POLITECNICO DI TORINO
Repository ISTITUZIONALE

Quantum-compliant users scheduling optimization in joint transmission mobile access networks

Original
Quantum-compliant users scheduling optimization in joint transmission mobile access networks / Volpe, Deborah; Cirillo, Giovanni Amedeo; Fantini, Roberto; Boella, Andrea; Mondo, Giovanni; Graziano, Mariagrazia; Turvani, Giovanna. - In: QUANTUM INFORMATION PROCESSING. - ISSN 1570-0755. - ELETTRONICO. - 23:7(2024). [10.1007/s11128-024-04471-1]

Availability:
This version is available at: 11583/2990740 since: 2024-07-12T16:53:14Z

Publisher:
Springer

Published
DOI:10.1007/s11128-024-04471-1

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

15 May 2025

Quantum Information Processing (2024) 23:262
https://doi.org/10.1007/s11128-024-04471-1

Quantum-compliant users scheduling optimization in joint transmission mobile access networks

Deborah Volpe[1] · Giovanni Amedeo Cirillo[1] · Roberto Fantini[2] · Andrea Boella[2] · Giovanni Mondo[2] · Mariagrazia Graziano[3] · Giovanna Turvani[1]

Received: 12 May 2023 / Accepted: 14 June 2024
© The Author(s) 2024

Abstract
Joint Transmission (JT) is the dynamic coordination of transmission and/or reception at multiple geographically separated sites to improve end-user service quality. When user equipment receives signals from multiple sites, downstream performance improves. An optimization problem arises in selecting the best user subset for JT within a multiple-input–multiple-output (MIMO) system. Unfortunately, a pure brute-force approach is not feasible due to exponential time growth with user combinations, unsuitable for real-time selection in mobile networks with users continuously changing in time. This article proposes quantum-compliant heuristics using quadratic unconstrained binary optimization (QUBO) for JT user scheduling. QUBO handles initial user selection, followed by brute-force exploration for the solution. Numerical results indicate that quantum-compliant methods decrease solution time without substantial accuracy loss compared to brute-force methods.

Keywords Joint transmission · Multiuser multiple-input–multiple-output · Multi-users scheduling optimization · Quadratic unconstrained binary optimization · Quantum computing · Hybrid quantum-classical algorithm

1 Introduction

Wireless Multiple-Input–Multiple-Output (MIMO) communication systems [1–4] involve multiple antennas at both the transmitter and receiver to enhance data throughput, reliability, and coverage. Different data streams are transmitted and received simultaneously, exploiting the spatial diversity of the wireless channel.

Joint Transmission (**JT**), commonly named Coordinated MultiPoint (CoMP) in the mobile access network context, is a multiuser MIMO system which aims to improve users' service quality by pre-compensating the **Inter-Cell Interference** (**ICI**) compo-

Deborah Volpe and Giovanni Amedeo Cirillo have contributed equally to this work.

Extended author information available on the last page of the article

Published online: 04 July 2024

nents, i.e., increasing the **Signal-to-Interference-and-Noise-Ratio** (**SINR**), through a **beamforming** technique. This consists of splitting information symbols among different **base stations** (**BSs**) to maximize the **transmission rate** (**TR**) of served users, making the channel's attenuation and phase-shifting effects negligible. However, attenuation could imply a preliminary signal amplification unsustainable by each BS, thus necessarily requiring the definition, for each symbol associated with a user to be served, of a **power scaling transmission factor** $p_s < 1$, which is also responsible for TR reduction.

An optimization problem, looking for the optimal subset of users to be served in each time slot, can be identified to make the most of the mechanism. From a practical perspective, it consists of finding the subset of users having the highest TR after defining the highest p_s values not exceeding the BSs maximum power. This subset can be theoretically identified by evaluating all those possible iteratively. For each of these, the optimal p_s values for the considered users must be found (e.g., exploiting a convex optimizer), and then the total TR can be computed. However, this **brute-force approach** can be computationally expensive and becomes unfeasible for many potential users because of subsets factorial scaling. It must be always reminded that in JT-based systems users change in time and the optimal subset in a given time frame must be detected in the shortest time possible. For this reason, it is important to define strategies for the solutions space exploration characterized by a good trade-off between the quality of the solution and the execution time.

Quantum computation [5] can offer advantages in tackling search and optimization problems. Thanks to quantum superposition and entanglement, it virtually performs a simultaneous evaluation of multiple possibilities, data, and combinations, thus significantly accelerating search processes. These features hold promise for addressing optimization problems, where quantum algorithms and procedures can find solutions with higher quality or in a lower time than classical methods. Reminding that quantum computing has already provided interesting results when solving problems concerning wireless communications [6–9], the investigation of the potentialities of this paradigm for users scheduling in JT-based MIMO systems can be considered reasonable.

This work is going to propose new approaches—one based on quantum-complaint heuristic (**QUBO-assisted**), the other on a simplification of the p_s exploration (**Naïve**)—and to compare them with the brute-force one in terms of total execution time and quality of the obtained solution. The analysis was performed with *ad-hoc* developed software prototypes, verifying the effectiveness of the proposed methods. A longer-term perspective of this work can be the definition of a **distributed computing system** for monitoring the JT mechanism in a quite small time window based on **application-specific hardware**.

The article is structured as follows. Sect. 2 introduces JT mechanism in wireless MIMO systems, the fundamentals of QUBO formulation are described in Sects. 3, 4 presents the proposed quantum-compliant strategies for JT scheduling optimization, Sect. 5 discusses results, and Sect. 6 concludes the article with some future perspectives.

2 Joint transmission theoretical foundations

In this section, the mathematical formalism and the working principles of JT systems are introduced. It emphasizes that it follows the mathematical formalism described in [10], which is based on a **phasor-domain** representation for sinusoidal signals and **normalizes electric powers** to an implicit resistance ($P_{\text{normalized}} = |V|^2$).

2.1 MIMO model

In a wireless MIMO system with N_{BS} **base stations** (**BSs**) (composed of a single antenna) and $S \leq N_{\text{BS}}$ receiving users, the n^{th} BS transmits a symbol x_n, which is received by a user s as $h_{sn}x_n$, where h_{sn} is a channel coefficient related to path-loss attenuation and random phase shift of the transmitted signal. The overall signal received by user s is given by:

$$y_s = \sum_{n=1}^{N_{\text{BS}}} h_{sn}x_n + \theta_s, \tag{1}$$

where θ_s is the intrinsic thermal noise. Therefore, the MIMO system transfer function can be represented in matrix form as:

$$Y = H \cdot X + \theta\,, \tag{2}$$

where $H \in \mathbb{C}^{S \times N_{\text{BS}}}$ is the **channel matrix**, X and Y are **transmitted** and **received** signal vectors and θ is the intrinsic **thermal noise** (**TN**) vector. Assuming that each BS transmits the symbol of one user, which implies $N_{\text{BS}} = S$ and $x_s = b_s$, the user s receives the signal:

$$y_s = \sum_{n=1}^{N_{\text{BS}}} h_{sn}x_n + \theta_s = \overbrace{h_{ss}x_s}^{\textbf{useful}} + \overbrace{\sum_{n=1, n\neq s}^{N_{\text{BS}}} h_{sn}x_n}^{\textbf{interference}} + \overbrace{\theta_s}^{\textbf{noise}}\,. \tag{3}$$

It is characterized by two unwanted components: the **interference** of the signals transmitted by other BSs and the **intrinsic thermal noise**,

$$\text{SINR} \triangleq \frac{P_{\text{useful}}}{P_{\text{interference}} + P_{\text{noise}}} = \frac{|h_{ss}x_s|^2}{\sum_{n=1, n\neq s}^{N_{\text{BS}}} |h_{sn}x_n|^2 + \sigma_\theta^2}\,, \tag{4}$$

where σ_θ^2 is the **TN power**. SINR is also used to determine the **transmission rate** (**TR**) over a fixed bandwidth B:

$$r \triangleq B \log_2 (1 + \text{SINR})\,. \tag{5}$$

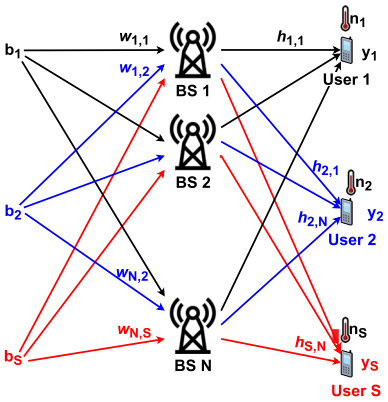


Fig. 1 Conceptual scheme of a MIMO system with Joint transmission. N BSs are exploited to transmit S symbols to S users. The symbol b_s to be transmitted to user s is associated with the base station n according to the element w_{ns} of the beamforming matrix, while the signal of the BS n is provided to user s according to the element h_{sn} of the channel matrix. The received signal y_s of each user depends on the linear combination of the BS signals and on the thermal noise n_s (Color figure online)

Maximizing SINR is crucial for maximizing the transmission rate because bandwidth is typically constant.

2.2 Join transmission mechanism

In JT-based networks, the goal is **to remove interference through compensation** by distributing user symbols (b_s)— equal to one in this work—to all base stations (BSs) through a technique called **beamforming** (Fig. 1). Therefore, each BS transmits a linear combination of user symbols:

$$x_n = \sum_{s=1}^{S} w_{ns} b_s \,, \tag{6}$$

thus making the received signals equal to:

$$Y = H \cdot X + \theta = H \cdot (W \cdot b) + \theta \,, \tag{7}$$

where $W \in \mathbb{C}^{N_{\mathrm{BS}} \times S}$ is the **beamforming matrix** engineered to compensate for interference and H depends on the environment ($H \times W = \mathbb{I}_{\dim\{S\}} \Leftrightarrow W = H^{\dagger}$, where $\mathbb{I}_{\dim\{S\}}$ is the identity matrix and $H^{\dagger}$ is **Moore-Penrose inverse matrix** of H).

Beamforming also implies signal amplification of the transmitted signal against channel attenuation, potentially exceeding the maximum supported power ($\sum_{s=1}^{S} |w_{ns} b_s|^2 \geq P_{\max_{\mathrm{BS}_n}}$), so a power partition factor (p_s) is introduced for each symbol. In this way, user s receives the following signal:

$$y_s = h_{ss} w_{ss} \sqrt{p_s} b_s + \theta_s \,, \tag{8}$$

and the TR can be written, according to the Shannon–Hartley theorem, as:

$$r \triangleq B \log_2 \left(1 + \frac{p_s |b_s|^2}{\sigma_\theta^2}\right). \tag{9}$$

For optimizing the performance of a mobile network, the goal is to maximize the overall TR of the network:

$$\text{find} \qquad \max \left\{r_{S_{\text{served}}}\right\} = \max \left\{B \sum_{s \in S_{\text{served}}} \log_2 \left(1 + \frac{p_s |b_s|^2}{\sigma_\theta^2}\right)\right\}. \tag{10}$$

In practice, there are more potential users (N_{UE}) than BSs (N_{BS}), and finding the best user subset involves iterating through all possible subsets and optimizing p_s for each user in each subset. Considering the increasing monotone metric in Eq. 9, the optimal p_s values of a given user subset can be identified with a **convex optimizer**.

