IEEE Access
Multidisciplinary Rapid Review Open Access Journal

Received 1 August 2023, accepted 26 August 2023, date of publication 11 September 2023, date of current version 15 September 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3312997

RESEARCH ARTICLE

Power Network Optimization: A Quantum Approach

GIUSEPPE COLUCCI[1], STAN VAN DER LINDE[2], AND FRANK PHILLIPSON[2,3]
[1]Quantum Quants, 3035 AW Rotterdam, The Netherlands
[2]Applied Cryptography & Quantum Algorithms, TNO, 2595 DA The Hague, The Netherlands
[3]School of Business and Economics, Maastricht University, 6211 LM Maastricht, The Netherlands

Corresponding author: Stan van der Linde (stan.vanderlinde@tno.nl)

This work was supported by Stichting Quantum Delta NL.

ABSTRACT Optimization of electricity surplus is a crucial element for transmission power networks since it leads to reducing costs as well as increasing efficiency across the network as a whole. In this paper, we show how to optimize such network costs using a quantum annealing approach. First, we define the Quadratic Unconstrained Binary Optimization (QUBO) problem for network partitioning. To achieve this, we introduce a method to translate inequality constraints with real-valued coefficients into approximate penalty functions. Next, we test the implementation on purely quantum and quantum-classical hybrid architectures. We then solve the problem using the D-Wave hybrid Constrained Quadratic Model (CQM) solver, the D-Wave hybrid Binary Quadratic Model (BQM) solver, as well as classical solvers available on Azure Quantum Cloud. Finally, we find that the value of the objective function obtained with the quantum-classical hybrid solvers is always lower compared to the classical approaches across a range of different problem sizes. This demonstrates that the quantum-classical hybrid methods outperform the classical methods in terms of solution quality.

INDEX TERMS Optimization, power networks, quantum annealing, quantum classical hybrid solver, quantum computing, QUBO.

I. INTRODUCTION

In recent years, the energy sector has witnessed a global shift toward clean energy, with companies around the world accelerating ambitious renewable energy goals, particularly in wind and solar power. As a result, the energy market is gradually transitioning from centrally planned, financed, and operated electricity systems to a more decentralized and real-time model. This shift involves the sharing of planning, finance, and operation responsibilities among an increasing number of parties. While decentralization brings benefits such as improved utilization of renewable energy sources, reduced fossil fuel use, and increased eco-efficiency [1], it also increases the complexity of electricity systems. Optimizing costs and operations of the entire network becomes increasingly challenging due to the growing number of energy players and the complex nature of the underlying power network.

Several methods have been implemented to enhance the efficiency and resilience of power networks [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. Graph theory has been proposed as an efficient methodology to identify the optimal design of electricity networks [5], [6], [7]. Recent advances in both quantum hardware and algorithm development have made it possible to solve problems that can be mapped onto graphs on quantum computers. More generally, quantum computers can efficiently solve combinatorial optimization problems, including NP-hard problems, for which efficient classical algorithms are unknown [13], [14], [15], [16].

The main goal of this paper is to demonstrate the current performance of one of the most promising quantum optimization approaches: quantum annealing. For this, we apply quantum annealing to the power network optimization problem of the energy industry. To our knowledge, this approach has not yet been applied to a large-scale problem in the energy field. This study therefore also serves as proof-of-concept for the use of quantum annealing for the power network optimization problem.

Apart from the significance of addressing an industry use case with quantum annealing, we also contribute to the theoretical aspect by improving the current approach to treating inequality constraints and QUBO penalty functions.

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Abdur Razzaque.

Specifically, we introduce a method to easily translate an inequality constraint with real-valued coefficients into an approximate penalty function for the quantum annealing Hamiltonian. This is a major result, as it is relevant for many use cases where the coefficients of the inequality constraints are not integer values (e.g., inventory management to reflect the physical storage capacity of a warehouse or the cost of storing inventory; resource allocation to reflect availability and costs of resources; machine scheduling to reflect capacity and efficiency of the machines).

This article is structured as follows. In Section II, we give a summary of the quantum annealing paradigm and outline the mathematical framework underlying the optimization problem we aim to solve. Then, in Section III, we define the power network optimization problem, with and without electricity sharing among nodes of the network. We adopt the graph partitioning problem formulation presented in [7]. In this problem formulation, a graph partitioning approach is used to construct minimal cost microgrids (partitions) for sharing electricity surplus within a power grid. In Section IV, we express the problem in the QUBO formulation, which will be used in the actual implementation and run on the quantum, classical and hybrid solvers. In Section V, we present the results of our graph partitioning model and implementation with synthetic datasets on quantum and hybrid implementations. Next, we show results of hybrid and classical approaches to an industrially relevant problem, by optimizing the electricity surplus of the transmission power network of Germany. Finally, in Section VI, we summarize our findings and outline possible future developments.

II. QUBO FORMULATION AND MATHEMATICAL TOOLBOX

The Adiabatic Quantum Optimization (AQO) model is a quantum computing model used to find solutions to combinatorial optimization problems. AQO utilizes the adiabatic theorem in quantum mechanics [17], [18] to adiabatically move towards the ground state of an interaction Hamiltonian, onto which an optimization problem is mapped. The hardware architecture that approximately implements the adiabatic quantum computer paradigm is the quantum annealer [19]. Unlike a quantum gate architecture, where a computation is defined by applying separate gates (operators) to each qubit, a quantum annealer defines a Hamiltonian operator that acts simultaneously on all qubits in the ground state. This operator adiabatically changes towards a Hamiltonian on which the cost function of a specific optimization problem is mapped [20]. Theoretically, the resulting state after the adiabatic evolution will be the ground state of the final Hamiltonian, i.e., the global minimum of the optimization problem mapped onto the final Hamiltonian.

In order to solve combinatorial optimization problems with quantum annealing hardware, one must map the cost function of the optimization problem to a Hamiltonian. The most common way to do this, is to use the Quadratic Unconstrained Binary Optimization (QUBO) formulation [21]. The QUBO formulation is easily transformed to describe a Hamiltonian. Most software suites which control quantum annealing hardware can use the QUBO formulation directly as input [21]. The QUBO formulation has the following form:

$$\min_{\boldsymbol{x}\in\{0,1\}^n} \boldsymbol{x}^\intercal Q\boldsymbol{x}, \tag{1}$$

where Q is a real valued matrix and $\boldsymbol{x}$ of length n and its transpose $\boldsymbol{x}^\intercal$ are binary vectors.

Most problems are not naturally expressed in the QUBO formulation, which requires the recasting of the problem. Many problems have already been reformulated into the QUBO form [20]. For those problems that are not yet in such a list, techniques have been distilled to rewrite combinatorial optimization problems into the QUBO form [22]. The specific techniques used in this paper are: (A) including equality constraints, (B) including inequality constraints and (C) pairwise degree reduction.

A. INCLUDING EQUALITY CONSTRAINTS

The QUBO formulation does not allow for any constraints. Nonetheless, many real world problems contain (in)equality constraints. These constraints are included into the objective function by means of a penalty function [22]. A penalty function for a constraint should be zero when the constraint is met and positive otherwise.

Let us construct such a penalty function for a general linear equality constraint as shown in (2)

$$\boldsymbol{a}^\intercal\boldsymbol{x} = b, \tag{2}$$

where $\boldsymbol{a}$ is a real constant vector and b is a real constant. If we take the left-hand side, subtract b from it and square it we get the following penalty function $P(\boldsymbol{x})$:

$$P(\boldsymbol{x}) = \left(\boldsymbol{a}^\intercal\boldsymbol{x} - b\right)^2. \tag{3}$$

Note that $P(\boldsymbol{x})$ is non-negative and that it is zero if and only if the constraint from (2) is met. Therefore, the addition of this function to the objective function will favour solutions that comply with the constraints in (2) (when assuming minimization).

B. INCLUDING INEQUALITY CONSTRAINTS

Including inequality constraints into the objective function requires more work compared to equality constraints. A general linear inequality constraint has the following form:

$$\boldsymbol{a}^\intercal\boldsymbol{x} \leq b. \tag{4}$$

If $\boldsymbol{a}$ and b are integer valued, then this equation is equivalent to an equality constraint with the inclusion of slack variables [22]. Subsequently, the latter equality constraint could then be included into the objective function with the penalty function shown in the previous section.

However, in many real life problems (including the one proposed in this paper), $\boldsymbol{a}$ and b in (4) are real valued. In this case we cannot transform the inequality constraint to an equality constraint. Instead we directly construct an (approximate) penalty function from the inequality constraint. We start by finding a (possible negative) lower bound of the left hand side $c \in \mathbb{R}$:

$$c \leq \boldsymbol{a}^\mathsf{T}\boldsymbol{x} \leq b, \tag{5}$$

where $\mathbb{R}$ is the set of real numbers. In general, c can be easily computed. If $\boldsymbol{a}$ contains at least one negative element, then c is the sum of all negative elements of $\boldsymbol{a}$, otherwise it is 0.

One approach to create a penalty function for (5) is to approximate (5) using K auxiliary binary variables z_i. The resulting penalty function of such an approach is

$$P(\boldsymbol{x}, z) = \left(\frac{2^K - \frac{1}{2}}{b-c} \left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c\right) - \sum_{i=0}^{K-1} 2^i z_i \right)^2, \tag{6}$$

of which the derivation can be found in Appendix A. The goal of this penalty function is to favour values of $\boldsymbol{x} \in \{0,1\}^n$ that comply with (4), as opposed to values of $\boldsymbol{x}$ that violate the constraint.

1) PROPERTIES AND ANALYSIS OF THE APPROXIMATE PENALTY FUNCTION

To our knowledge, the penalty function shown in (6) is a new result. Therefore, we analyse some of its properties in the form of mathematical theorems and propositions. The proofs of the theorems and propositions can be found in Appendix A. We start by showing that the penalty function can distinguish between values of $\boldsymbol{x}$ that meet the constraint and that violate the constraint.

