A hybrid Quantum-Classical Algorithm for Mixed-Integer Optimization in Power Systems

Petros Ellinas, Samuel Chevalier, Spyros Chatzivasileiadis,
Department of Wind and Energy Systems, Technical University of Denmark (DTU)
Elektrovej, 2800 Kgs. Lyngby, Denmark

Abstract—**Mixed Integer Linear Programming (MILP) can be considered the backbone of the modern power system optimization process, with a large application spectrum, from Unit Commitment and Optimal Transmission Switching to verifying Neural Networks for power system applications. The main issue of these formulations is the computational complexity of the solution algorithms, as they are considered NP-Hard problems. Quantum computing has been tested as a potential solution towards reducing the computational burden imposed by these problems, providing promising results, motivating the can be used to speedup the solution of MILPs. In this work, we present a general framework for solving power system optimization problems with a Quantum Computer (QC), which leverages mathematical tools and QCs' sampling ability to provide accelerated solutions. Our guiding applications are the optimal transmission switching and the verification of neural networks trained to solve a DC Optimal Power Flow. Specifically, using an accelerated version of Benders Decomposition , we split a given MILP into an Integer Master Problem and a linear Subproblem and solve it through a hybrid "quantum-classical" approach, getting the best of both worlds. We provide 2 use cases, and benchmark the developed framework against other classical and hybrid methodologies, to demonstrate the opportunities and challenges of hybrid quantum-classical algorithms for power system mixed integer optimization problems.**

Index Terms—**Quadratic unconstrained binary optimization, Quantum Computing, Benders decomposition, Optimal Transmission Switching, Neural Network Verification**

I. INTRODUCTION

Mixed-integer linear programs (MILPs) are pivotal in the realm of power systems engineering. These mathematical optimization methods, dubbed as such due to their inclusion of both continuous and discrete variables, are harnessed for the modeling and resolution of a diverse array of issues related to the operation, planning, and control of electrical power systems. For example, fundamental challenges such as unit commitment [1], investment planning [2], and optimal transmission switching (OTS) [3] are cast as MILPs. Furthermore, there is a growing trend toward utilizing additional tools, such as Neural Network constraint-violation verification [4], which can also be framed as MILPs.

Nevertheless, the integer aspect introduces a combinatorial element to the problem, as the solver must traverse various combinations of discrete choices. With an increase in the number of discrete variables and their potential values, the solution space grows exponentially, classifying the problem of finding the global solution of MILP models within the NP-Hard problem category.

Quantum computing is a promising direction to deliver a solution to the high computational complexity of MILPs. Recently, authors in [5],[6] have provided compelling evidence to support the notion that QCs can currently offer valuable capabilities, even with their inherent noise at the moment, and prior to the advent of fault-tolerant systems. This includes the currently available processors. Considering the rapid progress in the quantum computing field, we expect that these capabilities will increase substantially when effective fault-tolerant systems for QCs develop in the near future. Therefore, it is necessary to explore this track, as it can be a methodology that would help us solve a set of MILP power system problems that have so far been impossible to solve in an acceptable time window with reasonable computational recourses, and operate more efficient, safe, and resilient power grids.

Quantum computing has been investigated for the solution of MILPs, for instance, in [7] where the authors propose a Benders decomposition approach to solve a general form of MILPs, using a QC. However, according to [8] there are numerous challenges resulting from this method, such as the choice of inferior feasibility and optimality cuts and slow convergence towards the end of the algorithm. In [8], a multi-cut Benders decomposition is presented, during which the Quantum Processing Unit (QPU) is specifically allocated to execute a binary optimization subroutine focused on selecting the cuts that should be incorporated into the Master Problem (MP) to minimize the problem's size and accelerate the solution. However, this is done while both the MP and the Subproblems (SPs) are solved through classical methods, which slows the algorithm down significantly.

In the current literature, the exploration of quantum algorithms' application to power systems is still limited [9]. Among them, Ref. [10] presented the first successful solution of an AC power flow in a real QC. They successfully solved it for 3 and 5 bus systems, and they discussed scalability and noise issues of the current "noisy-era" QCs. Before that, Ref [11] proposed an algorithm to solve a power flow with quantum computing but only demonstrated in simulated quantum simulations. Ref. [12] put forward the idea of combining machine learning and quantum computing to potentially tackle security and reliability issues in large-scale power grid systems. Furthermore, in [13] authors use Quantum Annealing to solve a simple version of optimal phasor measurement unit placement, which is a crucial problem for the next-generation

power system design. They showed that in certain cases, the Quantum Annealing solver surpassed the classical solver CPLEX in performance.

Many studies have been focused in solving the Unit Commitment problem. In particular, [14] proposed a distributed algorithm for solving unit commitment. Furthermore, [15] proposed quantum-circuit-based algorithms for microgrid state estimation. Following the work of [16], authors of [17] have proposed a three-block alternating direction method, to solve the unit commitment problem.

In this work, we provide a framework for transforming MILP models of power system problems to a "quantum-classical solvable form". In contrast to alternative approaches, our methodology incorporates mathematical tools or harnesses the sampling capabilities of quantum solvers to expedite the solution process. Specifically, in quantum computing, measurements collapse the superposition of qubits' states to a specific outcome. The probabilities of different outcomes are determined by the amplitudes of the quantum states. Repeated measurements allow for the sampling of these outcomes according to their probabilities. Similarly, at the end of the annealing process, the quantum annealer samples from the final quantum state to obtain solutions to the optimization problem. The resulting combinatorial problems are solved solely in a QPU. The derived approximated solution, in some cases, has been empirically proven to converge faster to the optimal solution than the classical branch and bound method. Additionally, we offer a comprehensive Python code in the form of a tutorial available on our GitHub repository ([18]), which instructs users on how to convert any Mixed-Integer Linear Program (MILP) into the specified format. We benchmark this approach on OTS [3] and NN verification for DC-OPF [4], using both of the two prevailing quantum approaches: quantum annealing and quantum circuits.