The hardness of the task comes from the necessity of solving $N_{\text{subset}} = \binom{N_{\text{UE}}}{S} = \frac{N_{\text{UE}}!}{S!(N_{\text{UE}}-S)!}$ convex optimization problems and compare the obtained results for the selection of the best subset of users (**brute-force** method). Therefore, the complexity scales factorially with the number of users, thus making brute-force exploration unfeasible.

In this article, alternative methods are proposed to overcome this computational limitation, as quickly identifying the optimal user subset is essential in JT-based systems where users change over time.

3 Quadratic unconstrained binary optimization

QUBO modeling and quantum computational paradigms for QUBO problem solving are briefly introduced in the following. More details are available at [11–13].

3.1 Mathematical model

Quadratic Unconstrained Binary Optimization (**QUBO**) is a mathematical formulation capable of describing many real-world problems [11]. The term **Optimization** suggests the goal of this formulation: to minimize an objective function. As the term **Binary** implies, it employs unipolar binary variables (0/1). The **Quadratic** term refers to the highest power applied to variables, while **Unconstrained** indicates that the constraints on variables are not introduced in a standard way. The objective function can be written as:

$$\text{Obj}(c, a_i, b_{ij}, x_i) = c + \sum_i a_i \cdot x_i + \sum_{i<j} b_{ij} \cdot x_i x_j\,, \tag{11}$$

where $x_i \in [0, 1]$ is a binary variable, $x_i x_j$ is a coupler of two variables, a_i is a single-variable weight, b_{ij} is a strength which controls the influence of variables i and j and c is an offset, which can be neglected during the optimization.

In reality, QUBO formulation can take into account the variable constraints, introducing **quadratic penalties** to the objective function

$$\text{minimize/maximize} \quad y = f(\mathbf{x}) + \lambda g(\mathbf{x}) \,, \tag{12}$$

where λ is a positive penalty parameter to be multiplied by the constraint function $g(\mathbf{x})$. This trick allows the evaluation of constraints during the solver execution, but correctly sizing λ is critical and can affect the solution quality. In fact, a too-low value makes the constraint function negligible, while a too-high value flattens the objective function. According to [11], $\lambda g(\mathbf{x})$ should be a fraction ($\in [75\%; 150\%]$) of $f(\mathbf{x})$.

It is possible to prove that the objective function of a QUBO problem is equivalent to the **Ising** Hamiltonian, used in statistical mechanics for the description of ferromagnetism in terms of atomic spins:

$$\mathcal{H} = \sum_i h_i \cdot s_i + \sum_{i<j} J_{ij} \cdot s_i s_j \,. \tag{13}$$

Ising model involves bipolar binary variables ($s_i \in \{-1, +1\}$, associated with spin orientations) instead of unipolar ones, h_i is the single-spin external field, whose sign is related to the preferred orientation of a spin with respect to that of an applied external field, and J_{ij} magnitude and sign quantify the interaction between neighbor spin pairs and their preferred alignment (ferromagnetic or anti-ferromagnetic), respectively. It is possible to move from Ising to QUBO representation (and *vice-versa*) with the following relations:

$$\begin{aligned} x_i &= \frac{1 + s_i}{2} \,, \\ s_i &= 2x_i - 1 \,. \end{aligned} \tag{14}$$

Even though the conversion changes the values of the single-variable and two-variable coefficients, the optimal solution does not change. A QUBO solution variable equal to 0(1) has a corresponding Ising solution variable equal to -1(+1).

3.2 Solving QUBO models with quantum computers

QUBO and Ising are the currently most feasible formulations for exploiting **quantum-related solvers**. **Quantum computing** is a computational paradigm which exploits quantum mechanics for executing calculations. It is possible to define quantum algorithms capable of solving specific problems with a computational complexity lower than their corresponding classical counterparts. This is possible thanks to quantum principles [5] like:

- **Superposition**, allowing the implementation of a virtual parallel computation;

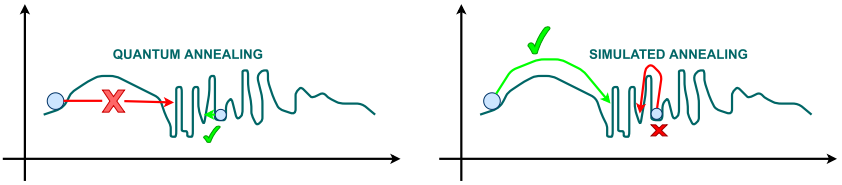


Fig. 2 Exploration capabilities and limits of quantum annealing (left) and simulated annealing (right). The first can efficiently explore the rough part of the energy landscape, while cannot easily overcome the wide energy barrier. On the contrary, simulated annealing exploits thermal fluctuations to overcome the wide energy barrier but shows more difficulties with high and narrow peaks (Color figure online)

- **Entanglement**, an intrinsic interaction between quantum systems exploitable to accelerate the convergence to the problem's solution;
- **Tunneling**, thanks to which quantum systems and particles can pass through high and narrow barriers instead of *jumping* them, as done by classic methods, such as hill climbing and simulated annealing.

The two most consolidated quantum computation paradigms are the general-purpose **quantum circuit model** and the special-purpose **quantum annealing**. In the first case, operations are executed in terms of quantum gates [5], a sort of extension to qubit (the unit of information of quantum computing) of logic gates employed in classical computing.

In quantum annealing [14], an objective function is mapped onto the energy profile of a quantum physical system, so its minimum-energy state can be associated with the problem's optimal solution. This can be achieved with quantum properties, like superposition, tunneling, and adiabatic evolution, differently from the classical simulated annealing, where the barrier is *jumped* instead of *tunneled*.

The barrier-overcoming mechanism is strictly related to the fact that the effectiveness of an optimization algorithm on a specific problem is strongly related to the characteristic of its energy profile [15]. For example, simulated annealing and other approaches based on local search can overcome wide and smooth barriers more easily than high and narrow ones, while, on the contrary, quantum annealing performs well with high and narrow peaks and is expected to be less effective in exploring vast and flat regions [16, 17].

The energy profiles associated with real-world problems are usually **heterogeneous**, as reported in Fig. 2. This implies that the probability of success of each solver depends on the compatibility of its exploration mechanism with the problem's energy profile region and that the definition of alternative QUBO models for the same problem could imply so different profiles that their best solver could change.

Current quantum annealers natively encode and solve Ising problems, so a QUBO problem can be solved by translating it into an Ising one —according to (14)—and handling the partial connectivity of the backend with minor embedding techniques [18]. Moreover, these are generally affected by errors that can be associated with decoherence phenomena; for this reason, **quantum-inspired** techniques, such as **digital annealing** [19], simulated quantum annealing [13] and simulated adiabatic bifurca-

tion [20], have been proposed to emulate the quantum annealing behavior on classical hardware, thus avoiding errors associated with qubits decoherence.

The same issues concerning qubit connectivity and decoherence are observed on quantum computers based on the circuit model. Moreover, the currently best procedures exploiting these quantum computers to solve QUBO/Ising problems require the assistance of classical computers [21–23]. In the context of this article, quantum annealing is employed as quantum QUBO solver, since real hardware based on this paradigm [24] is nowadays capable of processing more QUBO variables than the corresponding one based on the circuit model [25].

4 Proposed solutions

This work proposes alternative approaches for reducing the computational complexity and the time required for selecting the optimal subset of JT-served users. The first proposal (Sect. 4.1), called ***Naïve***, simplifies the p_s exploration for each subset, accepting sub-optimal solutions. The second approach performs a **quantum-compliant preliminary fast-approximated user selection**, which will be examined with the brute-force method. In this strategy, the user set reduction is done by exploiting a QUBO model (Sect. 4.2).

Before discussing the two proposed solutions, it is important to clarify that they operate at the single-BS-triplet level and in a specific sub-channel. As detailed in Sect. 5.1, in a specific sub-channel users could receive signals from more than three BSs belonging to different triplets. For this reason, in a specific sub-channel, users are preliminarily separated in such a way that each of them is associated with a single BS triplet. In particular, each user is associated with the triplet whose BSs are all capable of providing a signal $h_{sn}x_n \neq 0$ (Equation (1)) and the highest receivable power, according to TIM S.p.A. MIMO simulator. If the previous condition is not satisfied, the same procedure is applied by decreasing the highest number of BSs providing non-null received signals.

The described procedure can be repeated for all sub-channels of the analyzed MIMO system. Assuming that the optimization BS-triplet-level problems of each sub-channel can be analyzed independently on those of the other sub-channels, tasks can be also parallelized.

4.1 Naïve

The problem's computational complexity can be reduced, assuming all the S users in each subset have **the same** p_s. In this way, the p_s, allowing each BS to transmit at its maximum power, can be computed independently of the others; then, the power constraint is satisfied for all the N_{BS} available BSs by choosing the lowest p_s.

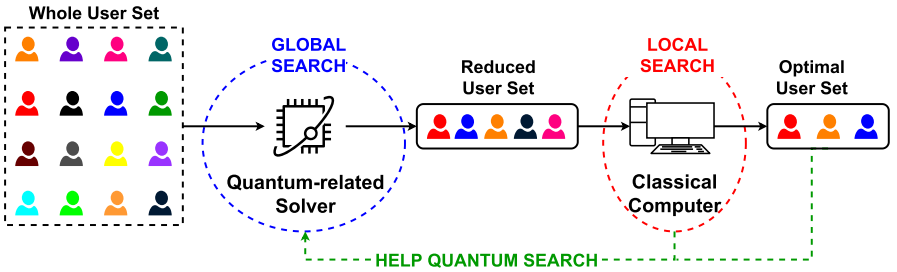


Fig. 3 Block scheme of the proposed quantum-compliant approach, where users are reduced by solving a QUBO problem and a classical computer evaluates all the reduced users' set combinations, identifying the optimal power partition factor and the best users to serve (Color figure online)

An algebraic example with three BSs and two users is reported in the following;

$$\begin{bmatrix} |w_{11}|^2 & |w_{12}|^2 \\ |w_{21}|^2 & |w_{22}|^2 \\ |w_{31}|^2 & |w_{32}|^2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} |w_{11}|^2 p_1 + |w_{12}|^2 p_2 \\ |w_{21}|^2 p_1 + |w_{22}|^2 p_2 \\ |w_{31}|^2 p_1 + |w_{32}|^2 p_2 \end{bmatrix} = \overset{\text{if } p_1 = p_2}{=} \begin{bmatrix} (|w_{11}|^2 + |w_{12}|^2)p \\ (|w_{21}|^2 + |w_{22}|^2)p \\ (|w_{31}|^2 + |w_{32}|^2)p \end{bmatrix} \leq \begin{bmatrix} P_{\max_{\text{BS}}} \\ P_{\max_{\text{BS}}} \\ P_{\max_{\text{BS}}} \end{bmatrix} \tag{15}$$

$$p = \min \left\{ \begin{bmatrix} P_{\max_{\text{BS}}}/(|w_{11}|^2 + |w_{12}|^2) \\ P_{\max_{\text{BS}}}/(|w_{21}|^2 + |w_{22}|^2) \\ P_{\max_{\text{BS}}}/(|w_{31}|^2 + |w_{32}|^2) \end{bmatrix} \right\}.$$

The total time required to find a solution is reduced because the convex optimizer is not involved. However, the obtained solution is **sub-optimal** because only one BS transmits at the maximum power, so exploiting all the antennas at their best is impossible.

