Hybrid quantum-classical multi-cut Benders approach with a power system application

Computers and Chemical Engineering 172 (2023) 108161

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Hybrid quantum-classical multi-cut Benders approach with a power system application

Nikolaos G. Paterakis

Eindhoven University of Technology, Department of Electrical Engineering, Eindhoven, 5600 MB, The Netherlands

ARTICLE INFO

Keywords:
Benders decomposition
Power systems
Quantum computing
Unit commitment

ABSTRACT

Leveraging the current generation of quantum devices to solve optimization problems of practical interest necessitates the development of hybrid quantum-classical (HQC) solution approaches. In this paper, a multi-cut Benders decomposition (BD) approach that exploits multiple feasible solutions of the master problem (MP) to generate multiple valid cuts is adapted, so as to be used as an HQC solver for general mixed-integer linear programming (MILP) problems. The use of different cut selection criteria and strategies to manage the size of the MP by eliciting a subset of cuts to be added in each iteration of the BD scheme using quantum computing is discussed. The HQC optimization algorithm is applied to the Unit Commitment (UC) problem. UC is a prototypical use case of optimization applied to electrical power systems, a critical sector that may benefit from advances in quantum computing. The proposed approach is demonstrated using the D-Wave Advantage 4.1 quantum annealer.

1. Introduction

Quantum computing (QC) is an emerging technology that may help address challenging computational problems. Although significant progress has been made in the recent years, universal error-corrected quantum computers have not been realized yet. Nonetheless, currently available quantum hardware, often called Noisy Intermediate-Scale Quantum (NISQ) (Preskill, 2018), allows applying quantum algorithms in order to identify problems and application areas in which QC can offer an advantage compared to classical computing.

The power and energy sector is an example of an area where computing has always been of paramount importance for system design and operation (Priesmann et al., 2019). However, the complexity induced by the modernization of the energy sector is posing computational challenges that may not be met by classical resources (Tovar-Facio et al., 2021). The energy sector is currently undergoing a transition from fossil-based to zero-carbon energy sources, triggered by global decarbonization targets. The electrical power sector is leading the energy transition through three innovation trends, namely the electrification of end-use sectors, the decentralization of energy resources and extensive digitalization (Lopion et al., 2018). In this context, QC and quantum informatics are expected to find significant applications. First, quantum cryptography can be leveraged in order to enhance the cyber security of power systems, especially those with many distributed energy resources (Tang et al., 2021). Also, QC has the potential to expand the current computational capabilities and facilitate the solution of complex analysis, optimization and data analytics problems in the energy domain (Eskandarpour et al., 2020; Olatunji et al., 2021; Giani and Eldredge, 2021), while maintaining a low energy footprint of computations (Elsayed et al., 2019).

1.1. Quantum computing for optimization

Among the many areas where a quantum speedup is sought, solving combinatorial optimization problems is one of the most prominent (Bernal et al., 2022). In the context of using QC for optimization, a commonly-studied class of combinatorial problems are Quadratic Unconstrained Binary Optimization (QUBO) problems (Kochenberger et al., 2014). A QUBO problem can be stated as $\min_{\mathbf{x}} \mathbf{x}^T \mathbf{Q}\mathbf{x}$ where $\mathbf{x} \in \{0,1\}^n$ and $\mathbf{Q} \in \mathbb{R}^{n \times n}$. QUBO problems can be mapped to Ising models ($\mathbf{x} \in \{-1,1\}^n$) through a linear transformation. The limitations of NISQ hardware have triggered the development of hybrid quantum–classical (HQC) algorithms that exploit both classical and quantum resources. Popular techniques include variational approaches such as the Variational Quantum Eigensolver (VQE) (Peruzzo et al., 2014) and the Quantum Approximate Optimization Algorithm (QAOA) (Farhi et al., 2014), as well as techniques based on Grover's algorithm (Gilliam et al., 2021). However, gate-based QC, which the aforementioned algorithms are designed to exploit, are still at an early development stage and, therefore, their application for solving practical-scale problems is limited (Nannicini, 2019). On the contrary,

E-mail address: n.paterakis@tue.nl.

https://doi.org/10.1016/j.compchemeng.2023.108161
Received 27 June 2022; Received in revised form 22 November 2022; Accepted 29 January 2023
Available online 6 February 2023

special-purpose quantum computers, namely quantum annealers (QA), currently surpass gate-based computers both in terms of number of qubits and qubit connectivity and can be used to obtain approximate solutions of larger QUBO instances (McGeoch, 2020).

Several problems can be modeled using QUBO (e.g., Grant et al., 2021; Harwood et al., 2021; Stollenwerk et al., 2019). However, most optimization problems of practical interest are constrained and contain both discrete and continuous variables. For instance, mixed-integer linear programming (MILP) problem formulations are often encountered in diverse application areas, such as logistics (Anghinolfi et al., 2016), the coordination of unmanned aerial vehicles (Yang et al., 2019) and power systems (Chen et al., 2016). In these cases, developing a monolithic QUBO formulation is neither a convenient nor efficient approach since the continuous variables would need to be discretized. In order to leverage a quantum processing unit (QPU) for solving mixed-integer problems of practical interest, decomposition-based HQC algorithms must be developed. Such HQC algorithms aim to decompose the original optimization problem into a part that can be assigned to the QPU, i.e., a QUBO problem or a problem that can be cast as a QUBO, and a part that can be solved efficiently using classical algorithms (e.g., a convex optimization problem).

Although HQC optimization algorithms are not expected to result in exponential speedups in the NISQ era, studying them is important for two reasons (Bass et al., 2021). First, they are an avenue for the application of QC to various types of optimization problems. Second, they offer the potential for quantifiable algorithmic performance improvements and reinforcing the value of using QC alongside classical resources in solving real-world problems in critical sectors, such as power and energy systems. Recently, the development of such approaches has gained attention and various problem-specific (Ajagekar et al., 2020; Braine et al., 2021) and general-purpose techniques have been proposed (Gambella and Simonetto, 2020; Chang et al., 2022; Zhao et al., 2022).

In Gambella and Simonetto (2020) a decomposition approach for mixed binary-continuous optimization problems with complicating constraints based on a multi-block version of the alternating direction method of multipliers (ADMM) was presented. The proposed method splits the original problem into a QUBO problem and constrained convex subproblems (SP). The QUBO problem was solved using VQE and QAOA, while the constrained convex SPs were solved with a classical solver. Despite being a general method of wide applicability, the convergence of the HQC algorithm is not guaranteed.

In Chang et al. (2022) and Zhao et al. (2022) the Benders decomposition (BD) scheme was used as the basis for a general-purpose HQC MILP solver. BD splits the problem into a master problem (MP) that contains both binary and continuous variables and is assigned to the QPU and a linear programming (LP) SP that is solved classically. In each iteration, the solution of the SP provides an upper bound to the objective function and generates a new constraint (cut) that is added to the MP to improve the lower bound. A primer on BD is presented in Section 2.2. Although BD is proven to converge to the global optimum of the original MILP problem, a major drawback of its direct application as an HQC algorithm is that it requires the discretization of the continuous variable that appears in the constraints of the MP (proxy for the SP value) at the expense of introducing an increasing number of ancillary qubits to represent the cuts as the iterations of the algorithm progress. The straightforward implementation of BD is known to require a large number of iterations for convergence, which would render the aforementioned approaches impractical considering the limitations of NISQ hardware. In this paper, a multi-cut implementation of BD is adopted and, instead of the MP, the QPU is assigned to solve QUBO problems that correspond to an acceleration subroutine that manages the size of the MP and feature a more predictable size.

1.2. Applications of quantum computing in the power & energy sector

Despite the projected benefits, technical studies that investigate the use of QC for addressing power and energy system computational problems are currently scarce. Relevant studies can be classified into three categories.

The first category of studies focused on power system optimization. In Jones et al. (2020), the solution of the phasor measurement unit (PMU) placement problem using the D-Wave Systems 2000Q QPU was investigated. The PMU placement problem was formulated as the minimum dominating set problem. It was found that for some instances the QA outperformed the classical solver CPLEX. The optimization model presented in Jones et al. (2020) is easily converted into a QUBO, however, it does not capture the complexity of the actual problem. In Ajagekar and You (2019), simplified formulations of the facility location–allocation, unit commitment (UC) and heat exchanger network synthesis problems were solved using both the D-Wave Systems 2000Q QPU and IBM Q gate-based quantum computers. It was reported that for some problem instances, contrary to the QC approach, the classical solver Gurobi failed to return an optimal solution within a given time limit. However, the problem formulations presented in Ajagekar and You (2019) could either be directly recast as QUBO problems or discretization of continuous variables was performed, at the expense of using a large number of ancillary qubits in order to represent the discretized variables. More recently, to address the limitations of the direct application of QC to solving mixed-integer power system optimization problems, HQC algorithms were applied to solve the UC problem (Chang et al., 2022; Koretsky et al., 2021; Mahroo and Kargarian, 2022; Nikmehr et al., 2022). Nevertheless, the aforementioned studies either applied a straightforward implementation of BD that requires a prohibitively large number of ancillary qubits or used QAOA and multi-block ADMM heuristics that do not provide convergence guarantees.

The second category of studies developed quantum algorithms for power system analysis. The main component of relevant approaches is the Harrow–Hassidim–Lloyd (HHL) algorithm for the solution of systems of linear equations. In Feng et al. (2021) and Sævarsson et al. (2022), a quantum power flow algorithm was introduced and tested. Although computational experiments were performed on small-scale test systems, it was argued in both studies that a quantum computational advantage may be attainable in the future, provided that a number of caveats related to the application of the HHL algorithm are addressed (Sævarsson et al., 2022; Aaronson, 2015). The HHL algorithm was also exploited in Eskandarpour et al. (2020) as part of a workflow to assess power system security under contingencies of increasing severity. Lastly, in Zhou et al. (2021), a quantum algorithm for simulating electromagnetic transients was presented. These studies advocate that QC may find impactful applications in reliability assessment of power grids, that is currently intractable for large-scale systems.

Applications of HQC algorithms that combine machine learning and quantum sampling are the subject of the last category of studies. For instance, in Ajagekar and You (2021) a methodology to detect and classify power system faults using conditional restricted Boltzmann machines and quantum generative training was proposed.

1.3. Contributions

The appealing convergence properties of BD motivate pursuing this technique as a template for developing general-purpose MILP HQC solvers. However, the straightforward implementation of the method does not efficiently exploit NISQ resources. In this paper, an alternative approach to implementing BD while leveraging quantum resources is investigated by developing a multi-cut version of the algorithm. Instead of discretizing the continuous variable of the MP to assign it to a QPU directly, the QPU is designated to handle a pure binary optimization subroutine that selects the cuts that enter the MP to manage its size and accelerate the BD scheme by tightening the approximation of the

MP, while both the MP and the SPs are solved classically. In contrast with the implementations in Chang et al. (2022) and Zhao et al. (2022), the number of ancillary qubits required in the proposed approach does not depend on the number of iterations needed for convergence. Cut selection involves the solution of NP-hard problems, potentially in every iteration of the algorithm. Small instances are expected to be solvable classically. However, as the problem instance size and the number of times cut selection is applied increase, investigating the potential for a quantum benefit is pertinent.

The contribution of this paper is threefold:

- A multi-cut BD scheme that is applicable to general MILP problems without decomposable SP structure is adapted such that both classical and quantum resources can be exploited.
- A cut selection procedure that is based on two different cut selection criteria (exclusion of infeasible MP solutions, MP variable coverage) and two different cut selection strategies (minimum set cover, maximum coverage) is proposed. The QUBO reformulation of the cut selection strategies, such that they can be executed using QC, is provided and implementation details are discussed.
- The proposed HQC optimization algorithm is applied to a prototypical power system optimization problem, namely the UC problem, and the computational viability of its solution using a commercially available QA is extensively discussed.

It is to be noted that in this paper, experiments are conducted using a QA. However, the quantum step can also be seamlessly executed on a gate-based QPU using, for instance, QAOA for the solution of the QUBO problem instances. This is due to the convergence properties of the proposed algorithm that are discussed in Section 3.5.

The remainder of this paper is organized as follows: in Section 2 the necessary theoretical background is established, while in Section 3 the proposed HQC multi-cut BD algorithm and the proposed cut selection procedure are detailed. Then, in Section 4, the UC problem formulation that is used for demonstration purposes is described. The setup of the numerical experiments is described in Section 5 and results are presented and discussed in Section 6. Finally, conclusions are drawn in Section 7.

2. Preliminaries

2.1. Quantum annealing

In this section, the process of solving an optimization problem using a QA is briefly reviewed. Further details can be found in various sources, including McGeoch (2020) and Johnson et al. (2011). QAs are based on the adiabatic quantum computing paradigm (Childs et al., 2001). Quantum annealing evolves a quantum state by applying a time-dependent Hamiltonian described by (1) over a time interval $[0, T_A]$, where T_A is the annealing time:

$$H(\tau) = A(\tau)H_0 + B(\tau)H_P, \ \tau \in [0, T_A] \tag{1}$$

The annealing path functions $A(\tau)$ and $B(\tau)$ are monotonic functions of time that satisfy the conditions $A(0) = 1$, $A(T_A) = 0$ and $B(0) = 0$, $B(T_A) = 1$ respectively. In other words, at the beginning of the annealing, the Hamiltonian is equivalent to H_0 and gradually transitions to H_P.

The initial Hamiltonian H_0 for a system of V qubits is described by (2), where σ_a^x is the Pauli-x operator applied to qubit a. The initial Hamiltonian sets the qubits into an equal superposition with respect to the computational basis z.

$$H_0 = -\sum_{a\in V} \sigma_a^x \tag{2}$$

The problem Hamiltonian H_P which is dominant at time T_A is described by (3) and represents the unconstrained optimization problem of interest in terms of an Ising model, whose ground state is the optimal solution. The parameters h, J are real numbers that depend on the problem to be solved (linear and quadratic biases) and σ_a^z is the Pauli-z operator applied to qubit a.

$$H_P = \sum_{a\in V} h_a \sigma_a^z + \sum_{a\in V} \sum_{b\in V, a\neq b} J_{a,b}\, \sigma_a^z \otimes \sigma_b^z \tag{3}$$

According to the quantum adiabatic theorem, starting from the ground state of H, if the transition $A : 1 \to 0$ and $B : 0 \to 1$ is sufficiently slow, the system will remain in the ground state throughout the transition. This implies that $H(T_A)$ should also be in the ground state, i.e., it represents the optimal solution of the problem (Farhi et al., 2000).

