

Master's Programme in Computer, Communication and Information Sciences

QUBO-Base Uplink Power Control in CF-mMIMO

Exploring Classical and Quantum Approaches

Vivian Phan

Author Vivian Phan

Title QUBO-Base Uplink Power Control in CF-mMIMO — Exploring Classical and Quantum Approaches

Degree programme Computer, Communication and Information Sciences

Major Machine Learning, Data Science and Artificial Intelligence

Supervisor Dr Matti Raasakka

Advisor Dr Karthik Upadhy

Collaborative partner Nokia Bell Labs

Date 29 September 2025

Number of pages 63

Language English

Abstract

Cell-Free Massive Multiple-Input Multiple-Output (CF-mMIMO) has emerged as a promising architecture for beyond-5G systems, offering gains in spectral efficiency, energy efficiency, and coverage uniformity by deploying distributed access points (APs) that jointly serve all users without cell boundaries. A key challenge in CF-mMIMO is uplink power control: under the max-min SINR fairness criterion, the problem is inherently NP-hard due to the strong coupling of user transmissions across APs.

This thesis addresses the problem by discretising user power levels and reformulating the optimisation as a Quadratic Unconstrained Binary Optimization (QUBO) model. The formulation is evaluated in a simulation framework against brute-force optimal solutions and greedy heuristics, using both classical and quantum solvers. The study highlights not only numerical performance of all the methods and solvers but also their optimality gap, allocation correctness, and simulation runtime scaling.

The results show that the QUBO approach achieves near-optimal SINR fairness with substantially better consistency than the greedy heuristic baselines, while highlighting its potential as a new paradigm that bridges classical optimisation with emerging quantum computing technologies.

Keywords optimisation , simulation , quantum computing , quantum information , wireless communication , MIMO technology , benchmarking

Preface

With deep gratitude, I thank my advisor, Karthik Upadhya, and my colleagues at Nokia Bell Labs for the opportunity, their guidance, and their immense patience. I am also grateful to my supervisor, Matti Raasakka, for his thoughtful feedback.

Finally, my heartfelt thanks to my partner and friends for accompanying me through this journey.

Espoo, 29 September 2025

Vivian Phan

Contents

| | |
|---|----------|
| Abstract | 2 |
| Preface | 3 |
| Contents | 4 |
| 1 Introduction | 6 |
| 1.1 Background and Motivation | 6 |
| 1.2 Problem Statement, Objectives, and Contributions | 7 |
| 1.3 Structure of the Thesis | 7 |
| 2 Theoretical Background | 9 |
| 2.1 Conventional Cellular Networks | 9 |
| 2.2 The Role of Massive MIMO | 9 |
| 2.2.1 Cell-free Massive MIMO | 9 |
| 2.3 Uplink Power Control in CF-mMIMO | 10 |
| 2.4 Quadratic Unconstrained Binary Optimization (QUBO) | 11 |
| 2.4.1 Overview and Relevance | 11 |
| 2.4.2 Equivalence to the Ising Model | 12 |
| 2.4.3 Mathematical Formulation | 12 |
| 2.4.4 Constraint Handling via Penalty Terms | 13 |
| 2.4.5 Unbalanced Penalization | 14 |
| 2.4.6 Solving Max-Min Problems Using QUBO and the Bisection Method | 16 |
| 2.5 Variational Quantum Algorithms (VQA) | 17 |
| 2.5.1 Cost Function | 18 |
| 2.5.2 Ansatz | 19 |
| 2.5.3 Gradients | 20 |
| 2.5.4 Optimizer | 20 |
| 2.6 Annealing | 21 |

| | | |
|----------|---|-----------|
| 2.6.1 | Simulated Annealing | 21 |
| 3 | System Models and Methods | 25 |
| 3.1 | System Model and Assumptions | 25 |
| 3.1.1 | Power control as a combinatorial optimization problem | 30 |
| 3.2 | QUBO Formulation of the Uplink Power Allocation Problem | 30 |
| 3.3 | QUBO Formulation of the Uplink Power Allocation Problem | 31 |
| 3.4 | Bisection Algorithm for Max-Min SINR Optimization | 33 |
| 3.4.1 | Classical Solvers (Gurobi, CPLEX) for QUBO | 34 |
| 3.4.2 | Variational Quantum Circuit | 35 |
| 3.5 | Greedy Power Allocation Method | 36 |
| 4 | Experimental Setups | 37 |
| 4.1 | Experimental Setup Details | 37 |
| 4.1.1 | Network Layout and Channel Model | 37 |
| 4.1.2 | Transmission Protocol and System Parameters | 38 |
| 4.1.3 | Power Levels | 38 |
| 4.1.4 | QUBO Constructions & Penalty Coefficients | 38 |
| 4.2 | Performance metrics | 40 |
| 5 | Results and Analysis | 41 |
| 5.1 | Performance Evaluation | 41 |
| 5.2 | Computational Complexity Analysis | 47 |
| 5.2.1 | Runtime Evaluation | 50 |
| 5.3 | Challenges & Discussion | 53 |
| 6 | Summary and Outlook | 56 |
| | References | 58 |

1 Introduction

1.1 Background and Motivation

The relentless growth in mobile data traffic and the demand for ubiquitous, high-quality wireless connectivity necessitate innovative network architectures [31, 46, 53]. Cell-Free Massive Multiple-Input Multiple-Output (CF-mMIMO) has emerged as a promising paradigm for beyond-5G and 6G networks. By distributing a large number of access points (APs) over a wide geographical area and jointly serving all user equipments (UEs) without explicit cell boundaries, CF-mMIMO offers substantial improvements in spectral efficiency, energy efficiency, and uniformity of service coverage [22, 27, 31, 44, 49]. These gains stem from inherent macro-diversity and strong interference mitigation capabilities.

Despite these advantages, guaranteeing fairness across users, particularly those in less favorable channel conditions, remains a critical challenge [21, 22, 49]. A widely adopted fairness criterion is the maximization of the minimum Signal-to-Interference-plus-Noise Ratio (SINR) across all users, often referred to as the **max-min SINR** problem [14, 19, 25, 39]. This metric ensures a baseline Quality of Service (QoS) while balancing resource allocation. However, the optimization problem is computationally difficult: uplink power control in CF-mMIMO is globally coupled since adjusting one UE's transmit power impacts the SINR of many others through distributed APs. Consequently, the computational burden grows rapidly with the number of UEs and APs, making the problem intractable for large-scale networks using conventional methods [22, 36].

To address this challenge, this thesis explores an alternative approach by formulating the discrete power control problem under the max-min SINR criterion as a Quadratic Unconstrained Binary Optimization (QUBO) problem. The motivation for this formulation lies in its compatibility with emerging computational paradigms, including quantum annealers and quantum-inspired algorithms such as the Quantum Approximate Optimization Algorithm (QAOA), which are designed to tackle NP-hard combinatorial optimization problems [12, 26]. These methods may offer scalability beyond classical approaches.

The choice of CF-mMIMO uplink power control as a case study is motivated by two considerations:

- The problem can be solved centrally, without requiring deployment of solvers at individual gNBs or APs.
- The optimization depends only on large-scale channel statistics due to channel hardening, reducing reliance on instantaneous CSI. This enables offloading computation to centralized cloud resources where averaged data can be used, avoiding excessive signaling overhead and mobility limitations [13, 31].

This makes CF-mMIMO power control both a practically relevant and theoretically well-suited candidate for exploring the potential of QUBO-based optimization.

1.2 Problem Statement, Objectives, and Contributions

The core problem addressed in this thesis is the design of uplink power control strategies for Cell-Free Massive MIMO systems under the *max-min SINR* criterion. This formulation seeks to maximize the minimum achievable SINR across all users, thereby ensuring fairness and a baseline Quality of Service (QoS). The challenge arises because uplink power control in CF-mMIMO is inherently a coupled optimization problem: the transmit power of each user impacts the SINR observed by many others due to the cooperative nature of distributed APs. When power levels are discretised, the problem becomes NP-hard [22, 36], making classical exact approaches computationally infeasible for realistic system sizes.

The main objective of this work is to investigate whether Quadratic Unconstrained Binary Optimization (QUBO) provides a viable framework for tackling this problem. By reformulating max-min SINR power control as a QUBO model, the study aims to leverage computational paradigms such as linear programming, simulated annealing and quantum algorithms, which are designed to address combinatorial optimization problems of this kind [12, 26]. The motivation is not only to evaluate performance relative to classical approaches but also to explore feasibility and the potential integration of quantum solutions and hardware in the longer term.

The main contributions of this thesis are as follows:

- **Problem reformulation:** The discrete uplink power control problem under the max-min SINR criterion is formulated as a QUBO model, enabling compatibility with quantum and quantum-inspired solvers.
- **Simulation framework:** A scalable environment is developed to evaluate the QUBO formulation using classical solvers (e.g., CPLEX, simulated annealing) and quantum-inspired methods (e.g., VQA).
- **Performance analysis:** The proposed approach is benchmarked against brute-force search and greedy heuristics, with comparisons in terms of SINR fairness, optimality gap, and runtime.
- **Scalability study:** Trade-offs between accuracy, problem size, and solver complexity are investigated, highlighting conditions under which QUBO-based approaches remain effective.

1.3 Structure of the Thesis

The remainder of this thesis is organized as follows:

- **Chapter 2: Theoretical Background** - Introduces the theoretical background of Cell-Free Massive MIMO and relevant knowledge regarding the mechanics behind each solver.
- **Chapter 3: Methodology** - Presents the system model, problem formulation, and the derivation of the QUBO-based approach.
- **Chapter 4: Experimental Setups** - Describes the simulation environment, parameter settings, and solver configurations used in the study.
- **Chapter 5: Results and Analysis** - Provides numerical evaluations, comparisons with baseline methods, and an assessment of performance, scalability, and limitations.
- **Chapter 6: Summary and Outlook** - Summarizes the main findings and discusses directions for extending this work, particularly toward quantum implementations.

2 Theoretical Background

2.1 Conventional Cellular Networks

Conventional mobile networks, often referred to as cellular wireless networks, are based on a cellular topology in which the coverage area is divided into distinct cells. Each cell is served by a dedicated base station (BS) responsible for managing communication with user equipment (UEs) within its region [5, 16, 51].

To meet increasing user demand for higher data rates, cellular networks have historically relied on network densification: deploying more base stations, shrinking cell sizes, and reusing spectrum more aggressively. This densification strategy, a key enabler of the gains achieved up to and including 5G, enhances spatial spectrum reuse and throughput. However, it has inherent limitations. As densification continues, inter-cell interference, cost, and spectrum constraints become major challenges. Eventually, the benefits of densification decrease, necessitating alternative architectural solutions [15].

2.2 The Role of Massive MIMO

Massive multiple-input multiple-output (massive MIMO) is a wireless communication technology that uses large arrays of antennas at each base station to serve many users simultaneously. In traditional cellular-based Massive MIMO, the infrastructure remains centralised: each user is connected to a single base station, which handles both data and control signals within its assigned coverage area.

While this approach improves spectral efficiency, it also has notable drawbacks. In particular, users at the cell edges suffer from inter-cell interference and uneven quality of service. These limitations are especially problematic in dense and heterogeneous environments [13, 31].

2.2.1 Cell-free Massive MIMO

Cell-Free Massive MIMO (CF-mMIMO) builds upon the Massive MIMO paradigm by distributing a large number of service antennas, referred to as access points (APs), across a wide geographical area [14].

Unlike traditional cellular networks, where each access point (AP) serves users within a fixed geographic cell, the APs in CF-mMIMO are not bound to any specific coverage area. Instead, they operate collaboratively, sharing synchronization and coordination information through distributed network nodes (backhaul links) to serve all users simultaneously across the entire network [13, 22, 31].

This joint transmission is carried out over the same time-frequency resources, with

the system typically operating under time-division duplexing (TDD), a scheme in which uplink and downlink transmissions occur over the same frequency band but at different time instances. This enables the exploitation of channel reciprocity, allowing the channel estimated during the uplink phase to be reused for downlink precoding [22].

For this purpose, each AP performs local channel estimation using uplink pilot signals transmitted by users. The details of the channel estimation process will be explained in 3.1. These locally obtained estimates are then used for both uplink data detection and downlink precoding, without requiring the exchange of instantaneous channel state information (CSI) between APs or with the central processing unit (CPU). The CPU's role is limited to managing slowly-varying control parameters, such as power control coefficients, and aggregating user data. [22].

This architecture eliminates entirely cell boundaries, leading to significant improvements in spectral efficiency and uniform service quality. Since each user is served by multiple geographically dispersed APs, the impact of path loss and interference is mitigated, particularly for users in traditionally underserved areas [13, 22, 31]. An illustration of how cell-free and cellular differs can be seen in Fig. 1.

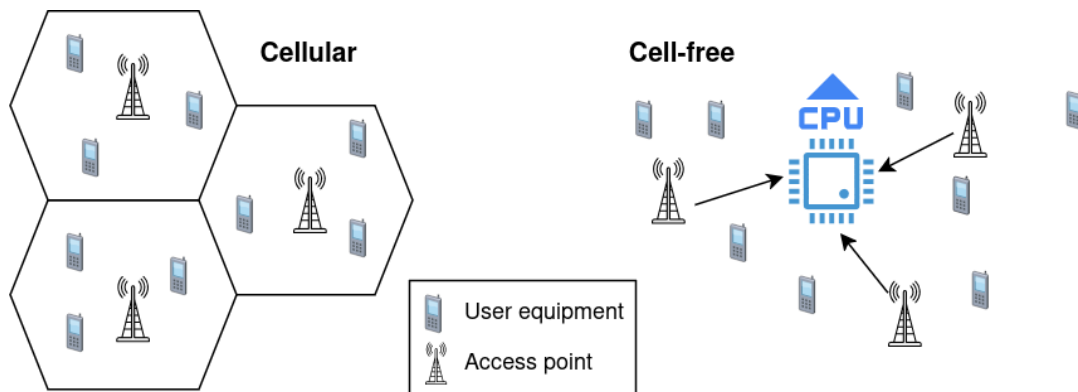


Figure 1: Comparison between conventional cellular and CF-mMIMO architectures. In the *cellular setup (left)*, each user is associated with a specific base station within a defined cell. In contrast, the *cell-free architecture (right)* eliminates cell boundaries; multiple distributed access points jointly serve all users under the coordination of a central processing unit (CPU).

2.3 Uplink Power Control in CF-mMIMO

Uplink power control refers to the process of dynamically adjusting the transmit power of user equipment (UE) when sending signals to a base station or access point (AP). The primary goal is to ensure that all users' signals arrive at the receiver with comparable strength. This uniformity facilitates accurate signal detection, minimises mutual interference, and promotes fairness by preventing stronger signals from overwhelming weaker ones. In general, power control balances the need for signal strength with

battery conservation and interference management [32, 38].

Since each UE is subject to transmit power constraints, either due to hardware limitations or regulatory policies, intelligent and adaptive power control mechanisms are crucial for maintaining overall system performance [22].

In Cell-Free Massive MIMO (CF-mMIMO) systems, uplink power control becomes even more critical due to the distributed nature of the architecture [33]. Unlike traditional cellular networks, where a user connects to a single base station, CF-mMIMO allows each user to be served simultaneously by multiple geographically distributed APs. This collaborative service model introduces new challenges in **joint power optimization across all users and APs** [22, 29, 47].

One of the key difficulties arises from the fast fluctuations of small-scale fading, which refers to rapid variations in signal amplitude and phase caused by constructive and destructive interference of multipath components. These fluctuations make real-time optimization of power control coefficients computationally demanding and potentially unstable [40].

2.4 Quadratic Unconstrained Binary Optimization (QUBO)

2.4.1 Overview and Relevance

Combinatorial optimization (CO) is a foundational area of research with extensive real-world applications spanning logistics, finance, scheduling, network design, and more [26]. It continues to be a focus of intense investigation across operations research, computer science, and data analytics due to the complexity and relevance of its problem classes.

Within this domain, the *Quadratic Unconstrained Binary Optimization* (QUBO) model [20, 26] has emerged as a unifying mathematical framework. It provides a compact representation for binary decision problems with quadratic objectives and can model a wide range of NP-hard problems. QUBO's generality makes it particularly attractive for emerging computing paradigms, including both quantum and neuromorphic architectures.

QUBO is the native format for quantum annealers such as those developed by D-Wave Systems, and it is also compatible with neuromorphic processors (e.g., IBM TrueNorth). Its practical relevance is underscored by significant industrial and governmental interest. Major technology companies like IBM, Google, Amazon, Microsoft, and Lockheed Martin, as well as national laboratories such as Los Alamos, Oak Ridge, Lawrence Livermore, and NASA Ames, are actively exploring QUBO-based optimization for both experimental and operational purposes [26].

Thanks to its unconstrained yet expressive structure, QUBO can encode a wide variety of constraints and logical rules through penalty terms. This makes it highly flexible, enabling practitioners to adapt the model to different problem domains while

maintaining compatibility with a growing ecosystem of solvers, including classical heuristics, hybrid algorithms, and quantum-inspired methods.

2.4.2 Equivalence to the Ising Model

A key reason for QUBO's broad applicability is its equivalence to the well-known *Ising model* in statistical physics. This equivalence enables the formulation of a wide range of NP-hard problems, including graph partitioning, number partitioning, set covering, satisfiability, and constrained spanning trees within the QUBO or Ising framework [12]. For example, Pakin [23] demonstrated how a shortest-path problem can be reformulated as an Ising Hamiltonian, where the ground state encodes the optimal path.

While the Ising model uses spin variables in $\{-1, +1\}^n$, it can be transformed into QUBO's binary form ($\{0, 1\}^n$) via a simple linear transformation: $x'_j = (x_j + 1)/2$. This translation makes Ising-based problems accessible to QUBO solvers and annealing algorithms, which mimic physical processes to find low-energy (optimal) states.