4.2 Quantum-compliant users set reduction

The second approach divides the problem into **two parts**:

- **Reduction of the user set** with a **QUBO** optimization problem to solve in a quantum or classical way;
- Application of the convex classical solver for optimizing the power partition factors on each combination of users in the reduced set, to find one with the highest **total transmission rate**.

Therefore, the QUBO solver allows a first screening of the users, based on their characteristics and the problem constraints, to analyze with the accurate but more complex brute-force approach.

This strategy, which is graphically represented in Fig. 3, and whose pseudocode is reported in Algorithm 1 significantly reduces the computational complexity and the execution time, without affecting significantly the quality of the final solution.

As shown in the aforementioned figure, a feedback mechanism could be inserted to improve the quality of the user reduction performed by the QUBO solver.

From a theoretical point of view, a *hybrid* approach, where the quantum-compliant method exploits the *Naïve* method in its second step, can be considered. However, this is not reported in this article because it is expected to require more execution time than the *Naïve*-only approach and to provide solutions with lower transmission rate than the canonical quantum-compliant method involving the convex optimizer.

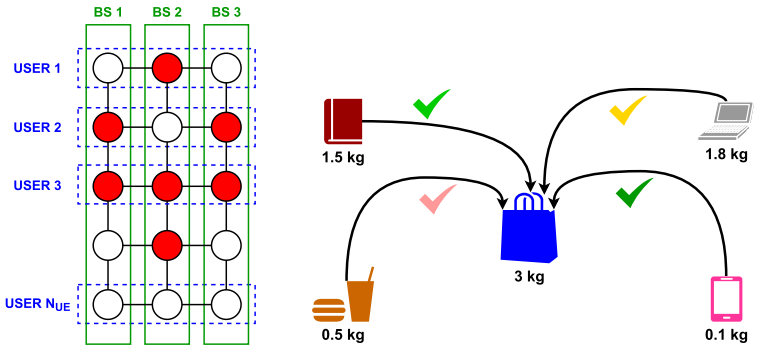
Algorithm 1 Quantum-compliant optimization method for a single BS triplet and a specific sub-channel

Input: $H, \sigma_n^2, P_{\max_{\mathrm{BS}_n}}$ and N_{UE}
Output: $S_{\mathrm{served}_{\mathrm{best}}}$ and $r_{\max}$
Users reduction:
generate QUBO ($H, \sigma_n^2, P_{\max_{\mathrm{BS}_n}}, N_{\mathrm{UE}}, N_{\mathrm{BS}}$)
solve QUBO with a classical or a quantum solver
obtain the users reduced set $N_{\mathrm{UE}_{\mathrm{reduced}}}$ from QUBO solution
Brute-force:
$r_{\max} = 0$
`/* For each users combination */`
for all S_{served} **do**
  $W \leftarrow H(S_{\mathrm{served}})^{\dagger}$
  compute $p_s(W, P_{\max_{\mathrm{BS}_n}})$
  $r_{S_{\mathrm{served}}} \leftarrow \sum_{s \in S_{\mathrm{served}}} B \log_2 \left(1 + \frac{p_s}{\sigma_{\hat{\theta}}^2}\right)$
  if $r_{S_{\mathrm{served}}} > r_{\max}$ **then**
    $r_{\max} \leftarrow r_{S_{\mathrm{served}}}$
    $S_{\mathrm{served}_{\mathrm{best}}} \leftarrow S_{\mathrm{served}}$
  end if
end for
Return: $S_{\mathrm{served}_{\mathrm{best}}}$ and $r_{\max}$

4.2.1 First QUBO formulation

The data structure of the proposed QUBO formulation is fundamentally a matrix $N_{\mathrm{UE}} \times N_{\mathrm{BS}}$ of unipolar binary variables (as shown in Fig. 4a), one for each user–BS pair, in which rows are devoted to users, while columns to BSs. The x_{sn} variable assumes value one if the n^{th} BS serves the s^{th} user.

Considering that the optimal subset of users has to be identified such that the **total TR is maximized without exceeding the BSs maximum power**, a similarity to the well-known **knapsack** problem (shown in Fig. 4b) [11] can be found. Indeed, it aims to define, for a set of objects X, each of which is labeled as x_i and is characterized by a weight w_i and a preference score p_i, the best subset to be put into a bag, guaranteeing that the total weight does not exceed a threshold W ($0 < \sum_{i=1}^{\dim(X)} w_i x_i \leq W$) while maximizing the total preference score $P = \sum_{i=1}^{\dim(X)} p_i x_i$.



(a) QUBO binary variables structure proposed.

(b) Graphical representation of the knapsack optimization problem.

Fig. 4 Structure of the binary variables chosen, which is, fundamentally, a $N_{\mathrm{UE}} \times N_{\mathrm{BS}}$ matrix of unipolar binary variables—x_{sn} is equal to one (colored red) if the n^{th} BS serves the s^{th} user in the optimal solution, and graphical representation of the principle of the knapsack problem exploited for the QUBO formulation (Color figure online)

For each BS of the group employed for the JT, i.e., for each column, the problem has to satisfy knapsack-like conditions:

- Maximization of the **TR**, which plays the role of the total preference scores;
- Verifying that the BS transmitted power does not exceed the maximum one, which is equivalent to assuring that the objects' total weight is lower than a certain threshold.

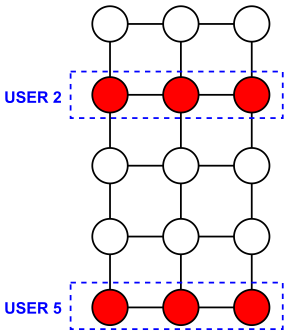
In the QUBO formulation, for optimizing the overall TR of the network, it was chosen to optimize the sum SNR:

$$\text{find} \qquad \max\left\{ \sum_{s \in S_{\mathrm{served}}} \frac{p_s |b_s|^2}{\sigma_\theta^2} \right\}. \tag{16}$$

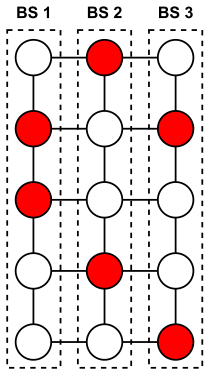
This choice was made because, under the Shannon law assumption, TR has a logarithmic dependence from SNR, which introduces a nonlinearity that is not compatible with QUBO formulation. Optimizing the sum of SNR of the network can be a simpler and more tractable approach. Moreover, by improving the SNR of each link in the network, the overall quality of the communication channels is expected to be improved.

Even if optimizing the sum of SNR of a mobile network may not directly guarantee the highest TR sum, it can be a reasonable approximation, since the optimization of the SNR sum usually leads to the optimization of the overall system's TR.
Moreover, since the BSs have to serve the same users, the complete QUBO formulation requires more than the combinations of N_{BS} parallel knapsack problems. In particular, all variables in a row must be all equal (as shown in Fig. 5a), thus ensuring that a user is served by all the BSs or not, and the sum of binary variables on a column must be equal to S (as shown in Fig. 5b), so that the BSs serve a total number of users equal to S.

Fig. 5 Constraints (Color figure online)



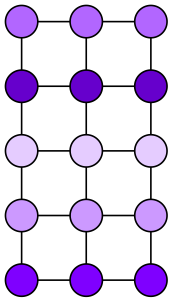
(a) Row constraint: all variables in a row must be all equal, thus ensuring that a user is served by all the BSs or not.



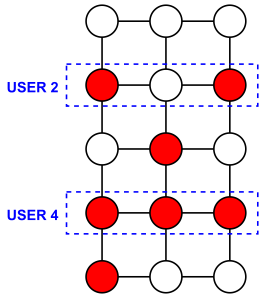
(b) Column constraint: the sum of binary variables on a column must be equal to S, so that the BSs serve a total number of users equal to S

.

Fig. 6 Graphical representation of the SNR optimization and a possible final solution (Color figure online)



(a) Graphical representation of the contribution associated with the maximization of SNR.



(b) Possible QUBO solution, in which the majority voter post-processing is required.

To summarize, the cost function is composed of four contributions:

1. **Maximize SNR**:

$$f_{\text{SNR}}(\mathbf{x}) = -\sum_{s} \frac{p_s}{\sigma_\theta^2} \sum_{n} x_{sn}\,, \tag{17}$$

where p_s is the power scaling factor of the s^{th} user, σ_θ^2 is the thermal noise power, and the minus sign is required to move the maximization problem to a minimization one. This term tends to force all x_{sn} to one. As observable in Fig. 6a, Boolean variables belonging to the same row/user have the same p_s. Moreover, different purple intensities refer to the different p_s, and consequently SNR, of users;

2. **Power constraint**, i.e., the BS maximum power must be not exceeded:

$$f_{\text{pow}}(\mathbf{x}) = \sum_n \left[\sum_s |w_{sn}|^2 p_s x_{sn} + \sum_{i \in A} \alpha_i a_i - P_{\max} \right]^2, \tag{18}$$

where $|w_{sn}|^2 p_s$ is the power required to the n^{th} BS to transmit the symbol b_s to the s^{th} user, $P_{\max}$ is the maximum power available for each base station, a_i is the i^{th} auxiliary variable required for expressing the inequality constraint (as presented in [11, 26] the number of auxiliary variables required depends from $P_{\max}$ value), and α_i is the weight of a_i;

3. **Row constraint**, i.e., all BSs have to serve the same users:

$$f_{\text{row}}(\mathbf{x}) = \sum_s \left[\left(\sum_n x_{sn} \right) - N_{\text{BS}} \right]^2; \tag{19}$$

4. **Column constraint**, i.e., the number of users served by each BS has to be equal to S:

$$f_{\text{column}}(\mathbf{x}) = \sum_n \left[\left(\sum_s x_{sn} \right) - S) \right]^2. \tag{20}$$

Therefore, the final objective function is:

$$\begin{aligned} f(\mathbf{x}) = \lambda_1 \sum_n \Bigg[\underbrace{\left(\sum_s x_{sn} \right) - S)}_{\textbf{column constraint}} \Bigg]^2 - \lambda_2 \underbrace{\sum_s \frac{p_s}{\sigma_\theta^2} \sum_n x_{sn}}_{\textbf{maximize SNR}} + \\ + \lambda_3 \sum_s \Bigg[\underbrace{\left(\sum_n x_{sn} \right) - N_{\text{BS}}}_{\textbf{row constraint}} \Bigg]^2 + \lambda_4 \underbrace{f_{pow}(\mathbf{x})}_{\textbf{power}}, \end{aligned} \tag{21}$$

where $\lambda_{1\ldots4}$ are the weights of the objective function terms to be properly sized, as mentioned in Paragraph 3. The principles behind the QUBO formulation are graphically resumed in Fig. 7.