Summarizing, the contributions of this paper are:

- We provide a framework for transforming MILP models of power system problems to a "quantum-classical solvable form", providing two different acceleration techniques. As part of the contributions of this paper, a comprehensive Python code, in the form of a tutorial, on how to solve any MILP problem in this framework is made publicly available in [18].
- We provide extensive numerical results for large enough instances that arrive at the limits of the capacity constraints of the current available QCs and we discuss the benefits of each method and the challenges that need to be addressed to exploit the future quantum advantage.

This paper is organized as follows. Section II covers Benders decomposition theory, its enhancements, and the conversion of the MP to a quantum solvable Quadratic Unconstrained Optimization problem (QUBO). In Section III, we detail the optimization models for benchmarking, while Section IV showcases the numerical experiments evaluating the framework's performance. Section V discusses the results and current challenges, and Section VI concludes.

II. PROBLEM FORMULATION

A variety of power system problems can be formulated in the form:

$$\max_{\boldsymbol{z},\boldsymbol{y}} \quad \mathbf{i}^T \boldsymbol{z} + \mathbf{c}^T \boldsymbol{y}$$
$$s.t. \quad \mathbf{A}\boldsymbol{z} + \mathbf{B}\boldsymbol{y} \leq \mathbf{b} \tag{1}$$
$$\boldsymbol{z} \in \mathbb{Z}, \boldsymbol{z} \in \{0,1\}^n, \boldsymbol{y} \in \mathbb{R}_+^p$$

where $\mathbf{i}^T$ and $\mathbf{c}^T$ are the coefficient matrices of integer and continuous variables in the objective function, and $\mathbf{A}$ and $\mathbf{B}$ correspond to coefficient matrices of integer and continuous variables in the constraints, respectively. Moreover, $\mathbf{b}$ is a vector of real numbers. These models are a special form of a bigger umbrella of the MILP models because they consist of binary variables $\mathbf{z}$ (instead of general integer variables) and continuous variables $\mathbf{y}$.

To exploit the speedup of quantum algorithms [19], [20] in solving combinatorial optimization problems, which are integer programs with binary variables, we should separate the aforementioned MILP models into an integer program and a continuous program. This can be done using Benders Decomposition (BD) [21]. The main idea of Benders is that for each feasible $\boldsymbol{z}$, the optimal $\boldsymbol{y}$ is determined via a linear program. Treating $\boldsymbol{y}$ as a function of $\boldsymbol{z}$, we use a scalar variable to represent the best $\boldsymbol{y}$ choice for a given $\boldsymbol{z}$, replacing its contribution to the objective. Initially, a coarse estimation of $\boldsymbol{y}$ is employed, followed by a sequence of dual solutions to tighten the approximation. This process converts the original MILP problem into a simpler MILP Master Problem and a linear Subproblem.

According to the above, we can first fix the binary variables in (1), transforming them to the vector $\boldsymbol{z}^{(n)}$. Therefore we can write the primal form of the SubProblem as a linear program:

$$\max_{\boldsymbol{y}} \quad \mathbf{i}^T \boldsymbol{z}^{(n)} + \mathbf{c}^T \boldsymbol{y} \tag{SP}$$
$$s.t. \quad \mathbf{B}\boldsymbol{y} \leq \mathbf{b} - \mathbf{A}\boldsymbol{z}^{(n)} \quad : \boldsymbol{\lambda} \tag{2}$$

where (n) index denotes the current iteration number. As (SP) is a convex linear program consisting only of continuous decision variables, a strong duality condition holds. Consequently we can define $\boldsymbol{\lambda}$ as the vector of dual variables associated to constraints (2) and so we can express its dual form as:

$$x(\boldsymbol{z}) = \min_{\boldsymbol{\lambda}} \quad \boldsymbol{\lambda}(\mathbf{b} - \mathbf{A}\boldsymbol{z}^{(n)})^T \tag{Dual-SP}$$
$$s.t. \quad \mathbf{B}\boldsymbol{\lambda} \geq \mathbf{c}$$
$$\boldsymbol{\lambda} \geq 0$$

At this point, we define extreme points set $\mathbf{T}$ and extreme rays set $\mathbf{K}$. We denote extreme point t as $\boldsymbol{p}^t$ and extreme ray k as $\boldsymbol{r}^k$ and they belong to the polyhedron:

$$Q = \{\boldsymbol{\lambda} \in \mathbb{R}_+^m \mid \mathbf{B}\boldsymbol{\lambda} \geq \mathbf{c}\} \tag{3}$$

Extreme rays and extreme points can be acquired from (Dual-SP)'s decision variables. Specifically, the (Dual-SP) is unbounded and $x(\mathbf{z}) = -\infty$, then $\boldsymbol{\lambda}$ expresses a dual ray $\boldsymbol{r}^{k'}$, rendering (SP) infeasible. This means that x does not allow a

feasible solution to (1), and that helps us to introduce a new feasibility cut in the MP as:

$$(\mathbf{b} - \mathbf{A}z)r^{k'} \geq 0 \quad k' \in K \tag{4}$$

However when (Dual-SP) is feasible, then dual variables $\boldsymbol{\lambda}$ denote an extreme point $\boldsymbol{p}^{t'}$, shaping an optimality cut as:

$$(\mathbf{b} - \mathbf{A}z)\boldsymbol{p}^{t'} \geq s \quad t' \in T \tag{5}$$

where s is a scalar value denoting the contribution of continuous variables to the MP:

$$\max_{z} \quad \mathbf{i}^T z + s \tag{MP}$$
$$s.t. \quad \boldsymbol{p}^t(\mathbf{b} - \mathbf{A}z) \geq s \quad \text{for } t \in T$$
$$\boldsymbol{r}^k(\mathbf{b} - \mathbf{A}z) \geq 0 \quad \text{for } k \in K$$

The algorithm iteratively reduces the optimality gap. The process is interrupted if the optimality gap is lower than a small positive number ϵ. It has been shown that Benders Decomposition converges to the global optimum of the initial MILP problem [8].