Since QAs are open systems, the final result may not be the optimal solution of the problem. For this reason, quantum annealing is applied multiple times (annealing-read cycle) in order to increase the probability of finding high quality solutions. In general, the process of solving an optimization problem using a QPU consists of three steps. First, the optimization problem must be expressed as an Ising model or, equivalently, a QUBO problem. The logical problem formulation may impose interactions between qubits that are not directly compatible with the physical topology of the QPU. For this reason, the second step consists of finding a minor embedding of the logical problem graph such that it is compatible with the sparse native topology of the QPU, i.e., the physical connectivity between qubits (Cai et al., 2014; Bernal et al., 2020). This is achieved by creating chains of physical qubits such that they behave as a single logical qubit. It is possible that at the end of the QA process a number of chains are broken, i.e., physical qubits that belong in the same chain are not in the same state. Note that finding a minor-embedding is a classical computational task and involves the solution of an NP-hard problem (Cai et al., 2014; Lobe and Lutz, 2021). Finally, the minor-embedded problem instance is submitted to the QPU together with a set of hyperparameters, QA is performed and a set of solutions is returned.

2.2. Benders decomposition

In this study, MILP problems of the form (4) are of interest:

$$\min_{\mathbf{x},\mathbf{y}} \quad \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y} \tag{4a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} \geq \mathbf{b} \tag{4b}$$

$$\mathbf{x} \in \mathbb{R}_+^n, \mathbf{y} \in \mathbb{Y}^m \tag{4c}$$

where $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{d} \in \mathbb{R}^m$, $\mathbf{b} \in \mathbb{R}^q$ are vectors and $\mathbf{A} \in \mathbb{R}^{q\times n}$, $\mathbf{B} \in \mathbb{R}^{q\times m}$ are matrices of coefficients, respectively. $\mathbb{Y}^m$ is a set of constraints involving only the decision variables $\mathbf{y}$. Without loss of generality, decision variables $\mathbf{x}$ are considered to be non-negative. Decision variables $\mathbf{y}$ may be considered complicating if they are involved in most of the problem constraints or render the optimization problem non-convex. If complicating variables are fixed, then the optimization problem is substantially simplified (Conejo et al., 2006).

BD is a popular technique that is applied to such problems (Benders, 1962). The central idea in BD is to decompose the original problem into an MP which contains all the integer variables and a SP in which the complicating variables $\mathbf{y}$ are fixed to tentative values (LP problem). The MP and the SP are iteratively solved. Tentative solutions $\hat{\mathbf{y}}$ (tentatively fixed variables are denoted by $\hat{\cdot}$) are provided by solving the MP, whereas the solution of the SP yields the so-called Benders cuts, i.e., constraints that are progressively added to the MP to restrict its solution space.

For a tentative solution of the MP, the SP is given by (5):

$$\min_{\mathbf{x}} \quad \mathbf{c}^T\mathbf{x} \tag{5a}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b} - \mathbf{B}\hat{\mathbf{y}} \tag{5b}$$

$$\mathbf{x} \in \mathbb{R}_+^n \tag{5c}$$

In practice, the dual of the SP (DSP) given by (6) is solved:

$$\max_{\mathbf{v}} \quad V_{DSP} = (\mathbf{b} - \mathbf{B}\hat{\mathbf{y}})^T \mathbf{v} \tag{6a}$$

$$\text{subject to} \quad \mathbf{A}^T \mathbf{v} \leq \mathbf{c} \tag{6b}$$

$$\mathbf{v} \in \mathbb{R}_+^q \tag{6c}$$

where $\mathbf{v}$ are the dual variables associated with constraints (5b). Notice that the feasible region of the DSP remains the same for different tentative values $\hat{\mathbf{y}}$.

The MP is given by (7):

$$\min_{\mathbf{y},\zeta} \quad V_{MP} = \zeta + \mathbf{d}^T \mathbf{y} \tag{7a}$$

$$\text{subject to} \quad \mathbf{y} \in \mathbb{Y}^m \tag{7b}$$

$$\zeta \geq \zeta^{low} \tag{7c}$$

$$(\mathbf{b} - \mathbf{B}\mathbf{y})^T \mathbf{v}_i \leq \zeta, \ \forall i \in D \tag{7d}$$

$$(\mathbf{b} - \mathbf{B}\mathbf{y})^T \mathbf{u}_j \leq 0, \ \forall j \in U \tag{7e}$$

In (7) ζ is a free variable that acts as a surrogate for the value of the DSP. To guarantee that the MP is always bounded, an arbitrarily low bound ζ^{low} on ζ is enforced via (7c). The set of constraints (7d), where $\mathbf{v}_i$ corresponds to an extreme point of (6), are referred to as Benders optimality cuts. If the DSP is unbounded, an extreme ray $\mathbf{u}_j$ can be extracted instead of an extreme point, giving rise to Benders feasibility cuts expressed by (7e). The sets of extreme points and extreme rays of the DSP are denoted by D and U, respectively.

In each iteration the value of the MP provides a lower bound of (4a), i.e., $LB = \hat{\zeta} + \mathbf{d}^T \hat{\mathbf{y}}$. The lower bound is monotonically increasing. A feasible solution of the DSP provides a valid upper bound $UB = (\mathbf{b} - \mathbf{B}\hat{\mathbf{y}})^T \hat{\mathbf{v}} + \mathbf{d}^T \hat{\mathbf{y}}$. The algorithm converges if $(UB - LB) < \epsilon$, where $\epsilon \rightarrow 0$ is a predetermined tolerance.

The maximum number of iterations required for BD to converge to the optimal solution of Problem (4) equals the total number of extreme points and extreme rays of the DSP, which corresponds to an exponential enumeration of all the solutions of the MP. Although finite, this number can be enormous. However, at optimality, the number of active MP constraints is limited and cannot exceed the number of the MP decision variables (Saharidis and Ierapetritou, 2013). BD exploits this observation and aims to identify the active MP constraints by generating and adding a single cut in each iteration and reach convergence without generating all the possible Benders cuts. Despite BD being one of the most widely applied decomposition techniques, the straightforward implementation that was presented in this section is often inefficient from a computational point of view. Several reasons, including poor feasibility and optimality cuts, initial iterations that slowly improve the bounds and slow convergence towards the end of the algorithm, have been recognized. Details about these problems, as well as a systematic review of the research on accelerating BD, can be found in Rahmaniani et al. (2017).

3. Methodology

3.1. Hybrid quantum-classical multi-cut Benders decomposition

In the standard BD described in Section 2.2 only a single cut is generated and added to the MP in each iteration. However, in order to accelerate BD, it is possible to generate and add multiple cuts to tighten the MP problem and improve the obtained lower bounds (Su et al., 2015; Tang et al., 2013; Saharidis et al., 2010; You and Grossmann, 2013; Asl and MirHassani, 2019). In this paper, the BD acceleration strategy that was introduced in Asl and MirHassani (2019) and is based on generating multiple cuts via multiple solutions (MCMS) of the MP is explored further. The advantage of this technique is that neither the SP need have a decomposable structure nor auxiliary SPs need to be devised to generate multiple cuts. In particular, multiple feasibility and optimality cuts are generated by exploiting multiple feasible (but not necessarily optimal) solutions of the MP that are available in each iteration. The generated cuts may be appended to the MP to restrict its solution space more drastically compared to adding a single cut and, therefore, reduce the number of iterations required for convergence. However, adding a large number of cuts increases the size of the MP rapidly and, as a result, may increase the total execution time of the algorithm. To achieve a trade-off between the increase in the size of the MP and the reduction in the number of major iterations of the decomposition algorithm, it is possible to add a subset of the cuts that are generated in each iteration. A cut selection subroutine based on pure binary optimization problems that may be solved using QC is developed, hence the modified algorithm is an HQC optimization algorithm (HQC-MCMS). A schematic illustration of HQC-MCMS is shown in Fig. 1.

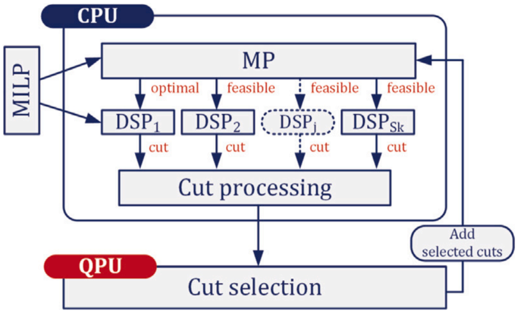


Fig. 1. Schematic representation of HQC-MCMS.

The HQC-MCMS procedure is described in Algorithm 1. Similar to the original BD algorithm, the procedure begins by setting the upper and lower bounds of the original objective function to $+\infty$ and $-\infty$, respectively (lines 1 and 2). In each iteration k, and while the difference between the upper and lower bounds remains higher than a predetermined tolerance ϵ, a number of steps are executed. First, the MP is solved and S_k feasible solutions are obtained. Modern solvers permit the extraction of high quality feasible solutions that are found while attempting to solve a MILP problem to optimality. Note that it may be the case that $S_k < S$, where S is the number of requested solutions. If the MP is proven to be infeasible, the optimization problem is also infeasible and the algorithm terminates. Otherwise, the lower bound is updated with the value V_{MP}^1 of the optimal solution (lines 5–9).

For each of the S_k available solutions the corresponding DSP is solved. Note that the DSP instances are independent and may be solved in parallel. If a DSP instance is infeasible, the algorithm terminates. If the DSP instance is unbounded, then a feasibility cut is generated and appended to the set of feasibility cuts G_k^F, whereas, if the DSP instance is optimal, then an optimality cut is generated and appended to the set of optimality cuts G_k^O (lines 10–19).

The crucial step of the HQC-MCMS algorithm is the cut selection procedure which is described by Algorithm 2 (Section 3.4) and comprises two elements, namely the cut selection criterion (Section 3.2) and the cut selection strategy (Section 3.3). This is the step where QC can be exploited. After the execution of the cut selection subroutine, the sets of the selected feasibility and optimality cuts, $G_k^{\prime F}$ and $G_k^{\prime O}$ respectively, are added to the MP to be solved in the next iteration (lines 20–27).

Finally, the upper bound is updated using the value of the DSP corresponding to the optimal solution of the MP, V_{DSP}^1, and the iteration counter is increased. The convergence of the HQC-MCMS algorithm is discussed in Section 3.5.

Algorithm 1: HQC-MCMS algorithm

```
Data: $\epsilon$ (tolerance), $S$ (maximum number of MP solutions to be
      extracted)
$UB \leftarrow +\infty$ ;
$LB \leftarrow -\infty$ ;
$k \leftarrow 1$ (iteration counter) ;
while $(UB - LB) > \epsilon$ do
    Solve MP$_k$ (7) and obtain $S_k \leq S$ solutions;
    if infeasible then
        Stop; Declare infeasibility;
    else
        $LB \leftarrow V^1_{MP}$ ;
        for $j \leftarrow 1$ to $S_k$ do                    ▷ In parallel
            Solve DSP$_{j,k+1}$ (6) for $S_j$ ;
            if infeasible then
                Stop; Declare infeasibility;
            else if unbounded then
                Add cut to $G^F_k$ ;
            else
                Add cut to $G^O_k$ ;
            end
        end
        if $G^F_k \neq \emptyset$ then
            Execute Algorithm 2 on a QPU;
            Add selected feasibility cuts $G'^F_k$ to MP$_{k+1}$;
        end
        if $G^O_k \neq \emptyset$ then
            Execute Algorithm 2 on a QPU;
            Add selected optimality cuts $G'^O_k$ to MP$_{k+1}$;
        end
        $UB \leftarrow \min\{UB, V^1_{DSP}\}$;
    end
    $k \leftarrow k + 1$;
end
```

3.2. Cut selection criteria

3.2.1. Criterion I: Cut selection based on the exclusion of infeasible solutions

In Asl and MirHassani (2019), cut selection was based on the observation that a subset of the generated feasibility cuts may exclude all the infeasible solutions of the MP that are identified in a given iteration. The feasibility cut that corresponds to the infeasible solution $\hat{\mathbf{y}}^i$ of the MP also excludes the infeasible solution $\hat{\mathbf{y}}^j$ if $(\mathbf{b} - \mathbf{B}\hat{\mathbf{y}}^j)^T \hat{\mathbf{u}}_i > 0$.

Let $|G^F_k|$ denote the cardinality of the set of all feasibility cuts that are generated in iteration k. Then, the $|G^F_k| \times |G^F_k|$ binary indicator matrix $\mathbf{E}$ can be constructed in order to compile information about the infeasible MP solutions that are excluded by each cut in the current iteration. Specifically, $\mathbf{E}_{ij} = 1$ if the feasibility cut i excludes the infeasible solution associated with the jth solution of the MP, otherwise $\mathbf{E}_{ij} = 0$. Note that the diagonal elements of $\mathbf{E}$ are always equal to one because, by definition, a feasibility cut excludes the infeasible MP solution based on which it was generated.

Cut selection based on this criterion presents two potential drawbacks. First, this criterion applies only to feasibility cuts. Second, it is possible that for a given problem instance one or more cuts can exclude all the infeasible MP solutions, rendering the application of this criterion trivial.

3.2.2. Criterion II: Cut selection based on MP variable coverage

It is often the case that the generated optimality and feasibility cuts are low-density. This means that many of the coefficients that correspond to MP decision variables in a given cut are either zero or near-zero relative to other coefficients. The contribution of low-density cuts to strengthening the MP tends to be limited (Saharidis et al., 2010). For this reason, several BD acceleration strategies are based on the idea of either generating high-density Pareto optimal cuts (Tang et al., 2013) or bundles of low-density cuts (Saharidis et al., 2010) such that more MP decision variables are covered. In this paper, instead of focusing on the generation of cuts such that the decision variables of the MP are covered, the cuts that are generated based on multiple solutions of the MP are inspected with the purpose of identifying a subset of feasibility (and/or optimality cuts) such that all or most of the MP decision variables are collectively covered. Note that a rather strict definition of MP variable coverage is adopted. A decision variable y_i of the MP is said to be covered in a given feasibility cut of the form $\sum_i y_i(\mathbf{B}^T\mathbf{u})_i \geq \mathbf{b}^T\mathbf{u}$, if for the ith row of the matrix $\mathbf{B}^T\mathbf{u}$ it holds $|(\mathbf{B}^T\mathbf{u})_i| > 0$. A similar definition applies to optimality cuts if $\mathbf{u}$ is replaced by $\mathbf{v}$.