As mentioned, f_{pow} allows the expression of the inequality contribution in QUBO, translating them as equality constraints by adding auxiliary variables whose number depends on the $P_{\max}$ values, as explained in [11, 26]. Python libraries for QUBO problem formulation, such as **qubovert** [27], can automatically do this transformation. However, introducing the auxiliary variables could dramatically increase the QUBO model and, consequently, the time required for its resolution, thus also reducing the advantage of employing the proposed solution. Therefore, this is the strongest limit of this QUBO formulation.

Unfortunately, the final solution could be similar to the one reported in Fig. 6b, where the elements of each row have different values. A cause could be the not-perfect sizing of penalty parameters. However, the proposed solver can activate a **majority**

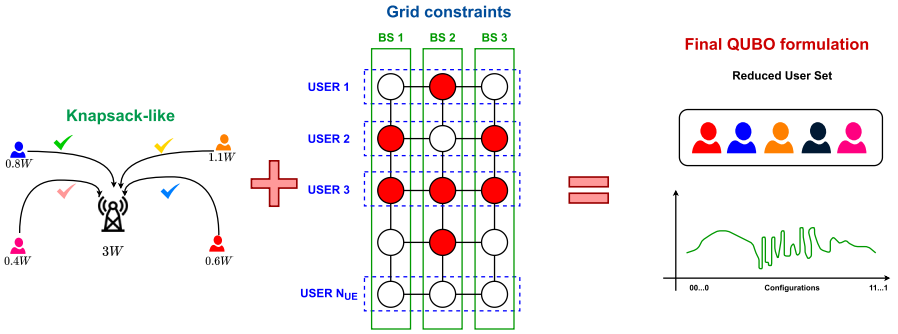


Fig. 7 Overview of QUBO model for users selection: graphical description of the knapsack-like problem associated with BS maximum power (**left**), QUBO variables data structure with row and column constraints (**middle**), an expected QUBO result, i.e., a subset of users, and an example of a problem cost function (**right**) (Color figure online)

voting mechanism, performing a sort of *error mitigation*: when more than half of the elements on a row are equal to one, the corresponding user is selected for the second step of the algorithm.

Another problem to be managed is when the reduced subset size is smaller than S. In this case, the solver modifies the weights λ_2 and λ_4 and tries to solve the QUBO problem again. This procedure, substantially equivalent to the so-called **sequential scaling** of QUBO penalty functions [28], can be repeated a couple of times, and if the number of users is still insufficient, the Naïve approach (Sect. 4.1) is employed. It is also involved if the convex solver in the second step gives unfeasible negative p_s.

4.2.2 Normalized QUBO formulation

The rapid increase of the QUBO problem dimension due to the insertion of auxiliary variables for representing the inequality constraints could reduce the efficiency of the proposed method, so evaluating a strategy for reducing the auxiliary variables number is crucial. First, it is necessary to remember that these are inserted to transform the inequality constraint into an equality one:

$$\sum_s |w_{sn}|^2 p_s x_{sn} + k = P_{\max}\,, \tag{22}$$

where the multi-level variable k can be described by exploiting binary expansions so that the number of inserted variables increases as $P_{\max}$ increases [26].

The first proposal for solving the presented problem is to perform a normalization of the power factors:

$$\sum_s \frac{|w_{sn}|^2 p_s x_{sn}}{P_{\max}} + k = \frac{P_{\max}}{P_{\max}}\,. \tag{23}$$

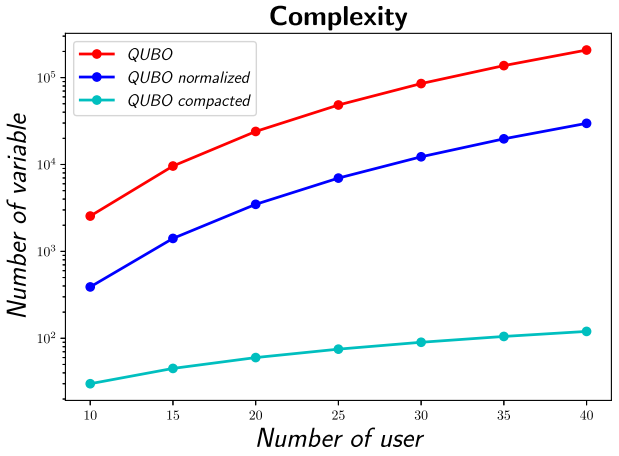


Fig. 8 Number of binary variables required as a function of the number of users for the first, normalized and compacted QUBO formulations (Color figure online)

In order to maintain all coherence, the constraint parameters must also be scaled by the same quantity.

The QUBO normalization significantly reduces the number of required variables (about one order of magnitude, as shown in Fig. 8), thus positively impacting on solving time.

Compacted QUBO formulation In order to further reduce the QUBO complexity, an *approximation* for the power constraint can be performed, taking into account that QUBO simply performs the first selection and the p_s will be optimized in the next step, considering the power constraint again. The proposed approximation applies an equality constraint instead of an inequality one. In particular, it is imposed that:

$$\sum_s |w_{sn}|^2 p_s x_{sn} = c \cdot P_{\max}\,, \tag{24}$$

where c is a reduction factor, e.g., 0.8, so that a much higher quadratic penalty is imposed on a solution that is further from the imposed value (as shown in Fig. 9). Solutions with a lower penalty are around the forced value $c \cdot P_{\max}$. Considering the proportionality between the SNR and the transmitted power, solutions with a transmitted power close to the maximum should be characterized by a higher SNR. This approach avoids the introduction of auxiliary variables, thus obtaining a QUBO model with only $N_{\mathrm{UE}} \times N_{\mathrm{BS}}$ variables (a reduction of two orders of magnitude with respect to the First formulation, as shown in Fig. 8).

Other QUBO formulations analyzed

Starting from the compact QUBO version, some constraints, which seem redundant considering the role played by the others, can be removed. In particular, three alternatives were considered:

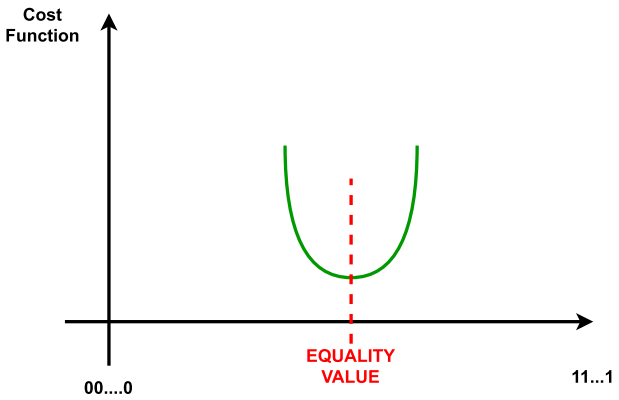


Fig. 9 Quadratic penalty imposed by an equality constraint (Color figure online)

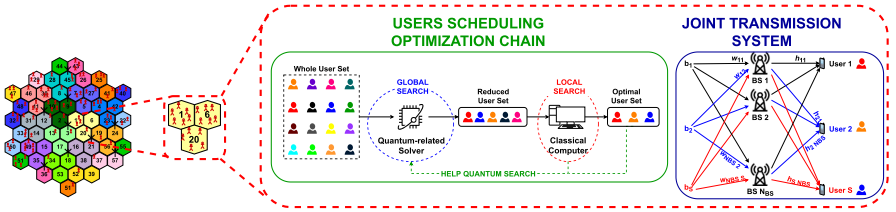


Fig. 10 Overall block scheme of the quantum-compliant solver for Joint transmission allocation problem, considering a single sub-channel, 19 BS triplets, and users grouped for triplet: the environment considered by the TIM S.p.A. simulator, with a focus on a BSs collaborating triplet, for which a dedicated JT analysis can be evaluated (**left**), block scheme of the proposed quantum-compliant approach, where users are reduced by solving a QUBO problem and a classical computer evaluates all the reduced user set combinations, identifying the optimal power partition factor and the best users to serve (**middle**), graphical description of the JT MIMO system, with the found optimal users served by BSs (**right**) (Color figure online)

- **Compacted QUBO without column constraint** (**QUBO NCC**): the removal of the column constraints was considered since the power limitations should be sufficient to select a number of users close to the wanted one.
- **Compacted QUBO without power constraint** (**QUBO NPC**): considering the observations in Sect. 4.2 and the presence of column constraints for limiting the number of users, the removal of the maximum power constraint can be tried.
- **Compacted QUBO without SNR optimization** (**QUBO NSNR**): since the current QUBO works with random p_s values not coming from previous experience, the objective function SNR contribution was tried to be removed.

5 Results

This section reports the most significant results for the proposed approaches for solving the JT optimization problem.

5.1 Setup overview

The tests conducted in this study involved the implementation of proposed strategies using Python and the **qubovert** [27] library for Quadratic unconstrained binary optimization (QUBO) formulation. The best power scaling (p_s) values were determined using the **Splitting Conic Solver** (**SCS**) [29] of the **CVXPy** [30] library. QUBO problems were solved using different solvers, including the **D-Wave Neal simulated annealer** (**SA**) [31],**D-Wave Leap Quantum Annealer**(**QA**) [24], accessible through a remote connection, and **digital annealer** (**DA**) solver [19], which was obtained by modifying from the D-Wave Neal SA code for fair comparison.

The SA tests were performed on a single-process Intel(R) Xeon(R) Gold 6134 CPU @ 3.20 GHz opta-core, Model 85, with a memory of about 103 GB [32], while tests comparing different QUBO solvers were carried out on a single-process Intel(R) Core(TM) i7-1165G7 @ 2.80 GHz, with a memory of 8 GB.

In all cases, independently on the employed QUBO solver, the majority-voting-based error mitigation technique was employed for the identification of the reduced user set for each BS triplet.