Theorem 1: Let $(\boldsymbol{a}, b) \in \mathbb{R}^{n+1}$ and define the set $F = \{\boldsymbol{x} \in \{0,1\}^n \mid \boldsymbol{a}^\mathsf{T}\boldsymbol{x} \leq b\}$. Let $c = \min F$ be a lower bound of $\boldsymbol{a}^\mathsf{T}\boldsymbol{x}$ and

$$P(\boldsymbol{x}, z) = \left(\frac{2^K - \frac{1}{2}}{b-c} \left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c\right) - \sum_{i=0}^{K-1} 2^i z_i \right)^2,$$

where $K \in \mathbb{N}$ and $\boldsymbol{z} \in \{0,1\}^K$. Then, for $\boldsymbol{x} \in \{0,1\}^n$ the following statements are equivalent:

1) $\boldsymbol{x} \in F$.
2) *There exists a $\boldsymbol{z} \in \{0,1\}^K$ such that $P(\boldsymbol{x}, \boldsymbol{z}) \leq 1/4$.*

For a fixed $\boldsymbol{x}$ we have that if $\boldsymbol{x}$ meets the constraint, then the minimum of the penalty function is between 0 and $1/4$. On the other hand, if $\boldsymbol{x}$ violates the constraint, then the minimum of the penalty function is strictly larger then $1/4$.

In the following proposition we shall show that this property is a direct consequence of the constant $(2^K - 1/2)/(b-c)$ in the definition of the penalty function.

Proposition 2: Let a, b, c and F be the same as defined in Theorem 1. Suppose $P(\boldsymbol{x}, z) = \left(\alpha_K(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c) - z\right)^2$, where $z \in \{0, 1, \ldots, 2^K - 1\}$. Then the following statements are equivalent:

1) $\boldsymbol{x} \in F$.
2) *There exists a z such that $P(\boldsymbol{x}, z) \leq 1/4$.*

if and only if $\alpha_K = (2^K - 1/2)/(b-c)$.

Another important property of this penalty function is the relative size of the penalty that is given by the penalty function. We will analyse this behaviour using the following proposition.

Proposition 3: Let $(\boldsymbol{a}, b) \in \mathbb{R}^{n+1}$ define the set $F = \{\boldsymbol{x} \in \{0,1\}^n \mid \boldsymbol{a}^\mathsf{T}\boldsymbol{x} \leq b\}$. Define $c := \min F$ (a lower bound of the constraint) and

$$P(\boldsymbol{x}, z) = \left(\alpha_K(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c) - \sum_{i=0}^{K-1} 2^i z_i \right)^2,$$

where $K \in \mathbb{N}$, $\boldsymbol{z} \in \{0,1\}^K$ and $\alpha_K = (2^K - 1/2)/(b-c)$. Then for $x \notin F$,

$$\min_{z \in \{0,1\}^K} P(\boldsymbol{x}, z) = \frac{1}{4} + \alpha_k \varepsilon + (\alpha_k \varepsilon)^2,$$

where $\varepsilon = \boldsymbol{a}^\mathsf{T}\boldsymbol{x} - b$.

$P(\boldsymbol{x}, z)$ with $\alpha_K = (2^K - 1/2)(/(b-c)$ scales quadratically in both α_K and the amount of violation. Because α_K scales exponentially in K, the sensitivity of the penalty function scales exponentially in K. The sensitivity of the penalty function can be increased by increasing K. However, one must be wary to set K to large values when implementing these penalties. Since α_K scales exponentially in K, the penalty can become numerically unstable for large values of K.

C. DEGREE REDUCTION

In the QUBO formulation only quadratic or lower order terms are allowed. However, often problem formulations contain higher order terms. These higher order terms can be reduced to quadratic terms by adding auxiliary binary variables. With pairwise degree reduction, we substitute a quadratic $x_i x_j$ term by a new variable z and we add the following penalty function to the objective

$$M(x_i, x_j, z) = x_i x_j - 2z(x_i + x_j) + 3z. \tag{7}$$

Note that $M(x_i, x_j, z)$ is non-negative and $M(x_i, x_j, z) = 0$ if and only if $x_i x_j = z$. So for example $x_1 x_2 x_3$ would become $x_1 z + \lambda M(x_2, x_3, z)$, where $\lambda > 0$ is a Lagrange multiplier.

III. GRAPH PARTITIONING PROBLEM FOR POWER GRIDS

Graph Partitioning (GP) is the determination of communities or clusters, within the nodes of a graph. GP has proven to be a useful method in network analysis [23]. In fact, GP approaches emerged to reduce the complexity of applications generally involving graph sizes arbitrarily large compared to the computational resources at hand, by dividing (*partitioning*) the graph into smaller subgraphs or sub-problems, and thus increase the computational performance [24].

Applications of GP include physical network design [25], VLSI design [26], telecommunication network design [27], load balancing of high performance computing (HPC) codes [28], distributed sparse matrix-vector multiplication [29], biological [30] and social networks [31], [32].

GP is an NP-hard problem (as a decision problem) [33], [34]. Algorithms to solve GP problems include exact algorithms, spectral partitioning, geometric partitioning, flow computations, etc. [35]. Exact algorithms are often used to solve small-size problems, whereas heuristic algorithms are needed for larger problems.

Recently, quantum computers have been used to model and solve the graph partitioning problem, see e.g., [36], [37], [38], and [39]. In this paper, we shall focus on solving the graph partitioning of a power network with quantum annealing, which uses quantum physics to find low-energy states of a problem and which can be mapped to the optimal or near-optimal solution of the optimization problem.

In the following we outline the mathematical formulation of the graph partitioning problem for a power grid, in which the performance metric is given by the *power grid cost*. We follow the formulation presented in [7], where minimal cost microgrids (partitions) are used to efficiently share electricity surplus. We first consider the case of no electricity sharing within partitions and then extend this formulation to allow for electricity sharing. Finally, we shall express the graph partitioning problem in the Hamiltonian/QUBO framework needed to implement and solve the problem on a quantum annealer.

A. MATHEMATICAL FORMULATION WITHOUT ELECTRICITY SHARING

In [7], the graph partition of power grids is modeled in the following manner. Suppose we have a graph $G = (V, E)$, where the vertices V represent geographical areas and the edges E represent transmission lines between two areas. The total electricity surplus in an area is then encoded in the weight of the vertices associated to the given area.

Let P be the number of partitions, or clusters, and $N = |V|$ be the number of vertices in the graph. Then the binary variable $v_{np} \in \mathbb{B}$ is 1 when vertex n is in partition p and 0 otherwise. Let k denote the threshold of self-sufficiency of the partition, as defined in [7]. Suppose $\alpha, \beta \in \mathbb{R}^+$ are coefficients of the cost for transmission lines within a single cluster and between clusters respectively. Then the optimization problem can be written as:

$$\min_{\boldsymbol{v}\in\mathbb{B}^{PN}} \sum_{p=1}^{P} \alpha \left(\sum_{n=1}^{N} v_{np} \right)^2 + \beta \left(|E| - \sum_{\{n,m\}\in E} v_{np} v_{mp} \right) \tag{8}$$

$$\text{s.t.} \sum_{p=1}^{P} v_{np} = 1, \quad n = 1, \ldots, N \tag{9}$$

$$\sum_{n=1}^{N} v_{np} w_n - k \sum_{n=1}^{N} v_{np} \leq 0, \quad p = 1, \ldots, P. \tag{10}$$

Equation (8) is the objective function of the optimization problem, which represent the cost of constructing a particular partition. The constraints described in (9) are called one-hot constraints. These constraints enforce each vertex to be in one and one only cluster. Lastly, the constraints in (10) are called balancing constraints. These ensure that the average surplus of a cluster is below the threshold k, so that the partition fulfills the *self-sufficiency* property.

B. MATHEMATICAL FORMULATION WITH ELECTRICITY SHARING

The mathematical formulation of the problem with electricity sharing is very similar to the problem outlined in the previous section. Both the objective function and one-hot constraint remain the same. The key difference is the balancing constraint, which now allows for sharing of energy surplus. This is done by adding a flow function between each partition $F_{pq}(\boldsymbol{v})$, such that $F_{pq}(\boldsymbol{v}) = -F_{qp}(\boldsymbol{v})$. The resulting mathematical formulation is given by

$$\min_{\boldsymbol{v}\in\mathbb{B}^{PN}} \sum_{p=1}^{P} \alpha \left(\sum_{n=1}^{N} v_{np} \right)^2 + \beta \left(|E| - \sum_{\{n,m\}\in E} v_{np} v_{mp} \right) \tag{11}$$

$$\text{s.t.} \sum_{p=1}^{P} v_{np} = 1, \quad n = 1, \ldots, N \tag{12}$$

$$\sum_{n=1}^{N} v_{np} w_n + \sum_{q=1}^{P} F_{pq}(\boldsymbol{v}) - k \sum_{n=1}^{N} v_{np} \leq 0, \quad p = 1, \ldots, P. \tag{13}$$

IV. QUBO FORMULATION OF GRAPH PARTITIONING

The optimization problem described in Section III contains constraints, which are not allowed in the QUBO formulation. To overcome this, constraints will be included into the objective function with the use of penalty terms to produce the following form:

$$\min H(\boldsymbol{x}) + \sum_{i=1}^{\#\text{constraints}} \lambda_i P_i(\boldsymbol{x}). \tag{14}$$

We will first show the QUBO formulation of the GP problem *without* electricity sharing, followed by the QUBO formulation of the GP problem *with* electricity sharing.