2.4.3 Mathematical Formulation

The QUBO (Quadratic Unconstrained Binary Optimization) model defines a binary optimisation problem with a quadratic objective function:

$$\min_{x \in \{0,1\}^n} x^\top Q x, \quad (1)$$

where $x \in \{0, 1\}^n$ is a binary vector of decision variables, and $Q \in \mathbb{R}^{n \times n}$ is a symmetric (or upper triangular) matrix encoding the cost structure of the problem. The diagonal entries of Q correspond to linear weights associated with individual variables, while the off-diagonal entries represent pairwise interaction terms between variables [26].

The objective function can be equivalently expressed as

$$x^\top Q x = \sum_{i=1}^n Q_{ii} x_i^2 + \sum_{i=1}^n \sum_{j=i+1}^n Q_{ij} x_i x_j. \quad (2)$$

Since each $x_i \in \{0, 1\}$, it follows that $x_i^2 = x_i$, and hence the diagonal terms effectively act as linear components. Therefore, the objective can be rewritten to explicitly separate linear and quadratic terms:

$$x^\top Q x = x^\top Q' x + q^\top x, \quad (3)$$

where Q' is a matrix with zeros on the diagonal (capturing only the interaction terms), and $q = \text{diag}(Q)$ contains the linear coefficients extracted from the diagonal of Q .

2.4.4 Constraint Handling via Penalty Terms

By definition, QUBO (Quadratic Unconstrained Binary Optimization) problems do not support constraints natively. Therefore, any constraints from the original problem must be incorporated into the objective function via *penalty terms* [26]. This is done by constructing quadratic penalty expressions that impose a cost on infeasible configurations, thus guiding the solution toward feasible regions [52].

In general, two types of constraints are considered:

- **Equality constraint:**

$$\sum_{i=1}^n c_i x_i = B \quad (4)$$

- **Inequality constraint:**

$$\sum_{i=1}^n l_i x_i \leq B, \quad l_i \in \mathbb{R} \quad (5)$$

To transform these constraints into the QUBO formulation, they are added to the cost function as penalty terms.

Equality Constraint An equality constraint of the form (4) can be embedded into the QUBO objective via a quadratic penalty term:

$$\lambda_0 \left(\sum_{i=1}^n c_i x_i - B \right)^2, \quad (6)$$

where $\lambda_0 > 0$ is a penalty coefficient that must be chosen sufficiently large to enforce the constraint, and $B \in \mathbb{R}$ is the target sum. This term is zero when the constraint is satisfied and grows quadratically with the violation.

This technique generalizes to a variety of linear equality and inequality constraints, and even logical conditions involving multiple binary variables. However, the choice of the penalty coefficient λ is critical: too small a value may allow constraint violations, while too large a value may overwhelm the objective, introduce numerical instability, or hinder convergence. Careful calibration or adaptive tuning strategies are often employed to balance feasibility with optimization quality.

Inequality Constraint via Slack Variable A common strategy to handle inequality constraints in QUBO formulations is to convert inequalities into equalities via *slack variables* $S \in \mathbb{Z}_{\geq 0}$ such that

$$\sum_{i=1}^n l_i x_i + S = B$$

holds [52]. This transformation turns the inequality (5) into an equality constraint suitable for QUBO encoding. A penalty term similar to (6) is then added:

$$\lambda_1 \left(\sum_{i=1}^n l_i x_i + S - B \right)^2, \quad (7)$$

where λ_1 is a penalty coefficient, and S is chosen such that the penalty is minimized when the inequality is satisfied.

To ensure that the slack variable does not exceed its required range, we define:

$$0 \leq S \leq B - \min_x \sum_{i=1}^n l_i x_i. \quad (8)$$

Binary Encoding of Slack Variable The slack variable S is represented in binary using N auxiliary binary variables $\{s_1, s_2, \dots, s_N\}$, where:

$$N = \log_2 \left(\max_x \left[B - \sum_{i=1}^n l_i x_i \right] \right) + 1. \quad (9)$$

Then the slack variable is encoded as:

$$S = \sum_{k=1}^N 2^{k-1} s_k, \quad s_k \in \{0, 1\}. \quad (10)$$

2.4.5 Unbalanced Penalization

While effective, slack variable approach significantly increases the number of binary variables. Consequently, the number of qubits and quantum operations also increases, rendering it impractical for near-term quantum hardware.

To address this issue, we adopt an alternative approach known as **unbalanced penalization** proposed by Montañez-Barrera [52]. This method incorporates inequality constraints directly into the QUBO cost function through asymmetric penalty terms, thus avoiding the need for slack variables.

Consider a generic inequality constraint of the form given by Eq. 5:

$$h(x) = C - \sum_{i=1}^n l_i x_i \geq 0, \quad (11)$$

where $h(x)$ represents a residual function. The idea is to penalize solutions that violate this constraint more severely than those that satisfy it.

To achieve this, we introduce a decay function of the form:

$$f(x) = e^{-h(x)}, \quad (12)$$

which remains close to zero when $h(x) \geq 0$, but increases rapidly when the constraint is violated (i.e., $h(x) < 0$). The behaviour of this function is illustrated in Figure 2.

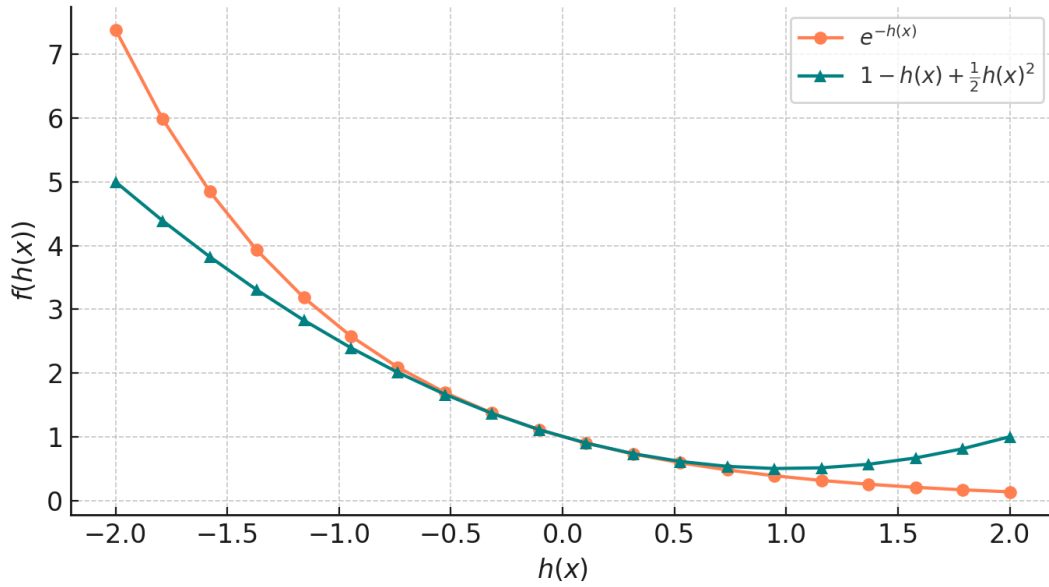


Figure 2: Exponential decay function $f(x) = e^{-h(x)}$ and its second-order Taylor approximation (dashed curve). The approximation captures the asymmetry of the penalty: negligible for feasible $h(x) \geq 0$, but steeply rising for infeasible $h(x) < 0$.

Since the exponential function cannot be directly expressed in QUBO form, we approximate it using a second-order Taylor expansion:

$$e^{-h(x)} \approx 1 - h(x) + \frac{1}{2}h(x)^2. \quad (13)$$

To generalise the penalty and allow flexibility in tuning constraint sensitivity, we rewrite the approximation with tunable parameters. The constant term can be omitted without affecting the optimisation result, as it does not influence the optimisation.

$$P_{\text{penalty}}(x) = -\lambda_1 h(x) + \lambda_2 h(x)^2, \quad (14)$$

where $\lambda_1, \lambda_2 > 0$ are penalty coefficients that can be adapted to balance the enforcement of constraints and the quality of the optimisation.

This unbalanced penalisation framework enables compact and efficient QUBO encodings for problems with inequality constraints, making it a practical alternative to slack-variable-based formulations in quantum and quantum-inspired optimisation.

2.4.6 Solving Max-Min Problems Using QUBO and the Bisection Method

As previously mentioned in 1, one common approach to ensure fairness in uplink power control is to formulate the problem as a max-min optimisation. A typical mathematical expression of this problem is

$$\max_{x \in \mathcal{X}} \min_k f_k(x),$$

where x denotes a vector of binary or discrete decision variables, $f_k(x)$ is an objective function associated with the user or constraint k , and \mathcal{X} is the feasible set. The goal is to find a configuration x that maximises the minimum performance metric across all k , thereby ensuring fairness.

This class of problems is challenging to address directly in QUBO framework due to the nested minimisation. QUBO (Eq. 1) requires a single scalar objective function expressed as a quadratic form over binary variables:

$$\min_{x \in \{0,1\}^n} x^\top Qx,$$

and it does not allow explicit constraints. Therefore, constraints must be encoded into the objective function via penalty terms, and the min-max structure must be reformulated.

To this end, the max-min problem can be transformed into a series of feasibility problems adapted from the **bisection method** [11]. The key idea is to introduce an auxiliary variable γ , which represents a threshold on the minimum performance across all users. The original problem is then equivalently reformulated as:

$$\text{Find } x \in \mathcal{X} \quad \text{such that} \quad f_k(x) \geq \gamma \quad \forall k.$$

This allows the use of bisection to iteratively search for the largest feasible γ . At each iteration, the following steps are performed:

1. A candidate value of γ is selected (typically the midpoint between the current upper and lower bounds).

2. A feasibility problem is solved to determine whether there exists a configuration $x \in \mathcal{X}$ such that all $f_k(x) \geq \gamma$.
3. If such a configuration exists, the lower bound on γ is increased; otherwise, the upper bound decreases.
4. The process is repeated until convergence to a desired precision.

To solve the feasibility problem at each step, a QUBO formulation is constructed by embedding the inequality constraints $f_k(x) \geq \gamma$ into the objective function using penalty terms as indicated in section 2.4.5, which penalises violations of the threshold. Then utilising the unbalance penalisation method, this penalty can be approximated as follows:

$$P_k(x; \gamma) \approx -\lambda_1(\gamma - f_k(x)) + \lambda_2(\gamma - f_k(x))^2,$$

The penalty coefficients $\lambda_1, \lambda_2 > 0$ are selected heuristically to control the trade-off between constraint satisfaction and optimization convergence.

Additional structural constraints, such as mutual exclusivity or one-hot encodings, are also incorporated as quadratic penalties. The final QUBO cost function at a fixed γ thus takes the form:

$$\min_{x \in \{0,1\}^n} \sum_k P_k(x; \gamma) + P_{\text{struct}}(x),$$

which is minimized using a QUBO solver (e.g., quantum annealing, simulated annealing, or classical optimizers).

This approach provides a scalable and systematic framework for addressing max-min optimization problems in binary domains, and it serves as the foundation for the uplink power control problem considered in this thesis.

2.5 Variational Quantum Algorithms (VQA)

To effectively utilize current NISQ-era quantum devices, which are limited by qubit count, restricted qubit connectivity, and both coherent and incoherent noise that constrain circuit depth, Variational Quantum Algorithms (VQAs) have emerged as a potential strategy for achieving practical performance [43]. They operate using parameterized quantum circuits, where parameters are iteratively updated by classical optimizers to approximate a solution. This variational approach can be viewed as a guided trial-and-error process. This approach has the added advantage of keeping the quantum circuit depth shallow and hence mitigating noise, in contrast to quantum algorithms developed for the fault-tolerant era.

Two important observations follow: first, because VQAs rely on heuristic methods, the optimization process may lead to solutions that are not globally optimal. Second, while VQAs do not provide exact answers, they are capable of efficiently generating approximations, even for problems that are computationally hard for classical algorithms [54].

A Variational Quantum Algorithm is typically composed of four key elements: a cost function, an ansatz, gradients, and an optimizer.

2.5.1 Cost Function

The cost function $C(\vec{\theta})$ encodes the specific objective of the problem being solved and guides the optimization process. It is generally defined as:

$$C(\vec{\theta}) = f(\rho_k, O_k, U(\vec{\theta})), \quad (15)$$

where f is a function depending on the parameter vector $\vec{\theta}$, which includes both continuous and discrete trainable variables; ρ_k denotes the input quantum states drawn from a training set; O_k represents the set of observables; and $U(\vec{\theta})$ is the parameterized quantum circuit (unitary).

To be effective within a *variational quantum algorithm (VQA)*, the cost function should satisfy the following criteria [43]:

- **Faithfulness:**
The cost function must be faithful, meaning that its minimum corresponds to a valid and meaningful solution of the target problem.
- **Efficient Estimability:**
The cost must be efficiently computable using quantum measurements, potentially combined with lightweight classical post-processing.
- **Quantum Advantage:**
The cost function should not be efficiently solvable by classical means; otherwise, the use of a quantum algorithm would offer no advantage.
- **Operational Meaningfulness:**
The value of the cost function should reflect solution quality—lower cost values should correspond to better solutions.
- **Trainability:**
The cost function should be trainable, in the sense that it allows efficient optimization of the parameter vector $\vec{\theta}$ using classical optimization routines.

2.5.2 Ansatz

In Variational Quantum Algorithms (VQAs), an *ansatz* defines the parameterized quantum circuit used to prepare trial quantum states. This circuit structure determines the expressibility of the state space, the optimization landscape, and the practical feasibility of implementation on near-term quantum devices.

For a cost function of the form Eq. 15, the parameters θ are encoded in a unitary operator $U(\theta)$ that acts on an initial quantum state. A general ansatz can be expressed as a sequence of L parameterized unitary blocks:

$$U(\theta) = U_L(\theta_L) \cdots U_2(\theta_2)U_1(\theta_1), \quad (16)$$

where each $U_l(\theta_l)$ is a unitary depending on the l -th parameter subset. A commonly used decomposition for each unitary block is:

$$U_l(\theta_l) = \prod_m e^{-i\theta_{l,m}H_m}W_m, \quad (17)$$

where H_m is a Hermitian operator and W_m is a fixed, unparameterized unitary. The subscript m indexes the individual components (e.g., gates or parameterized rotations) applied within the l -th layer. This layered structure allows for systematic construction of variational circuits using hardware-native gate sets and supports a wide variety of ansatz designs.

The general formulation in Eqs. (16) and (17) encompasses many commonly used ansätze, including hardware-efficient circuits, problem-inspired constructions, and structured ansätze. The specific implementations relevant to this thesis will be discussed in subsequent sections.

Figure 3 illustrate an example of an ansatz circuit consistent with the layered structure defined in Eq. (16).

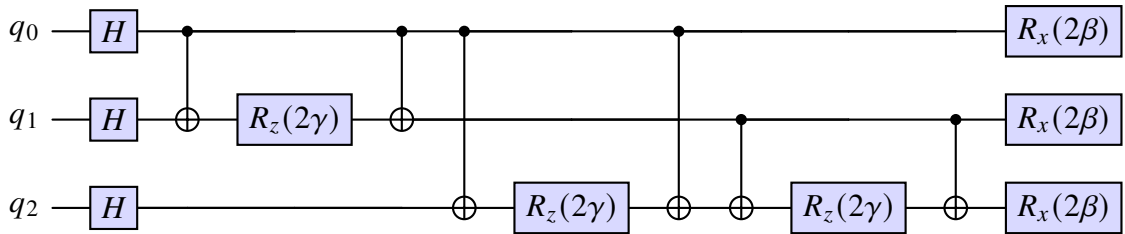


Figure 3: QAOA ansatz circuit for a 3-qubit ring MaxCut instance. The circuit prepares $|+\rangle^{\otimes 3}$, applies alternating cost unitaries generated by $\sum_{(i,j) \in E} Z_i Z_j$ (implemented via CX–RZ–CX per edge) with angles γ_ℓ , and mixer unitaries $\prod_i R_X(2\beta_\ell)$.

2.5.3 Gradients

After defining a parameterized ansatz $U(\boldsymbol{\theta})$, the optimization of the cost function relies on estimating gradients with respect to the parameters $\boldsymbol{\theta}$. Gradient-based optimization techniques, such as stochastic gradient descent or quasi-Newton methods, require evaluation of the partial derivatives $\partial C/\partial\theta_i$.

In VQAs, these gradients can be computed using quantum-compatible methods, the most common of which is the *parameter-shift rule* [43]. This method allows the exact derivative of an expectation value with respect to a gate parameter to be expressed as a linear combination of expectation values evaluated at shifted parameters:

$$\frac{\partial C(\boldsymbol{\theta})}{\partial\theta_i} = \frac{1}{2} [C(\boldsymbol{\theta}_i^+) - C(\boldsymbol{\theta}_i^-)],$$

where $\boldsymbol{\theta}_i^\pm$ denotes the parameter vector with θ_i shifted by $\pm\pi/2$, while all other parameters remain fixed.

This technique is valid when the ansatz includes gates of the form $e^{-i\theta_i P}$, where P is a Hermitian and typically Pauli operator. The parameter-shift rule is analytically exact, hardware-efficient, and avoids the numerical instability associated with finite-difference methods.

Alternative gradient estimation strategies include:

- **Finite-difference approximation:** Simple but sensitive to quantum noise and generally less robust.
- **Stochastic parameter shift and simultaneous perturbation:** More scalable for high-dimensional parameter spaces.
- **Analytic gradients via ancillary qubits:** Applicable to certain circuit structures with specialized hardware capabilities.

Accurate gradient estimation is essential for effective training of VQAs, particularly in high-dimensional parameter regimes where the optimization landscape may exhibit flat regions or barren plateaus.

2.5.4 Optimizer

In VQAs, a classical optimizer is used to iteratively update the parameter vector $\boldsymbol{\theta}$ of a quantum circuit in order to minimize a cost function. While quantum hardware is responsible for evaluating the cost through quantum measurements, it lacks the capability to adjust parameters, making classical optimization an essential component of the variational loop.

The associated optimization problems are generally considered NP-hard [42], often involving highly non-convex landscapes with many local minima. Moreover, VQAs

introduce additional challenges beyond those found in conventional classical optimization. These include the stochastic nature of function evaluations due to finite sampling, the impact of quantum hardware noise, and the phenomenon of barren plateaus, where gradients vanish exponentially with increasing system size.