A $|G^F_k| \times m$ binary indicator matrix $\mathbf{D}^F$ can be constructed after having identified which MP decision variables are covered by a given feasibility cut in the current iteration. Specifically, $\mathbf{D}^F_{ij} = 1$ if the jth variable of the MP is covered in cut i, otherwise $\mathbf{D}^F_{ij} = 0$. A similar matrix $\mathbf{D}^O$ ($|G^O_k| \times m$) may be constructed for optimality cuts, if cut selection is to be applied also to the set of optimality cuts. Note that some columns of $\mathbf{D}^F$ and $\mathbf{D}^O$ may be zero, that is, some decision variables may not be covered in any of the generated feasibility or optimality cuts.

Compared to Criterion I, cut selection based on the coverage of MP variables applies both to feasibility and optimality cuts and, therefore, may result in more effective MP size management. Moreover, from an implementation perspective, the construction of matrices $\mathbf{D}^F$ and $\mathbf{D}^O$ requires the evaluation of shorter expressions in comparison with $\mathbf{E}$.

3.3. Cut selection strategies

After having processed the available cuts according to one of the criteria that were presented in Section 3.2, a cut selection strategy must be applied in order to identify G'^F_k and/or G'^O_k. Two such strategies based on the minimum set cover problem and the maximum coverage problem, as well as their solution using a QPU are discussed next.

3.3.1. Strategy I: Minimum set cover for cut selection

The minimum set cover problem can be solved to identify a set of cuts with minimum cardinality that satisfy a given condition. If cut selection is based on Criterion I, then the minimum number of feasibility cuts that exclude all the infeasible solutions in the current iteration will be identified. Similarly, if cut selection is based on Criterion II, then the minimum number of feasibility (optimality) cuts that cover all the MP decision variables that can be covered in the current iteration will be identified.

Given a binary indicator matrix $\mathbf{M} \in \{0,1\}^{|I| \times |J|}$, where I is the set of rows and J is the set of columns, the minimum set cover problem is a pure binary optimization problem expressed by (8):

$$\min_{\chi} \quad \sum_{i \in I} \chi_i \tag{8a}$$

$$\text{subject to} \quad \sum_{i \in I} M_{ij}\chi_i \geq 1, \ \forall j \in J \tag{8b}$$

$$\chi \in \{0,1\}^I \tag{8c}$$

where χ_i is a binary variable that is equal to 1 if the cut i is selected and 0 otherwise. It is reiterated that the cut selection problem has to be solved in each iteration k. However, for notational simplicity, the iteration index k is dropped. Depending on the cut selection criterion, the columns of $\mathbf{M}$ represent either infeasible solutions or decision variables of the MP (matrix $\mathbf{M}$ is accordingly replaced by matrices $\mathbf{E}$, $\mathbf{D}^F$, $\mathbf{D}^O$). Note that when Criterion II is used, there is the possibility that a number of MP decision variables cannot be covered and, therefore, (8b) should be amended by subtracting a binary slack variable from the right-hand side in order to indicate that the constraint cannot be satisfied for a particular column j. The sum of the slack variables should also be penalized in (8a). However, this can be avoided by inspecting $\mathbf{M}$ and dropping all the columns of zeros.

The minimum set cover problem is known to be NP-hard (Grossman and Wool, 1997), i.e., it is intractable for large I and J. For this reason, various heuristics have been proposed in order to obtain approximate solutions. To find the minimum set cover using a QPU, (8) must be recast as a QUBO problem. First, (8b) is converted to an equality constraint by adding an integer slack variable on the right hand side of the constraint and using binary expansion (Tamura et al., 2021) with $\gamma_j = \lceil \log_2 \left(\sum_{i\in I} M_{ij}\right)\rceil,\ \forall j \in J$ ancillary qubits $s_{\alpha j}$ as in (9):

$$\sum_{i\in I} M_{ij}\chi_i = 1 + \sum_{\alpha=0}^{\alpha=\gamma_j-1} 2^{\alpha} s_{\alpha j},\ \forall j \in J \tag{9}$$

The QUBO problem formulation is given by (10), where H is the Hamiltonian of the problem, and $\mathcal{P}_{\mathcal{A}}$ and $\mathcal{P}_{B_j}, \forall j \in J$ are positive penalties that are heuristically determined such that $\mathcal{P}_{B_j} \gg \mathcal{P}_{\mathcal{A}}, \forall j \in J$ to avoid constraint violations:

$$\min_{\chi,s}\ H = H_A + H_B \tag{10a}$$

where

$$H_A = \mathcal{P}_{\mathcal{A}} \sum_{i\in I} \chi_i \tag{10b}$$

$$H_B = \sum_{j\in J} \mathcal{P}_{B_j} \left[\sum_{i\in I} \left(M_{ij}\chi_i\right) - 1 - \sum_{\alpha=0}^{\alpha=\gamma_j-1} 2^{\alpha} s_{\alpha j} \right]^2 \tag{10c}$$

The maximum number of qubits that are required in order to represent the problem is $|I| + |J|\lceil\log_2|I|\rceil$. If Criterion I is used, then this number translates to $|G_k^F|(1+\lceil\log_2|G_k^F|\rceil)$. This is the case if each infeasible MP solution is excluded by every feasibility cut. If Criterion II is used, then the maximum number of qubits that are needed is $|G_k^F| + m\lceil\log_2|G_k^F|\rceil$, if all the variables of the MP are covered by every single cut. A similar expression can be written for Criterion II applied to the set of optimality cuts.

It may be observed that for Criterion I the number of qubits that are necessary in order to represent Problem (10) is independent of the size of Problem (4). Instead it depends on the user-provided parameter S and is expected to be larger during the first iterations of the algorithm where mostly feasibility cuts are generated. In addition to that, inspection of the columns of matrix $\mathbf{M}$ can reveal rows (feasibility cuts) that exclude all the infeasible solutions in the current iteration and avoid triggering cut selection. The number of qubits in case Criterion II is applied depends on the number of MP variables and may be particularly large. However, for many practical applications the generated cuts are typically expected to be low-density, which implies that a relatively large number of columns of matrix $\mathbf{M}$ are expected to be dropped because all their entries are zero. In both cases, multiple cuts may exclude the same infeasible solutions or cover the same variables. Therefore, the rows of matrix $\mathbf{M}$ can also be inspected in order to remove duplicates, keeping the row that corresponds to a higher quality MP solution.

3.3.2. Strategy II: Maximum coverage for cut selection

The conservativeness of Strategy I may lead to a large number of cuts being added to the MP. To address this issue, the maximum coverage problem can be solved as an alternative cut selection strategy in order to select at most a predefined number of cuts such that, depending on the cut selection criterion that is applied, either the maximum number of infeasible solutions are excluded or the maximum number of MP decision variables are covered. To the best of the Author's knowledge, maximum coverage has not been used as a cut selection strategy in the context of BD before.

Given a matrix $\mathbf{M} \in \{0,1\}^{|I|\times|J|}$ ($\mathbf{M}$ is accordingly replaced by matrices $\mathbf{E}$, $\mathbf{D}^F$, $\mathbf{D}^O$) and a maximum number of cuts to be selected $\mathcal{M}(\le |I|)$, the maximum coverage problem (Takabe et al., 2018) is a pure binary optimization problem formulated in (11):

$$\max_{\chi,\phi}\ \sum_{j\in J} \phi_j \tag{11a}$$

$$\text{subject to}\ \sum_{i\in I} \chi_i \le \mathcal{M} \tag{11b}$$

$$\sum_{i\in I} M_{ij}\chi_i \ge \phi_j,\ \forall j \in J \tag{11c}$$

$$\chi \in \{0,1\}^I, \boldsymbol{\phi} \in \{0,1\}^J \tag{11d}$$

where χ_i is a binary variable that is equal to 1 if cut i is selected and ϕ_j is a binary variable that is equal to 1 if column j is covered. This means that, depending on the criterion that is used, either an infeasible solution of the MP is excluded, or an MP variable is covered by the selected cuts.

The QUBO problem formulation of the maximum coverage problem is given by (12), where H is the Hamiltonian of the problem and $\mathcal{P}_{\mathcal{A}}$, $\mathcal{P}_{C_j}, \forall j \in J$ and $\mathcal{P}_B$ are positive penalties that are heuristically determined such that $\mathcal{P}_B \gg \mathcal{P}_{\mathcal{A}}$ and $\mathcal{P}_{C_j} \gg \mathcal{P}_{\mathcal{A}}, \forall j \in J$. First, (11b) is converted into an equality constraint using integer slack variables on the left-hand side of the constraint and binary expansion introducing $\gamma = \lceil\log_2(\mathcal{M}+1)\rceil$ ancillary qubits s_α. Note that (11a) does not involve χ. Thus, cut combinations with $|G'^F_k|, |G'^O_k| \le \mathcal{M}$ that maximize coverage are indistinguishable from an optimization perspective and (11b) can be replaced by an equality constraint, avoiding the use of ancillary qubits to represent this constraint (Takabe et al., 2018). The only drawback of this simplification is that it prevents the discovery of a potentially smaller subset of cuts that also maximize coverage; however, if $\mathcal{M} \ll |I|$, (11b) can be expected to be binding. Similarly, (11c) can be converted into an equality constraint using $\gamma'_j = \lceil\log_2(\min(\mathcal{M}, (\sum_{i\in I} M_{ij})) + 1)\rceil,\ \forall j \in J$ ancillary qubits $s'_{\alpha j}$.

$$\min_{\chi,\phi,s,s'}\ H = H_A + H_B + H_C \tag{12a}$$

where

$$H_A = -\mathcal{P}_{\mathcal{A}} \sum_{j\in J} \phi_j \tag{12b}$$

$$H_B = \mathcal{P}_B \left(\sum_{i} \chi_i - \mathcal{M} + \sum_{\alpha=0}^{\alpha=\gamma-1} 2^{\alpha} s_{\alpha} \right)^2 \tag{12c}$$

$$H_C = \sum_{j\in J} \mathcal{P}_{C_j} \left[\sum_{i\in I} \left(M_{ij}\chi_i\right) - \phi_j - \sum_{\alpha=0}^{\alpha=\gamma'_j-1} 2^{\alpha} s'_{\alpha j} \right]^2 \tag{12d}$$

The maximum number of qubits that are required in order to represent the problem, assuming that (11b) is replaced by an equality constraint, is $|I|+|J|(1+\lceil\log_2(\mathcal{M}+1)\rceil)$. If Criterion I is used, this number translates to $|G_k^F| + |G_k^F|(1+\lceil\log_2(\mathcal{M}+1)\rceil)$. This is the case if each infeasible MP solution can be excluded by at least $\mathcal{M}$ cuts. If Criterion II is used, then the maximum number of qubits that are required is $|G_k^F| + m(1+\lceil\log_2(\mathcal{M}+1)\rceil)$ in case all the MP variables can be covered by at least $\mathcal{M}$ cuts. A similar expression can be written for Criterion II applied to the set of optimality cuts. Although these numbers may appear to be prohibitively large, in practice, the observations of Section 3.3.1 hold also when Strategy II is employed.

3.4. Cut selection procedure

The cut selection problem that is invoked in lines 21 and 25 of Algorithm 1 is described by Algorithm 2. If cut selection is based on Criterion I, the rows of matrix $\mathbf{E}$ are inspected in order to identify cuts that exclude the same infeasible solutions (line 3). If such duplicate rows are found, the row corresponding to the cut associated with an MP solution with smaller objective function value is kept, while the rest of the rows are dropped. Then, depending on the cut selection strategy, the corresponding optimization problem is solved (lines 5 and 8) and G'^F_k is returned. If Criterion II is used, matrices $\mathbf{D}^F$ and $\mathbf{D}^O$ are inspected depending on whether cut selection is applied both to feasibility and optimality cuts (lines 14 and 23). First, their rows are inspected similarly to the rows of **E**. Then, since it may not be possible to cover a number of MP variables by any of the cuts (all column entries are zero), the respective columns are dropped. Depending on the cut selection strategy, the corresponding optimization problem is solved (lines 25 and 28) using the modified matrix.

The inspection step may significantly reduce the size of the matrices before the cut selection optimization problems are solved. Additionally, all three matrices are inspected in order to identify whether a single cut can either exclude all the infeasible solutions (Criterion I) or cover all the MP variables that can be covered (Criterion II). If such a cut is found, the cut selection procedure terminates without triggering the solution of an optimization problem and the set of selected feasibility and/or optimality cuts that is returned contains only a single cut.

Algorithm 2: Cut selection procedure using a QPU

```
   Data: $G_k^F$, $G_k^O$, cutSelectionCriterion, cutSelectionStrategy,
         optSelect (boolean; whether to apply cut selection on
         optimality cuts), hyperparameters
1  if cutSelectionCriterion = Criterion I then
2      Construct E;
3      Inspect E;
4      if cutSelectionStrategy = Strategy I then
5          Solve (10) to select feasibility cuts;
6      end
7      if cutSelectionStrategy = Strategy II then
8          Solve (12) to select feasibility cuts;
9      end
10     Return $G_k^{\prime F}$;
11 end
12 if cutSelectionCriterion = Criterion II then
13     Construct $\mathbf{D}^F$;
14     Inspect $\mathbf{D}^F$;
15     if cutSelectionStrategy = Strategy I then
16         Solve (10) to select feasibility cuts;
17     end
18     if cutSelectionStrategy = Strategy II then
19         Solve (12) to select feasibility cuts;
20     end
21     if optSelect then
22         Construct $\mathbf{D}^O$;
23         Inspect $\mathbf{D}^O$;
24         if cutSelectionStrategy = Strategy I then
25             Solve (10) to select optimality cuts;
26         end
27         if cutSelectionStrategy = Strategy II then
28             Solve (12) to select optimality cuts;
29         end
30     end
31     Return $G_k^{\prime F}$ and $G_k^{\prime O}$;
32 end
```

3.5. Convergence of HQC-MCMS

Even though the solution of the cut selection problem using a QPU is not necessarily optimal (or even feasible) in every iteration, the HQC-MCMS algorithm is guaranteed to always converge to the global optimum of Problem (4) if at least one valid cut is added to the MP in each iteration.

Sets G_k^F and G_k^O contain only valid cuts that are generated based on multiple feasible solutions of the MP (Asl and MirHassani, 2019). Thus, any non-empty subsets $G_k^{\prime F}$ and $G_k^{\prime O}$ returned by Algorithm 2 contain only valid cuts, even if the solution to the QUBO problems (10) and (12) returned by the QPU is sub-optimal or infeasible. Convergence can also be guaranteed in case the cut selection problem is infeasible such that no cuts are selected in a given iteration. For instance, to prevent $G_k^{\prime F}$ and $G_k^{\prime O}$ from being empty, the feasibility or optimality cut that is generated at the current iteration using the optimal value of the MP in the DSP can be added in the next iteration.