A. QUBO FORMULATION OF GRAPH PARTITIONING WITHOUT ELECTRICITY SHARING

First, we derive the Hamiltonian for the objective function. The objective function shown in (8) consists of binary variables and is quadratic in nature. Hence, this objective function already complies with QUBO formalism and its Hamiltonian is given by

$$H(\boldsymbol{v}) = \sum_{p=1}^{P} \alpha \left(\sum_{n=1}^{N} v_{np} \right)^2 + \beta \left(|E| - \sum_{(n,m)\in E} v_{np} v_{mp} \right). \tag{15}$$

Secondly, we will derive a penalty function for the one-hot constraints shown in (9). Since these are linear equality constraints, we can use the technique described in

Subsection II-A to produce

$$P_{\text{oh}}(\boldsymbol{v}) = \sum_{n=1}^{N}\left(\sum_{p=1}^{P} v_{np} - 1\right)^2. \tag{16}$$

Lastly, we will derive a penalty function for the balancing constraint shown in (10). To do so, we will use the technique described in Subsection II-B for which we will need a lower bound on the constraint. A lower bound can be found by taking the summation over all negative $w_n - k$ values, i.e.,

$$c = \frac{1}{2}\sum_{n=1}^{N}(w_n - k - |w_n - k|). \tag{17}$$

This lower bound c and the upper bound of 0 produces the following penalty function:

$$P_{\text{bc}}(\boldsymbol{v}, \boldsymbol{x}; K) = \sum_{p=1}^{P}\left(\frac{2^K - \frac{1}{2}}{-c}\left(-c + \sum_{n=1}^{N} v_{np}(w_n - k)\right) - \sum_{a=0}^{K-1} 2^a x_{ap}\right)^2, \tag{18}$$

where x_{ap} are the auxiliary (slack) variables.

The final QUBO Hamiltonian representation of the whole problem is then

$$\min_{(\boldsymbol{v},\boldsymbol{x})\in\mathbb{B}^{PN}\times\mathbb{B}^{PK}} H(\boldsymbol{v}) + \lambda_{\text{oh}}P_{\text{oh}}(\boldsymbol{v}) + \lambda_{\text{bc}}P_{\text{bc}}(\boldsymbol{v}, \boldsymbol{x}; K), \tag{19}$$

where λ_{oh} and λ_{bc} are non-negative constants (Lagrange multipliers) and K is a positive integer.

From (19), it is evident that the number of variables for the problem with N nodes, P partitions and the hyperparameter K is $P(N + K)$.

B. QUBO FORMULATION OF GRAPH PARTITIONING WITH ELECTRICITY SHARING

The mathematical formulation of the GP problem with sharing has the same objective function and one-hot constraints. Therefore, we will use the same objective Hamiltonian and penalty function as in the previous section. What remains is constructing a penalty for the new balancing constraint. However, as will become clear, this altered balancing Hamiltonian will need new auxiliary variables for which the behaviour will be enforced by an additional constraint.

Let us start by defining a flow from partition p to partition q. We will construct a flow which can share everything from p to q, everything from q to p or nothing at all. This flow function has the following form:

$$F_{pq}(\boldsymbol{v}, \boldsymbol{f}) = \sum_{\{n,m\}\in E} w_{nm}(f_{nmp} - f_{nmq})(v_{np}v_{mq} + v_{mp}v_{nq}), \tag{20}$$

where f_{nmp} are new binary variables determining the flow between partition p and q. Note that $v_{np}v_{mq} + v_{mp}v_{nq}$ in (20) is 1 if $\{n, m\}$ is an edge between partition p and q and is zero otherwise.

Note that F is indeed a flow, since

$$\begin{aligned} F_{qp}(\boldsymbol{v}, \boldsymbol{f}) &= \sum_{\{n,m\}\in E} w_{nm}(f_{nmq} - f_{nmp})(v_{nq}v_{mp} + v_{mq}v_{np}) \\ &= -\sum_{\{n,m\}\in E} w_{nm}(f_{nmp} - f_{nmq})(v_{np}v_{mq} + v_{mp}v_{nq}) \\ &= -F_{pq}(\boldsymbol{v}, \boldsymbol{f}). \end{aligned} \tag{21}$$

Therefore, we also have the property $F_{pp}(\boldsymbol{v}, \boldsymbol{f}) = 0$.

If we include F from (20) into the balancing constraint, the constraint becomes cubic in nature. Hence, we will have to reduce the degree of this constraint by using the technique described in Subsection II-C. Let $\mathcal{P} = \{(p, q) \in \{1, \ldots, P\}^2 | p \neq q\}$, then we reduce the degree by adding auxiliary binary variables with the following properties

$$a_{nmpq} = v_{np}v_{mq}, \quad \forall\{n, m\} \in E, \forall(p, q) \in \mathcal{P}, \tag{22}$$

$$y_{nmpq} = f_{nmp}a_{nmpq}, \quad \forall\{n, m\} \in E, \forall(p, q) \in \mathcal{P}, \tag{23}$$

$$z_{nmpq} = f_{nmp}a_{nmqp}, \quad \forall\{n, m\} \in E, \forall(p, q) \in \mathcal{P}. \tag{24}$$

Hence, $F_{qp}(\boldsymbol{v}, \boldsymbol{f})$ becomes

$$\begin{aligned} F_{pq}(\boldsymbol{v}, \boldsymbol{f}) &= \sum_{\{n,m\}\in E} w_{nm}(f_{nmp} - f_{nmq})(v_{np}v_{mq} + v_{mp}v_{nq}) \\ &= \sum_{\{n,m\}\in E} w_{nm}(f_{nmp} - f_{nmq})(a_{nmpq} + a_{nmqp}) \\ &= \sum_{\{n,m\}\in E} w_{nm}(y_{nmpq} + z_{nmpq} - y_{nmqp} - z_{nmqp}) \\ &= G_{pq}(\boldsymbol{y}, \boldsymbol{z}) \end{aligned} \tag{25}$$

The penalty function of the balancing constraint with electricity sharing is then given by

$$\begin{aligned} P_{\text{bc}}(\boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \sum_{p=1}^{P}\Bigg(\frac{2^K - \frac{1}{2}}{c}\left(c + \sum_{n=1}^{N} v_{np}(w_n - k)\right) \\ + \sum_{\substack{q=1\\q\neq p}}^{P} G_{qp}(\boldsymbol{y}, \boldsymbol{z}) - \sum_{a=0}^{K-1} 2^a x_{ap}\Bigg)^2, \end{aligned} \tag{26}$$

where $c = \frac{1}{2}\sum_{n=1}^{N}(|w_n - k| - w_n + k) + \sum_{\{n,m\}\in E} w_{nm}$.

Lastly, we apply the penalty function for degree reduction as described in Subsection II-C to imply the required behaviour of the auxiliary variables:

$$\begin{aligned} P_{aux}(\boldsymbol{v}, \boldsymbol{f}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{a}) = \sum_{\{n,m\}\in E}\sum_{p,q\in\mathcal{P}} & M(v_{np}, v_{mq}, a_{nmpq}) \\ &+ M(f_{nmp}, a_{nmpq}, y_{nmpq}) \\ &+ M(f_{nmp}, a_{nmqp}, z_{nmpq}). \end{aligned} \tag{27}$$

With the new penalty terms, the QUBO problem for graph partitioning with electricity sharing is given by:

$$\begin{aligned} \min_{\boldsymbol{v},\boldsymbol{x},\boldsymbol{f},\boldsymbol{y},\boldsymbol{z},\boldsymbol{a}} & H(\boldsymbol{v}) + \lambda_{\text{oh}}P_{\text{oh}}(\boldsymbol{v}) + \lambda_{\text{bc}}P_{\text{bc}}(\boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}; K) \\ &+ \lambda_{\text{aux}}P_{aux}(\boldsymbol{v}, \boldsymbol{f}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{a}). \end{aligned} \tag{28}$$

The total number of variables of the QUBO formulation is less straightforward compared to the model without electricity sharing. We shall analyse this by counting the size of all the binary vectors $\boldsymbol{v}, \boldsymbol{x}, \boldsymbol{f}, \boldsymbol{y}, \boldsymbol{z}$ and $\boldsymbol{a}$. The size of $\boldsymbol{v}$ and $\boldsymbol{x}$ is the same as for the problem without sharing, PN and PK respectively. For each edge there is a f variable for each partition, hence the size of $\boldsymbol{f}$ is $P|E|$. Lastly, the size of $\boldsymbol{a}, \boldsymbol{y}$ and $\boldsymbol{z}$ is given by $\mathcal{P}$, which is $|E|(P^2 - 1)$. Therefore, the total number of variables for the QUBO with electricity sharing is $P(N + K + |E|) + 3|E|(P^2 - 1)$. Hence, when the network graph is not sparse, the number of variables is much larger for the model with sharing.

C. QUBO HYPERPARAMETERS

In the construction of the QUBO in the previous sections, the hyperparameters K and λ_i were introduced. These hyperparameters should be determined before solving the problem and comparing results. The K parameter has been discussed before in Section (II-B) and can be used to alter the sensitivity of the balancing constraint.

The other hyperparameters, λ_i, were determined with a grid-search. In this grid-search each grid point was evaluated using short runs of simulated annealing. For each grid point, two measures of quality were calculated: (i) the original objective value of (8) and (ii) the number of constraints violated. The best grid point had the lowest objective value for which no constraints were violated. To minimize computation time, we performed the grid-search in two stages: (i) a logarithmic stage and (ii) a linear stage. In the logarithmic stage a logarithmic grid was searched quickly to determine the order of magnitude of each λ_i. Next, a linear grid was searched to more precisely determine the best value of λ_i. Using this method we quickly found suitable hyperparameters λ_i.