A. Improvements

1) Optimization Method I: It has been reported that in some cases Benders Decomposition can have slow convergence times. However, as pointed out in references [22], [23], [24], there are multiple strategies, from the classical mathematical programming perspective, to enhance Benders decomposition. Consequently, we modify the conventional BD algorithm to tackle the slow convergence problems, by guiding the choice of binary variables and selecting efficient cuts.

Smart cuts or Pareto-optimal cuts selection were investigated in [22]. The authors of the aforementioned paper first introduced the term "domination" for cuts. Specifically a cut $l'(\mathbf{b} - \mathbf{A}z)$ dominating the cut $l''(\mathbf{b} - \mathbf{A}z)$ when:

$$l'(\mathbf{b} - \mathbf{A}z) \geq l''(\mathbf{b} - \mathbf{A}z) \quad \forall \quad z \in Z \tag{6}$$

for all z for a strict inequality for at least one solution z. Therefore they denote the pareto-optimal cut as the cut that does not get dominated by any other cuts.

Furthermore in [22], the definition of the core point was introduced and they connected it to the selection of Pareto optimal cuts. The core point of Z (denoted as $\overline{z}^{(n)}$) is any point from the set Z that lies within the relative interior of the convex hull Y_c of Y. Finally, [24], exploited the connection between core points and Pareto optimal cuts, proposing an independent optimization problem to generate Pareto optimal cuts as:

$$\min_{\boldsymbol{\lambda}} \quad \boldsymbol{\lambda}(\mathbf{b} - \mathbf{A}\overline{z}^{(n)})^T \tag{Pareto-SP}$$
$$s.t. \quad \mathbf{B}\boldsymbol{\lambda} \geq \mathbf{c}$$
$$\boldsymbol{\lambda} \geq 0$$

where $\overline{z}^{(n)}$ is a core point of Z. The optimal solution in the proposed problem is Pareto optimal.

Still, according to [25], [26], it is very hard to find a core point $\overline{z}^{(n)}$. To tackle that, [24] used the following core point approximation technique:

$$\overline{z}^{(n)} = \frac{1}{2}\overline{z}^{(n-1)} + \frac{1}{2}z^{(n-1)} \tag{7}$$

with a random $\overline{z}^{(0)}$. With this we can approximate a core point gradually, as in every iteration, we get a point between two solutions, which gradually approaches a core point.

Another problem of BD is the tailing effect [23]. Specifically, in [23], it has been observed that algorithms present an unstable behavior when they approximate the solution. Therefore, they proposed to add an extra regularizing term to the objective function. This term is the Hamming distance between $\overline{z}^{(n)}$ and $\overline{z}^{(n-1)}$. Consequently, we can write the resulting MP as:

$$\max_{z} \quad \mathbf{i}^T z + s + ||z - z^{(n-1)}|| \tag{MP-Opt}$$
$$s.t. \quad \boldsymbol{p}^t(\mathbf{b} - \mathbf{A}z) \geq s \quad \text{for } t \in T$$
$$\boldsymbol{r}^k(\mathbf{b} - \mathbf{A}z) \geq 0 \quad \text{for } k \in K$$

A benefit of this method is that we can remove previous cuts if the algorithm has to execute many iterations [23]. This can reduce the memory alongside the qubits requirements of the algorithm. The whole algorithm with optimization method I can be described by Algorithm 1.

Algorithm 1: Optimization Method I

Data: Problem (1)
Result: z^*, y^*
$n \leftarrow 1$;
$z, \overline{z}^{(0)} \leftarrow 0$;
while $|UB - LB| > \epsilon$ **do**
    Solve (Dual-SP) and get $x^{(n)}(z)$;
    $LB \leftarrow x_z^{(n)}$;
    $\overline{z}^{(n)} = \frac{1}{2}\overline{z}^{(n-1)} + \frac{1}{2}z^{(n-1)}$;
    Solve (Pareto-SP) and get $x(\overline{z}^{(n)}), \boldsymbol{\lambda}$;
    if $x(\overline{z}^{(n)})) = -\infty$ **then**
        Add extreme ray $K \cup \{k\}$;
    else
        Add extreme point $T \cup \{t\}$;
    end
    Solve (MP-Opt) and get $s(\boldsymbol{p}), z^{(n)}$;
    $UB \leftarrow s(\boldsymbol{p})$;
    $n \leftarrow n + 1$;
end

---

2) Optimization Method II: This optimization methodology exploits the ability of quantum algorithms to provide multiple solutions at the same time, with a technique called sampling. During our experiments, we found that the exchange of data between QPU and CPU is very inefficient. Consequently, to minimize the access time to the QPU, in every iteration we choose to run the (Dual-SP) for the best R solutions that the QPU provides. Therefore, Algorithm 2 leverages the ability of QCs to provide multiple solutions, to accelerate the solution of MILPs.

B. QUBO transformation

Despite the two reformulations in the linear SPs, the MP resulting from Benders Decomposition cannot be directly solved by a QC, as it is still a MILP, as it contains the scalar variables. As provided evidence e.g. [27] show that QC can solve certain combinatorial problems faster than classical computers [27], we try to transform the (MP-Opt) to a "quantum solvable" form: that is a QUBO. Note that Higher-Order Binary Optimization (HOBO) is applicable as it can be always transformed to a QUBO [28]. To this end, we write the canonical form of optimality and feasibility cuts constraints, adding scalar variables $a_1^t, a_2^k \geq 0$ as:

$$\boldsymbol{p}^t(\mathbf{b} - \mathbf{A}\boldsymbol{z}) + a_1^t == s \tag{8}$$

$$\boldsymbol{r}^k(\mathbf{b} - \mathbf{A}\boldsymbol{z}) + a_2^k == 0 \tag{9}$$

Then, following the table shown in [7], we add the penalty weights P_1, P_2 and we raise the constraints to the square, before writing the resulting QUBO as:

$$\max_{\boldsymbol{z}} \quad \mathbf{i}^T \boldsymbol{z} + s + ||\boldsymbol{z} - \boldsymbol{z}^{(n-1)}|| + \tag{MP-QUBO}$$

$$\sum_{t \in T} P_1(\boldsymbol{p}^t(\mathbf{b} - \mathbf{A}\boldsymbol{z}) - s + a_1^t)^2 +$$

$$\sum_{k \in K} P_2(\boldsymbol{r}^k(\mathbf{b} - \mathbf{A}\boldsymbol{z}) + a_2^k)^2$$