Naturally, the number of iterations required for convergence of the HQC-MCMS algorithm and the solution time are affected by the quality of the approximate QUBO solutions returned by the QPU. Relying on quantum resources for the cut selection step, a trajectory of larger MP instances or weaker approximations of the MP may be generated in comparison to that of solving the cut selection problem exactly, therefore possibly increasing the number of iterations and solution time. Nonetheless, as the scale of the cut selection problems increases, using a QPU to obtain approximate QUBO solutions of reasonable quality may prove a valuable option to limit the time spent on Algorithm 2, while effectively managing the size of the MP.

4. Use case: The unit commitment problem

The UC problem is a fundamental power system optimization problem that aims to schedule and dispatch the available generation and demand side resources such that a financial or operational objective is optimized (Zheng et al., 2015). Various classical solution techniques, including BD schemes, have been applied to variants of this problem (Wen et al., 2016; Nasri et al., 2016; Alemany et al., 2013; Fu et al., 2013, 2005; Wu and Shahidehpour, 2010). An interesting observation is that the Benders cuts are often low-density for the UC problem, which makes it a good candidate problem to demonstrate the effectiveness of multi-cut BD approaches (Wen et al., 2016; Fu et al., 2013; Wu and Shahidehpour, 2010). As it was discussed in Section 1.2, the UC problem has been studied also in the context of early QC applications in the power and energy sector. However, NISQ hardware limitations have constrained the application of quantum and HQC approaches to relatively simple problem formulations compared to those that have been solved by classical techniques. In Ajagekar and You (2019) a single-period formulation considering only power balance and generator output constraints was solved via discretization. The same formulation was adopted in Koretsky et al. (2021) and Mahroo and Kargarian (2022) that studied the application of QAOA and multi-block ADMM HQC heuristics respectively. An abstract formulation of the UC problem was presented in Nikmehr et al. (2022). QAOA was applied to simple instances of the UC formulation, while more complex variants were tackled using a multi-block ADMM HQC heuristic. Similarly, in Chang et al. (2022), a multi-period, network-constrained UC formulation was adopted and solved using a straightforward BD-based HQC algorithm with discretization of the MP variables, disregarding the commitment cost. However, the QA could not be applied even on a small 6-bus system due to the prohibitive number of physical qubits required to implement the algorithm.

To demonstrate the applicability of the solution approach that was presented in Section 3, a multi-period, network-constrained UC problem formulation is adopted and decomposed such that it is amenable to solution by the proposed HQC algorithm. First, in Section 4.1 the complete MILP problem formulation is presented. Then, the DSP and MP formulations are derived in Sections 4.2 and 4.3.

4.1. MILP problem formulation

The UC problem formulation is described by (13a)–(13o). The notation that is used is presented in Table 1. The dual variables associated with each constraint set are denoted by greek letters that are displayed in parentheses next to the corresponding equation. Note that all the dual variables associated with the inequality constraints are non-negative, whereas the dual variables associated with equality constraints are unrestricted.

$$\min_{\boldsymbol{P},\boldsymbol{\theta},\boldsymbol{u},\boldsymbol{y},\boldsymbol{z}} \quad TC = EC + CC$$

$$\text{where } EC = \sum_t \sum_g C_g P_{gt}$$

Table 1
Notation used in the UC formulation.

Sets and indices	
g (G)	Index (set) of generators
t (T)	Index (set) of time
i,j (I)	Indices (set) of buses
I^0	Set of the reference bus (singleton)
$l(L)$	Index (set) of loads
Parameters	
A_{ig}^{G}	Generator incidence matrix; 1 if generator g is connected to bus i
A_{il}^{L}	Load incidence matrix; 1 if load l is connected to bus i
C_g	Energy cost of generator g ($/MWh)
D_{lt}	Demand of load l in time t (MW)
NLC_g	No-load cost of generator g ($/h)
SUC_g	Start-up cost of generator g ($)
SDC_g	Shut-down cost of generator g ($)
P_g^{max}	Maximum power output of generator g (MW)
P_g^{min}	Minimum power output of generator g (MW)
P_g^{ini}	Initial power output of generator g (MW)
RU_g	Ramp-up rate of generator g (MW/h)
RD_g	Ramp-down rate of generator g (MW/h)
X_{ij}	Reactance of line (i,j) (pu)
F_{ij}^{max}	Maximum power flow through line (i,j) (MW)
B_{ij}	(i,j) admittance matrix element (pu)
u_g^{ini}	Initial state of generator g; 1 if generator g was online before the beginning of the scheduling horizon
ζ^{low}	Lower bound of the proxy variable ζ ($)
Decision variables	
P_{gt}	Power output of generator g in time t (MW)
θ_{it}	Voltage angle of bus i in time t (rad)
ζ	Proxy for the DSP value ($)
u_{gt}	Binary variable; 1 if generator g is online in time t
y_{gt}	Binary variable; 1 if generator g starts-up in time t
z_{gt}	Binary variable; 1 if generator g is shut-down in time t

$$CC = \sum_{t}\sum_{g}\left(NLC_g u_{gt} + SUC_g y_{gt} + SDC_g z_{gt}\right) \tag{13a}$$

subject to:

$$y_{gt} - z_{gt} = u_{gt} - u_{g(t-1)} \quad \forall g, t > 1 \tag{13b}$$

$$y_{gt} - z_{gt} = u_{gt} - u_g^{ini} \quad \forall g, t = 1 \tag{13c}$$

$$-P_{gt} \geq -P_g^{max} u_{gt} \quad \forall g, t \quad (\mu_{gt}^{+}) \tag{13d}$$

$$P_{gt} \geq P_g^{min} u_{gt} \quad \forall g, t \quad (\mu_{gt}^{-}) \tag{13e}$$

$$-P_{gt} + P_{g(t-1)} \geq -RU_g \quad \forall g, t > 1 \quad (\nu_{gt}^{+}) \tag{13f}$$

$$-P_{gt} \geq -RU_g - P_g^{ini} \quad \forall g, t = 1 \quad (\nu_g^{0+}) \tag{13g}$$

$$P_{gt} - P_{g(t-1)} \geq -RD_g \quad \forall g, t > 1 \quad (\nu_{gt}^{-}) \tag{13h}$$

$$P_{gt} \geq -RD_g + P_g^{ini} \quad \forall g, t = 1 \quad (\nu_g^{0-}) \tag{13i}$$

$$\frac{1}{X_{ij}}(\theta_{jt} - \theta_{it}) \geq -F_{ij}^{max} \quad \forall (i,j)|X_{ij} \neq 0, t \quad (\psi_{ijt}) \tag{13j}$$

$$\theta_{i,t} = 0 \quad \forall t, i \in I^0 \quad (\lambda_t^0) \tag{13k}$$

$$\sum_{j \in I} B_{ij}\theta_{jt} - \sum_{g \in G} P_{gt} A_{ig}^{G} = -\sum_{l \in L} D_{lt} A_{il}^{L} \quad \forall i, t \quad (\lambda_{it}) \tag{13l}$$

$$P_{gt} \geq 0 \quad \forall g, t \tag{13m}$$

$$\theta_{it} \in \mathbb{R} \quad \forall i, t \tag{13n}$$

$$u_{gt}, y_{gt}, z_{gt} \in \{0, 1\} \quad \forall g, t \tag{13o}$$

The objective function is expressed by (13a) and stands for the minimization of the total energy (EC) and commitment cost (CC) across the time horizon. Constraints (13b) and (13c) determine the generator commitment logic. The power output of generators is limited by (13d) and (13e). The intertemporal constraints (13f)–(13i) limit the change in the power output of generators in consecutive time periods according to their up and down ramp rates. Network constraints are modeled in terms of a DC power flow approximation. The power that flows through transmission lines is constrained by (13j), while (13k) fixes the voltage angle at the reference bus. The power balance at each bus is determined by (13l). Finally, (13m)–(13o) determine the domain of the decision variables.

In (13), $\boldsymbol{u}$ are considered to be the complicating variables. The MP includes constraints (13b), (13c) and (13o), as well as the necessary optimality and feasibility cut expressions. The objective function of the MP comprises only the component CC of (13a) and a proxy variable ζ. The MP provides a tentative generator commitment status $\hat{\boldsymbol{u}}$. The SP includes constraints (13d)–(13n), while its objective function includes the component EC of the original objective function (13a). Essentially, the SP represents the solution of a network-constrained economic dispatch.

4.2. Dual subproblem formulation

For a tentative solution $\hat{\boldsymbol{u}}$ of the MP, the SP is an LP problem. Therefore, its dual is also an LP problem (Conejo et al., 2006) that is expressed by (14a)–(14k).

$$\max_{\boldsymbol{\mu}^{+},\boldsymbol{\mu}^{-},\boldsymbol{\nu}^{+},\boldsymbol{\nu}^{-},\boldsymbol{\nu}^{0-},\boldsymbol{\nu}^{0+},\boldsymbol{\psi},\boldsymbol{\lambda},\boldsymbol{\lambda}^{0}} \quad UC^{DSP}$$

$$\text{where } UC^{DSP} = \sum_{t}\sum_{g}\left(\mu_{gt}^{-}P_g^{min} - \mu_{gt}^{+}P_g^{max}\right)\hat{u}_{gt}$$
$$-\sum_{t}\sum_{g}\left(\nu_{gt}^{+}RU_g + \nu_{gt}^{-}RD_g\right)$$
$$+\sum_{g}\left[\nu_g^{0+}\left(-P_g^{ini} - RU_g\right) + \nu_g^{0-}\left(P_g^{ini} - RD_g\right)\right]$$
$$-\sum_{t}\sum_{i}\left[\lambda_{it}\sum_{l}\left(D_{lt}A_{il}^{L}\right) + \sum_{j|X_{ij}\neq 0}\psi_{ijt}F_{ij}^{max}\right] \tag{14a}$$

subject to:

$$-\mu_{gt}^{+} + \mu_{gt}^{-} - \nu_g^{0+} + \nu_g^{0-} + \nu_{g(t+1)}^{+} - \nu_{g(t+1)}^{-}$$
$$-\sum_{i} A_{ig}^{G}\lambda_{it} \leq C_g \quad \forall g, t = 1 \tag{14b}$$

$$-\mu_{gt}^{+} + \mu_{gt}^{-} - \nu_{gt}^{+} + \nu_{gt}^{-}$$
$$-\sum_{i} A_{ig}^{G}\lambda_{it} \leq C_g \quad \forall g, t = |T| \tag{14c}$$

$$-\mu_{gt}^{+} + \mu_{gt}^{-} - \nu_{gt}^{+} + \nu_{g(t+1)}^{+} + \nu_{gt}^{-} - \nu_{g(t+1)}^{-}$$
$$-\sum_{i} A_{ig}^{G}\lambda_{it} \leq C_g \quad \forall g, t \in (1, |T|) \tag{14d}$$

$$\lambda_t^0 + \sum_{j}\left(\lambda_{jt}B_{ji}\right)$$
$$+\sum_{j|X_{ij}\neq 0}\left(\frac{\psi_{jit} - \psi_{ijt}}{X_{ij}}\right) = 0 \quad \forall i \in I^0, t \tag{14e}$$

$$\sum_{j}\left(\lambda_{jt}B_{ji}\right)$$
$$+\sum_{j|X_{ij}\neq 0}\left(\frac{\psi_{jit} - \psi_{ijt}}{X_{ij}}\right) = 0 \quad \forall i \in I - I^0, t \tag{14f}$$

$$\mu_{gt}^{+}, \mu_{gt}^{-}, \nu_{gt}^{+}, \nu_{gt}^{-} \geq 0 \quad \forall g, t \tag{14g}$$

$$\nu_g^{0-}, \nu_g^{0+} \geq 0 \quad \forall g \tag{14h}$$

$$\psi_{ijt} \geq 0 \quad \forall (i,j)|X_{ij} \neq 0, t \tag{14i}$$

$$\lambda_t^0 \in \mathbb{R} \quad \forall t \tag{14j}$$

$$\lambda_{it} \in \mathbb{R} \quad \forall i, t \tag{14k}$$

4.3. Master problem formulation

Given $G_k'^F$ and $G_k'^O$, the MP is a MILP problem that is expressed by (15a)–(15e).

$$\min_{\zeta,\boldsymbol{u},\boldsymbol{y},\boldsymbol{z}} \quad \zeta + \sum_t \sum_g \left(NLC_g u_{gt} + SUC_g y_{gt} + SDC_g z_{gt} \right) \tag{15a}$$

subject to:

(13b),(13c)

$$\left\{ \sum_t \sum_g \left(\hat{\mu}_{gt}^- P_g^{min} - \hat{\mu}_{gt}^+ P_g^{max} \right) u_{gt} \right.$$
$$- \sum_t \sum_g \left(\hat{\nu}_{gt}^+ RU_g + \hat{\nu}_{gt}^- RD_g \right)$$
$$+ \sum_g \left[\hat{\nu}_g^{0+} \left(-P_g^{ini} - RU_g \right) + \hat{\nu}_g^{0-} \left(P_g^{ini} - RD_g \right) \right]$$
$$- \sum_t \sum_i \left[\hat{\lambda}_{it} \sum_l \left(D_{lt} A_{il}^L \right) + \sum_{j | X_{ij} \neq 0} \hat{\psi}_{ijt} F_{ij}^{max} \right]$$
$$\left. \leq \zeta \right\}^{(\kappa)} \in \bigcup_{\kappa=2}^{k} G_\kappa'^O \tag{15b}$$

$$\left\{ \sum_t \sum_g \left(\hat{\mu}_{gt}^- P_g^{min} - \hat{\mu}_{gt}^+ P_g^{max} \right) u_{gt} \right.$$
$$- \sum_t \sum_g \left(\hat{\nu}_{gt}^+ RU_g + \hat{\nu}_{gt}^- RD_g \right)$$
$$+ \sum_g \left[\hat{\nu}_g^{0+} \left(-P_g^{ini} - RU_g \right) + \hat{\nu}_g^{0-} \left(P_g^{ini} - RD_g \right) \right]$$
$$- \sum_t \sum_i \left[\hat{\lambda}_{it} \sum_l \left(D_{lt} A_{il}^L \right) + \sum_{j | X_{ij} \neq 0} \hat{\psi}_{ijt} F_{ij}^{max} \right]$$
$$\left. \leq 0 \right\}^{(\kappa)} \in \bigcup_{\kappa=2}^{k} G_\kappa'^F \tag{15c}$$

$$\zeta \geq \zeta^{low} \tag{15d}$$

$$u_{gt}, y_{gt}, z_{gt} \in \{0, 1\} \quad \forall g, t \tag{15e}$$

The optimality and feasibility cuts that are appended to the MP up to the current iteration k are expressed by (15b) and (15c) respectively. Constraint (15d) is necessary in order to prevent the problem from being unbounded in the first iteration of the algorithm.