Classical optimization methods used in VQAs typically fall into two main categories: gradient-based and gradient-free approaches.

2.6 Annealing

Annealing originates from metallurgy, where a material is heated to a high temperature and then slowly cooled to reduce defects and reach a stable crystalline structure [3, 9, 30]. At high temperature, atoms can move freely and explore many configurations. As the material cools, the system settles into a low-energy, ordered state.

This process provides a natural analogy for optimisation: the system energy corresponds to the objective function, and atomic configurations represent candidate solutions. Heating enables a broad exploration of the search space, while gradual cooling helps escape local minima and converge to a low-energy configuration [3].

Simulated annealing implements this idea algorithmically through probabilistic updates. Quantum annealing replaces thermal fluctuations with quantum fluctuations [8]. Based on the quantum adiabatic theorem, if a quantum system begins in the ground state of a simple Hamiltonian and the Hamiltonian is varied slowly enough, the system will remain in its instantaneous ground state [6]. By starting with a superposition of candidate states and gradually transforming the Hamiltonian into one encoding the optimisation problem, quantum annealing leverages tunnelling and quantum parallelism to guide the system toward the ground state of the problem Hamiltonian, corresponding to the optimal solution [6, 50].

Since this thesis only implements simulated annealing, the detailed theoretical background will focus on this method.

2.6.1 Simulated Annealing

Simulated Annealing (SA) is one of the most well-known meta-heuristic optimization algorithms. It is particularly suited for combinatorial and black-box optimization problems, such as Quadratic Unconstrained Binary Optimization (QUBO), where the objective function is expensive to evaluate or lacks a closed-form expression [30]. Inspired by the physical annealing process in metallurgy, SA mimics the behavior of atoms cooling and settling into a low-energy crystalline configuration [3].

In optimization, this analogy translates into searching for the global minimum of a cost function by stochastically exploring the solution space. SA accepts worse solutions with a nonzero probability to escape local minima, controlled by a temperature

parameter that gradually decreases.

Algorithmic Framework At its core, SA is a stochastic local search method based on the Metropolis-Hastings algorithm [30]. Let $\mathbf{x} \in \{0, 1\}^n$ be the current solution and $E(\mathbf{x})$ its energy (cost). At each iteration, a neighboring solution \mathbf{x}' is proposed, and the energy difference is computed:

$$\Delta E = E(\mathbf{x}') - E(\mathbf{x}) \quad (18)$$

The candidate is accepted with probability [3, 4]:

$$P_{\text{accept}} = \begin{cases} 1, & \text{if } \Delta E \leq 0 \\ \exp\left(-\frac{\Delta E}{T}\right), & \text{if } \Delta E > 0 \end{cases} \quad (19)$$

where T is the current temperature. Over time, T is decreased according to a cooling schedule. At high temperatures, the algorithm accepts worse solutions more freely, allowing broad exploration. As $T \rightarrow 0$, the algorithm becomes greedier and focuses on local refinement.

Two core theoretical principles underpin SA's effectiveness: statistical equilibrium and asymptotic convergence.

Statistical Equilibrium The concept of statistical equilibrium in SA arises from statistical mechanics and the ergodic hypothesis [1]. It assumes that over time, a system explores all accessible configurations sufficiently so that ensemble averages match time averages.

At a fixed temperature T , the probability of being in state \mathbf{x}_i with energy E_i follows the Boltzmann distribution [2, 30]:

$$\pi_T(\mathbf{x}_i) = \frac{e^{-E_i/T}}{Z_T}, \quad Z_T = \sum_j e^{-E_j/T} \quad (20)$$

This distribution favours low-energy states but retains diversity at non-zero temperatures, enabling SA to balance exploration and exploitation effectively.

Asymptotic Convergence Simulated annealing can be modeled as a non-homogeneous Markov chain $\{X_k\}$ [4], where each iteration is a stochastic transition from one solution to another. The transition probabilities at temperature T_k are defined as [37]:

$$P_{ij}(T_k) = \begin{cases} G_{ij}(T_k)A_{ij}(T_k), & i \neq j \\ 1 - \sum_{l \neq i} P_{il}(T_k), & i = j \end{cases} \quad (21)$$

where:

- $G_{ij}(T_k)$: probability of generating state j from i ,
- $A_{ij}(T_k) = \exp\left(-\frac{(f(j)-f(i))^+}{T_k}\right)$: acceptance probability using the Metropolis rule.

Under mild conditions (irreducibility and ergodicity), this Markov chain admits a stationary distribution:

$$q_i(T) = \frac{e^{-f(i)/T}}{Z(T)}, \quad Z(T) = \sum_{j \in S} e^{-f(j)/T} \quad (22)$$

As $T \rightarrow 0$, this stationary distribution $q_i(T)$ converges to a uniform distribution over the set of global optima S_{opt} [30]:

$$\lim_{T \rightarrow 0^+} q_i(T) = q_i^* = \frac{1}{|S_{\text{opt}}|} \kappa(S_{\text{opt}})(i) \quad (23)$$

where the characteristic function $\kappa(S_{\text{opt}})$ is:

$$\kappa(S_{\text{opt}})(i) = \begin{cases} 1, & i \in S_{\text{opt}} \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

This result formalises the principle of *asymptotic convergence*, stating that the algorithm will, in the limit of vanishing temperature and infinite runtime, concentrate all probability mass on the globally optimal solutions.

Practical Limitation The theoretical convergence results require:

- An infinite number of iterations at each temperature,
- An infinitely slow cooling schedule,
- Exact attainment of equilibrium at each temperature.

In real-world settings, these conditions are infeasible. Therefore, SA is typically applied as a heuristic that balances solution quality and computational cost [3, 4, 30].

However, unlike exact solvers, SA does not require gradient information or convexity assumptions, and it can be implemented efficiently for large-scale problems. While it cannot guarantee global optimality under practical constraints, its ability to escape local minima and explore vast search spaces thanks to its probabilistic acceptance of worse moves during early iterations, making it a robust method for QUBO problems and other NP-hard combinatorial tasks.

Algorithm 1 Simulated Annealing for QUBO

```
1: Input: Initial solution  $x$ , initial temperature  $T_0$ , final temperature  $T_{\min}$ , cooling
   rate  $\alpha$ , QUBO matrix  $Q$ 
2: Output: Best solution  $x^*$  found
3:  $x^* \leftarrow x$ 
4:  $E^* \leftarrow x^\top Qx$ 
5:  $T \leftarrow T_0$ 
6: while  $T > T_{\min}$  do
7:   Generate neighbor  $x'$  by flipping one or more bits of  $x$ 
8:   Compute energy:  $E' = x'^\top Qx'$ 
9:    $\Delta E \leftarrow E' - E^*$ 
10:  if  $\Delta E \leq 0$  then
11:     $x \leftarrow x'$  ▷ Always accept better solution
12:     $x^* \leftarrow x', E^* \leftarrow E'$ 
13:  else
14:    Generate  $u \sim \mathcal{U}(0, 1)$ 
15:    if  $u < \exp(-\Delta E/T)$  then
16:       $x \leftarrow x'$  ▷ Accept worse solution probabilistically
17:    end if
18:  end if
19:   $T \leftarrow \alpha \cdot T$  ▷ Update temperature
20: end while
21: return  $x^*$ 
```

3 System Models and Methods

This chapter details the system model under consideration, the mathematical formulation of the max-min SINR problem, and the subsequent transformation into a QUBO framework suitable for solution via specialized classical and quantum-inspired optimization techniques. The methodological choices, including the strategy for handling constraints via unbalanced penalization and coefficient normalization within the QUBO, are discussed, along with the iterative binary search procedure employed to optimize the SINR target.

3.1 System Model and Assumptions

To formulate and analyze the uplink max-min Signal-to-Interference-plus-Noise Ratio (SINR) power control problem, we consider a Cell-Free Massive Multiple-Input Multiple-Output (CF-mMIMO) system. The system operates in Time Division Duplex (TDD) mode [22], primarily focusing on the uplink channel estimation and subsequent data transmission phases. The key components and assumptions of our system model are detailed below:

Network Layout. The system comprises L distributed Access Points (APs) and K single-antenna User Equipments (UEs). Both APs and UEs are assumed to be randomly distributed within a defined geographical area, for instance, a two-dimensional square area. To mitigate edge effects in simulations, a wrap-around deployment model can be employed. Each AP is equipped with N_{AP} antennas. All APs are connected via a high-capacity backhaul network to a Central Processing Unit (CPU), which is responsible for coordinating signal processing tasks, including joint data detection in the uplink [14, 22, 31]. In line with the ideal cell-free concept, it is assumed that all L APs cooperatively contribute to serving all K UEs, effectively removing cell boundaries and maximizing spatial diversity for uniformly high service quality [13, 14]. Thus, for any UE k , the set of serving APs \mathcal{M}_k encompasses all L APs.

Channel Model. Wireless signals are subject to fading, reflection, diffraction, and scattering, especially in environments with buildings, walls, or moving objects. These effects cause changes in:

- Signal strength (amplitude): How much the signal is weakened (attenuated).
- Signal phase: How much the signal gets delayed or shifted in phase.

Each AP needs to estimate these changes for each UE, so it can decode received data accurately and/or beamform effectively during downlink transmission [7].

To estimate these effects, UEs send pilot signals, which are known predefined sequences. By comparing what was transmitted (the known pilot) with what was received, the AP can infer the characteristics of the channel [48]. This includes [5, 7, 10]:

- **Path loss:** The overall attenuation due to distance and environment.
- **Small-scale fading:** Rapid changes due to multipath propagation.
- **Shadowing:** Large-scale attenuation due to obstacles.

The wireless channel between UE k and AP m is modeled as a complex vector $\mathbf{g}_{k,m} \in \mathbb{C}^{N_{\text{AP}}}$, which captures both large-scale and small-scale fading effects. Specifically, the channel is expressed as:

$$\mathbf{g}_{k,m} = \sqrt{\beta_{k,m}} \mathbf{h}_{k,m}, \quad (25)$$

where $\mathbf{h}_{k,m} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_{N_{\text{AP}}})$ models the fast (small-scale) fading via an independent and identically distributed (i.i.d.) complex Gaussian vector, and $\beta_{k,m}$ is a scalar coefficient accounting for large-scale fading effects.

The large-scale fading coefficient $\beta_{k,m}$ incorporates signal attenuation due to path loss and shadowing. Path loss models the average signal decay with distance between UE k and AP m , while shadowing accounts for signal blockage from environmental obstructions, commonly modeled as a log-normal random variable. The composite large-scale fading coefficient (in linear scale) is thus given by:

$$\beta_{k,m} = 10^{-\frac{\text{PL}_{k,m}}{10}} \cdot 10^{\frac{\sigma_{\text{sh}} z_{k,m}}{10}}, \quad (26)$$

where $\text{PL}_{k,m}$ represents the path loss in dB, σ_{sh} is the standard deviation of the shadowing in dB, and $z_{k,m} \sim \mathcal{N}(0, 1)$ is a standard Gaussian random variable. The small-scale fading vector $\mathbf{h}_{k,m}$ captures rapid signal fluctuations due to multipath propagation. The assumption of $\mathbf{h}_{k,m}$ following a standard complex Gaussian distribution corresponds to a Rayleigh fading environment, typical for scenarios lacking a dominant line-of-sight path [18, 22].

Uplink Transmission Protocol. The uplink transmission occurs in two phases within each coherence block: channel estimation and data transmission.

Channel Estimation Phase. In wireless communication systems, especially those based on Massive MIMO and Cell-Free Massive MIMO (CF-mMIMO), a crucial task is for the access points (APs) to understand the wireless channels between themselves and the users. This knowledge is essential for effective data transmission and reception.

To estimate the channel, users periodically transmit known signals called pilots (or pilot signals) to the APs [13][22]. Since these signals are known in advance, the APs can compare the received signal with the original pilot and estimate the quality

and characteristics of the channel, a process known as channel estimation. Accurate channel estimation is fundamental for enabling coherent communication and allowing the APs to coordinate their transmissions effectively [31] [17].

However, there is a challenge: the number of orthogonal pilot sequences (i.e. pilots that do not interfere with each other) is limited due to practical constraints such as time and frequency resources. When there are more users than orthogonal pilots available, some users must reuse the same pilot, leading to which is called pilot contamination [13][31]. Pilot contamination occurs when the channel estimates at the APs are corrupted by interference from multiple users transmitting the same pilot, which can degrade the system performance [17].

In CF-mMIMO, this issue becomes especially important because all APs can potentially serve all users. Therefore, how pilots are assigned has a significant impact on the ability of the network to accurately estimate channels and serve users efficiently [27]. Pilot assignment strategies aim to reduce interference between users with the same pilot by assigning pilots intelligently based on user location, channel conditions, or system demands.

Channel State Information (CSI) is acquired at the APs via uplink pilot transmission. A set of τ_p pilot sequences is available. Each UE k is assigned a pilot sequence $\phi_k \in \mathbb{C}^{\tau_p}$, satisfying $\|\phi_k\|^2 = \tau_p$ (assuming pilot symbols have unit energy and are transmitted over τ_p symbol durations for energy accumulation). During this phase, all UEs transmit their assigned pilot sequences concurrently with a fixed and known pilot power, denoted as η_{pilot} .

If the number of UEs K exceeds the number of available orthogonal pilot sequences τ_p , pilot contamination occurs, where multiple UEs share the same pilot sequence. This is captured by the pilot correlation $|\phi_i^H \phi_k|^2$, which is non-zero for $i \neq k$ if they use the same or non-orthogonal pilots.

Channel estimation is carried out at the APs using the linear minimum mean-square error (LMMSE) technique [22]. During the uplink training phase of duration τ_p time-frequency symbols, each UE transmits a pilot sequence $\phi_k \in \mathbb{C}^{\tau_p}$ satisfying $\|\phi_k\|^2 = 1$. The APs use the received pilot signals to estimate the channel coefficients. If pilot sequences are not orthogonal, *pilot contamination* occurs, causing interference among users during the estimation process.

The signal received at AP m for UE k during the training phase is

$$\hat{\mathbf{y}}_{k,m} = \sqrt{\eta_k} \mathbf{g}_{k,m} + \sum_{i \neq k} \sqrt{\eta_i} \mathbf{g}_{i,m} \phi_i^H \phi_k + \tilde{\mathbf{w}}_{k,m}, \quad (27)$$

where η_k is the transmit power used by UE k for pilot transmission, and $\tilde{\mathbf{w}}_{k,m} \sim \mathcal{CN}(0, \sigma_w^2 \mathbf{I})$ denotes additive thermal noise. The second term reflects the pilot contamination effect due to non-orthogonal pilots [22].

The LMMSE estimate of the channel $\mathbf{g}_{k,m}$ is then given by

$$\widehat{\mathbf{g}}_{k,m} = \alpha_{k,m} \widehat{\mathbf{y}}_{k,m}, \quad (28)$$

where the optimal scaling factor $\alpha_{k,m}$ is

$$\alpha_{k,m} = \frac{\sqrt{\eta_k} \beta_{k,m}}{\sum_{i=1}^K \eta_i \beta_{i,m} |\boldsymbol{\phi}_i^H \boldsymbol{\phi}_k|^2 + \sigma_w^2}. \quad (29)$$

Here, σ_w^2 represents the variance of the additive white Gaussian noise per receive antenna per symbol. If $\beta_{k,m}$ is defined as a gain-over-noise ratio, σ_w^2 is effectively normalized to 1 in this expression. The term $|\boldsymbol{\phi}_i^H \boldsymbol{\phi}_k|^2 / \tau_p$, where H denotes the Hermitian conjugate, simplifies to 1 if $i = k$ and to the squared normalized correlation for $i \neq k$. For perfectly orthogonal pilots assigned to different UEs, this correlation is zero.

Data Transmission Phase. Following channel estimation, UEs transmit their data symbols x_k^{UL} (with $\mathbb{E}[|x_k^{\text{UL}}|^2] = 1$) during the remaining part of the coherence block. The CPU forms a soft estimate of the uplink data symbol x_k^{UL} transmitted by UE k using signals from all APs in \mathcal{M}_k [18]. With

$$\widehat{x}_k^{\text{UL}} = \sum_{m \in \mathcal{M}_k} \widehat{\mathbf{g}}_{k,m}^H \mathbf{y}_m, \quad (30)$$

where $\mathbf{y}_m = \sum_{j=1}^K \sqrt{\eta_j^{\text{UL}}} \mathbf{g}_{j,m} x_j^{\text{UL}} + \mathbf{w}_m$ is the received signal at AP m , \mathbf{y}_m is substituted into the combining formula and expanding the terms leads to

$$\widehat{x}_k^{\text{UL}} = \underbrace{\sum_{m \in \mathcal{M}_k} \sqrt{\eta_k^{\text{UL}}} \widehat{\mathbf{g}}_{k,m}^H \mathbf{g}_{k,m} x_k^{\text{UL}}}_{\text{desired signal}} + \underbrace{\sum_{j \neq k} \sum_{m \in \mathcal{M}_k} \sqrt{\eta_j^{\text{UL}}} \widehat{\mathbf{g}}_{k,m}^H \mathbf{g}_{j,m} x_j^{\text{UL}}}_{\text{multi-user interference}} + \underbrace{\sum_{m \in \mathcal{M}_k} \widehat{\mathbf{g}}_{k,m}^H \mathbf{w}_m}_{\text{noise}}. \quad (31)$$

where $\boldsymbol{\eta}_k^{\text{UL}}$ is the K -dimensional vector collecting the uplink transmit powers of all UEs, and $\mathbf{w}_m \sim \mathcal{CN}(\mathbf{0}, \sigma_w^2 \mathbf{I}_{N_{\text{AP}}})$ is the N_{AP} -dimensional noise vector. The first term corresponds to the desired signal from user k , the second term represents inter-user interference due to simultaneous transmissions from other users $j \neq k$, and the last term accounts for thermal noise [29]. Although the true channels $\mathbf{g}_{k,m}$ are unknown in practice, their presence in the above expression is due to this substitution and is used only for analysis. It allows the decomposition into desired signal, multi-user interference, and noise terms, and facilitates performance evaluation through expectations over the channel distribution.