The tests exploit **synthetic** datasets and data from the **TIM S.p.A. MATLAB-based MIMO simulator**. The first scenario considers random sub-channel matrices for varying numbers of users (N_{UE}), while the second sub-channel matrices from the TIM S.p.A. simulator, which simulated a complex environment (shown in the leftmost part of Fig. 10). with 57 base stations, 570 users, and 52 sub-channels. The system-level simulator for mobile networks evaluates network-level performance indicators, such as cell capacity, user throughput distribution, radio resource utilization, and interference levels, considering a multiplicity of users and BSs. The propagation channel is computed by following the indication provided by 3GPP in the technical report TR 38.901 [33], describing the modeling of the radio channel in the radio bands used by 5 G technologies. The selected scenario is Urban Macro (UMa), composed of a hexagonal grid of 19 macro BS sites, with three sectors (i.e., cells) per site and an inter-site distance (ISD) of 500 m. BSs are assumed at 25 m, and a given number of pedestrian users, moving at a speed of 3 km/h, are randomly dropped in each cell (10 per cell in the considered scenario). As per TR 38.901, different propagation laws are used to compute path-loss and shadow for Line of Sight (LOS) and Non-Line of Sight users, with the probability of being in LOS, computed according to the technical report. Following the clustered delay line (CDL) approach, fast fading effects are included, considering a CDL-B channel profile.

All parameters were computed by considering each BS triplet independently from the others, through the method for associating each user with a single BS triplet introduced in Sect. 4.

The H matrix of each sub-channel was computed as the element-by-element product of the path-loss matrix in linear units L and the complex *dephasing* matrix P associated with the phase shift of each signal during its transmission from a BS to a user. Noise power was always considered constant and equal to -109dBm, for staying in a worst-case scenario.

5.2 Figures of merit

The following figures of merit were considered to evaluate the proposed approaches' effectiveness:

- The number of users combination to evaluate obtained as:

$$n_{c,\text{init}} = \binom{N_{\text{UE}}}{S} = \frac{N_{\text{UE}}!}{S!(N_{\text{UE}} - S)!} \tag{25}$$

  in the brute-force case and as:

$$n_c = \binom{N_{\text{UE}_{\text{reduced}}}}{S} = \frac{N_{\text{UE}_{\text{reduced}}}!}{S!(N_{\text{UE}_{\text{reduced}}} - S)!} \tag{26}$$

  in the QUBO-assisted case;
- The percentage of selected users from the original set:

$$p_{\text{sel}} = \frac{N_{\text{UE}_{\text{reduced}}}}{N_{\text{UE}}} \cdot 100\,; \tag{27}$$

- The complexity of the obtained QUBO formulation in terms of the number of binary variables exploited (v_q);
- The relative percentage transmission rate error:

$$\epsilon_r = \frac{Tr_{\text{brute-force}} - Tr_{\text{QUBO\_assisted}}}{Tr_{\text{brute-force}}} \cdot 100\,; \tag{28}$$

- The time required for solving the QUBO problem (t_q);
- The time required for solving the QUBO problem with the digital annealer normalized for the possible degree of parallelization ($t_{q_n} = \frac{t_q}{v_q}$);
- The time required for the overall optimization (t_t).

The first two figures of merit (n_c and p_{sel}) evaluate the effectiveness of different QUBO formulations in reducing the user set and selecting optimal subsets. Moreover, the comparison between $n_{c,\text{init}}$ and n_c quantifies the reduction in the number of **convex solver calls** required when employing the QUBO-assisted approach. Furthermore, the proposed QUBO formulations were compared in terms of **binary variables** amount v_q and **time required for their resolution** (QUBO-only) t_q.

Furthermore, the DA solver could be reasonably compared with the others by introducing a post-processing **normalized time** t_{q_n}. In fact, since this approach gives a computational advantage only involving parallelization (on its hand achievable with the required amount v_q only with dedicated hardware and not in software), the performed post-processing operation permits to estimate, with a software functionally-equivalent model, the QUBO solving time expected with the desired degree of parallel computation.

Finally, the **relative percentage TR error** ϵ_r and the **overall execution time** t_t compare the proposed approaches and the brute-force one in terms of both results quality and quickness.

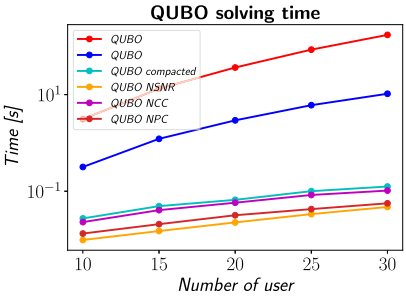
5.3 Comparison of different strategies

Table 1 reports some results obtained with the synthetic dataset, changing the number of involved users N_{UE}, with the original **brute-force approach** and the **QUBO-assisted** strategy, considering each proposed QUBO formulation solved with simulated annealing. This solver was chosen because it can be launched locally without requiring time-limited accesses to remote machines and can currently handle more binary variables than a quantum annealer, thus allowing the validation of the proposed QUBO model. For each N_{UE}, five different problems solved ten times were considered for averaging the results. The QUBO-assisted approach significantly reduces execution time while maintaining an acceptable relative error (ϵ_r). This advantage becomes more pronounced with an increasing number of users (as shown in Fig. 11). The variable reduction of QUBO formulation allows an appreciable reduction of the time required for solving it, which results in a reduction of the total execution time, even if less evident because, in the QUBO-assisted approach, t_q has a limited weight on t_t.

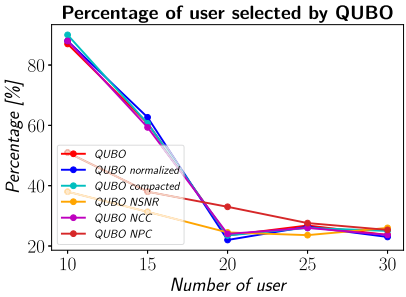
The complexity advantage of the proposed QUBO-assisted strategy is highlighted in Fig. 12a, reporting the number of saved convex solver calls as a function of the number of users. Additionally, Fig. 12b compares the number of convex optimizations required by the brute-force and the proposed QUBO-assisted approaches, varying the number of users. It is evident that the complexity reduction increases with the problem dimension, consistently with the time advantage demonstrated in Fig. 11.

Table 2 shows the results obtained by solving two synthetically generated problems for each N_{UE} ten times with the original **brute-force approach** and with the proposed **Naïve** and **QUBO-assisted** strategies, considering the first three formulations of Sect. 4.2. It is possible to notice that the **Naïve** approach allows a more relevant reduction of the execution time than the QUBO-assisted one but at the expense of the quality of the results (Fig. 13). Moreover, it is possible to ascertain that the total execution time t_t is usually increasing with the number of users for all the QUBO solvers and that p_{sel} usually tends to decrease with higher N_{UE}. The apparent exceptions in p_{sel} trends are expected to be overcome by increasing the number of repetitions for averaging the results and by improving the choice of QUBO penalty weights with a feedback mechanism. These trends also allow one to justify the t_t values obtained for Normalized QUBO and Compacted QUBO, which are, with $N_{\mathrm{UE}} = [35; 40]$, lower than those obtained with $N_{\mathrm{UE}} \leq 30$. In fact, the contribution of the QUBO solving time t_q in the calculation of t_t is, in these cases, more negligible than in the other QUBO-assisted strategies, so the total execution time mainly depends on the brute-force exploration time, which is lower for $N_{\mathrm{UE}} = [35; 40]$ because of a lower n_c.

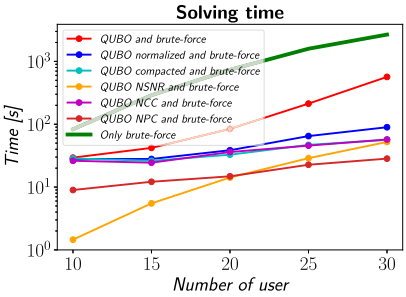
Table 3 presents the results obtained with data retrieved from the TIM S.p.A. simulator with all the analyzed approaches, with QUBO problems solved with simulated annealing. In particular, the results of the triplet of BSs 1-6-20 ($N_{\mathrm{UE}} = 30$) of Fig. 10 for different sub-channels are reported. According to the performed tests, the same overall trends in the results can be observed for other triplets and sub-channels. Also in this case, the **Naïve** approach gives the lowest execution time but the worst results, since the obtained transmission rates are about 40% lower than those obtained with the



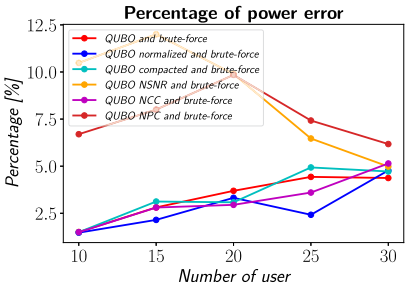
(a) QUBO solving time as a function of the number of users, exploiting simulated annealing solver.



(b) Percentage of users selected as a function of the number of users by QUBO formulations, exploiting simulated annealing solver.

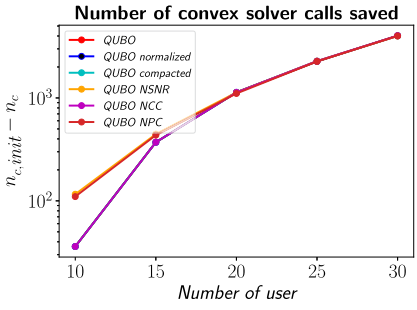


(c) Execution time of brute-force approach and QUBO-assisted approaches as a function of the number of users, exploiting simulated annealing solver for QUBO and splitting conic solver for convex optimization.

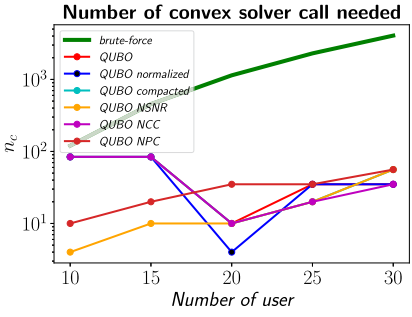


(d) Relative power error as a function of the number of users, exploiting simulated annealing solver.