However, (MP-QUBO) still consists of continuous variables, which cannot be understood by the QC. To tackle this problem we use a fundamental computer science technique, to approximate the continuous variables, called discretization or fixed-point approximation which is a two's-complement number system that encodes positive and negative numbers in a binary representation as:

$$s(\boldsymbol{p}) \approx \sum_{i=0}^{a_{cc}} 2^i \boldsymbol{p}_i^{pos} + \sum_{i=0}^{a_{cc}} 2^{-i} \boldsymbol{p}_i^{dec} - \sum_{i=0}^{a_{cc}} 2^i \boldsymbol{p}_i^{neg}$$

$$a_1^t \approx \sum_{i=0}^{e_{cc}^1} 2^i \boldsymbol{a}_i^{[t]}$$

$$a_2^k \approx \sum_{i=0}^{e_{cc}^2} 2^i \boldsymbol{a}_i^{[k]}$$

where a_{cc} are the number of qubits that are used to approximate either the positive, decimal, or negative part of s. Furthermore, t is the number of extreme points for a_1^t and k is the number of extreme rays. Notice that s is approximated with some negative, positive, and decimal numbers, but a_1^t, a_2^k are only approximated as positive integer variables. This does not degrade the quality of the solutions, because of the unconstrained nature of the resulting problem. The number of qubits required for approximating s is expressed by a_{cc} which, must follow the following equation:

$$a_{cc} \geq \log_2(1 + \max_{\boldsymbol{y}} \boldsymbol{c}^T \boldsymbol{y}) \tag{10}$$

According to (10), the number of qubits used to approximate the scalar variable s, which represents the contribution of the continuous function $\boldsymbol{c}^T \boldsymbol{y}$ to (MP-Opt)'s objective function,

must be sufficiently large to be greater than equal than the maximum value this function can take. Moreover, following (10):

$$e_{cc}^1 \geq \log_2(\max_{\boldsymbol{z}, s} \boldsymbol{p}^t(\mathbf{b} - \mathbf{A}\boldsymbol{z}) - s) \tag{11}$$

$$e_{cc}^2 \geq \log_2(\max_{\boldsymbol{z}} \boldsymbol{r}^k(\mathbf{b} - \mathbf{A}\boldsymbol{z})) \tag{12}$$

Note that the auxiliary variables a_1^t and a_2^k need to be allocated with a sufficient number of qubits to accurately approximate the maximum value of the sum of the remaining terms in equations (8) and (9), respectively.

Algorithm 2: Optimization Method II

Data: Problem (1), R
Result: $\boldsymbol{z}^*, \boldsymbol{y}^*$
$n \leftarrow 1$;
$\boldsymbol{z}, \overline{\boldsymbol{z}}^{(n)} \leftarrow 0$;
while $|UB - LB| > \epsilon$ **do**
  for $i \leftarrow 0$ to R **do**
    Solve (Dual-SP) and get $x_i^{(n)}(\boldsymbol{z}), \boldsymbol{\lambda}$;
    $LB \leftarrow x_i^{(n)}(\boldsymbol{z})$;
    if $|UB - LB| > \epsilon$ **then**
      Finish;
    end
    if $x(\boldsymbol{z}^{(n)})) = -\infty$ **then**
      Add extreme ray $K \cup \{k\}$;
    else
      Add extreme point $T \cup \{t\}$;
    end
  end
  Solve (MP-Opt) and get $s(\boldsymbol{p}), \boldsymbol{z}^{(n)}$;
  $UB \leftarrow s(\boldsymbol{p})$;
  $n \leftarrow n + 1$;
end

We note that according to [29], it is feasible to ensure convergence without the necessity of solving the MP to optimality in each iteration. However, this study does not delve into the practical implications of such an approach.

III. BENCHMARK MODELS

In this section, we outline the formulation of both the OTS and NN verification problem. We have selected these two problems as benchmarks because the former's solution is pivotal in Power Systems, while the latter presents a relatively novel problem with the potential to transform Power Systems Operation.

A. Optimal Transmission Switching (OTS)

The Optimal Transmission Switching (OTS) is a fundamental power systems problem. It involves maximizing the efficiency of the current system and addressing increasing demand using the already established infrastructure. The model of OTS is a mixture of an Optimal Power Flow (OPF), alongside

binary decisions for the optimal switching of transmission lines. Therefore, we can formulate the OTS problem as:

$$\min_{g_i,p_l,\theta_i,x_l} \sum_{i\in\mathcal{B}} c_i g_i \tag{13a}$$

$$s.t. \quad |p_l| \leq \overline{P}_l x_l \quad \forall l \in \mathcal{L} \tag{13b}$$

$$0 \leq g_i \leq \overline{P}_i \tag{13c}$$

$$g_i + \sum_{l\in\mathcal{L}_i^+} p_l x_l = \sum_{l\in\mathcal{L}_i^-} p_l x_l + D_i \tag{13d}$$

$$p_l \geq B_{i,j}(\theta_i - \theta_j) - M(1 - x_l) \tag{13e}$$

$$p_l \leq B_{i,j}(\theta_i - \theta_j) + M(1 - x_l) \tag{13f}$$

$$\sum_{l\in\mathcal{L}}(1 - x_l) \leq E \tag{13g}$$

$$g_i, \theta_i \in \mathbb{R} \quad \forall i,j \tag{13h}$$

$$x_l \in \{0,1\}, p_l \in \mathbb{R} \quad \forall l \in \mathcal{L} \tag{13i}$$

In this formulation, we have used the DC-OPF approximation as it serves our purposes. g_i expresses the generation dispatch of generators in node i and p_l expresses the power flow in line l, while $\overline{P}_l$ is the maximum power limit of the line. Furthermore, $\mathcal{L}_i^+$ is the set of lines that inject power to the node i, and $\mathcal{L}_i^-$ is the set of lines that export power from the node. Moreover, D_i represents the demand in bus i. Finally, θ_i represents the voltage magnitude in bus i, $B_{i,j}$ is the susceptance of line $l = (i,j)$ and x_l is the variable that denotes the status of the line i.e. for $x_l = 1$ the line is operational and for $x_l = 0$ the line does not transmit power.