Achievable Uplink Rate The Signal-to-Interference-plus-Noise Ratio (SINR) is a fundamental metric in wireless communication systems that quantifies the quality of a received signal in the presence of interference and noise. Specifically, it expresses the ratio between the power of the desired signal and the total power of the undesired components, namely, interference from other users and thermal noise.

In the uplink of a cell-free massive MIMO system, the SINR for user k is defined based on the received signal model at the central processing unit (CPU) after soft combining. Denoting the desired signal component as:

$$\text{Signal Power} = \eta_k^{\text{UL}} \left(\sum_{m \in \mathcal{M}_k} \gamma_{k,m} \right)^2,$$

where $\gamma_{k,m} = \sqrt{\eta_k} N_{\text{AP}} \alpha_{k,m} \beta_{k,m}$. This term captures the coherent power gain accumulated from all APs serving user k , where η_k^{UL} is the user's uplink transmit power and $\gamma_{k,m}$ encapsulates the channel gain after estimation [22].

The denominator of the SINR consists of three major components:

1. **Multi-user interference:** Caused by the simultaneous uplink transmission of other users $j \neq k$, whose signals are received non-orthogonally at the APs.
2. **Pilot contamination interference:** Arises when pilot sequences used for channel estimation are not perfectly orthogonal, resulting in correlated channel estimates that lead to coherent interference.
3. **Thermal noise:** Modelled as additive white Gaussian noise with variance σ_w^2 .

Combining these elements, the SINR for user k is embedded in the achievable rate expression as [22, 28, 34, 41]:

$$\text{SINR}_k = \frac{\eta_k^{\text{UL}} \left(\sum_{m \in \mathcal{M}_k} \gamma_{k,m} \right)^2}{\sum_{j=1}^K \eta_j^{\text{UL}} \sum_{m \in \mathcal{M}_k} \beta_{j,m} \gamma_{k,m} + \sum_{j=1, j \neq k}^K \eta_j^{\text{UL}} \left(\sum_{m \in \mathcal{M}_k} \gamma_{k,m} \frac{\beta_{j,m}}{\beta_{k,m}} \right)^2 |\phi_j^H \phi_k|^2 + \sigma_w^2 \sum_{m \in \mathcal{M}_k} \gamma_{k,m}}. \quad (32)$$

A high SINR implies that the desired signal dominates the interference and noise, enabling reliable decoding at higher data rates. Conversely, a low SINR indicates poor link quality, which limits the achievable data rate and increases the probability of decoding errors.

The SINR serves as a direct input to the uplink achievable rate formula:

$$R_k^{\text{UL}} = \log_2(1 + \text{SINR}_k), \quad (33)$$

where R_k^{UL} represents the maximum data rate (in bits per channel use) at which user k can reliably transmit information to the APs under current channel conditions.

3.1.1 Power control as a combinatorial optimization problem

To maximize the sum-rate, and to provide uniform service to all UEs [22] [35], the uplink power control scheme is defined as a max-min optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\eta}^{\text{UL}}} \quad & \min_{k=1,\dots,K} R_k^{\text{UL}}(\boldsymbol{\eta}^{\text{UL}}) \\ \text{s.t.} \quad & 0 \leq \eta_k^{\text{UL}} \leq P_{\max,k}^{\text{UL}}, \quad \forall k = 1, \dots, K, \end{aligned} \quad (34)$$

where $\boldsymbol{\eta}^{\text{UL}}$ is the K -dimensional vector of the uplink transmit power of all UEs, of which each UE k must select its uplink data transmit power, denoted η_k^{UL} , and $P_{\max,k}^{\text{UL}}$ is the maximum transmit power of the k -th user.

Since the function $\log(1+x)$ is a strictly increasing function for $x \geq 0$, therefore the max-min problem (Eq. 34) is equivalent to:

$$\begin{aligned} \max_{\boldsymbol{\eta}^{\text{UL}}} \quad & \min_{k=1,\dots,K} \text{SINR}_k^{\text{UL}}(\boldsymbol{\eta}^{\text{UL}}) \\ \text{s.t.} \quad & 0 \leq \eta_k^{\text{UL}} \leq P_{\max,k}^{\text{UL}}, \quad \forall k = 1, \dots, K. \end{aligned} \quad (35)$$

The SINR for user k is as noted in the Eq. 32, can be written as:

$$\text{SINR}_k = \frac{\eta_k^{\text{UL}} C_{S_k}}{\eta_k^{\text{UL}} C_{I1,k,k} + \sum_{\substack{j=1 \\ j \neq k}}^K \eta_j^{\text{UL}} (C_{I1,k,j} + C_{I2,k,j}) + N_k} = \frac{S_k(\eta_k^{\text{UL}})}{I_k(\boldsymbol{\eta}^{\text{UL}}) + N_k}, \quad (36)$$

where:

- C_{S_k} is the effective desired-signal gain for user k ,
- $C_{I1,k,k}$ is the self-interference coefficient from channel estimation error of user k ,
- $C_{I1,k,j} + C_{I2,k,j}$ is the total interference from user $j \neq k$ to user k ,
- N_k is the effective noise term after combining.

These coefficients are computed from the channel statistics and combining vectors for a given channel realization and remain constant during the optimization.

3.2 QUBO Formulation of the Uplink Power Allocation Problem

To encode the uplink max–min SINR problem into a Quadratic Unconstrained Binary Optimization (QUBO) form, we incorporate both the power selection constraints

and the SINR inequality constraints into a unified Hamiltonian. This formulation is compatible with quantum annealers and other QUBO solvers.

Power Selection and SINR Constraints

Each user $k \in \{1, \dots, K\}$ selects a transmit power $P^{(m)}$ from a discrete set of M_P predefined levels. Let $x_{k,m} \in \{0, 1\}$ be a binary variable that equals 1 if user k selects power level $P^{(m)}$, and 0 otherwise. The total transmit power for user k is:

$$\eta_k^{\text{UL}} = \sum_{m=1}^{M_P} x_{k,m} P^{(m)}.$$

The SINR constraint for each user k is given by:

$$\frac{S_k(\eta_k^{\text{UL}})}{I_k(\boldsymbol{\eta}^{\text{UL}}) + N_k} \geq t,$$

where t is a target SINR threshold, S_k is the desired signal power, I_k denotes interference, and N_k is the noise power. This can be rearranged as:

$$h_k(\mathbf{x}, t) = S_k(\eta_k^{\text{UL}}) - t \cdot [I_k(\boldsymbol{\eta}^{\text{UL}}) + N_k] \geq 0.$$

This expression is approximated in linear form using known coefficients from Eq. 36:

$$h_k(\mathbf{x}, t) = \left(\sum_{m=1}^{M_P} x_{k,m} P^{(m)} \right) (C_{S_k} - t C_{I1,k,k}) + \sum_{\substack{j=1 \\ j \neq k}}^K \left(\sum_{m=1}^{M_P} x_{j,m} P^{(m)} \right) [-t (C_{I1,k,j} + C_{I2,k,j})] - t N_k.$$

We penalize violations of this inequality using an asymmetric quadratic penalty function:

$$\text{Penalty}_k^{\text{ineq}}(\mathbf{x}, t) = -A_k \cdot h_k(\mathbf{x}, t) + B_k \cdot h_k^2(\mathbf{x}, t),$$

where $A_k, B_k > 0$ are user-specific penalty parameters controlling feasibility and numerical tractability.

3.3 QUBO Formulation of the Uplink Power Allocation Problem

To encode the uplink max–min SINR problem into a Quadratic Unconstrained Binary Optimization (QUBO) form, we incorporate both the power selection constraints and the SINR inequality constraints into a unified Hamiltonian. This formulation is compatible with quantum annealers and other QUBO solvers.

QUBO Construction

The QUBO matrix is defined as

$$Q(\mathbf{x}; t) = Q_{\text{select}}(\mathbf{x}) + Q_{\text{SINR}}(\mathbf{x}; t),$$

with components:

Power Selection Constraint Penalty Each user must select exactly one power level:

$$Q_{\text{select}}(\mathbf{x}) = \sum_{k=1}^K \Lambda_k \left(\sum_{m=1}^{M_P} x_{k,m} - 1 \right)^2,$$

where Λ_k is a large penalty weight for user k .

SINR Inequality Penalty The SINR constraint penalty term is:

$$Q_{\text{SINR}}(\mathbf{x}; t) = \sum_{k=1}^K \left[-A_k h_k(\mathbf{x}, t) + B_k h_k^2(\mathbf{x}, t) \right].$$

Each term $h_k(\mathbf{x}, t)$ can be expanded as:

$$h_k(\mathbf{x}, t) = \sum_{j=1}^K \sum_{m=1}^{M_P} \text{Coeff}_{k,j,m}(t) \cdot x_{j,m} + \text{Const}_k(t),$$

with:

$$\text{Coeff}_{k,j,m}(t) = \begin{cases} P^{(m)}(C_{S_k} - tC_{I1,k,k}), & \text{if } j = k, \\ P^{(m)}[-t(C_{I1,k,j} + C_{I2,k,j})], & \text{if } j \neq k, \end{cases} \quad \text{and} \quad \text{Const}_k(t) = -tN_k.$$

The full QUBO matrix becomes:

$$\begin{aligned} Q(\mathbf{x}; t) = & \sum_{k=1}^K \Lambda_k \left(\sum_{m=1}^{M_P} x_{k,m} - 1 \right)^2 \\ & + \sum_{k=1}^K \left[-A_k \left(\sum_{j,m} \text{Coeff}_{k,j,m} x_{j,m} + \text{Const}_k(t) \right) \right] \\ & + \sum_{k=1}^K B_k \left(\sum_{j,m} \text{Coeff}_{k,j,m} x_{j,m} + \text{Const}_k(t) \right)^2. \end{aligned}$$

The squared term contains:

- Linear terms from $\text{Coeff}_{k,j,m}^2 x_{j,m}$,
- Quadratic terms from $\text{Coeff}_{k,j_1,m_1} \cdot \text{Coeff}_{k,j_2,m_2} \cdot x_{j_1,m_1} x_{j_2,m_2}$ for $(j_1, m_1) \neq (j_2, m_2)$,
- Additional linear terms from cross products with the constant term,
- Constant terms $\text{Const}_k(t)^2$, which can be discarded in QUBO encoding.

Collecting all terms, the final QUBO objective can be written as:

$$\min_{\mathbf{x} \in \{0,1\}^n} \mathbf{x}^\top Q \mathbf{x} = x^\top Q' x + q^\top x, \quad (37)$$

where $Q \in \mathbb{R}^{n \times n}$ encodes both the quadratic interaction terms (Q') and the linear terms (q) arising from the objective of power selection and SINR-related penalties, as derived earlier in Eq. 37.

3.4 Bisection Algorithm for Max-Min SINR Optimization

To find the maximum achievable minimum SINR t under discrete power constraints, we proceed iteratively using the bisection method discussed in Sect. 2.4.6. At each iteration, a candidate threshold is chosen as the midpoint of the current interval:

$$t_{\text{candidate}} = (t_{\text{low}} + t_{\text{high}})/2.$$

A QUBO matrix is then constructed for this candidate threshold by evaluating all coefficients in the Hamiltonian $Q(\mathbf{x}; t_{\text{candidate}})$, and collecting the corresponding linear and quadratic terms to build the full cost matrix Q .

This QUBO is then solved using a suitable solver, such as CPLEX via Docplex, QAOA via Qiskit, or other hybrid methods, which returns a binary solution vector \mathbf{x}^* . The resulting solution is tested for feasibility against two criteria. First, the power selection constraint is verified by checking that each user selects exactly one power level. Second, the SINR constraint is validated by ensuring that the actual SINR achieved by each user is no less than $t_{\text{candidate}}$.

If both conditions are satisfied, the candidate threshold is considered feasible. The lower bound t_{low} is then updated to $t_{\text{candidate}}$, and \mathbf{x}^* is stored as the best-known solution. If the solution fails either constraint, the candidate is deemed infeasible and the upper bound is updated: $t_{\text{high}} \leftarrow t_{\text{candidate}}$. This process repeats until the difference $t_{\text{high}} - t_{\text{low}}$ falls below the specified convergence tolerance ε , or a maximum number of iterations is reached.

At termination, the algorithm returns the best feasible SINR threshold $t_{\text{best}} = t_{\text{low}}$, along with the corresponding optimal power assignment \mathbf{x}_{best} .

Algorithm 2 Bisection Method for Max-Min SINR via QUBO

```
1: Input: System constants, power levels  $\{P^{(M_p)}\}$ , bounds  $t_{\text{low}}, t_{\text{high}}$ , tolerance  $\varepsilon$ ,  
   penalty params  $\Lambda_k, A_k, B_k$ ,  
2: Initialize:  $\mathbf{x}_{\text{best}} \leftarrow \text{null}$   
3: while  $t_{\text{high}} - t_{\text{low}} \geq \varepsilon$ , or iteration limit not reached do  
4:    $t_{\text{candidate}} \leftarrow \frac{t_{\text{low}} + t_{\text{high}}}{2}$   
5:   Construct QUBO matrix  $Q(t_{\text{candidate}})$   
6:    $\mathbf{x}^* \leftarrow \arg \min_{\mathbf{x} \in \{0,1\}^n} \mathbf{x}^\top Q \mathbf{x}$   
7:   Check feasibility of  $\mathbf{x}^*$  at  $t_{\text{candidate}}$ :  
8:   if One power level per user and  $\text{SINR}_k \geq t_{\text{candidate}} \forall k$  then  
9:      $t_{\text{low}} \leftarrow t_{\text{candidate}}$   
10:     $\mathbf{x}_{\text{best}} \leftarrow \mathbf{x}^*$   
11:   else  
12:      $t_{\text{high}} \leftarrow t_{\text{candidate}}$   
13:   end if  
14: end while  
15: Output:  $t_{\text{best}} = t_{\text{low}}, \mathbf{x}_{\text{best}}$ 
```

3.4.1 Classical Solvers (Gurobi, CPLEX) for QUBO

To solve the QUBO-based uplink power control problem, we employ classical solvers such as **Gurobi** and **IBM CPLEX**. These solvers are widely used for large-scale combinatorial optimization tasks and are particularly effective for problems that can be expressed as *Mixed-Integer Programming (MIP)* models [45, 56].

QUBO problems, which is characterized by quadratic objective functions over binary variables, can be reformulated as MIP problems by encoding binary choices (e.g., power level selections) into integer variables and translating quadratic terms into equivalent linear constraints through auxiliary variables or solver-supported quadratic interfaces. This reformulation enables the use of MIP solvers.

Mixed-Integer Programming (MIP) MIP refers to a class of optimization problems that involve both integer and continuous decision variables. The general form of a MIP problem is:

$$\begin{aligned} \min_x \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b \\ & x_i \in \mathbb{Z}, \quad \forall i \in \mathcal{I} \\ & x_j \in \mathbb{R}, \quad \forall j \notin \mathcal{I} \end{aligned} \tag{38}$$

where:

- x is the vector of decision variables,
- \mathcal{I} is the index set of variables restricted to integer values (e.g., binary),
- A and b define the linear constraints,
- $c^\top x$ is the linear or quadratic objective function.

Both Gurobi and CPLEX provide highly optimized algorithms for solving MIP problems, including branch-and-bound, cutting planes, and heuristics. Models can be input via standard formats (e.g., .lp, .mps) or accessed through APIs such as `gurobipy` and `docplex`.

In this work, these solvers are used to obtain deterministic solutions from brute-force search to serve as performance benchmarks. Specifically, they provide upper bounds for evaluating the quality of approximate solutions generated by heuristic methods and quantum-inspired algorithms.

Despite their robustness and accuracy, classical solvers may face scalability challenges as the number of users K and the number of power levels M_P increase. This motivates the investigation of quantum and hybrid optimisation methods, which may offer more favourable computational properties for high-dimensional or highly structured instances.

3.4.2 Variational Quantum Circuit

We also solve the QUBO via a variational quantum circuit (VQC) method. The idea is to encode the QUBO Hamiltonian on n qubits, prepare a parameterized quantum state using a chosen ansatz, and train the circuit parameters with a classical optimizer to minimize the expected cost.

We map the binary variable $x_i \in \{0, 1\}$ from the QUBO formulation (Eq. 37)

$$H_{\text{Ising}} = \sum_{i < j} J_{ij} Z_i Z_j + \sum_i h_i Z_i + \text{const.} \quad (39)$$

with Pauli-Z operators Z_i acting on qubit i , and coefficients (J_{ij}, h_i) obtained from QUBO matrix Q via the binary-to-spin transformation.

The optimization problem becomes

$$\min_{\theta} \mathcal{L}(\theta) = \langle \psi(\theta) | H_{\text{Ising}} | \psi(\theta) \rangle, \quad (40)$$

where $|\psi(\theta)\rangle = U(\theta)|0\rangle^{\otimes n}$ is prepared by the chosen ansatz.

The parameters θ are updated using a classical optimizer (e.g., COBYLA or SPSA) to minimize $\mathcal{L}(\theta)$. The expectation value $\langle H_{\text{Ising}} \rangle$ is estimated from a finite number

of circuit executions (*shots*), introducing statistical noise that the optimizer must handle. After convergence, the bitstring corresponding to the lowest measured energy is selected as the solution to the QUBO.

3.5 Greedy Power Allocation Method

The Greedy Power Allocation algorithm is a heuristic method used to approximately solve the max-min SINR optimization problem in uplink power control. Its primary objective is to maximize the minimum SINR across all users by iteratively refining the transmit power vector. Given the discrete nature of power levels available to each user, this approach avoids the exponential complexity of exhaustive search by improving the current power configuration one user at a time.