D. IMPLEMENTATION ON D-WAVE ARCHITECTURE

The initial formulation of the power network optimization problem, as presented in this paper, was a constrained quadratic model (CQM). CQMs cannot be solved using a quantum annealer. We therefore reformulated the problem into a QUBO formulation, also known as a binary quadratic model (BQM). Although CQMs can currently not be solved with pure quantum approaches, they can be solved with hybrid approaches. Hybrid approaches solve the problem by using a combination of classical and quantum computation. To test the performance of current quantum annealing paradigms, the problem formulations were submitted to the D-Wave quantum computing system which offers both hybrid and pure quantum annealing solvers.

The D-Wave quantum computing system[1] is a hardware heuristic that minimizes Ising objective functions using a physically realized version of quantum annealing. Due to the mapping between QUBO and Ising variables [20], every QUBO problem can be translated to an Ising model which can then be embedded onto a D-Wave system.

[1] https://www.dwavesys.com/

We solved the QUBO with a pure quantum annealing approach, using the D-Wave Advantage system. The QUBO and CQM formulation were solved using the BQM and CQM solver from D-Wave's Leap hybrid solver service respectively. Note that the hybrid solvers use the D-Wave Advantage system as a back-end. An overview of the parameters used for all solvers can be found in Appendix B.

V. RESULTS AND DISCUSSION

In the preceding sections we have presented a method to construct QUBO penalty terms for real-valued linear inequality constraints. A noteworthy feature of this penalty term, is that we can freely choose the number of auxiliary variables K. Moreover, we show that the sensitivity scales exponentially in the number of auxiliary variables K. Due to the rapid scaling of the sensitivity, only a small amount of auxiliary variables is needed. This is a desirable property for quantum annealing, as the number of qubits is limited. Furthermore, we applied this penalty term construction method to obtain QUBOs for a power network optimisation problem with and without electricity sharing. Just like the penalty term construction method, the QUBOs that we presented in the preceding section are novel and can be used for further research. In what follows, we present the numerical results of the different solvers for the power network optimization problem without electricity sharing. This serves as a proof-of-concept of our QUBO, showing that these theoretical results can aid in the solution for relevant real-life problems such as power network optimization.

We tested the performance of the pure quantum and hybrid solvers of D-Wave and compared them with the performance of the classical solvers of the Microsoft Quantum-inspired optimization (QIO) provider[2] present on the Azure cloud. In (A) we explain how we tested the equivalence between the implementations of the CQM and BQM, which justifies the comparison between the hybrid and classical solvers in (C). In (B) we show that the QUBO formulation (or BQM) can be solved with a pure quantum annealer. For this purpose, we use a small synthetic data set with a known optimum.

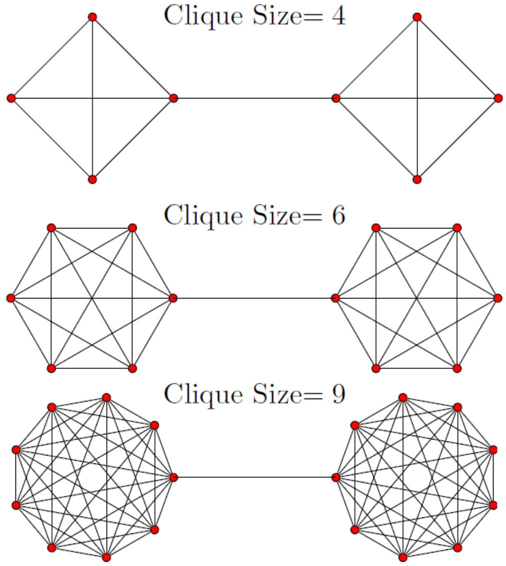
A. TESTING CQM AND BQM IMPLEMENTATIONS

To gain confidence in both our implementation and derivation of the CQM and BQM models, they were first tested on smaller problems. For this purpose, we designed two tests.

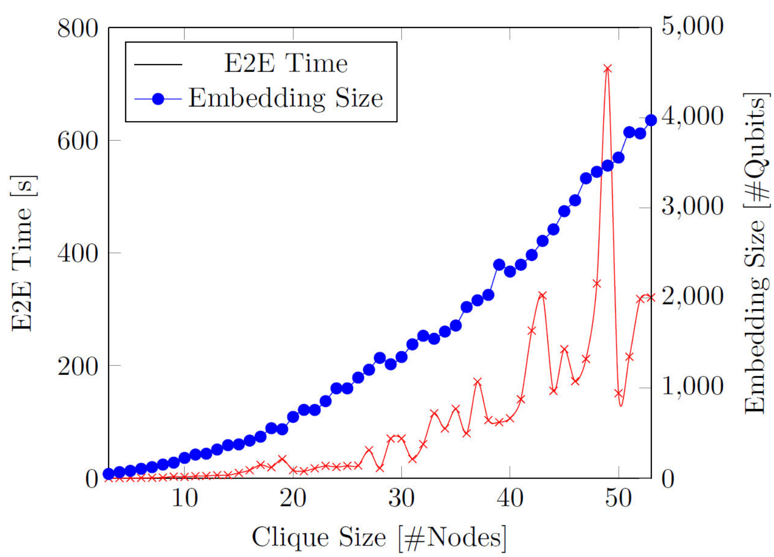
Firstly, the implementation was tested by constructing a square graph with known weights. Due to the problem size, it was possible to manually write out both the CQM and BQM models. These models were then compared to the CQM and BQM models implemented in our system, and they were found to be equal. Thus, we gained confidence in the implementation of our model.

Secondly, the equality of the optimal points of the CQM and BQM in practice was tested. In theory, the optimal

[2] https://learn.microsoft.com/en-us/azure/quantum/provider-microsoft-qio



(a)



(b)

FIGURE 1. (a) Shape of the graphs used for the QPU results. (b) On the left axis the E2E time versus clique size. The right axis shows the embedding size (in qubits) versus clique size.

point of the CQM and BQM should yield the same value when plugged into the original objective function shown in (8). We tested this using brute force algorithms. The original objective function was tested for 100 randomly generated graphs with 2 to 8 nodes. All 700 tested problems yielded a solution with an equal objective value for both BQM and CQM. Consequently, we gained confidence in the equivalence of the optimal points for the BQM and CQM.

B. QUANTUM ANNEALING ON A SYNTHETIC DATASET

The performance of the BQM using quantum annealing was tested on the D-Wave Advantage quantum annealer. To assess performance, a synthetic dataset was created. The use of a synthetic dataset is necessary in quantum computing benchmarking to ensure the validity of the algorithms used [40], [41], [42], [43], [44].

The dataset was created by connecting two cliques with a single edge (see Fig. 1a). This configuration generates a graph with two clearly defined partitions (the two cliques). Next, a surplus was added to nodes in the graph. The surplus possesses two key properties: (i) the average surplus in each clique is 0.45 and (ii) more than 1/3 of all possible partitions violate the balancing constraint with a threshold $k = 0.5$. Property (i) ensures that the balancing constraint is not violated when the two cliques are selected as partitions, while property (ii) ensures that combinations of variables exist where the balancing constraint can be violated.

The chosen dataset has two key advantages: it has a variable size, and the optimum solution is known beforehand. Hence, we can easily verify if the solver correctly finds the global optimum for different problem sizes. In [7], two different sets of cost parameters are considered, $(\alpha, \beta) = (1, 1)$ and $(\alpha, \beta) = (1, 10)$. Since the focus of the present analysis lies in proving the concept of the model on the quantum annealer, the specific values of these parameters are irrelevant. Thus, we choose α and β to be respectively 1 and 10. The threshold k was set to 0.5.

The problem was directly tested on the D-Wave Advantage system. The annealing time was set to 1000 μs, and the number of reads was set to 500. All other settings were left at their defaults. The problem was tested for clique sizes ranging from 3 to 53, beyond which no minor embedding could be found. The quantum annealer found the optimal solution in *all* instances at least once. The results are shown in Fig. 1b.

Fig. 1b shows the end-to-end time of solving the problem on the left axis. Since the actual time spent on the quantum annealer was the same for all these problems, almost all of this time was spent on finding an appropriate minor embedding. The embedding size is shown on the right axis as a function of clique size. Interestingly, no minor embedding could be found for a clique size larger than 53. This outcome was unexpected, considering that the minor embedding was achieved for a clique size of 3974, while the D-Wave Advantage system has 5640 qubits. This limitation in the quantum annealing implementation demonstrates the challenge of estimating the scalability of the problem size with the number of qubits in the quantum annealing architecture.

C. GERMAN TRANSMISSION POWER NETWORK DATA SET

Next we performed a benchmarking analysis on a large size energy network. For this analysis, we considered the transmission power data obtained with the open-source reference model of European transmission networks, SciGRID [45], which is based on the raw transmission data available in openstreetmap.org.[3]

[3]OpenStreetMap: https://www.openstreetmap.org/

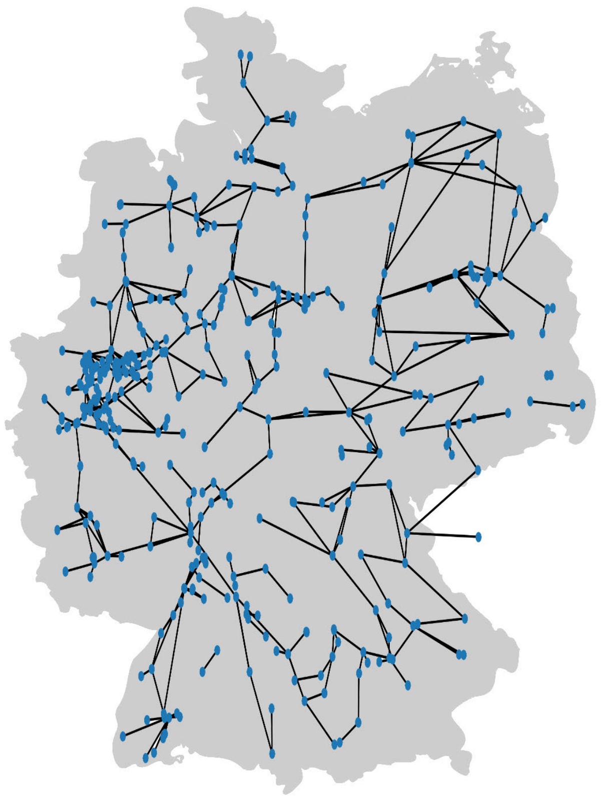


FIGURE 2. Graph derived from the SciGRID German transmission power network.