B. Neural Network verification for power systems

Neural Network constraint violation verification is a crucial step toward finding a scalable and trustworthy parametrized policy for problems in safety-critical systems. Recently, in [4], the authors proposed a direct MILP transformation of Neural networks that ensures precise worst-case performance guarantees, on constraints violations, across the entire range of possible inputs, for the DC-OPF problem:

$$\max_{\hat{\mathbf{z}}_k,\mathbf{z}_k,\mathbf{p}_d,\hat{\mathbf{p}}_g} p_g - \overline{p}_g \text{ or } \underline{p}_g - p_g \tag{14a}$$

$$s.t. \quad \underline{\mathbf{p}}_d \leq \mathbf{p}_d \leq \overline{\mathbf{p}}_d \tag{14b}$$

$$(\hat{\mathbf{p}}_g)^{slack} = \sum_{i\in\mathbf{B}}(\mathbf{M}_d\mathbf{p}_d)^i - \sum_{i\in\mathbf{B}\backslash slack}(\hat{\mathbf{p}}_g^{nsg})^i \tag{14c}$$

$$\hat{\mathbf{z}}_1 = \mathbf{W}_1\mathbf{p}_d + \mathbf{b}_1 \tag{14d}$$

$$\hat{\mathbf{z}}_{k+1} = \mathbf{W}_{k+1}\mathbf{z}_k + \mathbf{b}_{k+1} \tag{14e}$$

$$\hat{\mathbf{p}}_g^{nsg} = \mathbf{W}_{K+1}\mathbf{z}_K + \mathbf{b}_{K+1} \tag{14f}$$

$$\mathbf{z}_k = \boldsymbol{\sigma}(\hat{\mathbf{z}}_k) \tag{14g}$$

$$\hat{\mathbf{z}}_k, \mathbf{z}_k \in \mathbb{R} \quad \forall k \tag{14h}$$

$$\mathbf{p}_d \in \mathbb{R} \quad \forall d \tag{14i}$$

$$\hat{\mathbf{p}}_g \in \mathbb{R} \quad \forall g \tag{14j}$$

Constraints (14d)-(14j) are the exact representation of a Neural Network with K layers, and neurons with ReLU activation functions. In (14j), $\mathbf{z}_k$ expresses the output vector of neurons in layer k while $\hat{\mathbf{z}}_k$ expresses their input vector. Function $\sigma(.)$ is the activation function, applied elementwise on vector $\hat{\mathbf{z}}_k$.
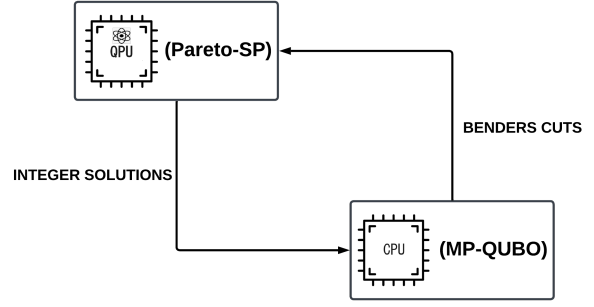

Fig. 1: Solution Workflow of Proposed Methodology

In this paper, and in most neural network implementations in recent years, we use the ReLU activation function, which can be expressed as $\max(\hat{\mathbf{z}}_k, 0)$. Following [4], we use the big-M method to convert this to a set of linear constraints with binaries (for more information, please see [4]). In (14e)-(14f), Matrix W represents the linear weights between layers K and $K + 1$, and vector $\mathbf{b}_{k+1}$ represents the biases in each neuron of layer $K + 1$, which are added before the application of the activation function $\sigma(.)$ in layer $K + 1$. Following this work we integrated in the aforementioned formulation the DC-OPF equations (14c). In particular, variables $\mathbf{p}_d$ and $\hat{\mathbf{p}}_g^{nsg}$ are the vectors of demand and generation (without the slack bus generation) respectively. Consequently, by changing the objective function, one could determine the maximum constraint violation of the trained NN, for any DC-OPF constraint. However, in the current formulation, the objective is to find the maximum upper or lower limit violation of the active power generation $p_g \in \hat{\mathbf{p}}_g$ of the generator g, where $\hat{\mathbf{p}}_g := \{\hat{\mathbf{p}}_g^{nsg}, (\hat{\mathbf{p}}_g)^{slack}\}$

A major drawback of this NN verification method is its scalability, although it is very accurate. Specifically, since Neural Networks can become very large, it is very hard computationally to run the resulting verification problems. The proposed framework could, theoretically, reduce the computational burden of this problem, in ideal settings.

IV. NUMERICAL EVALUATION

In this section, we present the numerical evaluation of our method, validating the proposed framework. For the quantum solution we used the Hybrid solver provided by D-Wave Leap [30] and for the classical solver we used Gurobi [31]. The solution workflow, showing the information exchange between CPU and QPU can be shown in Fig.1. We denote that for "Mean Iteration Time" in Table I, Table II and Table III, we include only CPU and QPU Access time [32]. Moreover, with "Minimum objective function until Iteration" on the y axis of Fig.2c-2h, we mean $min(UB)$ and $max(LB)$ found until a given iteration in the x axis.