Starting from an initial randomised power allocation, the algorithm evaluates each user's contribution to the minimum SINR by trying all discrete power levels for that user while holding others fixed. If an update improves the current minimum SINR (i.e., increases the worst-performing user's SINR), it is accepted. This process is repeated until no further improvement can be achieved.

Although it does not guarantee global optimality, this greedy method is computationally efficient and serves as a strong benchmark for comparing more advanced approaches, such as QUBO-based or quantum-inspired optimization techniques.

Algorithm 3 Greedy Power Allocation

```

1: Initialize:
2: Initialize power allocation  $\mathbf{P}_{\text{current}} = [p_0, p_1, \dots, p_{K-1}]$ 
3: Compute  $\text{min\_SINR}_{\text{current}} = \min_k \text{SINR}_k(\mathbf{P}_{\text{current}})$ 
4:  $\mathbf{P}_{\text{best}} \leftarrow \mathbf{P}_{\text{current}}$ 
5:  $\text{max\_min\_SINR\_so\_far} \leftarrow \text{min\_SINR}_{\text{current}}$ 
6: while improvement is found do
7:   for each user  $k = 1$  to  $K$  do
8:     for each power level  $p \in \{P^{(1)}, \dots, P^{(M_P)}\}$  do
9:       Temporarily set  $p_k = p$ , keeping other users' powers fixed
10:      Compute  $\text{min\_SINR}_{\text{current}} = \min_k \text{SINR}_k(\mathbf{P}_{\text{current}})$ 
11:      if  $\text{min\_SINR}_{\text{current}} > \text{max\_min\_SINR\_so\_far}$  then
12:        Accept this update:
13:         $\mathbf{P}_{\text{best}} \leftarrow \mathbf{P}_{\text{current}}$ 
14:         $\text{max\_min\_SINR\_so\_far} \leftarrow \text{min\_SINR}_{\text{current}}$ 
15:      end if
16:    end for
17:  end for
18: end while
19: return  $\mathbf{P}_{\text{best}}$ 

```

4 Experimental Setups

We evaluate the performance of the proposed QUBO-based power control method against brute-force search and greedy benchmarks in a simulated cell-free massive MIMO uplink scenario. The simulation environment and parameters are designed to be consistent with established models in the literature.

4.1 Experimental Setup Details

4.1.1 Network Layout and Channel Model

We consider a square coverage area of 100×100 square meters. To mitigate edge effects and emulate a larger, continuous network, a wrap-around topology is employed. In this approach, the finite simulation area is conceptually replicated in all directions, forming an infinite lattice of identical cells. This ensures that users near the edges experience the same average interference conditions as those in the centre, thus avoiding artificially favourable or unfavourable locations due to boundary effects.

Within this area, L Access Points (APs) and K single-antenna user equipments (UEs) are distributed uniformly at random for each independent setup realisation. For this study, we fix the number of APs to $L = 20$ and the number of users to $K = 4$. Each AP is equipped with $N = 4$ antennas.

The channel model incorporates both large-scale and small-scale fading, all of which are explicitly implemented in the simulation code. Large-scale fading, which includes path loss and shadow fading, is modeled as:

$$\beta_{lk} [\text{dB}] = \text{PL}_{lk} + \text{SF}_{lk}, \quad (41)$$

where PL_{lk} is the path loss and SF_{lk} is the shadow fading between the l -th AP and k -th user. The path loss is calculated using a log-distance path loss model based on a carrier frequency of $f_c = 3.5$ GHz, and the shadow fading follows a log-normal distribution with a standard deviation of $\sigma_{\text{sf}} = 8.2$ dB as defined in 3GPP documentation [24].

$$32.4 + 20 \log_{10}(f_c) + 31.9 \log_{10}(\text{distance})$$

Correlated shadowing between users and APs is included, with a spatial correlation distance of 100 meters. The small-scale fading is modeled as spatially correlated Rayleigh fading, where the spatial correlation matrix for each channel is generated using a local scattering model with an angular standard deviation (ASD) of 15 degrees around the nominal angle of arrival.

4.1.2 Transmission Protocol and System Parameters

The communication bandwidth is set to $B = 20$ MHz. The additive thermal noise power spectral density is -174 dBm/Hz, and each AP receiver front-end has a noise figure of 9 dB, resulting in a total noise power (σ_w^2) of -95 dBm.

The transmission frame structure follows a time-division duplex (TDD) protocol as previously explained in Section 2.2.1. We consider an uplink coherence block of $\tau_c = 200$ samples. The length of the orthogonal pilot sequence for channel estimation is set to $\tau_p = 20$. Since the number of users ($K = 4$) is less than τ_p , we assign orthogonal pilots to all users, ensuring there is no pilot contamination in this specific setup.

4.1.3 Power Levels

During the channel estimation phase, all users transmit their assigned pilot sequences with a fixed pilot power, $P_{\text{pilot}} = 100$ mW. The channel estimates are obtained using a Minimum Mean Square Error (MMSE) estimator.

For the data transmission phase, the power p_k for each user k is chosen from a discrete set \mathcal{P}_{set} within a power budget of $P_{\text{max}} = 100$ mW. To analyze the impact of power quantization, we evaluate three scenarios defined by the number of power levels N_p : D5 ($N_p = 5$), D10 ($N_p = 10$), and D20 ($N_p = 20$).

The set of available powers for a given scenario is uniformly discretized according to the formula:

$$\mathcal{P}_{\text{set}} = \left\{ i \cdot \frac{P_{\text{max}}}{N_p} \mid i = 1, 2, \dots, N_p \right\}.$$

In each scenario, the power levels are linearly spaced between the minimum and maximum power budget.

The objective is to find the power allocation vector that maximizes the minimum Signal-to-Interference-plus-Noise Ratio (SINR) among all K users. The SINR formulation precisely follows the structure presented in the Eq. 32 from the problem description.

4.1.4 QUBO Constructions & Penalty Coefficients

For the QUBO method, the unbalanced penalization scheme is used. A key challenge in formulating the QUBO for this problem is the potential for large variations in the magnitude of the SINR constraint terms, $h_k(X, t)$, across different users. To ensure numerical stability and balanced penalisation, each h_k term is normalised before being incorporated into the Hamiltonian.

The normalised constraint $h_{k,\text{norm}} = h_k/s_k$ is used, where the scaling factor s_k is dynamically estimated for each user k and target t . The factor s_k is chosen as the

maximum of the estimated absolute magnitudes of the signal, interference, and noise components of h_k , effectively representing the characteristic scale of the constraint. This normalisation brings all SINR constraints to a similar order of magnitude, allowing for a single set of penalty parameters ($\lambda = 1.0$, $A = 2.0$, and $B = 20.0$) to be applied effectively across all users.

Classical Solvers for QUBO We employ two classical optimization methods: IBM ILOG CPLEX and simulated annealing.

CPLEX is part of IBM’s optimization toolchain and offers native support for mixed-integer programming (MIP) formulations. Since QUBO problems can be reformulated as binary MIP models, CPLEX provides an efficient and reliable classical baseline. Its integration with Python via the `docplex` API also facilitates seamless incorporation into our experimental pipeline [60].

Simulated annealing is implemented using the `SimulatedAnnealing` sampler from the D-Wave Ocean SDK [55], which is designed to solve Ising and QUBO problems using thermal probabilistic search heuristics.

The results obtained from these classical solvers are used as baselines for comparison, providing reference points to assess other brute-force and greedy approaches, as well as other types of solver.

Quantum Solver for QUBO For the quantum approach, we employ the `EfficientSU2` circuit from Qiskit [57], which is a hardware-efficient ansatz consisting of alternating layers of parameterized single-qubit R_Y rotations and entangling gates. Specifically:

- Each layer applies $R_Y(\theta_{i,\ell})$, $R_Z(\theta_{i,\ell})$ rotations to every qubit i ,
- Followed by a chain of CX entangling gates (full entanglement across qubits),
- This pattern is repeated for a given depth L (number of repetitions).

The `EfficientSU2` ansatz is both expressive and hardware-friendly, making it suitable for near-term quantum devices. In these experiments, we used full entanglement and two layers to ensure sufficient expressivity for exploring the QUBO energy landscape.

Figure 4 illustrates the ansatz used in this implementation.

We consider two execution configurations:

1. **Noiseless simulation:** Executed on Qiskit’s `Aer_simulator` with 4096 shots [59]. The parameters θ are updated using the COBYLA optimizer with a maximum of 200 iterations.
2. **Fake backend simulation:** Executed using Qiskit’s `FakeMontrealV2` backend [58], which is a simulated quantum device model that emulates the properties of

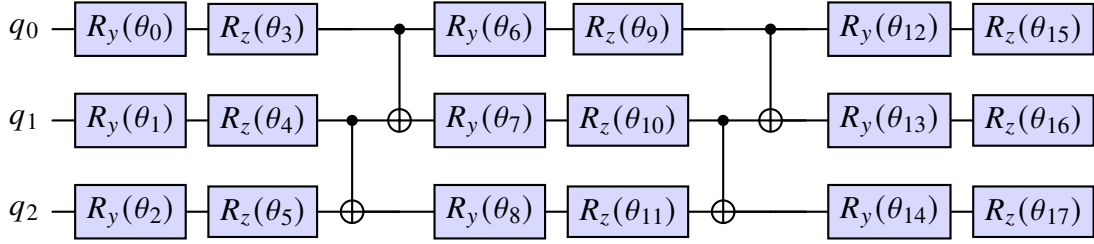


Figure 4: A hardware-efficient ansatz circuit generated using Qiskit’s `TwoLocal` class. The circuit consists of repeated layers of parameterized single-qubit rotations (R_Y and R_Z) and entangling gates (CX) applied in a linear topology. This ansatz is commonly used in variational quantum algorithms such as VQE due to its expressiveness and compatibility with near-term quantum hardware.

IBM’s `ibmq_montreal` system. The fake backend reproduces realistic hardware characteristics such as qubit connectivity, gate errors, and readout errors, but without the need for online access to the actual device. In this configuration, we use 10 shots for the estimator and 1000 shots for the sampler, combined with COBYLA for 50 iterations.

4.2 Performance metrics

To provide a comprehensive comparison between the QUBO-based method with CPLEX solver, the brute-force search, and the greedy algorithm, we evaluate their performance using the following key metrics across 1000 independent setups (random UE/AP placements) for each power discretization scenario.

1. **Max-Min SINR:** This is the primary metric for solution quality. It represents the highest possible minimum SINR that a given power allocation strategy can guarantee for all users.
2. **Consistency:** Measure how much the method results deviate from the optimal results through 1000 trials
3. **Computational Time:** We measure the total wall-clock time in seconds required for each method to find its final solution for a single setup. This metric is used to evaluate the scalability and practical feasibility of each approach.

5 Results and Analysis

5.1 Performance Evaluation

Overall Evaluation The performance of the QUBO-based approach with CPLEX solver is benchmarked against both a brute-force (optimal) baseline and a greedy heuristic across varying discretization sizes: D5, D10, and D20 (see Fig. 5). In each case, the horizontal axis denotes the minimum SINR achieved by brute-force search, while the vertical axis shows the corresponding values obtained by QUBO (red) or greedy method (green).

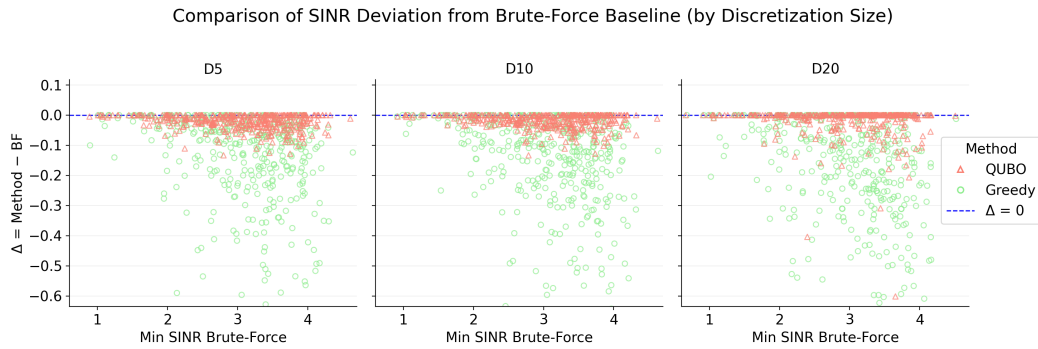


Figure 5: Comparison in deviation in minimum SINR relative to the brute-force (BF) optimum, plotted against the BF baseline and faceted by discretisation grid size (D5, D10, D20). Hollow markers show individual Monte Carlo realizations for QUBO (red triangles) and Greedy (green circles). The dashed line denotes the deviation $\Delta = 0$ (tie with BF); points below the line indicate worse-than-optimal minimum SINR.

Figure 5 shows the deviation from the brute-force optimum. Points, that lie on the dashed line, representing the optimal result, indicate that the method exactly matches the optimal minimum SINR. Those that lies underneath the line denote a shortfall.

QUBO (red triangles) concentrates tightly on optimal line for all discretisation sizes (D5, D10, D20). A large mass sits exactly on the zero line, implying that QUBO reproduces the brute-force solution in a substantial fraction of Monte Carlo realisations and is at most only slightly sub-optimal elsewhere.

Greedy (green circles) is systematically sub-optimal. Its cloud is centred below zero with a long negative tail. The dispersion grows as the discretisation grid becomes sparser (most pronounced in D20), suggesting that the greedy heuristic scales poorly as the search space enlarges.

Across discretisation sizes, QUBO is effectively optimal and robust, whereas the greedy approach suffers a consistent performance gap with occasional severe losses, especially on sparser grids. If computational budget allows, QUBO is the preferable choice when reliability near the optimum is required.

Consistency Evaluation Figure 6 presents the distribution of the optimality gap, which is defined as the percentage deviation from the brute-force solution for both the QUBO and Greedy methods under increasing grid sizes. A lower optimality gap indicates a better approximation to the optimal solution.

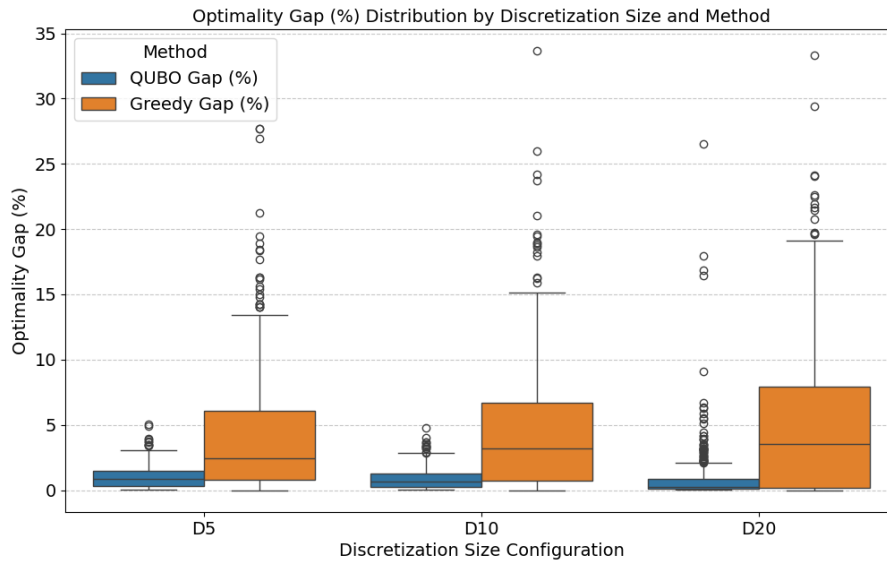


Figure 6: Distribution of the optimality gap (in %) for QUBO and greedy methods across different discretisation levels (D5, D10, D20).

Across all configurations (D5, D10, D20), the QUBO approach consistently achieves significantly lower optimality gaps. Moreover, the box plots for QUBO show a much tighter interquartile range and fewer extreme outliers, highlighting its high consistency and reliability in approximating near-optimal solutions.

In contrast, the greedy method displays a broader spread and a higher median gap, with performance degrading slightly as the problem size increases. This variability suggests that the greedy method is more susceptible to suboptimal decisions, especially in larger and more complex configurations.

These results reinforce the effectiveness of QUBO-based optimization in producing high-quality and stable solutions, particularly in resource allocation problems where robustness is essential.

Correctness Evaluation Correctness is defined as the element-wise agreement between a method’s power levels and the brute-force optimum (0–20% means very few elements matched; 100% means a perfect match).

Across all discretisation gaps, QUBO places substantially more mass in the higher-correctness bins (Fig. 7), with a pronounced spike at 100% that grows with grid

sparsity: about 30.5% (D5), 39.0% (D10) and 66.8% (D20) of set-ups are perfectly correct.

The greedy approach concentrates in the lowest bin, particularly at D5 and D10 (approx. 66% of set-ups in 0–20%), with only a modest tail towards high correctness. It improves somewhat at D20 but still lags QUBO (only 23.0% perfect).

As the grid becomes sparser, both methods improve, but the distribution shift is far stronger for QUBO: the low-correctness mass shrinks and the 100% bin dominates. Greedy retains a large share of poorly matched assignments even on the coarsest grid.

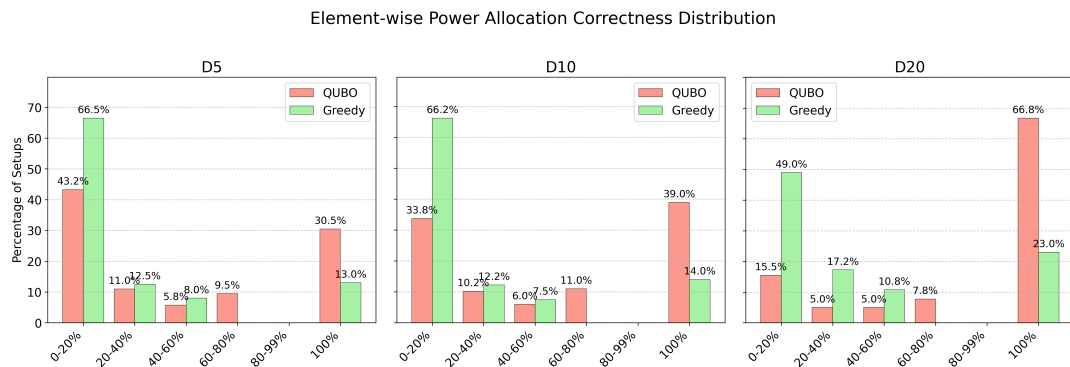


Figure 7: Element-wise power–allocation correctness relative to the BF optimum, shown as the distribution of set-ups across correctness bins. Panels correspond to discretisation gaps D5, D10 and D20 (larger D = sparser grid). Bars indicate the percentage of Monte Carlo set-ups in each bin for QUBO- and greedy method; labels above bars give exact percentages.