It should be noted that constraints involving only decision variables of the MP (e.g., generator minimum up and down time, reserve capacity constraints) can be added to the formulation of Section 4.1 directly, since they do not affect the DSP formulation or the expressions of the Benders cuts. Moreover, they do not influence the scalability of the cut selection step. This is due to the fact that the size of matrix **M** when Criterion I is used is independent of the MILP problem size, while under Criterion II the size of **M** depends only on the number of complicating variables and not on the number of MP constraints.

5. Numerical experiments

5.1. Implementation details

The HQC-MCMS algorithm was implemented in Python 3.9 using the Pyomo package (Bynum et al., 2021). All the classical MILP problems were solved using the Gurobi 9.5.0 solver (Gurobi Optimization, LLC, 2021) with a *MIPGap* of 0 % on a workstation with 2 Intel Xeon processors (24 cores, 3 GHz) and 128 GB of RAM. In order to find multiple solutions of the MP the solution pool functionality of Gurobi was used. The *PoolSearchMode* parameter was set to 1. This means that the solver continues the search for feasible solutions after the optimal solution has been found, however, no guarantees are provided about the quality of the additional feasible solutions. Note that when classical resources were utilized to solve the cut selection problem, the constrained optimization problem formulations (8) and (11) were used to obtain globally optimal solutions.

The QUBO problem instances were solved using the D-Wave Advantage 4.1 QA that was accessed via Amazon Braket. The D-Wave Advantage 4.1 QPU relies on the Pegasus topology and features more than 5000 qubits and 35 000 couplers (D-Wave Systems Inc., 2021). Embedding of the problem onto the physical QPU graph was performed using the *minorminer* package (D-Wave Systems Inc., 2017) with default settings. The chain strength value was set to 150% of the largest interaction coefficient observed in the problem Hamiltonian.

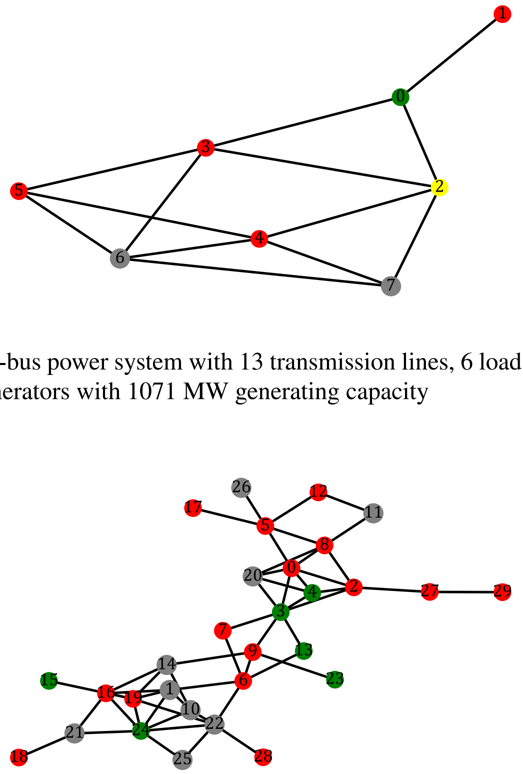
To derive values for the penalties in (10) and (12), the recommendations in Lucas (2014) were followed, such that the gain in H_A by violating a constraint is offset by the penalty incurred due to the violation. In the worst case, a feasible solution of problem (10) will have an objective function value of $H_A = \mathcal{P}_\mathcal{A} \sum_{i \in I} 1 = \mathcal{P}_\mathcal{A} |I|$, if all the cuts are added to cover all the columns of matrix **M**. Therefore, for problems of the type (10), $\mathcal{P}_\mathcal{A} = 1$ and $\mathcal{P}_{\mathcal{B}_j} = |I|$, $\forall j \in J$, where I, J are the sets of rows and columns of matrix **M** respectively. Similarly, for problem (12) the minimum value in the objective function is $H_A = -\mathcal{P}_\mathcal{A} \sum_{j \in J} 1 = -\mathcal{P}_\mathcal{A} |J|$, when all the columns of matrix **M** are covered by the selected cuts. As a result, for $\mathcal{P}_\mathcal{A} = 1$ the penalties could be set to $\mathcal{P}_{C_j} = |J|$, $\forall j \in J$ and $\mathcal{P}_\mathcal{B} = |J|$. In practice, it was found that for $\mathcal{P}_\mathcal{A} = 1$, increasing the penalty values to $\mathcal{P}_{C_j} = |I| + |J|$, $\forall j \in J$ and $\mathcal{P}_\mathcal{B} = |I| + |J|$ resulted in encountering less infeasible solutions. It is acknowledged that this approach is rather conservative and may provide weak lower bounds on the penalty values. Since the selection of the values of penalty weights may hinder the ability of the QPU to efficiently solve QUBO problems, it is desirable to reduce them to the extent possible (Quintero et al., 2022; García et al., 2022).

5.2. Input data

To investigate the applicability of the proposed methodology on the UC problem, the test systems displayed in Fig. 2 were randomly generated. The network topologies were generated based on the methodology that was proposed in Wang et al. (2008). First, buses are uniformly placed within a fixed area with width and height of 1. Then, given a distance requirement (in this study $[0, 0.4]$) between neighboring buses, the set of transmission lines are selected by sampling a Poisson distribution with its parameter set to 2.67. The reactance of a transmission line is considered proportional to its length. For simplicity a factor of 1/10 is assumed for all lines. Subsequently, the type of each bus is decided. It is assumed that 50% of the buses are load buses, 20% are generator buses and 30% have both loads and generators connected. In case the number of buses is such that the aforementioned percentages do not result in integers with a sum equal to the number of buses, the remaining buses are considered to be transfer buses (i.e., they do not connect loads or generators). The capacity of each line is sampled from a uniform distribution ranging from 15% to 35% of the total generating capacity of the system. The individual loads are assigned a percentage of the hourly normalized system load (with respect to the maximum generating capacity of the system) that is portrayed in Fig. 3 using the Dirichlet distribution. The Dirichlet distribution satisfies the requirements that each bus load fraction is positive and that the sum of the fractions is 1. Finally, generator parameters are constructed using the values presented in Table 2.

5.3. Simulation setup

For all the experiments the algorithm terminates if $\frac{UB-LB}{UB} 100\% < 0.5\%$. For the 8-bus system, the whole 24-hour load profile is used (72 MP decision variables are involved in the DSP), while 10 solutions of the MP are requested from the solver. For the 30-bus system 30 solutions of the MP are requested, however, considering the full 24 periods



(a) 8-bus power system with 13 transmission lines, 6 loads and 3 generators with 1071 MW generating capacity

(b) 30-bus power system with 51 transmission lines, 24 loads and 15 generators with 5395 MW generating capacity

Fig. 2. The test power systems. The nodes are color-coded. Red: load bus, Green: generation bus, Gray: load and generation bus, Yellow: transfer bus. Without loss of generality, Bus 0 is defined as the reference bus.

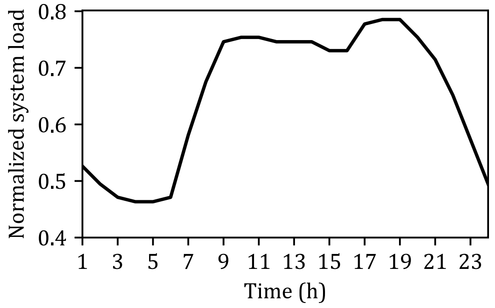


Fig. 3. Normalized system load. Extracted from (Ordoudis et al., 2016).

Table 2
Generator parameters. Ranges imply sampling a value from a uniform distribution.

Parameter	Value
P_g^{max}	$[60, 600]$
P_g^{min}	$[20\% P_g^{max}, 40\% P_g^{max}]$
RU_g	$\max(P_g^{min}, [20\% P_g^{max}, 40\% P_g^{max}])$
RD_g	RU_g
SUC_g	$[5, 1600]$
SDC_g	SUC_g
C_g	$[5, 30]$
NLC_g	$[3C_g, 6C_g]$
u_g^{ini}	$\{0, 1\}$
P_g^{ini}	$P_g^{min} u_g^{ini}$

the heuristic would not result in finding a feasible minor-embedding within the timeout limit of 1000s for all the cases. In particular, this concerned instances of the cut selection problem applied to optimality cuts using Criterion II. Although failure of the heuristic to find a minor embedding does not prove that it is impossible to minor-embed a given problem, as it will be demonstrated in Section 6, the relevant problems feature logical qubits with a high node degree, which implies that a large number of physical qubits need to be chained in order to implement the required logical qubits and interactions, potentially exceeding the capacity of the Pegasus graph. For this reason, only the first 8 time periods were considered (120 MP decision variables involved in the DSP). It is to be noted that the size of the UC problem instances studied is in line with –and even exceeds– current research (Chang et al., 2022; Koretsky et al., 2021; Mahroo and Kargarian, 2022; Nikmehr et al., 2022). For both power systems, Strategy II was applied using $\mathcal{M} = 3$ (both for feasibility and optimality cuts).

For all problems that were submitted to the QPU the annealing time was set to 10 μs and the anneal-read cycle was repeated 1000 times. Note that a single solution needs to be chosen in order to select the cuts that enter the MP in each iteration of the HQC-MCMS algorithm. The lowest-energy feasible solution returned by the QPU was implemented. If none of the solutions in the returned sample set were feasible, the solution with the lowest number of constraint violations was implemented, provided that $G_k^{\prime F}$ and $G_k^{\prime O}$ were not empty.

6. Results and discussion

6.1. Performance comparison of cut selection criteria and strategies

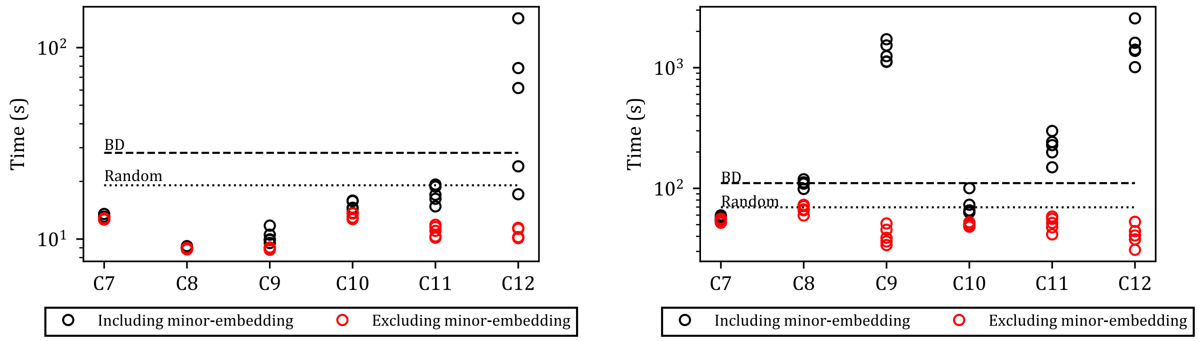
In order to assess the performance of the HQC-MCMS algorithm the cut selection procedure described in Algorithm 2 is executed both on a classical computer and a QA. The straightforward implementation of BD and the addition of all the available cuts to the MP are used as benchmarks of the performance of the multi-cut strategies. Moreover, a case in which the optimal solution and three randomly selected solutions of the MP are used to generate cuts is presented in order to establish the fact that Algorithm 2 performs a non-trivial cut selection.

Detailed computational characteristics for the two test systems are presented in Table 3. For ease of reference, the different combinations of cut selection strategies and cut selection criteria are denoted by C1–C12. Specifically, the number of iterations until convergence is achieved, the total time required to solve the MILP problem, as well as the time that is spent on each component of the algorithm are provided. The values presented in these tables are averaged across 5 executions of the algorithm for each case. Further details concerning the execution time of the individual runs of the algorithm for C7–C12 are shown in Fig. 4. The time for completing a quantum computation each time cut selection is triggered is given by $T_Q \approx T_P + \rho(T_A + T_R)$, where T_Q is the QPU access time, T_P is the quantum programming time, ρ is the number of times an anneal-read cycle is repeated, T_A is the annealing time and T_R is the time required to read a measurement. It is conventional to report only ρT_A as an equivalent to the classical CPU time (Jones et al., 2020), however, the total T_Q is reported in Table 3 for the cut selection component for the sake of completeness, since it corresponds to the wall clock time. Recognizing that in contrast with the exact solution to the cut selection problem reached by Gurobi, the solution of the QA is not necessarily optimal, timing results should be interpreted as an indication of the practical performance of the HQC-MCMS algorithm using the specific QA hardware rather than a benchmark between the QA and a state-of-the-art classical solver in search for optimal solutions to (10) and (12). Other metrics would be more suitable for such studies (King et al., 2015).

Table 3
Number of iterations, total solution time and timing of individual components for the two test systems. The values are averaged across five executions of the algorithm for each case. The convergence criterion is $\frac{UB-LB}{UB}100\% < 0.5\%$. All times are in (s). The columns *BD*, *MCMS* and *Random* stand for the straightforward implementation of BD, the classical MCMS algorithm without cut selection and the random cut selection benchmark, respectively. In cases C1–C6, the cut selection problem is solved with classical resources, while in cases C7–C12 the cut selection subroutine is assigned to the QPU. Cut selection Strategies I and II refer to the minimum set cover and maximum coverage problems, respectively. Cut selection Criteria I and II refer to the exclusion of infeasible solutions of the MP and MP variable coverage, respectively. Finally, where applicable, the application of cut selection to the set of optimality cuts (parameter *optSelect* in Algorithm 2) is denoted by True (T) or False (F).