In the following section, we will compare the proposed methodology with the following:

- Single Step MILP Optimization as shown in (1) (SSO)
- Conventional Benders Decomposed problem solved by CPUs only (C-BD-C)

- Accelerated Benders with optimization method I Decomposed problem solved by CPUs only (BD-C-I)
- Accelerated Benders with optimization method I Decomposed problem solved by CPU and QPU (BD-QC-I)
- Accelerated Benders with optimization method II Decomposed problem solved by CPU and QPU running the (Dual-SP), R times per Benders Iteration (BD-QC-II-R). We provide numerical results for R=2,3,5,10

We note that the Quantum Annealing solutions used in this section and Quantum Approximate Optimization Solutions benchmarked in the next section, are heuristic solutions. Therefore the results of the Branch and Bound Algorithm of Gurobi for the solution of (SSO) are used as a ground truth, to assess the quality of the benchmarked solutions.

A. Benders Benchmarks

Initially, we assess the efficacy of the enhancements introduced in comparison to the conventional Benders decomposition method. Therefore, we benchmark the conventional Benders decomposition (C-BD-C) against the proposed accelerated versions (BD-C-I). Figures 2a and 2b, illustrate the disparity in convergence between the accelerated (BD-C-I) and the conventional (C-BD-C) approaches for the NN verification problem and the OTS problem, respectively. The results demonstrate the substantial positive influence of the algorithmic modifications on both performance and convergence speed. Specifically, for NN verification, the proposed enhancements accelerated the algorithm by 61%, and in the case of OTS, the acceleration was 38%. Furthermore, it is noteworthy that the trailing effect was significantly mitigated in the BD-C-I approach.

Moreover, comparing the iterations needed for (BD-QC-II-R) with (BD-QC-I) we can view that for both NN verification and OTS there was a reduction in the number of iterations needed to converge. This happens because in every iteration we produce more cuts. However, as Tables I,II and III show, in the case of NN verification (BD-QC-II-R) methodology does worse than (BD-QC-I) in all cases, as every iteration of the former needs more time than every iteration in the latter.
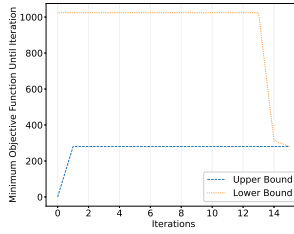
B. OTS Benchmarks

1) IEEE 6 Bus Benchmark: Initially, we conducted experiments using the IEEE 6-Bus system [33], which consists of 11 lines, to assess the effectiveness of two methodologies, (BD-QC-I) and (BD-QC-II-2), in comparison to the established benchmarks of (C-BD-C) and (SSO). For this evaluation, we executed each benchmark five times while setting the parameter E in (13g) to 5. Furthermore, the parameters $a_{cc}, e^1_{cc}, e^2_{cc}$ were set to 10. The average solution times for these instances are presented in Table I.

Notably, it is evident that (BD-QC-II-2) outperforms the other methodologies, indicating a clear quantum advantage in this context. Additionally, (BD-QC-I) appears to be superior to the conventional approach represented by (C-BD-C), but inferior of (SSO).
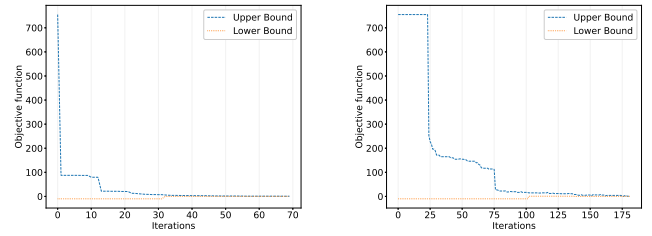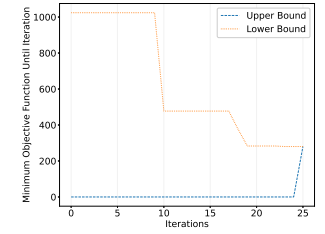

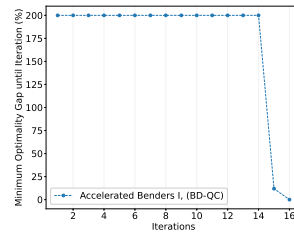(a) (BD-C-I) solving NN verification
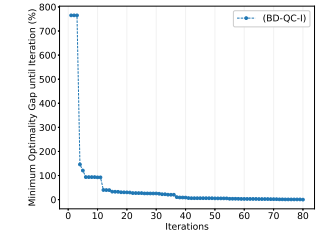(b) (C-BD-C) solving NN verification
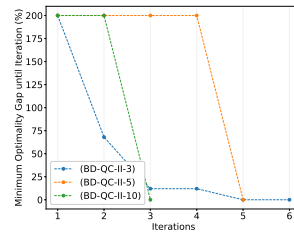

(c) (BD-C-I) solving OTS
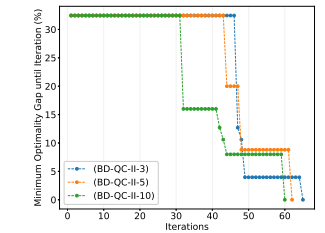(d) (C-BD-C) solving OTS

(e) (BD-QC-I) solving OTS for IEEE 14 Bus
(f) (BD-QC-I) solving NN Verification

(g) (BD-QC-II) solving OTS for IEEE 14 Bus
(h) (BD-QC-II) solving NN verification

Fig. 2

2) IEEE 14 Bus Benchmark: Assessing the scalability of the hybrid classical-quantum algorithms, we expand our numerical results, by solving larger instances of OTS, on the IEEE 14 Bus system. This evaluation involved optimizing 20 binary variables, with the parameter E set to 17. Moreover, the parameters $a_{cc}, e^1_{cc}, e^2_{cc}$ were set to 18. Table II presents the results for the OTS solutions on the 14-bus system. It is essential to highlight that all methods successfully achieved the optimal value. What is particularly striking is that some methods relying on harnessing QCs gave close results to those relying solely on conventional CPUs. In this regard, (SSO) emerges as a standout solution. Intriguingly, (BD-QC-II-5)

TABLE I: Benchmarks comparison for the Optimal Transmission Switching problem for the IEEE 6-Bus system

6-Bus OTS Results			
Method	Mean Iteration Time (s)	Iterations	Objective Value (MW)
SSO	0.281	0	280.32
C-BD-C	0.043	8	280.32
BD-QC-I	0.050	7	280.32
BD-QC-II-2	0.056	3	280.32

outperforms the other benchmarks except (SSO), which has a comparable solution time. It is worth noting that, (BD-C-II-5) outperforms its counterparts, because the others either result in unnecessary cuts or result in too many QPU calls. Therefore, parameter R became a tunable hyperparameter.

