, S.; Krenss, J.; Wiesner, S.; and Fruth, T. 2023. EnMAP MPS: Challenges, Enhancements and Evaluations of the Early Mission Phase. In *17th International Conference on Space Operations, SpaceOps 2023*.

Makarov, A.; Pérez-Herradón, C.; Franceschetto, G.; Taddei, M. M.; Osaba, E.; Del Barrio, P.; Villar-Rodriguez, E.; and Oregi, I. 2024. Quantum optimization methods for satellite mission planning. *IEEE Access*.

Perron, L.; and Furnon, V. 2024. OR-Tools.

Prüfer, S.; Lenzen, C.; Wiesner, S.; Krenss, J.; Wörle, M. T.; and Mrowka, F. 2021. Use Cases and Algorithms of the EnMAP Mission Planning System. In *16th International Conference on Space Operations (SpaceOps 2021)*.

Quantum, M. A.; Aghaee, M.; Alcaraz Ramirez, A.; Alam, Z.; Ali, R.; Andrzejczuk, M.; Antipov, A.; Astafev, M.;

Barzegar, A.; Bauer, B.; et al. 2025. Interferometric single-shot parity measurement in InAs–Al hybrid devices. *Nature*, 638(8051): 651–655.

Quetschlich, N.; Koch, V.; Burgholzer, L.; and Wille, R. 2023. A hybrid classical quantum computing approach to the satellite mission planning problem. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 1, 642–647. IEEE.

Rainjonneau, S.; Tokarev, I.; Iudin, S.; Rayaprolu, S.; Pinto, K.; Lemtiuzhnikova, D.; Koblan, M.; Barashov, E.; Kordzanganeh, M.; Pflitsch, M.; et al. 2023. Quantum algorithms applied to satellite mission planning for Earth observation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16: 7062–7075.

Stollenwerk, T.; Michaud, V.; Lobe, E.; Picard, M.; Basermann, A.; and Botter, T. 2021. Agile earth observation satellite scheduling with a quantum annealer. *IEEE Transactions on Aerospace and Electronic Systems*, 57(5): 3520–3528.

Storch, T.; Honold, H.-P.; Chabrilat, S.; Habermeyer, M.; Tucker, P.; Brell, M.; Ohndorf, A.; Wirth, K.; Betz, M.; Kuchler, M.; Mühle, H.; Carmona, E.; Baur, S.; Mücke, M.; Löw, S.; Schulze, D.; Zimmermann, S.; Lenzen, C.; Wiesner, S.; Aida, S.; Kahle, R.; Willburger, P.; Hartung, S.; Dietrich, D.; Plesia, N.; Tegler, M.; Schork, K.; Alonso, K.; Marshall, D.; Gerasch, B.; Schwind, P.; Pato, M.; Schneider, M.; de los Reyes, R.; Langheinrich, M.; Wenzel, J.; Bachmann, M.; Holzwarth, S.; Pinnel, N.; Guanter, L.; Segl, K.; Scheffler, D.; Foerster, S.; Bohn, N.; Bracher, A.; Soppa, M. A.; Gascon, F.; Green, R.; Kokaly, R.; Moreno, J.; Ong, C.; Sornig, M.; Wernitz, R.; Bagschik, K.; Reintsema, D.; La Porta, L.; Schickling, A.; and Fischer, S. 2023. The EnMAP imaging spectroscopy mission towards operations. *Remote Sensing of Environment*, 294: 113632.

Wörle, M. T.; Lenzen, C.; Göttfert, T.; Spörl, A.; Grishchekin, B.; Mrowka, F.; and Wickler, M. 2014. The Incremental Planning System – GSOC’s Next Generation Mission Planning Framework. In *13th International Conference on Space Operations*.