Fig. 11 Graphical view of the results of Table 1 (Color figure online)



(a) Number of saved convex solver execution due to QUBO pre-selection of the users



(b) Number of convex solver execution required

Fig. 12 Reduction of the computational complexity achieved with the proposed QUBO-assisted user selection (Color figure online)

Table 1 Results obtained with the synthetic dataset, changing the number of involved users N_{UE}, with the original **brute-force approach** and the **QUBO-assisted** strategy, considering each obtained QUBO formulation solved with simulated annealing

N_{UE}	Brute-force	First QUBO			Normalized QUBO			Compacted QUBO			QUBO NCC			QUBO NPC			QUBO NSNR		
	$n_{c,\mathrm{init}}$	v_q	ϵ_r [%]	n_c	v_q	ϵ_r [%]	n_c	v_q	ϵ_r [%]	n_c	v_q	ϵ_r [%]	n_c	v_q	ϵ_r [%]	n_c	v_q	ϵ_r [%]	n_c
10	120	2550	1.5	84	390	1.5	84	30	1.5	84	**30**	**1.5**	**84**	30	6.7	10	30	10.5	4
15	455	9600	2.8	84	1410	2.1	84	45	3.1	84	**45**	**2.8**	**84**	45	8.0	20	45	12.0	10
20	1140	24,000	3.7	10	3475	3.3	4	60	3.1	10	**60**	**2.9**	**10**	60	9.8	35	60	9.9	10
25	2300	48,375	4.4	35	6975	2.4	35	75	4.9	20	**75**	**3.6**	**20**	75	7.4	35	75	6.5	20
30	4060	85,350	4.4	35	12,270	4.8	35	**90**	**4.7**	**56**	90	5.1	35	90	6.2	56	90	5.0	56

N_{UE}	Brute-force	First QUBO			Normalized QUBO			Compacted QUBO			QUBO NCC			QUBO NPC			QUBO NSNR		
	t_t [s]	t_q [s]	t_t [s]	p_{sel} [%]	t_q [s]	t_t [s]	p_{sel} [%]	t_q [s]	t_t [s]	p_{sel} [%]	t_q [s]	t_t [s]	p_{sel} [%]	t_q [s]	t_t [s]	p_{sel} [%]	t_q [s]	t_t [s]	p_{sel} [%]
10	83.4	3.11	29.4	87	0.32	28.0	88	0.03	28.5	90	**0.02**	**9.0**	**88**	0.01	26.2	51	0.01	1.5	38
15	289.4	13.33	42.0	61	1.22	28.0	63	0.05	26.2	61	**0.04**	**21.4**	**59**	0.02	12.1	38	0.01	5.5	31
20	736.3	36.40	84.3	24	2.93	38.6	22	0.07	32.8	24	**0.05**	**36.1**	**24**	0.03	14.8	33	0.02	14.2	25
25	1589.7	85.32	212.9	27	6.04	64.8	26	0.10	46.9	26	**0.08**	**45.6**	**26**	0.04	22.7	28	0.03	28.8	24
30	2671.5	172.40	566.9	24	10.41	89.6	23	**0.13**	**56.3**	**25**	0.10	45.6	24	0.06	28.4	25	0.05	52.2	26

Bold values indicate the best results

For each N_{UE}, five different problems solved ten times were considered for averaging the results. In this table and in the following ones, n_c is the **number of users combinations to be brute-force explored after QUBO preliminary selection**, $n_{c,\mathrm{init}}$ is the **number of users combinations to be explored with the brute-force exploration (without QUBO preliminary selection)**, v_q is the **total number of QUBO variables**, ϵ_r is the **relative percentage TR error**, p_{sel} is the **scaling of the selected users from the original set obtained with QUBO**, t_t is the **overall execution time**, and t_q is the **QUBO solving time**

Table 2 Results obtained by solving two randomly generated problems for each N_{UE} ten times with the original **brute-force approach**, the proposed **Naïve** and the **QUBO-assisted** strategies (QUBO solved with simulated annealing), considering the first three formulations of Sect. 4.2

N_{UE}	Brute-force		Naïve		First QUBO						Normalized QUBO						Compacted QUBO					
	$n_{c,\text{init}}$	t_t[s]	t_t[s]	ϵ_r [%]	v_q	p_{sel}[%]	n_c	t_q[s]	t_t[s]	ϵ_r [%]	v_q	p_{sel}[%]	n_c	t_q[s]	t_t[s]	ϵ_r [%]	v_q	p_{sel}[%]	n_c	t_q[s]	t_t[s]	ϵ_r [%]
10	120	183.14	0.03	61.01	2550	54	10	1.58	24.77	19.82	390	54	10	0.84	24.42	19.82	**30**	**54**	**10**	**0.06**	**23.87**	**19.82**
15	455	689.03	0.11	64.11	9600	46	12	5.75	73.94	15.66	1410	46	12	0.84	62.03	15.66	**45**	**46**	**12**	**0.06**	**55.80**	**15.66**
20	1140	1741.08	0.28	62.16	24,000	34	35	15.13	106.00	16.29	3480	34	35	6.56	68.41	16.29	**60**	**32**	**20**	**0.13**	**59.14**	**16.29**
25	2300	3310.42	0.54	63.66	48,375	28	35	30.06	216.82	14.97	6975	28	35	13.88	90.09	14.97	**75**	**28**	**35**	**0.17**	**69.36**	**14.97**
30	4060	5857.01	0.93	63.48	85,350	30	84	60.86	654.29	15.78	12,270	34	120	23.64	164.01	15.78	**90**	**30**	**84**	**0.24**	**155.87**	**15.78**
35	6545	9361.08	1.49	63.49	137,550	15	10	104.61	1473.30	15.15	19,740	18	20	38.08	135.76	15.15	**105**	**15**	**10**	**0.30**	**29.33**	**15.15**
40	9880	13675.75	2.15	62.93	207,600	13	10	159.53	3297.47	14.22	29,760	14	20	53.48	195.68	14.22	**120**	**13**	**10**	**0.37**	**48.08**	**14.22**

Bold values indicate the best results

Table 3 Results obtained by solving the problems with data retrieved from the TIM S.p.A. simulator, with all the analyzed approaches

N_c	Brute-force	Naïve	First QUBO			Normalized QUBO			Compacted QUBO			QUBO NCC			QUBO NPC			QUBO NSNR		
	$n_{c,\text{init}}$	ϵ_r[%]	v_q	ϵ_r[%]	n_c	v_q	ϵ_r[%]	n_c	v_q	ϵ_r[%]	n_c	v_q	ϵ_r[%]	n_c	v_q	ϵ_r[%]	n_c	v_q	ϵ_r[%]	n_c
1	4060	42.3	5025	12.0	1140	802	12.0	1540	90	12.0	1140	90	12.0	4060	90	7.1	56	**90**	**6.0**	**35**
2	4060	43.1	4804	12.0	1140	768	12.0	1330	90	12.0	1330	90	12.0	3654	90	5.1	56	**90**	**2.8**	**20**
3	4060	41.5	4458	12.4	1140	716	12.4	1540	90	12.4	1140	90	12.4	3276	90	5.1	56	**90**	**2.9**	**35**

N_c	Brute-force	Naïve	First QUBO			Normalized QUBO			Compacted QUBO			QUBO NCC			QUBO NPC			QUBO NSNR		
	t_t[s]	t_t[s]	t_q[s]	t_t[s]	p_{sel}[%]	t_q[s]	t_t[s]	p_{sel}[%]	t_q[s]	t_t[s]	p_{sel}[%]	t_q[s]	t_t[s]	p_{sel}[%]	t_q[s]	t_t[s]	p_{sel}[%]	t_q[s]	t_t[s]	p_{sel}[%]
1	6040	1	13.6	1977	67	1.2	2697	74	0.3	1925	66	0.1	7060	100	0.1	98	26	**0.1**	**61**	**22**
2	6255	1	13.6	1924	66	1.2	2380	70	0.3	2507	71	0.2	6538	97	0.1	89	25	**0.1**	**49**	**21**
3	6051	1	12.1	2127	68	1.1	2627	73	0.2	2023	67	0.1	5698	94	0.1	117	27	**0.1**	**63**	**22**

Bold values indicate the best results

QUBO problems were solved with simulated annealing. In particular, the results of the triplet of BSs 1-6-20 for different sub-channels (identifier N_c) are reported

Table 4 QUBO-assisted results obtained with a classical (SA), a quantum (QA) and a quantum-inspired (DA) solvers

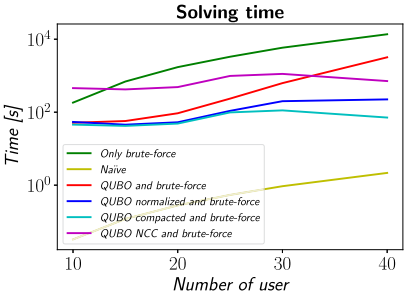
Simulated Annealing																		
N_{UE}	First QUBO			Normalized QUBO			Compacted QUBO			QUBO NCC			QUBO NPC			QUBO NSNR		
	ϵ_r[%]	t_q[s]	p_{sel}[%]	ϵ_r	t_q[s]	p_{sel}[%]	ϵ_r[%]	t_q[s]	p_{sel}[%]	ϵ_r[%]	t_q[s]	p_{sel}[%]	ϵ_r[%]	t_q[s]	p_{sel}[%]	ϵ_r[%]	t_q[s]	p_{sel}[%]
4	**7.9**	0.004	75	**7.9**	0.004	75	10.4	0.002	75	12.9	0.002	75	10.8	0.002	85	**7.9**	0.001	75
5	**0.2**	0.012	92	**0.2**	0.004	92	**0.2**	0.002	88	**0.2**	0.002	84	5.2	0.002	80	**0.2**	0.002	60
6	**2.6**	0.025	100	2.6	0.011	97	**2.6**	0.004	100	2.6	0.004	100	**2.6**	0.002	73	**0.8**	0.002	80
7	5.5	0.041	100	5.5	0.11	100	5.5	0.005	100	5.5	0.006	100	3.0	0.002	66	**1.5**	0.002	54
8	**2.6**	0.057	38	**2.5**	0.018	40	**2.6**	0.003	38	**2.6**	0.004	38	**8.05**	0.002	65	5.7	0.002	75
9	30.3	0.84	33	30.3	0.027	33	30.3	0.003	33	30.3	0.003	33	18.0	0.002	60	16.9	0.003	60
10	**4.0**	0.110	64	**4.0**	0.039	66	**4.0**	0.006	60	**4.0**	0.006	66	13.1	0.004	54	17.8	0.006	34
11	**0.5**	0.200	55	1.0	0.061	62	0.1	0.006	55	1.0	0.003	60	4.0	0.002	49	17.6	0.010	31
12	**1.7**	0.293	58	**1.7**	0.093	57	**1.7**	0.005	58	**1.7**	0.008	58	2.2	0.005	48	21.3	0.005	30
13	**1.4**	0.420	32	1.4	0.141	31	1.4	0.009	31	1.4	0.008	34	11.6	0.005	41	19.6	0.007	28
14	2.2	0.559	40	**2.1**	0.265	43	**2.1**	0.011	43	2.2	0.009	40	6.4	0.007	41	3.7	0.006	37
15	4.1	0.643	43	4.1	0.222	43	4.1	0.010	40	4.1	0.10	41	10.8	0.007	37	5.7	0.006	25
Digital Annealing																		
N_{UE}	First QUBO			Normalized QUBO			Compacted QUBO			QUBO NCC			QUBO NPC			QUBO NSNR		
	ϵ_r[%]	t_{qn}[s]	p_{sel}[%]	ϵ_r[%]	t_{qn}[s]	p_{sel}[%]	ϵ_r[%]	t_{qn} [s]	p_{sel}[%]	ϵ_r[%]	t_{qn}[s]	p_{sel}[%]	ϵ_r[%]	t_{qn}[s]	p_{sel}[%]	ϵ_r[%]	t_{qn} [s]	p_{sel}[%]
4	7.9	0.038	75	7.9	0.194	15	10.4	0.189	15	**7.9**	0.200	75	**4.7**	0.185	90	7.9	0.161	15
5	0.2	0.047	84	0.2	0.233	18	0.2	0.229	17	0.2	0.218	88	9.8	0.218	68	8.1	0.189	12
6	2.6	0.051	83	2.6	0.235	19	2.6	0.217	20	2.6	0.220	100	9.3	0.212	70	14.9	0.222	16
7	4.5	0.052	97	**3.4**	0.212	20	4.5	0.211	20	5.5	0.196	97	0.7	0.184	69	6.6	0.201	11
8	5.5	0.052	57	4.6	0.236	8	3.7	0.232	7.5	5.7	0.231	43	11.3	0.212	60	16.4	0.230	15
9	**8.7**	0.052	55.6	7.8	0.236	6.7	8.9	0.232	6.7	3.6	0.232	51	**9.2**	0.212	58	**11.7**	0.230	12
10	6.0	0.053	66	4.2	0.253	13	10.5	0.212	12	4.1	0.214	58	**5.9**	0.194	72	**8.8**	0.212	6.8