				MCMS with cut selection						HQC-MCMS					
	BD	MCMS	Random	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Strategy	–	–	–		I			II			I			II	
Criterion	–	–	–	I	II	II	I	II	II	I	II	II	I	II	II
optSelect	–	–	–	–	F	T	–	F	T	–	F	T	–	F	T
							8-bus system								
Iterations	23.00	7.00	17.60	13.00	7.00	7.00	13.00	9.00	9.00	13.00	7.00	7.00	13.40	10.00	9.80
Total time	28.20	8.66	19.10	13.12	9.22	9.24	12.95	10.65	10.78	13.14	9.06	10.33	14.87	17.22	64.62
MP solution	1.74	2.17	3.77	1.77	1.63	1.30	1.71	1.86	1.75	1.76	1.58	1.39	1.89	2.10	1.68
DSP solution	26.46	6.49	15.33	8.36	6.50	6.61	8.34	7.15	7.13	8.37	6.58	6.63	8.54	7.55	7.47
M construction	–	–	–	2.37	0.32	0.42	2.29	0.49	0.54	2.27	0.33	0.40	2.30	0.52	0.58
Cut selection	–	–	–	0.62	0.78	0.92	0.62	1.16	1.35	0.33	0.39	0.50	0.43	0.82	0.94
Minor-emb. (feas.)	–	–	–	–	–	–	–	–	–	0.41	0.17	0.18	1.71	6.23	7.62
Minor-emb. (opt.)	–	–	–	–	–	–	–	–	–	–	–	1.23	–	–	46.34
							30-bus system								
Iterations	51.00	10.00*	24.40	9.00	11.00	10.00	9.00	9.00	9.00	9.00	10.60	8.40	9.00	10.20	11.40
Total time	110.50	80.56	69.65	53.22	64.73	35.01	52.86	38.11	29.16	56.31	109.92	1348.55	73.14	223.53	1594.90
MP solution	29.00	62.86	37.58	27.74	40.67	12.25	26.51	19.19	8.71	28.40	45.88	18.04	23.33	30.20	12.24
DSP solution	81.51	17.70	32.07	16.12	20.45	17.82	16.10	16.17	15.83	15.95	18.77	19.07	16.29	17.91	23.40
M construction	–	–	–	8.77	1.92	2.56	9.64	1.74	2.31	8.91	1.63	2.11	9.60	1.84	3.04
Cut selection	–	–	–	0.59	1.70	2.38	0.61	1.00	2.30	0.37	1.02	1.91	0.44	1.08	2.61
Minor-emb. (feas.)	–	–	–	–	–	–	–	–	–	2.67	42.62	53.43	23.49	172.51	360.04
Minor-emb. (opt.)	–	–	–	–	–	–	–	–	–	–	–	1253.99	–	–	1193.57

*For $\epsilon \to 0$, MCMS without cut selection yields a lower bound on the number of iterations.



(a) 8-bus system

(b) 30-bus system

Fig. 4. HQC-MCMS execution time for different cut selection strategies and criteria using QA, compared to the straightforward implementation of BD and the random benchmark. Markers correspond to the five independent executions of the algorithm for each case. The dashed line shows the average execution time of BD. The dotted line indicates the average execution time of the random benchmark. The vertical axis is in logarithmic scale.

6.1.1. HQC-MCMS performance on the 8-bus system

In Table 3 it can be observed that all multi-cut strategies result in a reduction in the number of iterations with respect to the straightforward implementation of BD of up to 69.6%. Since the 8-bus power system problem instance is small, adding all the generated cuts at each iteration to the MP results in the lowest total execution time, despite the increase in the MP solution time. The reason for this is that all cases in which cut selection is applied, additional components are executed, namely the construction of the indicator matrix **M** and the solution of the cut selection problem. For this small problem instance, the significant reduction in the DSP solution time, which in turn depends on the number of iterations, is dominant. Nonetheless, applying any of the cut selection strategies and criteria results in a comparable acceleration with respect to the straightforward implementation of BD when classical resources are used.

For the 8-bus system, the average cut selection time is by up to 50% lower using QA for cases C7–C12 in comparison with their classical counterparts C1–C6. However, when QA is used, additional time is required in order to map the problem to the QPU topology, which can significantly influence the overall performance of the HQC-MCMS algorithm. It can be seen that when Strategy I is used for cut selection, significantly less time is required in order to find a minor embedding in comparison with Strategy II. Also, applying cut selection using D^F resulted in faster minor embedding in comparison with D^O. Nonetheless, for cases C7–C11 the HQC-MCMS algorithm performed better, in terms of total solution time, than the straightforward implementation of BD, while C8 also performed better than its classical counterpart C2. Finally, it is to be noted that although, on average, the total solution time that is reported in C12 exceeds that of the straightforward implementation of BD, the performance of the minor embedding heuristic is highly variable in this case (standard deviation of 50.31 s) and two instances were found to perform better than BD, as it can be seen in Fig. 4(a).

6.1.2. HQC-MCMS performance on the 30-bus system

A similar analysis is performed for the 30-bus system based on the results that are presented in Table 3. First, it can be observed that all the multi-cut solution approaches result in a reduction in the number of iterations with respect to the straightforward BD implementation of up to 83.5%. Contrary to the 8-bus system, adding all the available cuts in each iteration to the MP results in a proliferation of the MP solution time that is sufficient to render it a less performant option than the application of any of the cut selection strategies when the classical solver is used. Both for Strategies I and II the best performance was observed when Criterion II was used by applying cut selection both on feasibility and optimality cuts (C3 and C6) due to the maximum reduction in the size of the MP. However, the corresponding cases using QA (C9 and C12) were the least performant ones due to the excessive time that was required in order to map the corresponding problems to the QPU topology. This is confirmed by the results that are shown in Fig. 4(b).

Similarly to the results for the 8-bus power system, except for cases C11 and C12, cut selection was faster when QA was used also for the 30-bus power system by up to 40%. The worse performance of cut selection in C11 and C12 can be attributed to the higher average number of iterations that were required for convergence to the specified tolerance in comparison with their classical counterparts C8 and C9. Despite the significant time that is spent on minor-embedding, C7 and C10 are characterized by a lower solution time in comparison with adding all the available cuts in the MP, while C8 was slightly faster than the straightforward BD implementation.

The results on the 30-bus test system reveal a significant difference in the time that is required for minor-embedding the logical problem graphs to the QPU topology when different cut selection strategies and cut selection criteria are applied. Minor-embedding time exhibits strong dependence on the complexity of the cut selection problem that is solved using QA, as well as on the size of matrix **M** after inspection.

First, Strategy II relies on a more complex optimization problem compared to Strategy I, with two sets of decision variables and two sets of constraints. Specifically, the components of χ are always fully-connected due to constraint (12c), whereas in Strategy I, the number of interactions between components of χ is determined only by the structure of matrix **M**, i.e., the characteristics of the cuts. Moreover, in Strategy II, the components of χ interact also with components of $\boldsymbol{\phi}$. As a consequence, the resulting problem graphs tend to have a higher maximum node degree compared to those of Strategy I. Also, because under Criterion II there are more constraints in comparison with Criterion I ($|G|\times|T| \gg S$), the mean degree of the QUBO problems in the former case tends to be higher due to more interactions between the decision variables and ancillary qubits. Since the physical qubits on the hardware graph have a fixed degree (15 in the case of the Pegasus QPU graph), embedding problems with high node degrees is more computationally challenging. This is confirmed by the results portrayed in Figs. 5(a) and 5(b) where the minor-embedding time of the individual cut selection instances is plotted against the maximum degree of the node degree distribution of the logical problem graph.

The second observation is that when Criterion I is used (C7 and C10), the number of logical qubits is higher compared to Criterion II (C8, C9 and C11, C12 respectively). This is observed in Figs. 6(a) and 6(b) where the maximum number of logical qubits that are required to model the cut selection problem is plotted versus the actual number of logical qubits after inspecting matrix **M**. From Fig. 6(b) it is also evident that the size of QUBO instances is generally larger for optimality cuts in comparison with feasibility cuts. This may be attributed to the different density of feasibility and optimality cuts (the number of non-zero MP variable coefficients), which is higher for the latter. For instance, for the worst-performing instance of C9, the lowest average density of feasibility cuts was 7.92% and the highest 12.50%, whereas for the optimality cuts the lowest average density was 35.83% and the highest 55.65%. As a consequence, preprocessing of matrix **M** is more effective in terms of reducing the size of the cut selection QUBO problem for feasibility cuts.

Table 4
Largest QUBO problem instance submitted to the QPU for each case (8-bus system).

	Decision variables	Ancillary qubits	Interactions
C7	5	16	78
C8	7	0	0
C9	4	40	184
C10	15	18	124
C11	28	21	84
C12	48	83	634

6.2. Quality of the solutions obtained using quantum annealing

As expected, all executions of the HQC-MCMS algorithm terminated within the specified optimality tolerance. To evaluate the qualitative characteristics of the solutions that are returned by QA, the increase in the number of constraints of the MP due to the addition of feasibility and optimality cuts across iterations is displayed for the 8-bus and 30-bus test systems in Figs. 7 and 8 respectively.

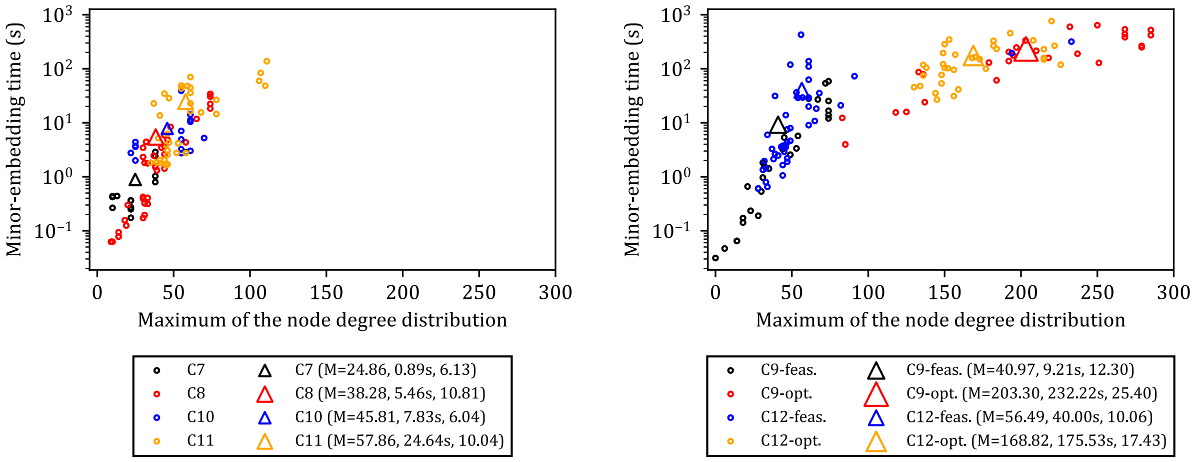
6.2.1. Solution characteristics for the 8-bus system

For the 8-bus system it can be seen in Fig. 7 that the increase in the number of MP constraints when cut selection problem is solved using QA corresponds to that of the classical solution for cases C7–C9. This is the reason why no significant differences are observed in the MP solution time in comparison with their classical counterparts, while the same number of iterations are performed. On the contrary, differences are observed when Strategy II is employed in combination with any of the cut selection criteria (C10–C12). Although in all three cases there are instances of the quantum step which result in an optimal cut selection trajectory, the average number of iterations is increased due to the sub-optimal trajectories that are generated in some of the runs. It should be noted that the size of the problems that are submitted to the QPU is relatively small, with the largest instance solved using QA in each case reported in Table 4. For this reason, the lowest-energy solutions tend to satisfy the constraints of the cut selection strategies. It is also worth mentioning that for all the problems related to this test system that were submitted to the QPU, only a single decision was made based on a sample with broken logical chains (chain break fraction of 0.76%). The comparable performance of the classical and quantum resources shows that HQC-MCMS is a viable decomposition-based HQC algorithm for small-scale optimization problems.

6.2.2. Solution characteristics for the 30-bus system

For the 30-bus test system, the results portrayed in Fig. 8 are indicative of the performance of QA as a heuristic for cut selection when HQC-MCMS is applied to larger-scale optimization problems. For Strategy I, although in C7 the application of QA resulted, in each iteration, in the selection of a comparable number of cuts with its classical counterpart, using Criterion II results in trajectories that significantly increase the size of the MP. This is reflected in the increased MP solution time, but also in the reduced average number of iterations required for convergence in C8 and C9. For Strategy II, the opposite behavior is observed. With the exception of C10 in which the results are consistent with the optimal results obtained by classical optimization, C11 and C12 are characterized by a higher number of iterations in comparison with their classical counterparts. This is an indication that a larger number of sub-optimal cut choices are made by QA. Nonetheless, despite the cut selection trajectories generated by QA in C9 and C12 departing significantly from the optimal, the cut selection remains effective in terms of MP size management for Criterion II applied both to feasibility and optimality cuts. This is evident from the lower average time that is spent on solving the MP compared to the benchmarks, as reported in Table 3.

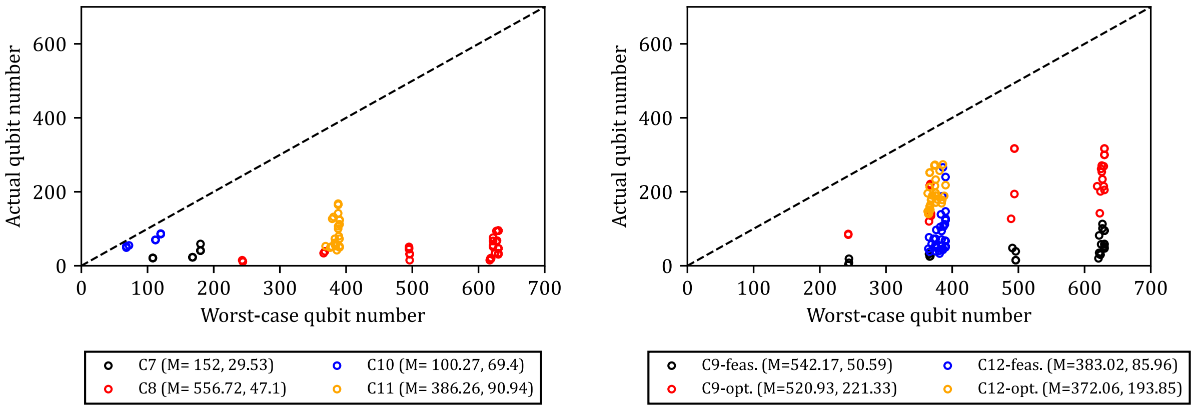
To provide further insight into the quality of the QA solutions for the larger 30-bus system, additional details for different cut selection



(a) Cut selection applied only to feasibility cuts

(b) Cut selection applied to feasibility and optimality cuts

Fig. 5. Minor-embedding time versus the maximum of the node degree distribution of the logical cut selection problem graph under different cut selection strategies and criteria (30-bus system). Means across all the instances are depicted with triangle markers and mean values are presented in the legend in the format (x, y, z), where z is the mean of the node degree distribution. The size of the triangle markers is proportional to the average value of the mean of the node degree distribution. The vertical axis is in logarithmic scale.