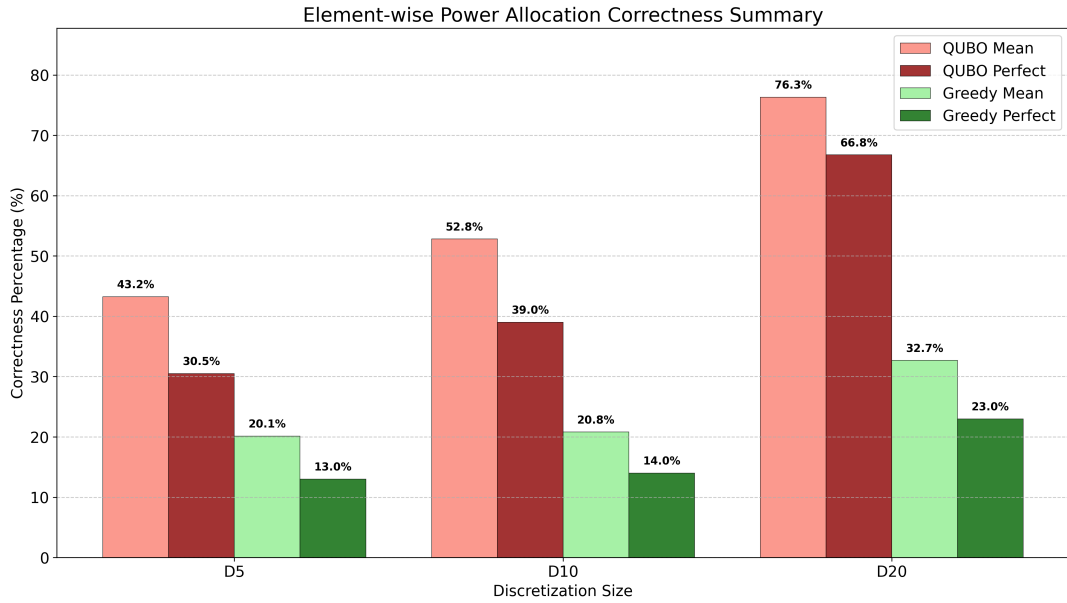


Figure 8: Summary of element-wise power-allocation correctness versus discretisation gap. “Mean” is the average percentage of elements matching the BF allocation per set-up. “Perfect” is the proportion of set-ups with 100% element-wise agreement. Results are shown for QUBO and greedy method at D5, D10 and D20 (larger D = sparser grid).

The averages (Fig. 8) confirm the distributional picture. QUBO’s mean element-wise correctness rises from 43.2% (D5) to 76.3% (D20); Greedy increases only from 20.1% to 32.7%. The perfect-match rate (100% correctness) scales steeply for QUBO, from 30.5% to 66.8%, whereas Greedy moves from 13.0% to 23.0%.

Increasing the discretisation gap (sparser grids, fewer candidate power levels) reduces the search space. QUBO capitalises on this and frequently recovers the exact BF allocation, while the greedy heuristic remains prone to local decisions that misassign many elements.

From Fig. 5 and Fig. 8, it can be seen that element-wise correctness and minimum-SINR quality are only loosely coupled. The minimum SINR is determined by the *bottleneck user*, whose SINR is minimum, while all others are *non-limiting* users with their SINR is above the minimum.

In our results, QUBO often attains close to optimal result even when correctness is moderate, showing that many allocations differing from BF still yield the same bottleneck SINR, mismatches on non-limiting users or compensating changes leave the objective unchanged. In contrast, the low correctness of the greedy method coincides more frequently with, reflecting the sensitivity to errors of the bottleneck user, although some low correctness cases still achieve near-optimal SINR. Thus, correctness is helpful for diagnosing alignment with BF decisions, but it is not a reliable proxy for SINR performance.

Comparison Between Different Solvers Figure 9 compares the performance of the three solvers across 100 independent Monte Carlo runs. For the CPLEX solver, the optimality gaps are tightly clustered around 0-1% for both D25 and D30 scenarios, with minimal spread and only a few minor outliers, indicating consistently optimal performance.

Interestingly, the simulated annealing method exhibits an even narrower spread and achieves slightly better SINR values compared to those of CPLEX. This suggests that its probabilistic exploration is highly effective for this class of problem, despite being a metaheuristic approach.

For the noiseless quantum simulator (AerSim), the median optimality gaps are in the low single digits approximately 1–2% for D30, and slightly higher for D25 with moderate variability. Most runs fall within a few percent of the optimum, though occasional outliers reach into the high single digits or low teens. This small suboptimality arises from finite measurement statistics (4096 shots) and imperfect parameter optimisation using COBYLA.

In contrast, when using the noisy quantum circuit (FakeBackend / Noisy VQC), both the median gaps and variability increase significantly. Median gaps rise to approximately 3–5%, and the spread widens considerably, with whiskers extending beyond 10% and some outliers reaching over 16%. These degradations are attributed to accumulated gate and readout errors, low shot estimation (10-shot estimator with 1000-shot sampler), and a reduced number of optimiser iterations (50), all of which compound to reduce solution quality and stability under realistic noise conditions.

Additionally, both quantum simulation setups (Sect. 4.1.4) only use two layers, thus limit the expressivity of the ansatz. It can be assumed that the accuracy of the results would improve by increasing the number of layers.

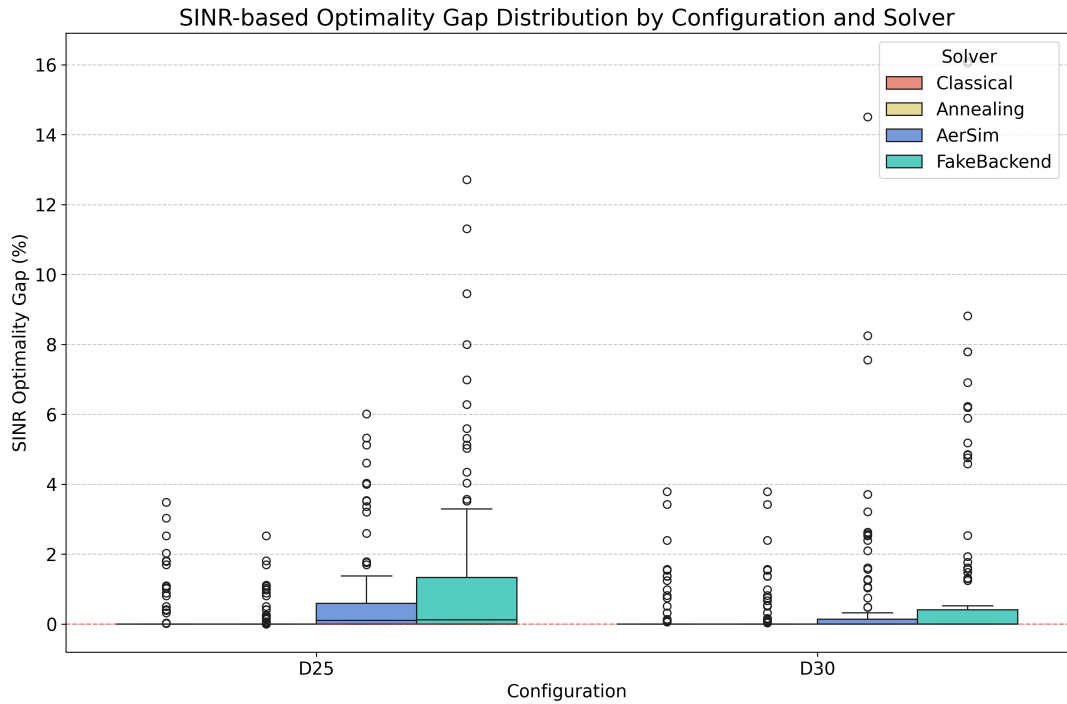


Figure 9: Optimality-gap distributions (%) for two discretization levels, D25 and D30 (larger D = sparser power grid), across three solvers. CPLEX (red) is the IBM provided solver; Annealing(yellow) is the simulated annealing process (1000 samples); AerSim (blue) is a noiseless Qiskit simulation (4096 shots, COBYLA 200 iters); FakeBackend (green) emulates realistic device noise using FakeMontrealV2 (estimator: 10 shots, sampler: 1000 shots, COBYLA 50 iters). Boxes show median and interquartile range; whiskers denote $1.5 \times \text{IQR}$, outliers are circles. The dashed red line marks zero gap to the optimal results.

Discretisation Effect Table 1 presents the effect of discretisation resolution on the mean minimum SINR achieved by three optimization methods: brute-force (BF), QUBO, and greedy method. As the power level discretization getting denser, the overall SINR values increases across all methods due to finer and more precise adjustment to the power level.

Table 1: Effect of discretisation gap on the mean minimum SINR across optimization methods.

| Nof. Variables | Method | Mean (Min SINR) |
|----------------|--------|-----------------|
| 4×5 | BF | 3.051 |
| | QUBO | 3.021 |
| | Greedy | 2.919 |
| 4×10 | BF | 3.043 |
| | QUBO | 3.023 |
| | Greedy | 2.896 |
| 4×20 | BF | 2.967 |
| | QUBO | 2.938 |
| | Greedy | 2.804 |

5.2 Computational Complexity Analysis

To evaluate the scalability and practical feasibility of the investigated power allocation strategies, we conduct an analytical assessment of their computational complexity. The primary parameters governing the problem size are the number of users (K) and the number of discrete power levels available to each user (P). The number of access points (L) primarily influences a one-time precomputation cost.

Brute-Force Search The brute-force method guarantees global optimality by exhaustively evaluating every possible power allocation vector.

- **Total complexity:** $P^K \cdot K^2$
- **Complexity:** $O(n^k)$
- **Analysis:** The total search space consists of P^K unique power allocation combinations. For each combination, the algorithm must calculate the SINR for all K users to find the minimum. The calculation of a single user's SINR involves summing interference terms from all other users, an $O(K)$ operation. Therefore, evaluating the minimum SINR for one power allocation vector requires $O(K^2)$ computations. The total complexity is the product of the search space size and the cost per evaluation.
- **Scalability:** The exponential dependence on K renders the brute-force approach computationally intractable for all but the smallest problem instances, serving primarily as a theoretical benchmark for solution quality.

Greedy Algorithm The greedy algorithm operates on an iterative best-response principle, offering a low-complexity heuristic.

- **Total Complexity:** $N_{\text{passes}} \cdot P \cdot K^3$
- **Complexity:** $O(n^3)$
- **Analysis:** The algorithm proceeds in passes (N_{passes}) over all users. In a single pass, each of the K users is considered for a power level update. To determine the best response for one user, the algorithm tests all P available power levels. For each trial power level, it must recalculate the SINRs for all K users to evaluate the new system-wide minimum SINR, which costs $O(K^2)$. Thus, one full pass over all users has a complexity of $O(K \cdot P \cdot K^2) = O(P \cdot K^3)$.
- **Scalability:** While the number of passes (N_{passes}) is not guaranteed to be a small constant, greedy algorithms often converge within a modest number of iterations in practice. Assuming N_{passes} is small or grows sub-polynomially, the overall complexity is polynomial. This polynomial scaling makes the greedy method highly scalable and suitable for large-scale problems, albeit without guarantees of finding the globally optimal solution.

QUBO-based Method The QUBO-based methodology translates the problem into a sequence of Quadratic Unconstrained Binary Optimization problems solved within a binary search framework. Its complexity is multifaceted.

- **Overall Complexity:** $N_{\text{bs}} \cdot (K^3 P^2 + T_{\text{solve}}(K \cdot P))$
- **Overall Complexity:** $O(n^n)$ in small instances, $O(c^n)$ in the worst case.
- **Analysis:** The process consists of three main stages:
 1. **Precomputation:** A one-time calculation of the constant SINR coefficients ($C_{S,k}$, $C_{I1,kj}$, etc.) has a complexity of $O(LK^2)$. This cost is amortized and typically not the bottleneck.
 2. **Binary Search Loop:** The algorithm performs N_{bs} iterations (a small, fixed number) to find the optimal SINR target t .
 3. **Per-Iteration Cost:** Within each iteration, the primary costs are:
 - *QUBO Matrix Construction:* The formulation of the Q matrix requires iterating through all pairs of the $N_{\text{vars}} = K \cdot P$ binary variables for each of the K SINR constraints. The dominant term arises from constructing the quadratic part of the Hamiltonian, resulting in a complexity of $O(K \cdot N_{\text{vars}}^2) = O(K \cdot (KP)^2) = O(K^3 P^2)$.
 - *QUBO Solving:* This is the most critical step. Solving a QUBO instance is NP-hard, with a worst-case complexity of $O(\exp(N_{\text{vars}}))$. Let $T_{\text{solve}}(N_{\text{vars}})$ be the practical time taken by the solver. State-of-the-art classical solvers (e.g., CPLEX, Gurobi) can often solve instances with hundreds of variables efficiently, but their performance remains problem-dependent and can face an exponential wall.

- **Scalability:** The method’s scalability is not dictated by the polynomial construction cost but by the practical difficulty of solving the generated QUBO instances, represented by $T_{\text{solve}}(K \cdot P)$. The growth of the number of variables, $N_{\text{vars}} = K \cdot P$, is the primary limiter.

Table 2: Comparative Summary of Computational Complexity

| Method | Complexity | Key Scaling Factor | Optimality |
|-------------|---------------------------|--|------------------------|
| Brute-Force | $O(n^k)$ | Exponential in k | Guaranteed Global |
| Greedy | $O(n^3)$ | Polynomial in K and P | Heuristic (Local) |
| QUBO | $O(T_{\text{solve}}(KP))$ | NP-hard in $N_{\text{vars}} = KP$ | Heuristic (Global aim) |

Figure 10 illustrates the computational cost scaling of various solution approaches as the number of variables increases. The data points are not empirical runtimes; they are values computed from the analytic cost models derived above.

These formulas instantiate the big- O expressions with fixed constants to produce concrete y-values for plotting. The choice of $\alpha = 1.15$ is illustrative (any $\alpha > 1$ yields exponential growth); it is selected so that the exponential curve is visible in the shown range and crosses the polynomial model within the plot. The precomputation term that depends on the number of access points L is omitted from the figure since it is a one-time cost and does not affect the comparative trends.

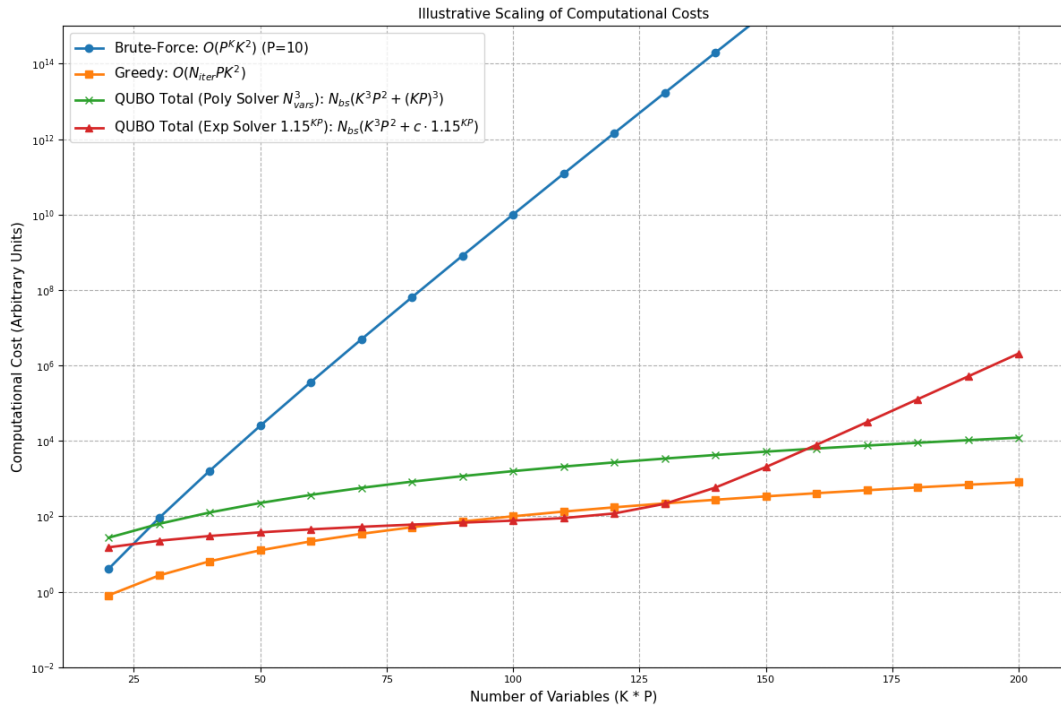


Figure 10: Illustrative scaling of computational costs (log-scale) for different solution methods as a function of problem size, represented by the number of variables $K \times P$.

The brute-force method, while yielding exact solutions, exhibits exponential complexity and becomes computationally prohibitive even for moderate problem sizes. In contrast, the greedy method maintains low computational cost across all sizes due to its iterative, heuristic nature, but this comes at the expense of suboptimal performance, as shown in earlier evaluations.

This analysis reveals a clear trade-off between computational cost and solution quality. The QUBO method serves as an intermediate approach, translating the problem into a well-studied NP-hard format where the computational burden is shifted from an explicit exponential search to the implicit, and often more manageable, complexity of a dedicated QUBO solver.