Table 4 continued

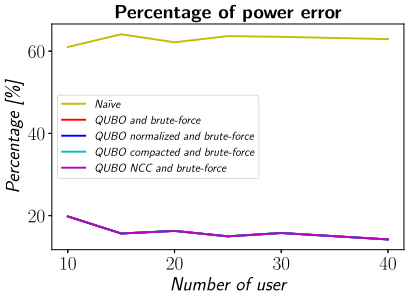
Digital Annealing																	
N_{UE}	First QUBO			Normalized QUBO			Compacted QUBO			QUBO NCC			QUBO NPC			QUBO NSNR		
	ϵ_r[%]	t_{qn}[s]	p_{sel}[%]	ϵ_r[%]	t_{qn}[s]	p_{sel}[%]	ϵ_r[%]	t_{qn} [s]	p_{sel}[%]	ϵ_r[%]	t_{qn}[s]	p_{sel}[%]	ϵ_r[%]	t_{qn}[s]	p_{sel}[%]	ϵ_r[%]	t_{qn}[s]	p_{sel}[%]
11	2.4	0.059	78	**0.2**	0.266	11	**0.0**	0.186	12	1.6	0.185	72	**3.5**	0.171	55	3.4	0.184	6
12	2.0	0.055	73	2.0	0.256	11	1.8	0.220	12	2.1	0.220	71	**1.8**	0.208	55	13.3	0.215	6
13	2.1	0.064	63	**1.3**	0.270	6	4.7	0.203	6	2.6	0.220	57	**2.2**	0.217	57	**5.2**	0.207	6
14	**2.1**	0.099	59	2.3	0.260	9	2.4	0.215	9	2.7	0.215	64	**2.5**	0.208	60	**3.0**	0.223	7
15	**1.0**	0.085	71	1.9	0.625	8	3.3	0.236	8	**1.1**	0.212	59	**2.8**	0.189	48	**0.8**	0.213	5
Quantum Annealing																		
N_{UE}	First QUBO			Normalized QUBO			Compacted QUBO			QUBO NCC			QUBO NPC			QUBO NSNR		
	ϵ_r[%]	t_q[s]	p_{sel}[%]	ϵ_r[%]	t_q[s]	p_{sel}[%]	ϵ_r[%]	t_q[s]	p_{sel}[%]	ϵ_r[%]	t_q[s]	p_{sel}[%]	ϵ_r[%]	t_q[s]	p_{sel}[%]	ϵ_r[%]	t_q[s]	p_{sel}[%]
4	7.9	5.069	75	7.9	3.302	75	**7.9**	0.002	75	7.9	2.202	75	6.4	2.165	90	7.9	2.182	75
5	0.2	16.833	100	0.2	2.487	88	0.2	0.002	96	0.2	10.072	88	**1.3**	2.138	80	8.2	6.586	60
6	3.6	103.727	67	**1.2**	4.471	90	3.0	0.004	80	**1.2**	5.65	80	3.9	2.729	73	3.4	2.857	60
7	**0.6**	95.77	69	4.5	4.782	91	**4.5**	0.005	91	**3.4**	2.766	43	**0.3**	3.503	83	4.8	15.441	69
8	–	–	–	4.1	7.326	57	3.8	0.003	63	4.9	3.9622	63	14.7	4.213	60	**3.9**	3.525	88
9	–	–	–	**2.7**	14.761	64	**1.64**	0.003	71	**3.2**	3.609	80	16.0	5.520	49	23.0	3.692	30
10	–	–	–	14.5	33.501	56	6.2	0.006	60	4.0	8.540	68	10.0	2.923	52	12.7	12.997	30
11	–	–	–	0.4	62.506	65	0.1	0.006	69	**0.4**	3.828	73	5.7	14.636	49	8.1	3.491	31
12	–	–	–	1.9	114.926	52	6.8	0.005	62	1.8	3.370	78	8.7	3.172	43	**2.3**	3.481	37
13	–	–	–	1.3	227.680	68	**1.3**	0.009	85	**1.3**	3.503	80	7.4	3.812	45	12.4	8.128	28
14	–	–	–	2.5	451.048	56	2.2	0.011	68	**2.1**	4.400	71	6.2	4.902	40	19.4	4.861	26
15	–	–	–	**0.2**	260.662	53	**2.6**	0.009	77	2.6	4.183	73	7.7	5.265	39	8.4	4.536	29

Bold values indicate the best results

The metric t_{qn} is the post-processing normalized time associated with the Digital Annealing solver. Since the focus of this table is to compare the effectiveness of different solvers in evaluating QUBO, only the QUBO time (for considering the solver speed), the error rate, and the percentage of selected users (both for estimating the quality of the obtained solution) are reported



(a) Execution time of brute-force approach and QUBO-assisted approaches as a function of the number of users, exploiting simulated annealing solver for QUBO and splitting conic solver for convex optimization.



(b) Relative power error as a function of the number of users, exploiting simulated annealing solver.

Fig. 13 Graphical view of the results of Table 2

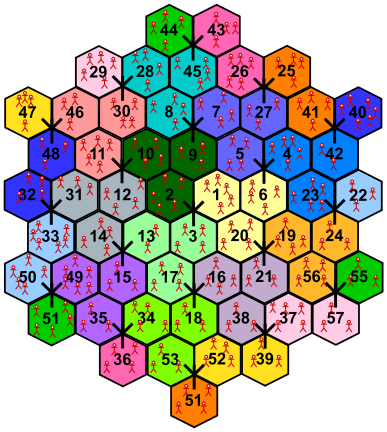
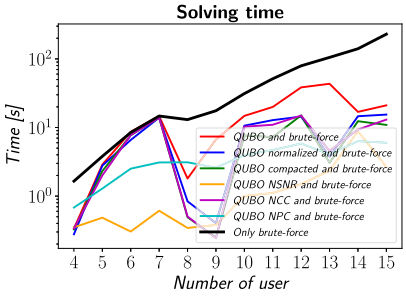


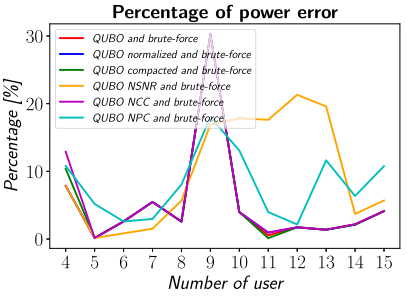
Fig. 14 TIM S.p.A. MATLAB-based MIMO simulator data. BSs are separated in 19 triplets and simulations can be performed on each of the 52 sub-channels

brute-force exploration. On the other hand, apart from NCC, QUBO models allow a reduction of the selected users for brute-force exploration of more than 25%, thus also reducing t_t. From the transmission rate perspective, ϵ_r never exceeds 12.4%, which is about three times lower than the value obtained with the **Naïve** approach. In general, the QUBO NSNR seems to be the best approach, since it achieves the best trade-off between low execution time (in terms of p_{sel} or t_t) and transmission rate (ϵ_r, never exceeding 6%) (Fig. 14).

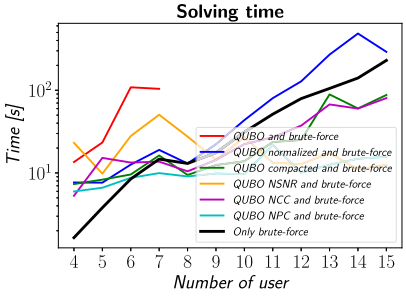
Finally, Table 4 proves how the chosen QUBO solver influences the solution time and the quality of the results (Fig. 15). Results are associated with tests performed by solving one synthetically generated problem for each N_{UE} ten times. The synthetic data were chosen because they permit to granularly change the number of users and the application of QA also for the First QUBO formulation. It is possible to recognize the flexibility of QUBO formulation, which can be solved by a **classical optimizer** (SA), a **quantum** one (QA) and a **quantum-inspired** one (DA). In evaluating QA results,



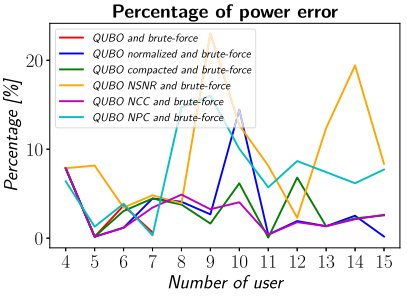
(a) Execution time of brute-force and QUBO-assisted approaches as a function of the number of users, exploiting simulated annealing.



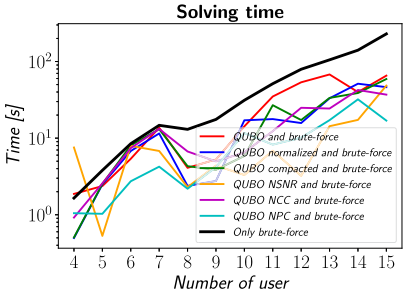
(b) Relative power error as a function of the number of users, exploiting simulated annealing.



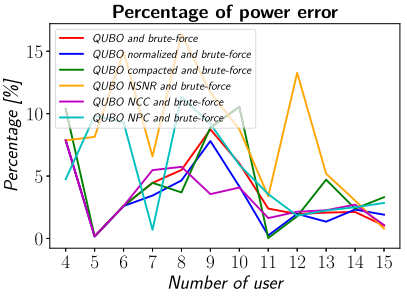
(c) Execution time of brute-force and QUBO-assisted approaches as a function of the number of users, exploiting quantum annealer.



(d) Relative power error as a function of the number of users, exploiting quantum annealer.



(e) Execution time of brute-force and QUBO-assisted approaches as a function of the number of users, exploiting digital annealer.



(f) Relative power error as a function of the number of users, exploiting digital annealer.