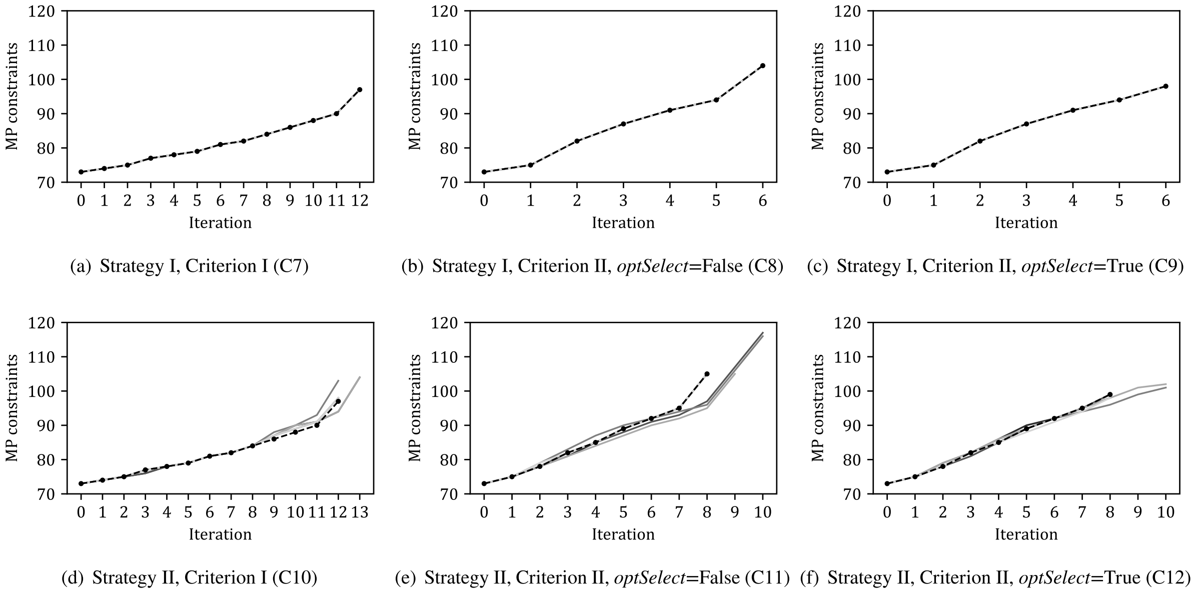
(a) Cut selection applied only to feasibility cuts

(b) Cut selection applied to feasibility and optimality cuts

Fig. 6. The actual number of logical qubits after the inspection of matrix **M** versus the worst-case number of qubits required to represent the cut selection problem under different cut selection strategies and criteria (30-bus system). Mean values across all the instances are presented in the legend in the format (x, y). The dashed line $x = y$ is used a visual aid to illustrate the effectiveness of the inspection step; if a point lies on $x = y$, then the actual number of required logical qubits corresponds to the worst-case logical qubit number.

Table 5
QA solution details for the 30-bus system across all executions of the HQC-MCMS algorithm. The first column reports the total number of times the QPU was used to perform cut selection in each case. The second column presents the number of times broken chains were observed in the implemented solution, while the third column reports the maximum chain break fraction. The representative chain length is provided in the fourth column. In the fifth column, the number of times the set of solutions returned by the QPU contained only infeasible solutions is reported. The number of violated constraints and the total number of constraints in the instance that presented the most violations are given in the sixth and seventh columns respectively. The characteristics of the largest problem instance in terms of logical qubits that was solved in each case are given in columns eight to ten.

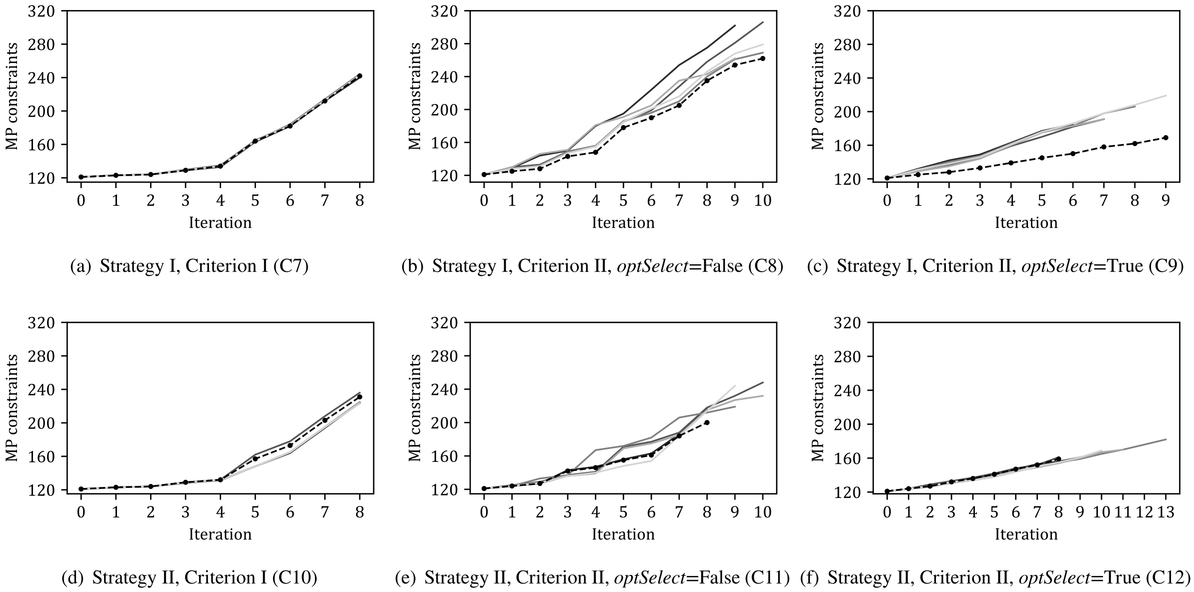
		Minor embedding				Worst case infeasibility		Largest problem instance		
	QPU calls	Decisions with broken chains	Max. chain break fraction (%)	Representative chain length	Infeasible solutions	Violated constraints	Constraints	Decision variables	Ancillary qubits	Interactions
C7	15	1	1.69	5	–	–	–	5	54	294
C8	39	1	1.05	7.75	–	–	–	17	78	1087
C9 (feas.)	29	3	1.72	7.25	–	–	–	18	95	752
C9 (opt.)	27	19	5.09	34.2	2	1	43	30	287	7975
C10	15	0	–	9.33	–	–	–	35	52	376
C11	35	9	4.83	14	4	1	33	69	99	600
C12 (feas.)	45	12	7.35	13.71	5	2	22	117	149	861
C12 (opt.)	34	17	6.42	27.87	1	1	28	109	165	5083



(a) Strategy I, Criterion I (C7) (b) Strategy I, Criterion II, *optSelect*=False (C8) (c) Strategy I, Criterion II, *optSelect*=True (C9)

(d) Strategy II, Criterion I (C10) (e) Strategy II, Criterion II, *optSelect*=False (C11) (f) Strategy II, Criterion II, *optSelect*=True (C12)

Fig. 7. Number of MP constraints in each iteration of the HQC-MCMS algorithm applied to the 8-bus system for different cut selection strategies and criteria. The trajectories corresponding to each of the 5 runs using a QPU for cut selection are represented by the gray lines. The optimal solution trajectory found using classical optimization is marked with the dashed black line.



(a) Strategy I, Criterion I (C7) (b) Strategy I, Criterion II, *optSelect*=False (C8) (c) Strategy I, Criterion II, *optSelect*=True (C9)

(d) Strategy II, Criterion I (C10) (e) Strategy II, Criterion II, *optSelect*=False (C11) (f) Strategy II, Criterion II, *optSelect*=True (C12)

Fig. 8. Number of MP constraints in each iteration of the HQC-MCMS algorithm applied to the 30-bus system for different cut selection strategies and criteria. The trajectories corresponding to each of the 5 runs using a QPU for cut selection are represented by the gray lines. The optimal solution trajectory found using classical optimization is marked with the dashed black line.

strategies and criteria are presented in Table 5 for all the executions of the HQC-MCMS using the QPU. The first observation is that the deterioration of the solution quality observed in Fig. 8 across different cases is related to the size of the QUBO instances that are assigned to the QPU. Furthermore, the size of the problem impacts the qualitative characteristics of minor-embedding. The minor embedding of larger QUBO problems tends to be characterized by longer physical qubit chains. To quantify this observation, a representative chain length is calculated. Specifically, for each minor embedding that is found in a single run if cut selection is triggered, the maximum qubit chain length is recorded and the results are averaged. The maximum across the five runs is considered as the representative chain length. As expected from the results in Figs. 5 and 6, the chain length was found to be significantly higher under Criterion II, in particular when cut selection is applied to optimality cuts. For this reason, larger problem instances tend to result in more decisions being made based on samples with broken logical chains and higher maximum chain break fractions. Notably, QA managed to discover feasible solutions in 95% of the solved cut selection problem instances, whereas in the few cases where a cut selection was made based on an infeasible solution, only a few of the problem constraints were not satisfied. Summarizing Table 5, Strategy I is associated with more favorable solution characteristics than Strategy II. The same can be said about Criterion I in comparison with Criterion II, especially when cut selection is applied to optimality cuts.

6.3. Practical limitations and future outlook

Based on the computational experience with the UC problem, two limitations of the proposed HQC algorithm can be identified. First, minor-embedding has to be repeated in each iteration where the QPU is called because a different matrix **M** is available. Although performing cut selection using quantum resources may be a computationally efficient procedure itself, the impact of the time that is required in order to map the problem to the hardware graph can adversely impact the overall performance of the algorithm by introducing a classical computation bottleneck, as it can be seen in several instances in Table 3. To overcome this limitation, either more efficient minor-embedding heuristics or a systematic way to exploit previously generated minor embeddings need to be developed. One pragmatic solution is to precalculate a minor embedding of the largest fully-connected QUBO graph that can be embedded onto the QPU and use it to trivially map any QUBO of smaller size. However, this approach introduces the risk of rejecting larger non fully-connected QUBO instances that may still be embeddable onto a given QPU, effectively restricting the size of problems that can be solved by the current generation of QA. For perspective, the largest complete graph that is known to be embeddable onto the Pegasus P16 graph is K_{185}, while for the previous generation Chimera C16 graph it is K_{65} (Zbinden et al., 2020). Note that although the average node degree of the logical QUBO graphs of Problems (10) and (12) can be particularly high, they are never complete graphs. The reason for this is that the ancillary qubits that are introduced to model a particular constraint induce interactions only with other ancillary qubits and components of the decision variables χ and $\boldsymbol{\phi}$ that are involved in that constraint.

The second shortcoming of the proposed algorithm in the context of NISQ hardware is the dependence of the size of matrix **M** under Criterion II on the number of complicating variables of the MILP problem. This imposes a limit on the size of MILP problems for which quantum resources can be used, if the QUBO problems (10) and (12) are to be minor-embedded directly onto the QPU. In the case of the UC problem, the limiting factor would be the number of generators and time periods, since the number of columns of **M** and, therefore, the required number of logical qubits in the worst case, depends linearly on the number $|G|\times|T|$ of complicating variables u_{gt} (generator status). No limitation is induced by the number of buses, transmission lines and loads, since the relevant decision variables appear only in the SPs. On the contrary, the size of matrix **M** under Criterion I depends solely on the number of MP solutions, which is determined by the user-selected parameter S and is independent of the size of the MILP problem (13). As a consequence, although Criterion II appears to be computationally advantageous in comparison with Criterion I when classical computing resources are used, the latter may find wider applicability to solving MILP problems by using NISQ hardware to perform cut selection.

Establishing bounds on the maximum number S of cuts that can be generated during an iteration and the number of complicating variables m a MILP problem can have such that the resulting QUBO problems (10) and (12) can be directly embedded onto a QPU is particularly challenging. Clearly, S and m are limited by the embeddability of the largest instance of the cut selection problem that must be solved in an execution of the HQC-MCMS algorithm. However, the structure of matrix **M** and, therefore, the actual number of logical qubits that need to be introduced and interactions between them depend on the characteristics of the cuts, which are not known in advance. Moreover, the fact that cuts tend to be low-density implies that, in practice, **M** is sparse and its size significantly smaller than $S\times S$ (Criterion I) and $S\times m$ (Criterion II) because of the application of the preprocessing technique. For such arbitrary QUBO problems, graph properties they must fulfill in order to determine whether it is embeddable onto the QPU graph are not easy to recognize (Lobe and Lutz, 2021). Nonetheless, an interesting extension to the HQC-MCMS algorithm so that it can be applied to problems with large numbers of complicating variables would be to decompose QUBO-based cut selection problem instances that are too large to directly minor-embed onto the QPU (Bass et al., 2021).

Despite the aforementioned limitations, contingent on algorithmic improvements and expecting the availability of more densely-connected QPU topologies in the future, the proposed approach may find wide applicability since its convergence is resilient to the heuristic nature of QC and it does not require the MILP problem to possess SPs with special structure.

7. Conclusion

In this paper, a hybrid quantum–classical (HQC) multi-cut Benders decomposition (BD) strategy to solve general mixed-integer linear programming (MILP) problems to optimality was presented. The proposed approach exploits multiple feasible solutions of the master problem (MP) in order to generate multiple feasibility and optimality cuts. Adding multiple cuts to the MP improves the convergence rate of BD. However, the increase in the size of the MP may adversely impact solution time. In order to manage the size of the MP and exploit the availability of multiple cuts, a cut selection procedure that can be assigned to a quantum computer was developed. Two different criteria and two different cut selection strategies based on pure binary optimization problems that can be solved using quantum computing were studied. The HQC algorithm was applied to the Unit Commitment problem and computational experiments were conducted using the D-Wave Advantage 4.1 quantum annealer. Results on two test power systems showed that although it is viable for quantum resources to be used as an alternative to classical resources for cut selection for small-scale problems, current hardware limitations must be overcome and the efficiency of minor-embedding techniques should be improved before effectively applying the proposed approach to large-scale problem instances. Future research will focus on further improving the proposed HQC algorithm according to the limitations that were identified in Section 6.3, applying it on different use cases, and pursuing the generalization of the proposed methodology to encompass different types of optimization problems.