5.2.1 Runtime Evaluation

To better illustrate the computational cost, we evaluate the wall-clock runtime of different methods, as well as the runtimes associated with the various QUBO solvers. It is important to note that the quantum solvers considered here (AerSim and Fake-Backend) are simulations. Therefore, their runtimes do not correspond to execution times on actual quantum hardware. Instead, they should be interpreted as indicative reference points for assessing algorithmic complexity and relative solver behaviour in a controlled setting.

Runtime Across Methods As shown in Figure 11, moving from a denser to a sparser grid (D5, D10, D20) reduces the search space, so the Brute-Force search drops from ≈ 0.5 s to ≈ 11 ms. Greedy method is almost insensitive to the grid (≈ 1 - 1.3 ms across panels), reflecting its low polynomial cost per pass.

Runtime is dominated by QUBO construction and solver iterations. It is highest on the densest grid with D5 at approximately 3.5 min, and falls to tens of seconds for the sparser settings (D10: ≈ 14 s; D20: ≈ 19 s). The slight non-monotonic D10 to D20 change is consistent with fixed overheads and instance-to-instance variability after the big reduction from D5.

For these problem sizes, Greedy method is the fastest by far, Brute-Force can be competitive only when the grid is sparse, and the QUBO approach is the slowest under our current (classical) pipeline despite offering a unified formulation amenable to quantum or specialised solvers.

Wallclock Runtime Comparison by Method and Discretization Size

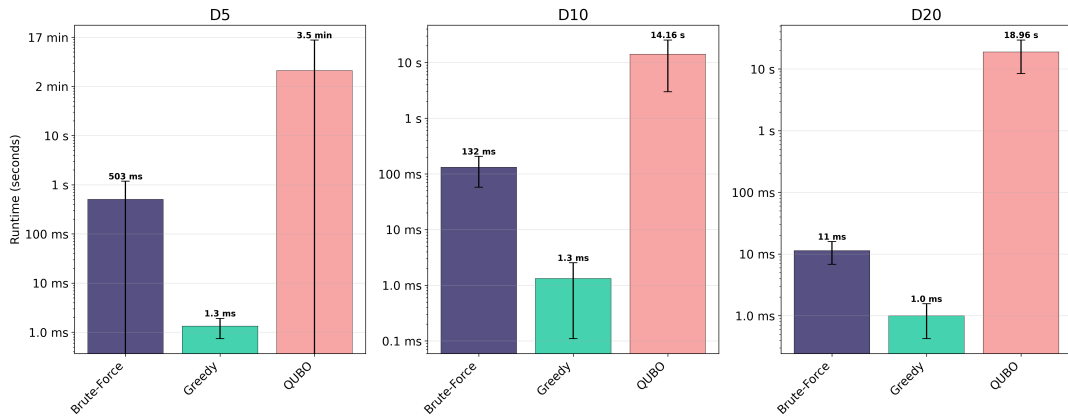


Figure 11: Wall-clock runtime by method and discretization level (D5 = densest grid, D20 = sparsest). Bars show mean runtime; error bars indicate run-to-run variability. Note the log-scale y-axis.

From the analytical time complexity model mentioned earlier, the per-iteration cost of the QUBO pipeline is dominated by the solver cost as $T_{\text{solve}}(K \times P)$, where $N_{\text{vars}} = K \times P$ and T_{solve} can grow *exponentially* with N_{vars} in the worst case.

A denser grid (D5) implies larger P and thus larger N_{vars} , placing the runtime in the regime where the $T_{\text{solve}}(K \times P)$ term dominates. This explains the larger drop in wallclock time when moving from D5 to D10: a modest reduction in P contracts N_{vars} and produces an order-of-magnitude decrease, consistent with exponential sensitivity.

By contrast, the change in runtime from D10 to D20 change is comparatively small and occasionally non-monotonic. Here the problem is already below the steep exponential region, so fixed overheads and instance-to-instance variability (e.g., binary-search depth, solver iterations, transpilation, cache/OS jitter) mask further gains. This pattern simultaneously (i) supports the exponential component of the complexity model for larger N_{vars} and (ii) shows *diminishing returns* that further coarsening from D10 only improves runtime marginally, while potentially hurting the optimality of the results.

Denser discretizations improve optimality but can inflate runtime sharply for exhaustive and QUBO-based methods. If computational time is limited, coarser grids substantially cut the computational cost for QUBO without sacrificing the quality of the solution.

Runtime Across QUBO Solvers Figure 12 compares the wallclock runtimes of the four solver methods across two problem sizes: D25 and D30. These results align with the complexity expectations discussed in Section 5.2, where the runtime cost is largely dictated by the number of binary variables and the noise/sampling overhead in quantum solvers.

Wallclock Runtime Comparison: Four Solver Methods

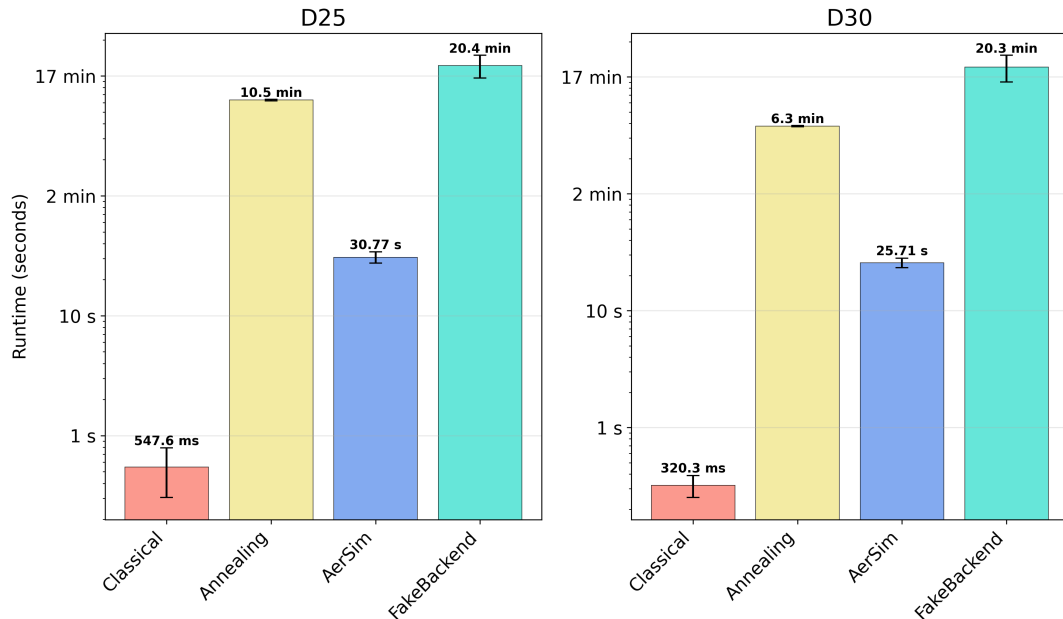


Figure 12: Wall-clock runtime comparison for the same two discretization levels (D25, D30). Bars show mean runtime; error bars indicate run-to-run variability (log-scaled y-axis).

Classical solvers such as CPLEX incur negligible runtime, completing in sub-second time for both setups (547.6 ms for D25 and 320.3 ms for D30), due to the absence of quantum-related sampling or noise estimation. Simulated annealing, although still a classical method, is notably slower, taking 10.5 minutes for D25 and 6.3 minutes for D30. This reflects the iterative nature of the algorithm and its dependence on thermal sampling.

For quantum-based solvers, the noiseless variational quantum circuit (AerSim) demonstrates moderate runtimes of approximately 30.8 (D25) and 25.7 seconds (D30). As the density of the grid increases (from D30 to D25), more binary variables are required, leading to increased sampling costs to estimate expectation values. On real quantum hardware, we would expect shorter runtimes, with circuit compilation being the dominant cost.

The FakeBackend (a hardware-mimicking VQC with realistic noise) exhibits the longest runtimes in both cases, which were around 20 minutes per instance. This is expected due to the need to simulate device characterisation rather than run on actual hardware, the low-shot readout (10-shot estimator, 1000-shot sampler), and limited optimiser iterations. Despite slightly fewer variables in D30, noise and sampling costs dominate execution time in both scenarios.

These observations emphasize the trade-off between discretization granularity and

computational cost. A denser power grid improves solution quality by reducing discretization error (as shown in Section 5.1), but it also increases circuit width, runtime, and noise sensitivity. In practice, this motivates the use of adaptive or coarse-to-fine discretization strategies, starting with a sparse grid and refining locally as needed, to balance optimality against runtime constraints.

Finally, from a solution quality perspective, simulated annealing achieves the closest approximation to the global optimum among all solvers, with CPLEX following closely. The noiseless AerSim quantum solver also performs competitively, though with small suboptimality due to finite sampling and optimizer limitations. The FakeBackend exhibits larger gaps and variability, attributable to realistic noise and limited optimizer resources. These findings suggest that improvements such as deeper circuits, more measurement shots, or error mitigation could help bridge the performance gap in hardware-based VQCs. Moreover, the strong performance of simulated annealing motivates further investigation into quantum annealing hardware as a complementary approach.

5.3 Challenges & Discussion

Classical simulation of quantum circuits is memory-bound Although the QUBO encoding grows only linearly in the number of binary variables, $N_{\text{vars}} = K \times P$ for fixed power levels P , emulating a *quantum* ansatz for this QUBO on a classical computer is quickly impractical. A statevector simulator must store 2^n complex amplitudes for an n -qubit circuit. Under the one-to-one mapping $n = N_{\text{vars}}$, this yields

$$\text{Memory (bytes)} = 2^n \times 16,$$

for double-precision complex numbers. As shown in Fig. 13, a single workstation reaches a hard wall at roughly 30–35 qubits (e.g., $n = 30$ requires ≈ 16 GiB; $n = 36$ exceeds 1 TiB). This constraint pertains only to the *classical simulation of quantum circuits*; it does not apply to classical QUBO solvers (e.g., CPLEX) or to executing quantum circuits on quantum hardware.

QUBO Resource Scaling (P=4)

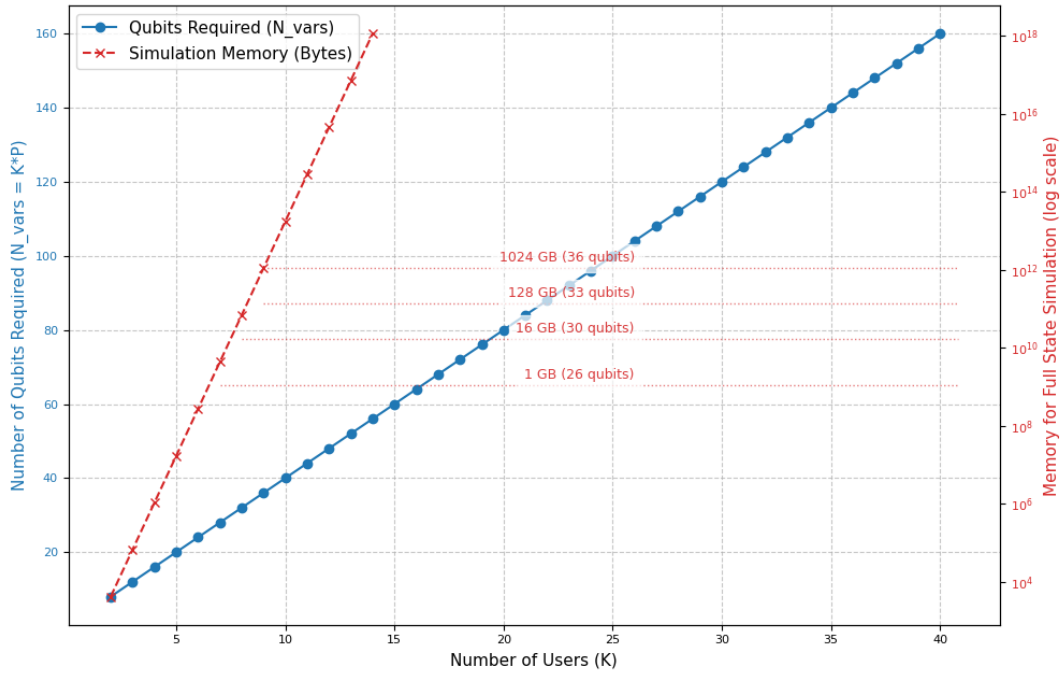


Figure 13: QUBO/quantum resource scaling with respect to the number of users K for a fixed number of power levels $P = 4$. **Left axis (blue):** number of qubits or binary variables $N_{\text{vars}} = K \times P$; this is the relevant resource when using a QUBO solver. **Right axis (red, log-scale):** RAM required by a *classical statevector simulator* if a quantum ansatz with $n = N_{\text{vars}}$ qubits were simulated, showing exponential growth and common memory thresholds.

Restricted access and operational limits of quantum hardware Access to present-day quantum processors is limited and, when available, subject to technical and operational constraints: number of qubits, queue times and shot budgets restrict the number of runs, instance sizes and hyperparameter sweeps. Calibration drift and device variability mean performance depends on when and on which qubits a circuit is executed. Transpilation overheads and limited connectivity increase effective circuit depth, and noise (finite coherence, gate/readout infidelity) caps depth and wall-clock runtime. Together, these factors materially limit the size and quality of experiments that can be executed in practice.

Scope limits for classical solver While classical QUBO solvers (e.g., CPLEX) do not face the statevector memory blow-up, their solve times grow rapidly with N_{vars} (Fig. 11). As a result, running a sufficient number of Monte Carlo replications at production scale becomes computationally costly. Our experiments therefore target smaller instances with focus on methodological rather than representative.

SINR objective and penalty formulation The QUBO formulation uses heuristic penalty weights to encode constraints, which introduces formulation-dependent bias and makes solution quality sensitive to scaling. We did not undertake a systematic penalty-tuning study as it is not within the scope of this study.

Moreover, the minimum-SINR objective is addressed via a sequence of thresholded feasibility QUBOs parameterised by a target t (e.g., within a binary search). For any given t , the ground state, if found, certifies only feasibility at that threshold; it does not by itself guarantee the global optimum over t . Misscaled penalties or solver suboptimality during the search can produce low-energy states that are infeasible or correspond to suboptimal t , so global optimality cannot be guaranteed.

6 Summary and Outlook

This thesis studied uplink power control through a QUBO encoding of the max-min SINR objective and evaluated solution quality and runtime against brute force and a greedy method baseline as well as comparing performance of different QUBO solvers for this problem. We considered discretised power grids with power levels as well as number of user and assessed performance across multiple Monte Carlo realisations using three complementary views:

- Δ vs. Brute-force (Method–Brute-force) to quantify optimality loss,
- Element-wise correctness relative to the brute-force allocation
- Wall-clock runtime.

Across various discretisation levels, the QUBO method concentrated around $\Delta \approx 0$ and often matched the optimal results exactly, while the greedy method showed substantial underperformance across all setups and poorer consistency.

Low element-wise agreement does not necessarily imply poor minimum-SINR: mismatches on non-limiting users (those with SINR above the bottleneck) or compensating changes can leave the bottleneck unchanged, explaining why QUBO can deliver $\Delta \approx 0$ with only moderate correctness.

Coarsening the grid increases exact matches for QUBO and reduces search complexity, but exhibits diminishing returns beyond roughly D10: additional coarsening yields small runtime gains and can erode optimality.

The greedy method is fastest but least accurate, while brute-force sits in the middle and accelerates on sparser grids; QUBO is slowest with a heavy-tailed distribution.

It is important to note that the reported wall-clock runtimes for quantum methods are intended to study the complexity of the problem rather than to estimate execution time on actual quantum hardware. On physical devices, the computational step is typically faster, assuming that we have direct access to the hardware. As a result, direct runtime comparison with hardware is not meaningful. Instead, the runtime results here serve as evidence of the underlying scaling behaviour and computational complexity of the problem across different solvers.

Value relative to prior work Beyond numerical results, the value of this work lies in exploring a new paradigm for the uplink power allocation problem by reformulating it within the QUBO framework. This approach opens the door to leveraging emerging quantum and quantum-inspired solvers, in contrast to the conventional convex or heuristic-based methods commonly studied in the literature.

In addition, the study contributes a benchmarking protocol for QUBO-based wireless optimisation: a bottleneck-aware performance metric, correctness diagnostics that

disentangle allocation similarity from objective value, and runtime visualisations that provide insights into the exponential complexity scaling of the problem across different methods and solvers.

Limitations and scope Experiments used reduced, stylised instances (e.g., small K and coarse discretisation) rather than production scale Massive MIMO. Classical QUBO solvers do not suffer the statevector memory blow-up, but solve times still grow steeply with N_{vars} , making large Monte Carlo studies expensive (Fig. 11). Our QUBO employs heuristic penalty weights and a threshold search on t . However, we did not perform a systematic penalty tuning study, and global optimality on t cannot be guaranteed.

On the quantum side, classical simulation of quantum circuits is memory-bound (Fig. 13), and access to real hardware is limited by queue times, shot budgets, calibration drift, connectivity/transpilation overheads, noise, and usable qubit count. These experiments provide methodological insights but do not claim transferability to full-scale deployments.

Future work To expand this work, the framework could be scaled to larger network sizes and extended to finer-grained or even continuous power control levels, which would bring the formulation closer to practical deployment scenarios.

Moreover, robustness studies should be conducted under different conditions, for example: incorporating user mobility, bursty traffic, or heterogeneous quality-of-service requirements.

Third, the penalty selection scheme used in the QUBO formulation could be made adaptive, reducing the need for manual tuning and improving performance across different problem instances.

Finally, implementation on quantum hardware represents a key long-term goal. In particular, testing on existing platforms such as D-Wave’s quantum annealers or gate-based superconducting devices available through IBM and other providers would provide valuable information on the viability of QUBO-based formulations on real quantum hardware.