In particular, we performed the analysis on the SciGRID model output tables with vertices and links obtained from running the model on Germany raw data. The SciGRID output tables are: the table *vertices*, which contains the geographical center positions of German electrical substations, together with information on voltage level, frequency, name and operator; the table *links*, which contains the connections between two substations, with information on properties of the transmission line, e.g., voltage, number of cables in the circuit, resistance, and maximum current.

Given their structure, the SciGRID output tables can be used to build the graph of the German transmission power network (Fig. 2) and define the mapping to the problem in (8-10) and (11-13), where the table *vertices* is used to define the nodes and the table *links* defines the edges of the transmission network.

Additionally, for the complete definition of the optimization problem, we need to choose the values of the cost parameters α and β, and the electricity surplus at each node. These parameters are not defined or available in the SciGRID data set. In [7] two different sets of cost parameters are considered, $(\alpha, \beta) = (1, 1)$ and $(\alpha, \beta) = (1, 10)$. Since the focus of the present analysis is a benchmark of the quantum optimization approaches, the specific value of these parameters is not relevant. Thus, for the sake of the analysis, we choose α and β to be respectively 1 and 10.

TABLE 1. Hyperparameters for the QUBO of the German network for different partition sizes P, where $\alpha = 1$, $\beta = 10$, $k = 0.5$ and $K = 10$.

P	2	3	4	5	6	7	8	9
λ_{oh}	500	300	270	250	240	230	230	290
λ_{bc}	0.01	0.08	0.2	0.2	0.4	0.9	0.9	1.0

TABLE 2. Objective function value for the German power network for diferent number of partitions P. Each partition was solved with D-Waves Constrained Quadtaric Model Solver (CQM) and Binary Quadratic Model Solver (BQM), as well as Azures Parallel Tempering (PT), Simulated Annealing (SA), Substochastic Monte Carlo (MC) and Tabu Search (TS). Each solver has a time limit of 10 seconds.

P	CQM	BQM	PT	SA	MC	TS
2	101,362	102,232	103,022	102,842	103,845	103,484
3	74,824	75,350	78,035	78,142	77,376	77,180
4	64,272	67,196	69,082	69,135	67,999	70,247
5	60,130	63,762	**65,088**	**65,036**	**66,218**	**67,133**
6	**59,190**	**63,110**	80,095	78,808	83,875	72,645
7	60,048	64,240	136,392	139,403	134,969	79,644
8	62,094	66,386	150,925	154,054	199,356	85,948
9	64,882	69,180	174,162	178,692	325,600	115,515

Finally, the electricity surplus at each node (i.e., the weight of the node) is drawn from a uniform $[0, 1)$ distribution. In fact, an estimation of a realistic electricity surplus is out of scope for this paper and does not add value to the benchmark, as it would only influence the choice of the hyperparameters and the value of the objective function. In our test data, the average surplus across all nodes was approximately 0.49. In order to make the problem sufficiently difficult, the threshold was set to $k = 0.5$.

Table 1 shows the hyperparameters obtained with the grid-search algorithm outlined in Subsection IV-C for the case of the German transmission power network. $K = 10$ was chosen, because it was found to be sufficiently large to detect small errors in the balancing constraint, whilst being small enough to be safe in terms of overflow values. We defined eight different graph partitioning problems, corresponding to different number of partitions (2 to 9) and calculated the hyperparameters for each of these problems. Note that the higher the number of partitions, the higher the number of variables involved becomes, which also drives up the computational time.

Table 2 and Fig. 3 shows the comparison of the minimum of the objective function found by different solvers used to solve the graph partitioning problem on the German transmission power network.

For this comparison we first ran the problem on the CQM and BQM hybrid solvers. Then, we implemented the same binary quadratic problem in the Microsoft QIO framework and solved it by means of Parallel Tempering (PT), Simulated Annealing (SA), Substochastic Monte Carlo (MC) and Tabu Search (TS). Parallel Tempering and Simulated Annealing map the problem onto a thermodynamic system and search for the optimum by exchanging configurations at different temperatures. Substochastic Monte Carlo is a diffusion Monte Carlo algorithm inspired by adiabatic quantum computation. Tabu Search is a metaheuristic optimization

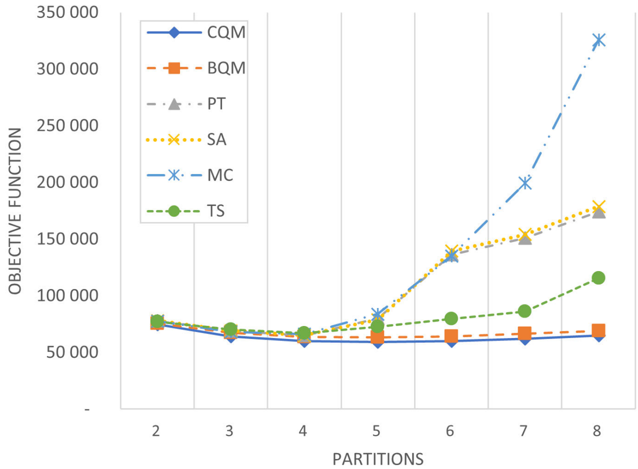


FIGURE 3. Change in objective function value for different solvers.

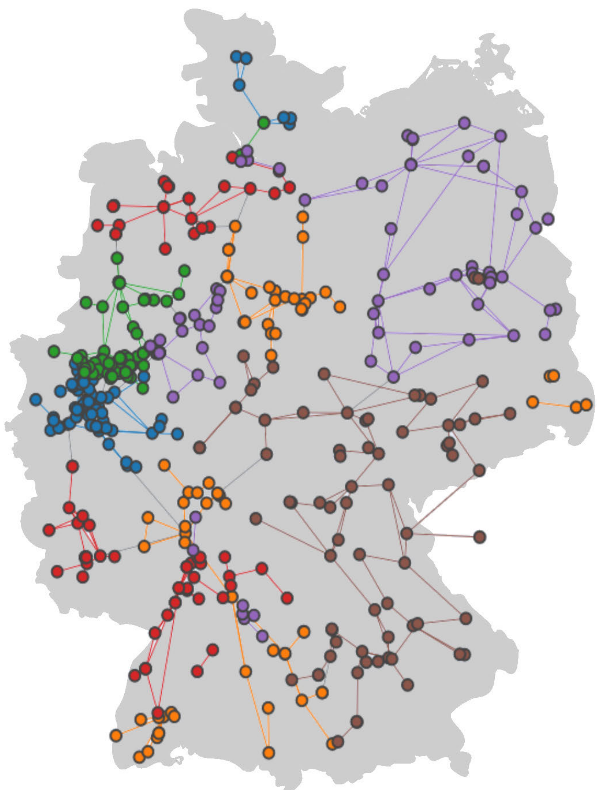


FIGURE 4. Best solution found by the Hybrid CQM Sampler (Number of partitions [P] is 6).

approach based on a local search method, which looks at neighboring configurations to move across the solution space. For the sake of a fair comparison, all approaches, hybrid and quantum-inspired, are run with a time limit of 10 seconds all other settings were left at their defaults. Table 2 shows the objective value of the optimal solution, thus it provides an estimation of the quality of the solution. The column P indicates the number of partitions in which the network is clustered.

Clearly the BQM and CQM implementations outperform the quantum-inspired methods in terms of quality. For all numbers of partitions, the CQM provides the lowest objective value. Additionally, all quantum-inspired methods show a deterioration in the quality of the solution for large number of partitions. This is clearly visible in the behavior of the objective function value for different partition sizes shown in Fig. 3. In fact, the value of the objective function at the minimum for the classical solvers shows a large variability across different partition sizes, compared to the hybrid solvers (CQM and BQM). For PT, SA, MC and TS the objective function explodes for partition sizes larger than 5. This is probably due to the fact that hybrid solvers can handle a larger number of variables and constraints.

Fig. 4 shows an example of solution obtained with the Hybrid CQM sampler for $P = 6$ partitions. Note that some clusters include points that are far from each other. This is the result of the simplification that the edge cost, β in (8), is a constant. A possible improvement of this assumption would be to define β as a function of the geographical distance between nodes.

VI. CONCLUSION

In this paper, we demonstrated how to optimize the costs of constructing an energy network with microgrids (partitions) for sharing energy surplus using quantum annealing. While previous studies have focused on utilizing quantum annealing to solve various optimization problems, this paper stands out by addressing a crucial use case in the energy sector: power network optimization. The proposed method for encoding the problem into a QUBO formulation is novel and provides a way to handle complex optimization problems that involve linear and nonlinear constraints with real-valued coefficients.

Following the approach presented in [7], we first mapped the optimization problem to a graph partitioning problem, where the objective is to determine the optimal set of communities of power substations. The problem was mathematically defined as a QUBO problem, both with and without electricity sharing among the network nodes. Additionally, we introduced a grid-search approach to determine the optimal hyperparameters for the optimization problem. The model and implementation for the case of no-electricity sharing were tested on both a real quantum annealer and hybrid solvers. The tests were conducted on small-sized random graphs, and the results were verified using an exhaustive search algorithm (exact solution).