As shown in Fig.2c and Fig.2e (BD-QC-I) takes one iteration more than (BD-C-I), which can be possibly attributed to the inherent noise associated with NISQ. Finally, in Fig.2g, the behavior of the methods (BD-QC-II) is depicted.

TABLE II: Benchmarks comparison for the Optimal Transmission Switching problem for the IEEE 14-Bus system

14-Bus OTS Results			
Method	Mean Iteration Time (s)	Iterations	Objective Value (MW)
SSO	0.38	0	280.32
BD-C-I	0.04	15	280.32
BD-QC-I	0.05	16	280.32
BD-QC-II-3	0.07	6	280.32
BD-QC-II-5	0.08	5	280.32
BD-QC-II-10	0.28	3	280.32

C. NN Verification Benchmarks

Neural Network verification was more challenging to solve, as we can see from the increased number of iterations needed to converge. The tested neural networks contained 3 input neurons, and 3 output neurons alongside 2 hidden layers with 20 neurons each. The NN was trained to determine the DC-OPF result applied on the IEEE 9-bus system. Parameters $a_{cc}, e_{cc}^1, e_{cc}^2$ were set to 20. Moreover, In Fig.2f we can observe that the algorithm converges in 79 iterations, in contrast with (BD-C-I) method which only needed 70 iterations to converge. Similar to the previous test case in section IV-B, (SSO) was the best benchmark. In this instance, we can observe that (BD-QC-I) outperforms all (BD-QC-II) methods, in solution time. It becomes, thus, obvious that different decomposition techniques are most suitable for delivering the best results in different problems. Finally, parameter R is crucial for the performance enhancement of the method (BD-QC-II).

TABLE III: Benchmarks comparison for the NN Verification problem

NN Results			
Method	Mean Iteration Time (s)	Iterations	Objective Value (MW)
SSO	0.19	0	0.74
BD-C-I	0.05	69	0.74
BD-QC-I	0.06	79	0.74
BD-QC-II-3	0.09	65	0.74
BD-QC-II-5	0.12	63	0.74
BD-QC-II-10	0.16	60	0.74

V. Discussion

Besides our experiments with Quantum Annealers (such as D-Wave LEAP), which we presented in the previous Section, we also explored implementing the aforementioned simulations using circuit-based QCs by leveraging the Quantum Approximate Optimization Algorithm (QAOA). These experimental runs were conducted on IBM's latest available Eagle r3 Processor, with capacity of 127 qubits. Circuit-based QCs are considered general purpose QCs, and are expected to become one of the main quantum computing technologies in the future. In contrast, Quantum Annealers have been tailored to solve specifically optimization problems. Still, circuit-based QCs are less mature compared to the Quantum Annealers, when it comes to optimization problems, and suffer from increased average error per gate. Beyond a certain circuit depth, which is between 60-100 gates at the moment (October 2023), real circuit-based QCs become too noisy. The circuit depth required for our QAOA implementation to solve (MP-QUBO) was 500 gates, which hampered our efforts to obtain a solution. Hence, it appears that Quantum Annealing might be a more suitable alternative for solving unconstrained binary optimization problems at this stage. This claim is supported by the recent work [34], which empirically proves that Quantum Annealing technologies will constantly perform better than circuit based QAOA for combinatorial optimization problems solution. Therefore, in this work, we showed that quantum annealing can already deliver faster solutions for smaller power system sizes, and has the potential to scale to larger problems in the near future. At the same time, as a research community we must also concentrate our efforts in formulating optimization procedures with reduced circuit depth for circuit-based QCs. Considering the major efforts taking place in the quantum computing community, in the near future improved error mitigation strategies will appear which will address several of the current scalability challenges of quantum computing. The power systems community shall be ready with the suitable formulations for quantum optimization tools in order to directly exploit the potential for faster computing times.

As far as the methods we presented in this paper are concerned, besides circuit depth, one additional key factor to consider when we design a quantum optimization method is how our chosen method affects the qubit requirements. As we use Benders decomposition method, the need for qubits increases linearly with the number of constraints in (MP-Opt). This happens because we introduce integer variables such as a_1^t and a_2^k in each constraint, which in turn affects the number of qubits required, as explained in (10) to (12). Developing approaches that could arrive at similar results with a reduced number of qubits will lead both to scalable solutions and faster computing times. To facilitate these efforts, we have placed the Python code for all the presented approaches online, in the form of a tutorial, so that the research community may use this as a starting point for future improvements.

VI. Conclusion

This paper introduces a framework for transforming Mixed Integer Linear Programs (MILPs) for power system problems

to a "quantum-classical solvable form" in order to exploit the benefits of quantum optimization solvers. MILPs play a pivotal role in power system operation and planning, including e.g. unit commitment and optimal transmission switching, while they have recently emerged as a tool for trustworthy machine learning, e.g. for solving neural network verification problems for power systems. We introduce a general framework employing a hybrid "quantum-classical" approach and specifically utilizing an accelerated version of Benders decomposition. We apply this framework to two use cases: Optimal Transmission Switching and Neural Network verification for DC-OPF. We show that for small problem sizes, Quantum Annealers already achieve faster computation times than classical computing. Our experiments with larger problems in Quantum Annealers and real circuit-based QCs showed that future work needs to explore how to exploit the advantages of quantum optimization through scalable methods that require reduced circuit sizes. Therefore, to facilitate future efforts in the power systems research community, one of the contributions of this paper is that all our code, in the form of a tutorial in Python, is online available in [18].