References

- [1] Ludwig Boltzmann. *Vorlesungen über Gastheorie*. Original from Harvard University, digitized by the Internet Archive. Leipzig: J. A. Barth, 1896. URL: <https://archive.org/details/vorlesungenber00boltooft>.
- [2] Lev Davidovich Landau and Evgeny Mikhailovich Lifshitz. *Statistical Physics*. 3rd ed. Vol. 5. Course of Theoretical Physics. Originally published in 1976. Oxford: Pergamon Press, 1980. ISBN: 0-7506-3372-7.
- [3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by Simulated Annealing”. In: *Science* 220.4598 (1983), pp. 671–680.
- [4] D. Bertsimas and J. Tsitsiklis. “Simulated Annealing”. In: *Statistical Science* 8.1 (1993), pp. 10–15. DOI: [10.1214/ss/1177011077](https://doi.org/10.1214/ss/1177011077).
- [5] Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 1996.
- [6] Edward Farhi et al. “A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem”. In: *Science* 292.5516 (Apr. 2001), pp. 472–475. ISSN: 1095-9203. DOI: [10.1126/science.1057726](https://doi.org/10.1126/science.1057726). URL: <http://dx.doi.org/10.1126/science.1057726>.
- [7] Jerry D. Gibson, ed. *The Communications Handbook*. 2nd. CRC Press, 2002. DOI: [10.1201/9781420041163](https://doi.org/10.1201/9781420041163). URL: <https://doi.org/10.1201/9781420041163>.
- [8] Gabriel Aeppli and Thomas F. Rosenbaum. “Experiments on Quantum Annealing”. In: *Quantum Annealing and Other Optimization Methods*. Ed. by Arnab Das and Bikas K. Chakrabarti. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 157–169. ISBN: 978-3-540-31515-5. DOI: [10.1007/11526216_6](https://doi.org/10.1007/11526216_6). URL: https://doi.org/10.1007/11526216_6.
- [9] Steven E. Hughes. “Chapter 4 - Materials and Their Weldability”. In: *A Quick Guide to Welding and Weld Inspection*. Ed. by Steven E. Hughes. Woodhead Publishing Series in Welding and Other Joining Technologies. Woodhead Publishing, 2009, pp. 36–48. ISBN: 978-1-84569-641-2. DOI: <https://doi.org/10.1016/B978-1-84569-641-2.50004-0>. URL: <https://www.sciencedirect.com/science/article/pii/B9781845696412500040>.
- [10] *Multipath propagation and parameterization of its characteristics*. ITU-R Recommendation P.1407. Available at: <https://www.itu.int/rec/R-REC-P.1407/en>. ITU, 2009.
- [11] R.L. Burden and J.D. Faires. *Numerical Analysis*. Brooks/Cole, Cengage Learning, 2011. ISBN: 9780538735643. URL: <https://books.google.fi/books?id=KlfrjCDayHwC>.
- [12] Andrew Lucas. “Ising formulations of many NP-problems”. In: *Frontiers in Physics* 2 (2014). ISSN: 2296-424X. DOI: [10.3389/fphy.2014.00005](https://doi.org/10.3389/fphy.2014.00005). URL: <http://dx.doi.org/10.3389/fphy.2014.00005>.

- [13] Elina Nayebi et al. “Cell-Free Massive MIMO systems”. In: *2015 49th Asilomar Conference on Signals, Systems and Computers*. 2015, pp. 695–699. DOI: [10.1109/ACSSC.2015.7421222](https://doi.org/10.1109/ACSSC.2015.7421222).
- [14] Hien Quoc Ngo et al. “Cell-Free Massive MIMO: Uniformly great service for everyone”. In: *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. 2015, pp. 201–205. DOI: [10.1109/SPAWC.2015.7227028](https://doi.org/10.1109/SPAWC.2015.7227028).
- [15] Jeffrey G. Andrews et al. “Are we approaching the fundamental limits of wireless network densification?” In: *IEEE Communications Magazine* 54.10 (2016), pp. 184–190. DOI: [10.1109/MCOM.2016.7588290](https://doi.org/10.1109/MCOM.2016.7588290).
- [16] Guowang Miao et al. *Fundamentals of Mobile Data Networks*. Cambridge University Press, 2016. ISBN: 978-1107143210.
- [17] Elina Nayebi et al. “Performance of cell-free massive MIMO systems with MMSE and LSFDR receivers”. In: *2016 50th Asilomar Conference on Signals, Systems and Computers*. 2016, pp. 203–207. DOI: [10.1109/ACSSC.2016.7869024](https://doi.org/10.1109/ACSSC.2016.7869024).
- [18] Stefano Buzzi and Carmen D’Andrea. “Cell-Free Massive MIMO: User-Centric Approach”. In: *IEEE Wireless Communications Letters* 6.6 (2017), pp. 706–709. DOI: [10.1109/LWC.2017.2734893](https://doi.org/10.1109/LWC.2017.2734893).
- [19] Stefano Buzzi and Alessio Zappone. “Downlink power control in user-centric and cell-free massive MIMO wireless networks”. In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–6. DOI: [10.1109/PIMRC.2017.8292293](https://doi.org/10.1109/PIMRC.2017.8292293).
- [20] Mark Lewis and Fred Glover. *Quadratic Unconstrained Binary Optimization Problem Preprocessing: Theory and Empirical Analysis*. 2017. arXiv: [1705.09844](https://arxiv.org/abs/1705.09844) [cs.AI]. URL: <https://arxiv.org/abs/1705.09844>.
- [21] Elina Nayebi et al. “Precoding and Power Optimization in Cell-Free Massive MIMO Systems”. In: *IEEE Transactions on Wireless Communications* 16.7 (2017), pp. 4445–4459. DOI: [10.1109/TWC.2017.2698449](https://doi.org/10.1109/TWC.2017.2698449).
- [22] Hien Quoc Ngo et al. “Cell-Free Massive MIMO Versus Small Cells”. In: *IEEE Transactions on Wireless Communications* 16.3 (2017), pp. 1834–1850. DOI: [10.1109/TWC.2017.2655515](https://doi.org/10.1109/TWC.2017.2655515).
- [23] Scott Pakin. “Navigating a Maze using a Quantum Annealer”. In: *Proceedings of the Second International Workshop on Post Moores Era Supercomputing. PMES’17*. Denver, CO, USA: Association for Computing Machinery, 2017, pp. 30–36. ISBN: 9781450351263. DOI: [10.1145/3149526.3149532](https://doi.org/10.1145/3149526.3149532). URL: <https://doi.org/10.1145/3149526.3149532>.
- [24] 3GPP. *Study on channel model for frequencies from 0.5 to 100 GHz*. 3GPP TR 38.901. Version 15.0.0. Release 15. ETSI, July 2018. URL: https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/15.00.00_60/tr_138901v150000p.pdf.

- [25] Manijeh Bashar et al. *Enhanced Max-Min SINR for Uplink Cell-Free Massive MIMO Systems*. 2018. arXiv: [1801.10188 \[cs.IT\]](https://arxiv.org/abs/1801.10188). URL: <https://arxiv.org/abs/1801.10188>.
- [26] Fred W. Glover and Gary A. Kochenberger. “A Tutorial on Formulating QUBO Models”. In: *CoRR* abs/1811.11538 (2018). arXiv: [1811.11538](https://arxiv.org/abs/1811.11538). URL: <http://arxiv.org/abs/1811.11538>.
- [27] Hien Quoc Ngo et al. “On the Total Energy Efficiency of Cell-Free Massive MIMO”. In: *IEEE Transactions on Green Communications and Networking* 2.1 (2018), pp. 25–39. DOI: [10.1109/TGCN.2017.2770215](https://doi.org/10.1109/TGCN.2017.2770215).
- [28] E. Björnson and L. Sanguinetti. “Making Cell-Free Massive MIMO Competitive with MMSE Processing and Centralized Implementation”. In: *IEEE Transactions on Wireless Communications* 19.1 (Jan. 2019), pp. 77–90. DOI: [10.1109/TWC.2019.2949476](https://doi.org/10.1109/TWC.2019.2949476).
- [29] Carmen D’Andrea et al. “Uplink Power Control in Cell-Free Massive MIMO via Deep Learning”. In: *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. 2019, pp. 554–558. DOI: [10.1109/CAMSAP45676.2019.9022520](https://doi.org/10.1109/CAMSAP45676.2019.9022520).
- [30] Daniel Delahaye, Supatcha Chaimatanan, and Marcel Mongeau. “Simulated annealing: From basics to applications”. In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau and Jean-Yves Potvin. Vol. 272. International Series in Operations Research & Management Science (ISOR). Springer, 2019, 1–35. ISBN 978-3-319-91085-7. DOI: [10.1007/978-3-319-91086-4_1](https://doi.org/10.1007/978-3-319-91086-4_1). URL: <https://enac.hal.science/hal-01887543>.
- [31] Giovanni Interdonato et al. “Ubiquitous cell-free Massive MIMO communications”. In: *EURASIP Journal on Wireless Communications and Networking* 2019.1 (Aug. 2019), p. 197. ISSN: 1687-1499. DOI: [10.1186/s13638-019-1507-0](https://doi.org/10.1186/s13638-019-1507-0). URL: <https://doi.org/10.1186/s13638-019-1507-0>.
- [32] Chris Johnson. *5G New Radio in Bullets*. 1st. CreateSpace Independent Publishing Platform, 2019. ISBN: 978-1794567859.
- [33] Rasoul Nikbakht and Angel Lozano. “Uplink Fractional Power Control for Cell-Free Wireless Networks”. In: *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 2019, pp. 1–5. DOI: [10.1109/ICC.2019.8761096](https://doi.org/10.1109/ICC.2019.8761096).
- [34] J. Zhang et al. “Cell-Free Massive MIMO: A New Next-Generation Paradigm”. In: *IEEE Access* 7 (July 2019), pp. 99878–99888. DOI: [10.1109/ACCESS.2019.2928312](https://doi.org/10.1109/ACCESS.2019.2928312).
- [35] Xuebing Bai et al. “Sum-Rate Maximization in Cell-Free Massive MIMO with Low-Resolution ADCs and ZF Receiver”. In: *2020 IEEE 20th International Conference on Communication Technology (ICCT)*. 2020, pp. 259–263. DOI: [10.1109/ICCT50939.2020.9295959](https://doi.org/10.1109/ICCT50939.2020.9295959).

- [36] Emil Björnson and Luca Sanguinetti. “Scalable Cell-Free Massive MIMO Systems”. In: *IEEE Transactions on Communications* 68.7 (2020), pp. 4247–4261. DOI: [10.1109/TCOMM.2020.2987311](https://doi.org/10.1109/TCOMM.2020.2987311).
- [37] Pierre Brémaud. “Non-homogeneous Markov Chains”. In: *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues*. Cham: Springer International Publishing, 2020, pp. 399–422. ISBN: 978-3-030-45982-6. DOI: [10.1007/978-3-030-45982-6_12](https://doi.org/10.1007/978-3-030-45982-6_12). URL: https://doi.org/10.1007/978-3-030-45982-6_12.
- [38] ETSI TS 138 213 V16.2.0: 5G; NR; Physical layer procedures for control (3GPP TS 38.213 version 16.2.0). Tech. rep. Section 7.1.1. ETSI, 2020. URL: https://www.etsi.org/deliver/etsi_ts/138200_138299/138213/16.02.00_60/ts_138213v160200p.pdf.
- [39] Muhammad Farooq, Hien Quoc Ngo, and Le Nam Tran. *A Low-Complexity Approach for Max-Min Fairness in Uplink Cell-Free Massive MIMO*. 2020. arXiv: [2011.05076](https://arxiv.org/abs/2011.05076) [eess.SP]. URL: <https://arxiv.org/abs/2011.05076>.
- [40] Amin Ghazanfari et al. “Enhanced Fairness and Scalability of Power Control Schemes in Multi-Cell Massive MIMO”. In: *IEEE Transactions on Communications* 68.5 (2020), pp. 2878–2890. DOI: [10.1109/TCOMM.2020.2970058](https://doi.org/10.1109/TCOMM.2020.2970058).
- [41] J. Zhang et al. “Prospective Multiple Antenna Technologies for Beyond 5G”. In: *IEEE Journal on Selected Areas in Communications* 38.8 (Aug. 2020), pp. 1637–1660. DOI: [10.1109/JSAC.2020.3000407](https://doi.org/10.1109/JSAC.2020.3000407).
- [42] Lennart Bittel and Martin Kliesch. “Training Variational Quantum Algorithms Is NP-Hard”. In: *Physical Review Letters* 127.12 (Sept. 2021). ISSN: 1079-7114. DOI: [10.1103/physrevlett.127.120502](https://doi.org/10.1103/physrevlett.127.120502). URL: <http://dx.doi.org/10.1103/PhysRevLett.127.120502>.
- [43] M. Cerezo et al. “Variational quantum algorithms”. In: *Nature Reviews Physics* 3.9 (Aug. 2021), pp. 625–644. ISSN: 2522-5820. DOI: [10.1038/s42254-021-00348-9](https://doi.org/10.1038/s42254-021-00348-9). URL: <http://dx.doi.org/10.1038/s42254-021-00348-9>.
- [44] Özlem Tugfe Demir, Emil Björnson, and Luca Sanguinetti. “Foundations of User-Centric Cell-Free Massive MIMO”. In: *Foundations and Trends® in Signal Processing* 14.3–4 (2021), pp. 162–472. ISSN: 1932-8354. DOI: [10.1561/2000000109](https://doi.org/10.1561/2000000109). URL: <http://dx.doi.org/10.1561/2000000109>.
- [45] IBM. *IBM CPLEX Optimizer for z/OS*. IBM Documentation — CPLEX Optimizer for z/OS 12.10.0. 2021. URL: <https://www.ibm.com/docs/en/cofz/12.10.0?topic=mc-what-is-cplex> (visited on 09/29/2025).
- [46] Ki-Hong Park, Mohamed-Slim Alouini, and Yunfei Chen. *On-Demand Networking for Ubiquitous Connectivity and Network Resilience: A Network-in-a-Box Solution*. 2021. arXiv: [2110.01726](https://arxiv.org/abs/2110.01726) [cs.NI]. URL: <https://arxiv.org/abs/2110.01726>.

- [47] Yongshun Zhang et al. “Deep Learning-Based Power Control for Uplink Cell-Free Massive MIMO Systems”. In: *2021 IEEE Global Communications Conference (GLOBECOM)*. 2021, pp. 1–6. DOI: [10.1109/GLOBECOM46510.2021.9685827](https://doi.org/10.1109/GLOBECOM46510.2021.9685827).
- [48] Shuaifei Chen et al. “A survey on user-centric cell-free massive MIMO systems”. In: *Digital Communications and Networks* 8.5 (2022), pp. 695–719. ISSN: 2352-8648. DOI: <https://doi.org/10.1016/j.dcan.2021.12.005>. URL: <https://www.sciencedirect.com/science/article/pii/S2352864821001024>.
- [49] Trang C. Mai, Hien Quoc Ngo, and Le-Nam Tran. “Energy Efficiency Maximization in Large-Scale Cell-Free Massive MIMO: A Projected Gradient Approach”. In: *IEEE Transactions on Wireless Communications* 21.8 (2022), pp. 6357–6371. DOI: [10.1109/TWC.2022.3148531](https://doi.org/10.1109/TWC.2022.3148531).
- [50] Atanu Rajak et al. “Quantum annealing: an overview”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 381.2241 (Dec. 2022). ISSN: 1471-2962. DOI: [10.1098/rsta.2021.0417](https://doi.org/10.1098/rsta.2021.0417). URL: <http://dx.doi.org/10.1098/rsta.2021.0417>.
- [51] European Information Technologies Certification (EITC). *Cellular Networks, Cells, and Base Stations*. Accessed: 2024-11-22. 2024. URL: <https://www.eitc.org/e-learning/mobile-and-wireless-communications/cellular-networks-cells-and-base-stations>.
- [52] J A Montañez-Barrera et al. “Unbalanced penalization: a new approach to encode inequality constraints of combinatorial problems for quantum optimization algorithms”. In: *Quantum Science and Technology* 9.2 (Apr. 2024), p. 025022. ISSN: 2058-9565. DOI: [10.1088/2058-9565/ad35e4](https://doi.org/10.1088/2058-9565/ad35e4). URL: <http://dx.doi.org/10.1088/2058-9565/ad35e4>.
- [53] GSMA Intelligence. *The Mobile Economy Europe 2025*. GSMA Intelligence Report. GSMA, Jan. 2025. URL: <https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2025/01/0125-Mobile-Economy-Europe-2025-web.pdf>.
- [54] QuEra Computing Inc. *What is Variational Quantum Algorithm*. Accessed: 2025-06-18. QuEra Computing. June 2025. URL: <https://www.quera.com/glossary/variational-quantum-algorithm>.
- [55] D-Wave Systems Inc. *dwave-samplers: Simulated Annealing*. API Reference — Samplers. D-Wave Ocean SDK. URL: https://docs.dwavequantum.com/en/latest/ocean/api_ref_samplers/index.html#simulated-annealing (visited on 09/29/2025).
- [56] Gurobi Optimization, LLC. *Gurobi Optimizer*. Gurobi Optimization, LLC. URL: <https://www.gurobi.com/solutions/gurobi-optimizer/> (visited on 09/29/2025).
- [57] IBM Quantum. *EfficientSU2*. IBM Quantum Documentation. URL: <https://quantum.cloud.ibm.com/docs/en/api/qiskit/qiskit.circuit.library.EfficientSU2> (visited on 09/29/2025).

- [58] IBM Quantum. *Fake provider*. Qiskit IBM Runtime Documentation. URL: <https://quantum.cloud.ibm.com/docs/en/api/qiskit-ibm-runtime/fake-provider> (visited on 09/29/2025).
- [59] IBM Quantum. *Simulate stabilizer circuits*. IBM Quantum Documentation. URL: <https://quantum.cloud.ibm.com/docs/en/guides/simulate-stabilizer-circuits> (visited on 09/29/2025).
- [60] Qiskit Community. *Using Classical Optimization Solvers and Models with Qiskit Optimization*. Qiskit Optimization documentation (e.g., v0.7.0). Qiskit. URL: https://qiskit-community.github.io/qiskit-optimization/tutorials/11_using_classical_optimization_solvers_and_models.html (visited on 09/29/2025).