For the numerical results, we considered a real-sized problem using data from the German transmission power network. The optimization problem was implemented for the case of no-electricity sharing and executed on the D-Wave hybrid CQM and BQM solvers. The quantum solver results were compared to those obtained from classical approaches available on Azure, specifically Parallel Tempering, Simulated Annealing, Substochastic Monte Carlo, and Tabu Search present in the Microsoft QIO framework. The numerical results revealed that, given the same computation time, the quantum approaches (CQM and BQM) outperformed the classical algorithms in terms of solution quality. The objective function value of the *quantum* solutions was consistently lower than that of the classical approaches across a range of problem sizes. Furthermore, the solution quality of the classical approaches deteriorated much faster than that of the quantum approaches as the problem complexity (number of variables) increased. This effect suggests that hybrid-quantum approaches might be more suitable for representing and solving large-scale optimization problems.

A. FUTURE CHALLENGES

The objective of this paper was to present the current potential of quantum optimization in the energy industry and benchmark the results against existing classical approaches. Further refinements, improvements and extensions of the problem addressed and of the model used are possible.

First, further investigation is required to implement the QUBO that was formulated for the optimization problem with electricity sharing and verify that the computational advantage demonstrated in this paper holds for the electricity sharing problem as well. Additionally, a comparison with other classical solvers currently used for real business applications (e.g., Gurobi or COIN-OR Couenne) would provide further confidence for the proposed methodology.

Second, from a more technical perspective, the optimization of hyperparameters can be improved. Alternatives to the grid search algorithms [46], such as random search, Bayesian Optimization or metaheuristics algorithms (e.g. genetic algorithms) can be tested to improve speed or quality of the solution.

Finally, it should be noted that other quantum optimization algorithms are available which do not leverage quantum annealing but gate-based quantum computing. Examples of these algorithms are VQE [47] and QAOA [48]. However, the performance of these algorithms is limited by the current state of quantum hardware [49] and only small size problems can be solved with these methods. Nevertheless, due to the continuous development and breakthroughs in quantum hardware we believe it is important to regularly monitor the maximum size of the problem that can be solved with gate-based optimization quantum algorithms.

The objective of this paper was to present the current potential of quantum optimization in the energy industry and benchmark the results against existing classical approaches. Given the current geopolitical situation and the associated significance of energy sources, we hope to inspire researchers and companies to validate our findings and test this approach on other use cases in the energy sector.

APPENDIX A
APPROXIMATE PENALTY FUNCTION FOR INEQUALITY CONSTRAINT

The appendix focuses on the approximate penalty function introduced in Subsection II-B. The first section will show how this penalty function was derived. The second section will provide the proofs of the theorems and propositions shown in provided Subsection II-B.

A. DERIVATION ON THE APPROXIMATE PENALTY FUNCTION

In this section we will focus on the derivation of the penalty function introduced in Subsection II-B. The goal is to construct a penalty function that favours values of $\boldsymbol{x}$ that comply with the constraint

$$\boldsymbol{a}^\mathsf{T}\boldsymbol{x} \le b \tag{29}$$

over values of $\boldsymbol{x}$ that violate the constraint. Let c be the tight lower bound of $\boldsymbol{a}^\mathsf{T}\boldsymbol{x}$. Since $\boldsymbol{x}$ is a binary vector and $\boldsymbol{a}$ and b are constants, such a c always exists. Furthermore, c can be easily found. If all values in $\boldsymbol{a}$ are non-negative, then the minimum of $\boldsymbol{a}^\mathsf{T}\boldsymbol{x}$ is when $\boldsymbol{x} = \boldsymbol{0}$ and therefore $c = 0$. In the case that $\boldsymbol{a}$ contains negative values, then $\boldsymbol{a}^\mathsf{T}\boldsymbol{x}$ is minimized when $\boldsymbol{x}$ is one for all elements corresponding with negative values in $\boldsymbol{a}$ and zero otherwise. In this case we have that c is the sum of all negative values. Using this lower bound c we can write (29) as

$$c \le \boldsymbol{a}^\mathsf{T}\boldsymbol{x} \le b. \tag{30}$$

Next, we subtract c from all sides to produce

$$0 \le \boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c \le b - c. \tag{31}$$

If $\boldsymbol{a}$, b and c are integer valued, one could replace the inequality constraint with an equality constraint by adding slack variables, which results in:

$$\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c = \sum_{i=0}^{I-2} 2^i z_i + z_{I-1}(2^I - 1 - b + c), \tag{32}$$

which subsequently produces the following penalty function:

$$P_{\text{int}}(\boldsymbol{x}, z) = \left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c - \sum_{i=0}^{I-2} 2^i z_i - z_{I-1}(2^I - 1 - b + c)\right)^2. \tag{33}$$

However, when $\boldsymbol{a}$, b or c are arbitrary non integer values, there can be $\boldsymbol{x}$ that comply with the constraint for which (33) does not hold for any z. For example, any $\boldsymbol{a}^T\boldsymbol{x} = n - 1/2$, where $n < b$ is an integer, we have that the smallest value of P_{int} is $1/4$. Furthermore, when $\boldsymbol{a}^T\boldsymbol{x} = b + 1/2$, which clearly violates the constraint, then there exists a $\boldsymbol{z}$, such that $P_{\text{int}} = 1/4$. Hence, P_{int} cannot differentiate well between values that comply with the constraint and values that violate the constraint. Therefore, if $\boldsymbol{a}$, b or c are arbitrary real valued constants, then (33) is not a good approximation of the constraint in (29).

We can overcome the problem by multiplying both sides by $(2^K - 1)/(b - c)$, where K is an integer valued constant, to produce

$$0 \le \frac{2^K - \frac{1}{2}}{b - c}\left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c\right) \le 2^K - \frac{1}{2}.$$

The equation above can then be *approximated* with the equality

$$\frac{2^K - \frac{1}{2}}{b-c}\left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c\right) = \sum_{i=0}^{K-1} 2^i z_i.$$

Using the technique described in Subsection II-A, we can transform this equality constraint to the final penalty function:

$$P(\boldsymbol{x}, z) = \left(\frac{2^K - \frac{1}{2}}{b-c}\left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c\right) - \sum_{i=0}^{K-1} 2^i z_i\right)^2. \tag{34}$$

In the next section we will prove that the penalty function in (34) does indeed favour values of $\boldsymbol{x}$ that comply with the constraint over values that violate it.

B. PROOF OF THEOREMS AND PROPOSITIONS

First we show the proof of Theorem 1, which proofs the intended behaviour of the penalty function.

Proof of Theorem 1: Since $\sum_{i=0}^{K-1} 2^i z_i$ is a bijection from $\{0, 1\}^K$ to $\{0, 1, \ldots, 2^K - 1\}$, we shall replace $\sum_{i=0}^{K-1} 2^i z_i$ by $\hat{z} \in \{0, 1, \ldots, 2^K - 1\}$ in this proof. First we will proof (1) $\Rightarrow$ (2). Suppose that $\boldsymbol{x} \in F$, then $\boldsymbol{a}^\mathsf{T}\boldsymbol{x} \leq b$ and we have

$$\frac{2^K - \frac{1}{2}}{b-c}\left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c\right) \leq \frac{2^K - \frac{1}{2}}{b-c}(b-c) = 2^K - \frac{1}{2}.$$

Because c is a lower bound of $\boldsymbol{a}^\mathsf{T}\boldsymbol{x}$ we know

$$\frac{2^K - \frac{1}{2}}{b-c}\left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c\right) \geq \frac{2^K - \frac{1}{2}}{b-c}(c-c) = 0.$$

Define

$$\hat{x} := \frac{2^K - \frac{1}{2}}{b-c}\left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c\right),$$

then $\hat{x} \in [0, 2^K - \frac{1}{2}]$ and we see

$$\min_{z \in \{0,1\}^K} P(\boldsymbol{x}, z) = \min_{\hat{z} \in \{0,1,\ldots,2^K-1\}} (\hat{x} - \hat{z})^2 \leq \frac{1}{4}.$$

Hence, there is a $\boldsymbol{z} \in \{0, 1\}^K$ such that $P(\boldsymbol{x}, \boldsymbol{z}) \leq 1/4$.

Next, we will complete the proof by showing (2) $\Rightarrow$ (1). Let $\boldsymbol{x} \in \{0, 1\}^n$ and suppose that we have $z \in \{0, 1\}^K$ such that $P(\boldsymbol{x}, \boldsymbol{z}) \leq 1/4$. Therefore,

$$\frac{2^K - \frac{1}{2}}{b-c}\left(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c\right) - \hat{z} \leq \frac{1}{2}$$

Next we add $\hat{z}$ to both sides of the equation, multiply both sides by $\frac{b-c}{2^K - \frac{1}{2}}$ and finally add c to both sides, which produces

$$\boldsymbol{a}^\mathsf{T}\boldsymbol{x} \leq c + (\frac{1}{2} + \hat{z})\frac{b-c}{2^K - \frac{1}{2}}$$

Rewriting the right hand side gives

$$\frac{c(2^K - \frac{1}{2}) + (\frac{1}{2} + \hat{z})(b-c)}{2^K - \frac{1}{2}}$$

Since $\hat{z} \in \{0, 1, \ldots, 2^K - 1\}$ we know $\hat{z} \leq 2^K - 1$, hence

$$\begin{aligned} \frac{c(2^K - \frac{1}{2}) + (\frac{1}{2} + z)(b-c)}{2^K - \frac{1}{2}} &\leq \frac{c(2^K - \frac{1}{2}) + (2^K - \frac{1}{2})(b-c)}{2^K - \frac{1}{2}} \\ &= b. \end{aligned}$$

Therefore, $\boldsymbol{a}^\mathsf{T}\boldsymbol{x} \leq b$ and we are done. □

The consequence of Theorem 1 is that $P(\boldsymbol{x}, z)$ distinguishes between values of $\boldsymbol{x}$ that meet the constraint and that violate the constraint. For a fixed $\boldsymbol{x}$ we have that if $\boldsymbol{x}$ meets the constraint, then the minimum of the penalty function is between 0 and 1/4. On the other hand, if $\boldsymbol{x}$ violates the constraint, then the minimum of the penalty function is strictly larger then 1/4.