REFERENCES

[1] A. Bhardwaj, V. K. Kamboj, V. K. Shukla, B. Singh, and P. Khurana, "Unit commitment in electrical power system-a literature review," in *2012 IEEE international power engineering and optimization conference Melaka, Malaysia*, pp. 275–280, IEEE, 2012.

[2] E. Samani and F. Aminifar, "Tri-level robust investment planning of ders in distribution networks with ac constraints," *IEEE Transactions on Power Systems*, vol. 34, no. 5, pp. 3749–3757, 2019.

[3] A. Hinneck and D. Pozo, "Optimal transmission switching problem solving:parallelization and benchmarks," 2021.

[4] A. Venzke, G. Qu, S. Low, and S. Chatzivasileiadis, "Learning optimal power flow: Worst-case guarantees for neural networks," 2020.

[5] B. Tasseff, T. Albash, Z. Morrell, M. Vuffray, A. Y. Lokhov, S. Misra, and C. Coffrin, "On the emerging potential of quantum annealing hardware for combinatorial optimization," 2022.

[6] R. Brown, D. E. B. Neira, D. Venturelli, and M. Pavone, "A copositive framework for analysis of hybrid ising-classical algorithms," 2024.

[7] Z. Zhao, L. Fan, and Z. Han, "Hybrid quantum benders decomposition for mixed-integer linear programming," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, p. 2536–2540, IEEE Press, 2022.

[8] N. G. Paterakis, "Hybrid quantum-classical multi-cut benders approach with a power system application," *Computers and Chemical Engineering*, vol. 172, p. 108161, 2023.

[9] S. Golestan, M. Habibi, S. Mousazadeh Mousavi, J. Guerrero, and J. Vasquez, "Quantum computation in power systems: An overview of recent advances," *Energy Reports*, vol. 9, pp. 584–596, 2023.

[10] B. Sævarsson, S. Chatzivasileiadis, H. Jóhannsson, and J. Østergaard, "Quantum computing for power flow algorithms: Testing on real quantum computers," 2022.

[11] F. Feng, Y. Zhou, and P. Zhang, "Quantum power flow," 2021.

[12] Y. Zhou and P. Zhang, "Noise-resilient quantum machine learning for stability assessment of power systems," *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 475–487, 2022.

[13] E. B. Jones, E. Kapit, C.-Y. Chang, D. Biagioni, D. Vaidhynathan, P. Graf, and W. Jones, "On the computational viability of quantum optimization for pmu placement," 2020.

[14] N. Nikmehr, P. Zhang, and M. A. Bragin, "Quantum distributed unit commitment: An application in microgrids," *IEEE Transactions on Power Systems*, vol. 37, no. 5, pp. 3592–3603, 2022.

[15] F. Feng, P. Zhang, Y. Zhou, and Z. Tang, "Quantum microgrid state estimation," *Electric Power Systems Research*, vol. 212, p. 108386, 2022.

[16] C. Gambella and A. Simonetto, "Multiblock admm heuristics for mixed-binary optimization on classical and quantum computers," *IEEE Transactions on Quantum Engineering*, vol. 1, pp. 1–22, 2020.

[17] R. Mahroo and A. Kargarian, "Hybrid quantum-classical unit commitment," in *2022 IEEE Texas Power and Energy Conference (TPEC)*, IEEE, Feb. 2022.

[18] "Code." https://github.com/elpetros99/HybridQuantumMILP.git.

[19] R. D. Somma, D. Nagaj, and M. Kieferová, "Quantum speedup by quantum annealing," *Physical review letters*, vol. 109, no. 5, p. 050501, 2012.

[20] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices," *Physical Review X*, vol. 10, no. 2, p. 021067, 2020.

[21] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The benders decomposition algorithm: A literature review," *European Journal of Operational Research*, vol. 259, no. 3, pp. 801–817, 2017.

[22] T. L. Magnanti and R. T. Wong, "Accelerating benders decomposition: Algorithmic enhancement and model selection criteria," *Operations research*, vol. 29, no. 3, pp. 464–484, 1981.

[23] A. Ruszczynski and A. Swietanowski, "Accelerating the regularized decomposition method for two stage stochastic linear problems," *European Journal of Operational Research*, vol. 101, no. 2, pp. 328–342, 1997.

[24] N. Papadakos, "Practical enhancements to the magnanti–wong method," *Operations Research Letters*, vol. 36, no. 4, pp. 444–449, 2008.

[25] A. Mercier, J.-F. Cordeau, and F. Soumis, "A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem," *Computers and Operations Research*, vol. 32, no. 6, pp. 1451–1476, 2005.

[26] A. Mercier and F. Soumis, "An integrated aircraft routing, crew scheduling and flight retiming model," *Computers & Operations Research*, vol. 34, no. 8, pp. 2251–2265, 2007.

[27] W.-L. Chang, R. Wong, W.-Y. Chung, Y.-H. Chen, J.-C. Chen, and A. V. Vasilakos, "Quantum speedup for the maximum cut problem," 2023.

[28] A. Mandal, A. Roy, S. Upadhyay, and H. Ushijima-Mwesigwa, "Compressed quadratization of higher order binary optimization problems," 2020.

[29] A. M. Geoffrion and G. W. Graves, "Multicommodity distribution system design by benders decomposition," *Management Science*, vol. 20, no. 5, pp. 822–844, 1974.

[30] "D-wave leap." https://cloud.dwavesys.com/leap/. Accessed on 2023-09-06.

[31] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023.

[32] "Operation and timing — d-wave system documentation documentation."

[33] A. J. W. B. F. Wollenberg and G. B. Sheblé, *Power generation, operation, and control.* 2014.

[34] E. Pelofske, A. Bärtschi, and S. Eidenbenz, *Quantum Annealing vs. QAOA: 127 Qubit Higher-Order Ising Problems on NISQ Computers*, p. 240–258. Springer Nature Switzerland, 2023.