CRediT authorship contribution statement

Nikolaos G. Paterakis: Conceptualization, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Aaronson, S., 2015. Read the fine print. Nat. Phys. 11, 291–293.
Ajagekar, A., Humble, T., You, F., 2020. Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. Comput. Chem. Eng. 132, 106630.
Ajagekar, A., You, F., 2019. Quantum computing for energy systems optimization: Challenges and opportunities. Energy 179, 76–89.
Ajagekar, A., You, F., 2021. Quantum computing based hybrid deep learning for fault diagnosis in electrical power systems. Appl. Energy 303, 117628.
Alemany, J.M., Magnago, F., Moitre, D., 2013. Benders decomposition applied to security constrained unit commitment. IEEE Lat. Am. Trans. 11 (1), 421–425.
Anghinolfi, D., Paolucci, M., Robba, M., 2016. Optimal planning of door-to-door multiple materials separated waste collection. IEEE Trans. Autom. Sci. Eng. 13 (4), 1448–1457.
Asl, N.B., MirHassani, S.A., 2019. Accelerating benders decomposition: multiple cuts via multiple solutions. J. Comb. Optim. 37, 806–826.
Bass, G., Henderson, M., Heath, J., Dulny III, J., 2021. Optimizing the optimizer: decomposition techniques for quantum annealing. Quantum Mach. Intell. 10.
Benders, J., 1962. Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4, 238–252.

Bernal, D.E., Ajagekar, A., Harwood, S.M., Stober, S.T., Trenev, D., You, F., 2022. Perspectives of quantum computing for chemical engineering. AIChE J. 68 (6), e17651.

Bernal, D.E., Booth, K.E.C., Dridi, R., Alghassi, H., Tayur, S., Venturelli, D., 2020. Integer programming techniques for minor-embedding in quantum annealers. In: Hebrard, E., Musliu, N. (Eds.), Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Springer International Publishing, Cham, pp. 112–129.

Braine, L., Egger, D.J., Glick, J., Woerner, S., 2021. Quantum algorithms for mixed binary optimization applied to transaction settlement. IEEE Trans. Quantum Eng. 2, 3101208, 1–8.

Bynum, M.L., Hackebeil, G.A., Hart, W.E., Laird, C.D., Nicholson, B.L., Siirola, J.D., Watson, J.-P., Woodruff, D.L., 2021. Pyomo–Optimization Modeling in Python, Vol. 67, third ed. Springer Science & Business Media.

Cai, J., Macready, W.G., Roy, A., 2014. A practical heuristic for finding graph minors. arXiv:1406.2741.

Chang, C.-Y., Jones, E., Yao, Y., Graf, P., Jain, R., 2022. On hybrid quantum and classical computing algorithms for mixed-integer programming. arXiv:2010.07852.

Chen, Y., Liu, F., Liu, B., Wei, W., Mei, S., 2016. An efficient MILP approximation for the hydro-thermal unit commitment. IEEE Trans. Power Syst. 31 (4), 3318–3319.

Childs, A.M., Farhi, E., Preskill, J., 2001. Robustness of adiabatic quantum computation. Phys. Rev. A 65, 012322.

Conejo, A.J., Castillo, E., Minguez, R., Garcia-Bertrand, R., 2006. Decomposition Techniques in Mathematical Programming. Springer-Verlag Berlin Heidelberg.

D-Wave Systems Inc., 2017. minorminer GitHub repository. URL: https://github.com/dwavesystems/minorminer.

D-Wave Systems Inc., 2021. QPU-specific physical properties: Advantage_system4.1.

Elsayed, N., Maida, A.S., Bayoumi, M., 2019. A review of quantum computer energy efficiency. In: Proc. 2019 IEEE Green Technologies Conference. GreenTech, Lafayette, LA, USA, pp. 1–3.

Eskandarpour, R., Bahadur Ghosh, K.J., Khodaei, A., Paaso, A., Zhang, L., 2020. Quantum-enhanced grid of the future: A primer. IEEE Access 8, 188993–189002.

Eskandarpour, R., Gokhale, P., Khodaei, A., Chong, F.T., Passo, A., Bahramirad, S., 2020. Quantum computing for enhancing grid security. IEEE Trans. Power Syst. 35 (5), 4135–4137.

Farhi, E., Goldstone, J., Gutmann, S., 2014. A quantum approximate optimization algorithm. arXiv:1411.4028.

Farhi, E., Goldstone, J., Gutmann, S., Sipser, M., 2000. Quantum computation by adiabatic evolution. arXiv:quant-ph/0001106.

Feng, F., Zhou, Y., Zhang, P., 2021. Quantum power flow. IEEE Trans. Power Syst. 36 (4), 3810–3812.

Fu, Y., Li, Z., Wu, L., 2013. Modeling and solution of the large-scale security-constrained unit commitment. IEEE Trans. Power Syst. 28 (4), 3524–3533.

Fu, Y., Shahidehpour, M., Li, Z., 2005. Security-constrained unit commitment with AC constraints. IEEE Trans. Power Syst. 20 (2), 1001–1013.

Gambella, C., Simonetto, A., 2020. Multiblock ADMM heuristics for mixed-binary optimization on classical and quantum computers. IEEE Trans. Quantum Eng. 1, 3102022, 1–22.

García, M.D., Ayodele, M., Moraglio, A., 2022. Exact and sequential penalty weights in quadratic unconstrained binary optimisation with a digital annealer. In: Proc. 2022 Genetic and Evolutionary Computation Conference. pp. 184–187.

Giani, A., Eldredge, Z., 2021. Quantum computing opportunities in renewable energy. SN Comput. Sci. 2, 393.

Gilliam, A., Woerner, S., Gonciulea, C., 2021. Grover adaptive search for constrained polynomial binary optimization. Quantum 5, 428.

Grant, E., Humble, T.S., Stump, B., 2021. Benchmarking quantum annealing controls with portfolio optimization. Phys. Rev. Appl. 15, 014012.

Grossman, T., Wool, A., 1997. Computational experience with approximation algorithms for the set covering problem. European J. Oper. Res. 101 (1), 81–92.

Gurobi Optimization, LLC, 2021. Gurobi Optimizer Reference Manual. URL: https://www.gurobi.com.

Harwood, S., Gambella, C., Trenev, D., Simonetto, A., Bernal, D., Greenberg, D., 2021. Formulating and solving routing problems on quantum computers. IEEE Trans. Quantum Eng. 2, 3100118, 1–17.

Johnson, M.W., Amin, M.H.S., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A.J., Johansson, J., Bunyk, P., Chapple, E.M., Enderud, C., Hilton, J.P., Karimi, K., Ladizinsky, E., Ladizinsky, N., Oh, T., Perminov, I., Rich, C., Thom, M.C., Tolkacheva, E., Truncik, C.J.S., Uchaikin, S., Wang, J., Wilson, B., Rose, G., 2011. Quantum annealing with manufactured spins. Nature 473, 194–198.

Jones, E.B., Kapit, E., Chang, C.-Y., Biagioni, D., Vaidhynathan, D., Graf, P., Jones, W., 2020. On the computational viability of quantum optimization for PMU placement. In: Proc. 2020 IEEE Power & Energy Society General Meeting. PESGM, Montreal, QC, Canada, pp. 1–5.

King, J., Yarkoni, S., Nevisi, M.M., Hilton, J.P., McGeoch, C.C., 2015. Benchmarking a quantum annealing processor with the time-to-target metric. arXiv:1508.05087.

Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y., 2014. The unconstrained binary quadratic programming problem: A survey. J. Comb. Optim. 28, 58–81.

Koretsky, S., Gokhale, P., Baker, J.M., Viszlai, J., Zheng, H., Gurung, N., Burg, R., Paaso, E.A., Khodaei, A., Eskandarpour, R., Chong, F.T., 2021. Adapting quantum approximation optimization algorithm (QAOA) for unit commitment. In: Proc. 2021 IEEE International Conference on Quantum Computing and Engineering. QCE, Broomfield, CO, USA, pp. 181–187.

Lobe, E., Lutz, A., 2021. Minor embedding in broken Chimera and Pegasus graphs is NP-complete. arXiv:2110.08325.

Lopion, P., Markewitz, P., Robinius, M., Stolten, D., 2018. A review of current challenges and trends in energy systems modeling. Renew. Sustain. Energy Rev. 96, 156–166.

Lucas, A., 2014. Ising formulations of many NP problems. Front. Phys. 2.

Mahroo, R., Kargarian, A., 2022. Hybrid quantum-classical unit commitment. In: Proc. 2022 IEEE Texas Power and Energy Conference. TPEC, College Station, TX, USA, pp. 1–5.

McGeoch, C.C., 2020. Theory versus practice in annealing-based quantum computing. Theoret. Comput. Sci. 816, 169–183.

Nannicini, G., 2019. Performance of hybrid quantum-classical variational heuristics for combinatorial optimization. Phys. Rev. E 99, 013304.

Nasri, A., Kazempour, S.J., Conejo, A.J., Ghandhari, M., 2016. Network-constrained AC unit commitment under uncertainty: A Benders' decomposition approach. IEEE Trans. Power Syst. 31 (1), 412–422.

Nikmehr, N., Zhang, P., Bragin, M., 2022. Quantum distributed unit commitment: an application to microgrids. IEEE Trans. Power Syst. 37, 3592–3603.

Olatunji, O.O., Adedeji, P.A., Madushele, N., 2021. Chapter 22 - quantum computing in renewable energy exploration: status, opportunities, and challenges. In: Azar, A.T., Kamal, N.A. (Eds.), Design, Analysis, and Applications of Renewable Energy Systems. In: Advances in Nonlinear Dynamics and Chaos (ANDC), Academic Press, pp. 549–572.

Ordoudis, C., Pinson, P., Morales González, J., Zugno, M., 2016. An Updated Version of the IEEE RTS 24-Bus System for Electricity Market and Power System Operation Studies. Technical Report, Technical University of Denmark, URL: https://orbit.dtu.dk/en/publications/an-updated-version-of-the-ieee-rts-24-bus-system-for-electricity-.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A., O'Brien, J.L., 2014. A variational eigenvalue solver on a photonic quantum processor. Nature Commun. 5, 4213.

Preskill, J., 2018. Quantum computing in the NISQ era and beyond. Quantum 2, 79.

Priesmann, J., Nolting, L., Praktiknjo, A., 2019. Are complex energy system models more accurate? An intra-model comparison of power system optimization models. Appl. Energy 255, 113783.

Quintero, R., Bernal, D., Terlaky, T., Zuluaga, L.F., 2022. Characterization of QUBO reformulations for the maximum k-colorable subgraph problem. Quantum Inf. Process 21.

Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W., 2017. The Benders decomposition algorithm: A literature review. European J. Oper. Res. 259 (3), 801–817.

Sævarsson, B., Chatzivasileiadis, S., Jóhannsson, H., Østergaard, J., 2022. Quantum computing for power flow algorithms: Testing on real quantum computers. In: Proc. 11th Bulk Power Systems Dynamics and Control Symposium. IREP 2022, Banff, Canada, pp. 1–8.

Saharidis, G., Ierapetritou, M., 2013. Speed-up Benders decomposition using maximum density cut (MDC) generation. Ann. Oper. Res. 210, 101–123.

Saharidis, G.K.D., Minoux, M., Ierapetritou, M., 2010. Accelerating Benders method using covering cut bundle generation. Int. Trans. Oper. Res. 17, 221–237.

Stollenwerk, T., Lobe, E., Jung, M., 2019. Flight gate assignment with a quantum annealer. In: Feld, S., Linnhoff-Popien, C. (Eds.), Quantum Technology and Optimization Problems. Springer International Publishing, Cham, pp. 99–110.

Su, L., Tang, L., Grossmann, I.E., 2015. Computational strategies for improved MINLP algorithms. Comput. Chem. Eng. 75, 40–48.

Takabe, S., Maehara, T., Hukushima, K., 2018. Typical approximation performance for maximum coverage problem. Phys. Rev. E 97, 022138.

Tamura, K., Shirai, T., Katsura, H., Tanaka, S., Togawa, N., 2021. Performance comparison of typical binary-integer encodings in an ising machine. IEEE Access 9, 81032–81039.

Tang, L., Jiang, W., Saharidis, G.K.D., 2013. An improved Benders decomposition algorithm for the logistics facility location problem with capacity expansions. Ann. Oper. Res. 210, 165–190.

Tang, Z., Qin, Y., Jiang, Z., Krawec, W.O., Zhang, P., 2021. Quantum-secure microgrid. IEEE Trans. Power Syst. 36 (2), 1250–1263.

Tovar-Facio, J., Martín, M., Ponce-Ortega, J.M., 2021. Sustainable energy transition: modeling and optimization. Curr. Opin. Chem. Eng. 31, 100661.

Wang, Z., Thomas, R.J., Scaglione, A., 2008. Generating random topology power grids. In: Proc. 41st Annual Hawaii International Conference on System Sciences. HICSS 2008, Waikoloa, HI, USA, p. 183.

Wen, Y., Guo, C., Pandžić, H., Kirschen, D.S., 2016. Enhanced security-constrained unit commitment with emerging utility-scale energy storage. IEEE Trans. Power Syst. 31 (1), 652–662.

Wu, L., Shahidehpour, M., 2010. Accelerating the Benders decomposition for network-constrained unit commitment problems. Energy Syst. 1, 339–376.

Yang, J., Xu, X., Yin, D., Ma, Z., Shen, L., 2019. A space mapping based 0–1 linear model for onboard conflict resolution of heterogeneous unmanned aerial vehicles. IEEE Trans. Veh. Technol. 68 (8), 7455–7465.

You, F., Grossmann, I.E., 2013. Multicut Benders decomposition algorithm for process supply chain planning under uncertainty. Ann. Oper. Res. 210, 191–211.

Zbinden, S., Bärtschi, A., Djidjev, H., Eidenbenz, S., 2020. Embedding algorithms for quantum annealers with Chimera and Pegasus connection topologies. In: Proc. 2020 International Conference on High Performance Computing. pp. 187–206.

Zhao, Z., Fan, L., Han, Z., 2022. Hybrid quantum Benders' decomposition for mixed-integer linear programming. In: Proc. 2022 IEEE Wireless Communications and Networking Conference. WCNC, Austin, TX, USA.

Zheng, Q.P., Wang, J., Liu, A.L., 2015. Stochastic optimization for unit commitment—A review. IEEE Trans. Power Syst. 30 (4), 1913–1924.

Zhou, Y., Feng, F., Zhang, P., 2021. Quantum electromagnetic transients program. IEEE Trans. Power Syst. 36 (4), 3813–3816.