Next, we prove the proposition about the uniqueness of the penalty function.

Proof of Proposition 2: The first direction of the proof is the case of Theorem 1. For the other direction we divide the problem into four cases: (i) $\alpha_K > (2^K - 1/2)/(b-c)$, (ii) $0 < \alpha_K < (2^K - 1/2)/(b-c)$, (iii) $\alpha_K = 0$ and (iv) $\alpha_K < 0$.

We start with the case (i) $\alpha_K > (2^K - 1/2)/(b-c)$. Suppose $\boldsymbol{x}^\mathsf{T}\boldsymbol{a} = b$, then $\boldsymbol{x} \in F$. Therefore,

$$\begin{aligned} \alpha(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c) - z &> \frac{2^K - \frac{1}{2}}{b-c}(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c) - z \\ &= 2^K - \frac{1}{2} - z \\ &\geq 2^K - \frac{1}{2} - (2^K - 1) \\ &= \frac{1}{2}. \end{aligned}$$

Hence, for all $z \in \{0, 1, \ldots, 2^K - 1\}$ we have $P(\boldsymbol{x}, z) > 1/4$.

For case (ii) $0 < \alpha_K < (2^K - 1/2)/(b-c)$ we set $\boldsymbol{a}^T\boldsymbol{x} = b + 1/\alpha_K$. Clearly, $\boldsymbol{x} \notin F$, but we will show that there is a $z \in \{0, 1, \ldots, 2^K - 1\}$ such that $P(\boldsymbol{x}, z) \leq 1/4$. By our choice of $\boldsymbol{x}$ we have that $\alpha(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c) = \alpha(b-c) + 1$. Furthermore, by the bounds on α we get $0 < \alpha(b-c) < 2^K - 1/2$. Combining both gives us

$$0 < \alpha(\boldsymbol{a}^\mathsf{T}\boldsymbol{x} - c) < 2^K + \frac{1}{2}.$$

Hence there exists a $z \in \{0, 1, \ldots, 2^K - 1\}$ such that $P(\boldsymbol{x}, z) < 1/4$.

For case (iii) where $\alpha_K = 0$ we take any $x \notin F$ and set $z = 0$. Then $P(\boldsymbol{x}, z) = 0$.

For case (iv) we have $\alpha_K < 0$. Set $\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{x} = c - 1/\alpha_K$, where we assume that $b \geq c - 1/\alpha_K$. Therefore, $\boldsymbol{x} \in F$. We then see

$$\max_{z \in \{0,1,\ldots,2^K-1\}} \alpha_K(\boldsymbol{a}^T\boldsymbol{x} - c) - z = \alpha_K(\boldsymbol{a}^T\boldsymbol{x} - c) = -1$$

Hence, for all $z \in \{0, 1, \ldots, 2^K - 1\}$ we have $P(\boldsymbol{x}, z) \geq 1$. □

During the proof of Proposition 2 it becomes clear that $\alpha_K < (2^K - 1/2)/(b-c)$ will not produce a useful penalty function as the penalty function does not distinguish between values of $\boldsymbol{x}$ that meet the constrain or ones that violate the constraint. For the case $\alpha_K > (2^K - 1/2)/(b-c)$, an argument can be made that it does differentiate between values of $\boldsymbol{x}$ that

TABLE 3. Settings used for the DWaveSampler.

Argument Name	Value
anneal_offsets	None
anneal_schedule	None
annealing_time	1000
answer_mode	histogram
auto_scale	True
flux_biases	None
flux_drift_compensation	True
h_gain_schedule	[[0,1],[1000,1]]
initial_state	None
max_answers	500
num_reads	500
num_spin_reversal_transforms	0
programming_thermalization	1000
readout_thermalization	0
reduce_intersample_correlation	False
reinitialize_state = True	

comply with the constraint and $\boldsymbol{x}$ that violate the constraint. However, values of $\boldsymbol{x} \in F$, where $\boldsymbol{a}^T\boldsymbol{x}$ is close to b, will have much larger values for $\min_z P(\boldsymbol{x}, z)$, compared to values of $\boldsymbol{x}$, where $\boldsymbol{a}^\intercal\boldsymbol{x}$ is close to c. Hence, a penalty function where $\alpha_K > (2^K - 1/2)/(b-c)$ gives a worse approximation then a penalty function with $\alpha_K = (2^K - 1/2)/(b-c)$.

Lastly, we show the proof of the size of the penalty when a constraint is violated.

Proof of Proposition 3: Let $x \notin F$, then $\boldsymbol{a}^\intercal\boldsymbol{x} > b$. Therefore, we know $\alpha_K(\boldsymbol{a}^T\boldsymbol{x} - c) > 2^K - 1/2$. Hence, the $P(\boldsymbol{x}, z)$ is minimized when all z_i values are 1 and the total sum $\sum_{i=0}^{K-1} 2^i z_i$ becomes $2^K - 1$. Thus, we get:

$$\min_{z\in\{0,1\}^K} P(\boldsymbol{x}, z) = (\alpha_K(\boldsymbol{a}^T\boldsymbol{x} - c) - 2^K + 1)^2.$$

Next, we substitute $\boldsymbol{a}^\intercal\boldsymbol{x} = \varepsilon + b$ and use some basic algebra to produce

$$\min_{z\in\{0,1\}^K} P(\boldsymbol{x}, z) = (\alpha_K(b-c) + \alpha_K\varepsilon - 2^K + 1)^2.$$

Note that $\alpha_K(b-c) = 2^K - 1/2$. Using this property then produces the final result

$$\begin{aligned}\min_{z\in\{0,1\}^K} P(\boldsymbol{x}, z) &= (2^K - \frac{1}{2} + \alpha_K\varepsilon - 2^K + 1)^2 \\ &= (\frac{1}{2} + \alpha_K\varepsilon)^2 \\ &= \frac{1}{4} + \alpha_k\varepsilon + (\alpha_k\varepsilon)^2\end{aligned}$$

□

APPENDIX B
SOLVER PARAMETERS

For this research, the Advantage_system4.1 was used. The annealig time was set to 1000μs and the number of reads was set to 500. All others settings were left at default, which is equal to the settings shown in Table 3.

For the hybrid BQM and CQM solvers of DWave, there is just one parameter to set, which is the time limit. This was set to 10 seconds.

TABLE 4. Settings for the BQM and CQM hybrid solver from DWave.

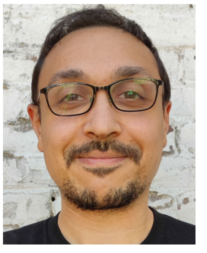
Argument Name	Value
time_limit	10

Similarly to the hybrid BQM and CQM solvers, all the used solvers from Microsoft Azure Quantum Inspired Optimization were given a timeout of 10s. All other settings were left at the default, which is equal to setting all other settings to None.

REFERENCES

[1] N. Soininen and K. Huhta, "Law for decentralized electricity systems: Lessons from complexity theory," in *Sustainable Energy Democracy and the Law*. Leiden, The Netherlands: Brill Nijhoff, 2021, pp. 77–98.
[2] F. Shahrokhi and D. W. Matula, "The maximum concurrent flow problem," *J. ACM*, vol. 37, no. 2, pp. 318–334, Apr. 1990.
[3] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *J. ACM*, vol. 19, no. 2, pp. 248–264, Apr. 1972.
[4] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, "Semidefinite programming for optimal power flow problems," *Int. J. Electr. Power Energy Syst.*, vol. 30, nos. 6–7, pp. 383–392, Jul. 2008.
[5] P. H. Nguyen, W. L. Kling, G. Georgiadis, M. Papatriantafilou, L. A. Tuan, and L. Bertling, "Application of the graph theory in managing power flows in future electric networks," in *New Frontiers Graph Theory*, Y. Zhang, Ed. Rijeka, Croatia: IntechOpen, 2012, ch. 12.
[6] Y. Yamagata and H. Seya, "Proposal for a local electricity-sharing system: A case study of Yokohama city, Japan," *IET Intell. Transp. Syst.*, vol. 9, no. 1, pp. 38–49, Feb. 2015.
[7] T. Tanjo, K. Minami, and H. Maruyama, "Graph partitioning of power grids considering electricity sharing," *Int. J. Smart Grid Clean Energy*, vol. 5, no. 2, pp. 112–120, Apr. 2016.
[8] B. Ghaddar, J. Marecek, and M. Mevissen, "Optimal power flow as a polynomial optimization problem," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 539–546, Jan. 2016.
[9] A. L. Bella, P. Klaus, G. Ferrari-Trecate, and R. Scattolini, "Supervised MPC control of large-scale electricity networks via clustering methods," 2020, *arXiv:2004.14117*.
[10] A. L. Bella, P. Klaus, G. Ferrari-Trecate, and R. Scattolini, "Supervised model predictive control of large-scale electricity networks via clustering methods," *Optim. Control Appl. Methods*, vol. 43, no. 1, pp. 44–64, Jan. 2022.
[11] A. Safdarian, P. H. Divshali, M. Baranauskas, A. Keski-Koukkari, and A. Kulmala, "Coalitional game theory based value sharing in energy communities," *IEEE Access*, vol. 9, pp. 78266–78275, 2021.
[12] N. S. Hadjidimitriou, A. Frangioni, A. Lodi, and T. Koch, *Mathematical Optimization for Efficient and Robust Energy Networks* (AIRO Springer Series). Cham, Switzerland: Springer, 2021.
[13] I. Hen and M. S. Sarandy, "Driver Hamiltonians for constrained optimization in quantum annealing," *Phys. Rev. A, Gen. Phys.*, vol. 93, no. 6, Jun. 2016, Art. no. 062312.
[14] I. Hen and F. M. Spedalieri, "Quantum annealing for constrained optimization," *Phys. Rev. Appl.*, vol. 5, no. 3, Mar. 2016, Art. no. 034007.
[15] E. Zahedinejad and A. Zaribafiyan, "Combinatorial optimization on gate model quantum computers: A survey," 2017, *arXiv:1708.05294*.
[16] S. Hadfield, Z. Wang, B. O'Gorman, E. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, p. 34, Feb. 2019.
[17] D. J. Griffiths and D. F. Schroeter, *Introduction to Quantum Mechanics*. Cambridge, U.K.: Cambridge Univ. Press, 2018.
[18] A. Messiah, *Quantum Mechanics*. Chelmsford, MA, USA: Courier Corporation, 2014.
[19] C. C. McGeoch, "Theory versus practice in annealing-based quantum computing," *Theor. Comput. Sci.*, vol. 816, pp. 169–183, May 2020.
[20] A. Lucas, "Ising formulations of many NP problems," *Frontiers Phys.*, vol. 2, p. 5, Feb. 2014.
[21] S. E. Venegas-Andraca, W. Cruz-Santos, C. McGeoch, and M. Lanzagorta, "A cross-disciplinary introduction to quantum annealing-based algorithms," *Contemp. Phys.*, vol. 59, no. 2, pp. 174–197, Apr. 2018.

[22] F. Glover, G. Kochenberger, R. Hennig, and Y. Du, "Quantum bridge analytics I: A tutorial on formulating and using qubo models," *Ann. Oper. Res.*, vol. 314, pp. 141–183, Apr. 2022.
[23] C.-E. Bichot and P. Siarry, *Graph Partitioning*. Hoboken, NJ, USA: Wiley, 2013.
[24] K. Masato and N. Hiromi, *Divide-and-Conquer Approaches to Quantum Chemistry: Theory and Implementation*. Dordrecht, The Netherlands: Springer, 2011, pp. 97–127.
[25] F. Rosato, "Heuristic graph partitioning with preferred cluster sizes and application to the generation of realistic distribution grid topologies," in *Proc. IEEE 15th Int. Conf. Compat., Power Electron. Power Eng. (CPE-POWERENG)*, Jul. 2021, pp. 1–7.
[26] A. B. Kahng, J. Lienig, I. L. Markov, and J. Hu, *VLSI Physical Design From Graph Partitioning to Timing Closure*, vol. 312. Cham, Switzerland: Springer, 2011.
[27] M. Toril, I. Molina-Fernández, V. Wille, and C. Walshaw, "Analysis of heuristic graph partitioning methods for the assignment of packet control units in GERAN," *Wireless Pers. Commun.*, vol. 60, no. 4, pp. 611–633, Oct. 2011.
[28] C. Márquez, E. César, and J. Sorribes, "Graph-based automatic dynamic load balancing for HPC agent-based simulations," in *Proc. Eur. Conf. Parallel Process.*, S. Hunold, A. Costan, D. Giménez, A. Iosup, L. Ricci, M. Engracia, G. Requena, V. Scarano, A. L. Varbanescu, S. L. Scott, S. Lankes, J. Weidendorfer, and M. Alexander, Eds. Cham, Switzerland: Springer, 2015, pp. 405–416.
[29] U. V. Catalyurek and C. Aykanat, "Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 10, no. 7, pp. 673–693, Jul. 1999.
[30] S. Navlakha and C. Kingsford, "Exploring biological network dynamics with ensembles of graph partitions," in *Proc. Pacific Symp. Biocomputing. Pacific Symp. Biocomput.*, 2010, pp. 77–166.
[31] C. Tsourakakis, C. Gkantsidis, B. Radunovic, and M. Vojnovic, "FENNEL: Streaming graph partitioning for massive scale graphs," in *Proc. 7th ACM Int. Conf. Web Search Data Mining*, Feb. 2014, pp. 333–342.
[32] T. Lopes, V. Ströele, M. Dantas, R. Braga, and J.-F. Mehaut, "A parallel graph partitioning approach to enhance community detection in social networks," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–6.
[33] L. Hyafil and R. L. Rivest, "Graph partitioning and constructing optimal decision trees are polynomial complete problems," *Laboratoire de Recherche en Informatique et Automatique, Rapport de Recherche*, vol. 33, pp. 1–9, Oct. 1973.
[34] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theor. Comput. Sci.*, vol. 1, no. 3, pp. 237–267, Feb. 1976.
[35] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz, "Recent advances in graph partitioning," in *Algorithm Engineering*. Cham, Switzerland: Springer, 2016.
[36] H. Ushijima-Mwesigwa, C. F. A. Negre, and S. M. Mniszewski, "Graph partitioning using quantum annealing on the D-wave system," in *Proc. 2nd Int. Workshop Post Moores Era Supercomput.*, Nov. 2017, pp. 22–29.
[37] S. Pramanik and M. G. Chandra, "Quantum-assisted graph clustering and quadratic unconstrained D-ary optimisation," 2020, *arXiv:2004.02608*.
[38] C. F. A. Negre, H. Ushijima-Mwesigwa, and S. M. Mniszewski, "Detecting multiple communities using quantum annealing on the D-wave system," *PLoS ONE*, vol. 15, no. 2, Feb. 2020, Art. no. e0227538.
[39] U. Chukwu, R. Dridi, J. Berwald, M. Booth, J. Dawson, D. Le, M. Wainger, and S. P. Reinhardt, "Constrained-optimization approach delivers superior classical performance for graph partitioning via quantum-ready method," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Sep. 2020, pp. 1–6.
[40] S. Greer and D. O'Malley, "An approach to seismic inversion with quantum annealing," in *Proc. SEG Tech. Program Expanded Abstr.*, Sep. 2020, pp. 2845–2849.
[41] S. Mugel, C. Kuchkovsky, E. Sánchez, S. Fernández-Lorenzo, J. Luis-Hita, E. Lizaso, and R. Orús, "Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks," *Phys. Rev. Res.*, vol. 4, no. 1, Jan. 2022, Art. no. 013006.
[42] F. Phillipson and H. S. Bhatia, "Portfolio optimisation using the D-wave quantum annealer," 2020, *arXiv:2012.01121*.
[43] F. Phillipson and I. Chiscop, "A quantum approach for tactical capacity management of distributed electricity generation," in *Proc. Int. Conf. Innov. Community Services*, F. Phillipson, G. Eichler, C. Erfurth, and G. Fahrnberger, Eds. Cham, Switzerland: Springer, 2022, pp. 323–333.
[44] M. Dukalski, D. Rovetta, S. van der Linde, M. Möller, N. Neumann, and F. Phillipson, "Quantum computer-assisted global optimization in geophysics illustrated with stack-power maximization for refraction residual statics estimation," *Geophysics*, vol. 88, no. 2, pp. V75–V91, Mar. 2023.
[45] M. Carsten, M. Wided, and K. David, "SciGRID—An open source reference model for the European transmission network (v0.2)," SciGRID, Jul. 2016.
[46] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.
[47] J. Tilly, "The variational quantum Eigensolver: A review of methods and best practices," *Phys. Rep.*, vol. 986, pp. 1–128, Nov. 2022.
[48] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014, *arXiv:1411.4028*.
[49] F. Leymann and J. Barzen, "The bitter truth about gate-based quantum algorithms in the NISQ era," *Quantum Sci. Technol.*, vol. 5, no. 4, Sep. 2020, Art. no. 044007.



GIUSEPPE COLUCCI received the B.Sc. and M.Sc. degrees in physics from the University of Bari Aldo Moro, Italy, in 2008 and 2010, respectively, and the Ph.D. degree in theoretical physics from Goethe University, Frankfurt am Main, Germany, in 2014.

From 2014 to 2016, he was a Software Engineer with UL TS, Leiden, The Netherlands, and then as a Quantitative Analyst with Rabobank, Utrecht, The Netherlands, from 2016 to 2018. He currently invests the role of Senior ALM Specialist with de Volksbank, Utrecht, The Netherlands. He is also the Founder of Quantum Quants, Rotterdam, The Netherlands, established in 2020, which helps companies to get insights on quantum computing and develop business applications of quantum computing. He published several articles in theoretical nuclear physics and quantum computing. His research interest includes applications of quantum computing to business use cases, with a focus on optimization in the financial and energy sectors.

STAN VAN DER LINDE received the degree in applied mathematics from the Delft University of Technology, in 2021. For the master's thesis, he worked on using quantum annealing for a geophysical application.

His specialization is in computational science and engineering. During his studies, he focused on numerical mathematics, optimization, and quantum information courses. Since 2022, he has been a Junior Scientist with the Applied Cryptography & Quantum Algorithms Department, TNO. Within this department, he works on bridging the gap between industry and academia in the field of quantum computing. To this end, he researches (hybrid) quantum computing methods that could be used on industry-sized problems.

FRANK PHILLIPSON received the degree in econometrics and operations research from VU Amsterdam, in 1996, the P.D.Eng. degree in applied mathematics from the Delft University of Technology, in 1998, and the Ph.D. degree in business analytics from VU Amsterdam, in 2014.

From 1998 to 2003, he was employed with KPN (Royal Dutch Telecom) Research. In 2004, he joined TNO, the Netherlands Organization for Applied Scientific Research. He is currently a Senior Scientist with the Department of Applied Cryptography & Quantum Algorithms, TNO, where he is responsible for the research into applications on quantum computing devices. He is also a part-time Full Professor of computational operations research with Maastricht University, researching the use of quantum computers and other accelerating hardware for solving optimization problems.