Fig. 15 Graphical view of the results of Table 4

it is necessary to consider that the solving time also includes the one required for minor embedding (which increases with the number of involved variables), for remote connection and any waiting one spent in the execution queue (the effective annealing time was set to 100μs). Furthermore, **the First QUBO formulation was solved by QA only until the number of users was lower than eight** because current devices cannot embed larger problems. The best solver (in terms of the quality of the results) depends on the QUBO formulation considered. This is coherent with the expectation because, as already discussed when Fig. 2 of Sect. 3.2 has been commented, modifying the QUBO formulation changes the energy profile of the problem and, consequently, the effectiveness of classical and quantum explorations [13, 15]. According to the performed tests, SA is most frequently characterized by the lowest ϵ_r with the QUBO formulations with all constraints. This could be traced back to the fact that the presence of auxiliary variables or the current choice of λ values for the constraints penalty functions imply a problem's profile characterized by wide barriers, not necessarily high. On the other hand, the reduction of variables obtained with normalization and the removal of some constraints imply an overall quadratic profile with narrower barriers, which are more suitable for QA and DA solvers. In particular, QA seems to be the best solver for QUBO NCC formulation, while DA is the best for NPC and NSNR formulations.

It is also interesting to observe that QA and DA achieve, for low N_{UE} values, results in terms of ϵ_r very close to SA with QUBO formulations without constraints removals, usually with a lower p_{sel}, thus having a smaller reduced user set and a consequent brute-force exploration with less iterations. For this reason, in a performance-complexity trade-off evaluation, the quantum and quantum-inspired methods can be considered competitive with respect to a classical QUBO solver.

6 Conclusions

This work proposed new strategies for optimizing the JT mechanism, in terms of TR, according to the potential users and the constraints on BS transmitting power. The results show that the QUBO-assisted mechanism achieves a better trade-off between execution time and the quality of the solution, compared to the brute-force and Naïve approaches. Indeed, the first gives the best solution but its execution time scales factorially, while the other gives the lowest solving time but gives results with poor quality. At the same time, QUBO-assisted solvers guarantee a significant reduction of the execution time without excessively degrading the performance. Coherently with the expectation, the Compacted, NPC and NSNR QUBO formulations, having the lowest number of binary variables, give the lowest execution time with performance comparable to First and Normalized formulations. Moreover, the performance obtained employing SA, QA, and DA solvers was compared, showing that the best solver strictly depends on QUBO formulation.

QUBO-assisted method can be undoutedly improved, exploiting alternative techniques for λ tuning. For example, **scaled-sequential scaling** and **binary-search scaling** [28] can be considered when the identified candidate users set is too low. Moreover, a feedback mechanism should be implemented (green arrow in Fig. 10) for

adjusting the penalty weights of the different contributions in the QUBO formulation, exploiting the previous experience and performing the QUBO optimization with the p_s values computed in the previous time slot, instead of generating them randomly. This approach can be considered reasonable assuming that the users' spatial motion should be limited during the time-slots of evaluation of the optimization problem, thus not dramatically changing their contribution in the channel matrix H. Finally, it is interesting to remind that the state of art presents some methods for a preliminary λ estimation [34–37], thus allowing reasonable initial weights for the constraints of the optimization problem.

The performed analysis with software prototypes lays the fundamentals, in the long term perspective, for a distributed computing system which can optimize in a quite small time window the JT mechanism. In particular, the idea is to have a hardware (Field-Programmable Gate Array or an Application-Specific Integrated Circuit [38]) solver in each BS or in each group of interacting BSs, which allows the optimization of the group of collaborating transmitting antennas. The design of a specialized architecture should guarantee the reactivity of the system, acquire information from sensors and work in a relatively small window of time. The flexibility of QUBO allows for the use of classical, quantum-emulated, or quantum-inspired solvers. Remote connections to quantum annealers are also anticipated as scaling improvements are expected.

Improvements are not strictly related to the definition of the QUBO model, but also concern benchmarking. In fact, attempts to compare the proposed QUBO-assisted method with alternative optimization formulations, such as linear programming [39, 40], and the solvers with other classical approaches, such as CPLEX and Gurobi, can be done.

The proposed final distributed optimization mechanism should significantly improve the communication quality thanks to JT, thus guaranteeing a more comfortable social interaction between users of wireless mobile networks.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request and after finding a proper agreement with TIM S.p.A.

Declarations

Conflict of interest The authors declared that they have no conflict of interest to this work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Jensen, M.A., Wallace, J.W.: A review of antennas and propagation for mimo wireless communications. IEEE Trans. Antennas Propag. **52**(11), 2810–2824 (2004). https://doi.org/10.1109/TAP.2004.835272
2. Bliss, D.W., Forsythe, K.W., Chan, A.M.: MIMO wireless communication. Linc. Lab. J. **15**(1), 97–126 (2005)
3. Biglieri, E., Calderbank, R., Constantinides, A., Goldsmith, A., Paulraj, A., Poor, H.V.: MIMO Wireless Communications. Cambridge University Press (2007). https://doi.org/10.1017/CBO9780511618420
4. Poularikas, A.D.: MIMO System Technology for Wireless Communications. CRC Press (2018). https://doi.org/10.1201/9781315222011
5. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press. https://doi.org/10.1017/CBO9780511976667
6. Suriya, M.: Machine learning and quantum computing for 5G/6G communication networks-a survey. Int. J. Intell. Netw. **3**, 197–203 (2022). https://doi.org/10.1016/j.ijin.2022.11.004
7. Phillipson, F.: Quantum computing in telecommunication-a survey. Mathematics **11**(15), 3423 (2023). https://doi.org/10.3390/math11153423
8. Wilson, B., Goh, E., Guillaume, A., Alimo, R., Claudet, T., Venkataram, H.: Automating antenna scheduling problems using quantum computing and deep reinforcement learning. In: IGARSS 2022–2022 IEEE International Geoscience and Remote Sensing Symposium, pp. 4915–4918 (2022). https://doi.org/10.1109/IGARSS46834.2022.9884342
9. Barillaro, G., Boella, A., Gandino, F., Vakili, M.G., Giusto, E., Mondo, G., Montrucchio, B., Scarabosio, A., Scionti, A., Terzo, O., Vitali, G.: Comparison of heuristic approaches to pci planning for quantum computers. In: 2023 IEEE International Conference on Consumer Electronics (ICCE), pp. 1–6 (2023). https://doi.org/10.1109/ICCE56470.2023.10043394
10. Li, J., Chen, X., Botella, C., Svensson, T., Eriksson, T.: Resource allocation for ofdma systems with multi-cell joint transmission. In: 2012 IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 179–183 (2012). https://doi.org/10.1109/SPAWC.2012.6292883
11. Glover, F., Kochenberger, G., Du, Y.: A tutorial on formulating and using qubo models. arXiv preprint arXiv:1811.11538 (2018) https://doi.org/10.48550/arXiv.1811.11538
12. Combarro, E.F., González-Castillo, S., Di Meglio, A.: A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms. Packt Publishing Ltd (2023)
13. Volpe, D., Cirillo, G.A., Zamboni, M., Turvani, G.: Integration of simulated quantum annealing in parallel tempering and population annealing for heterogeneous-profile qubo exploration. IEEE Access **11**, 30390–30441 (2023). https://doi.org/10.1109/ACCESS.2023.3260765
14. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse ising model. Phys. Rev. E **58**, 5355–5363 (1998). https://doi.org/10.1103/PhysRevE.58.5355
15. Chancellor, N.: Modernizing quantum annealing using local searches. New J. Phys. **19**(2), 023024 (2017). https://doi.org/10.1088/1367-2630/aa59c4
16. Le Bellac, M.: Quantum Physics. Cambridge University Press (2011)
17. Denchev, V.S., Boixo, S., Isakov, S.V., Ding, N., Babbush, R., Smelyanskiy, V., Martinis, J., Neven, H.: What is the computational value of finite-range tunneling? Phys. Rev. X **6**(3), 031015 (2016). https://doi.org/10.1103/PhysRevX.6.031015
18. Boothby, K., Bunyk, P., Raymond, J., Roy, A.: Next-Generation Topology of D-Wave Quantum Processors (2020)
19. Aramon, M., Rosenberg, G., Valiante, E., Miyazawa, T., Tamura, H., Katzgraber, H.G.: Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. Front. Phys. **7**, 48 (2019). https://doi.org/10.3389/fphy.2019.00048
20. Goto, H., Tatsumura, K., Dixon, A.R.: Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems. Sci. Adv. **5**(4), 2372 (2019). https://doi.org/10.1126/sciadv.aav2372
21. Tilly, J., Chen, H., Cao, S., Picozzi, D., Setia, K., Li, Y., Grant, E., Wossnig, L., Rungger, I., Booth, G.H., Tennyson, J.: The Variational Quantum Eigensolver: a review of methods and best practices. arXiv (2021). https://doi.org/10.48550/ARXIV.2111.05176
22. Zhou, L., Wang, S.-T., Choi, S., Pichler, H., Lukin, M.D.: Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. Phys. Rev. X **10**, 021067 (2020). https://doi.org/10.1103/PhysRevX.10.021067

23. Giuffrida, L., Volpe, D., Cirillo, G.A., Zamboni, M., Turvani, G.: Engineering grover adaptive search: exploring the degrees of freedom for efficient qubo solving. IEEE J. Emerg. Sel. Top. Circuits Syst. **12**(3), 614–623 (2022). https://doi.org/10.1109/JETCAS.2022.3202566
24. Johnson, M.W., Amin, M.H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., Harris, R., Berkley, A.J., Johansson, J., Bunyk, P.: Quantum annealing with manufactured spins. Nature **473**(7346), 194–198 (2011). https://doi.org/10.1038/nature10012
25. Humble, T.S., Thapliyal, H., Muñoz-Coreas, E., Mohiyaddin, F.A., Bennink, R.S.: Quantum computing circuits and devices. IEEE Des. Test **36**(3), 69–94 (2019). https://doi.org/10.1109/MDAT.2019.2907130
26. Verma, A., Lewis, M.: Variable reduction for quadratic unconstrained binary optimization. arXiv preprint arXiv:2105.07032 (2021) https://doi.org/10.48550/arXiv.2105.07032
27. Iosue, J.T.: qubovert Documentation. [Online at https://qubovert.readthedocs.io/en/stable/,; accessed 10-May-2022] (2019)
28. García, M.D., Ayodele, M., Moraglio, A.: Exact and sequential penalty weights in quadratic unconstrained binary optimisation with a digital annealer. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 184–187 (2022). https://doi.org/10.1145/3520304.3528925
29. O'Donoghue, B.: Operator splitting for a homogeneous embedding of the linear complementarity problem. SIAM J. Optim. **31**(3), 1999–2023 (2021). https://doi.org/10.1137/20M1366307
30. Diamond, S., Boyd, S.: Cvxpy: A python-embedded modeling language for convex optimization. J. Mach. Learn. Res. **17**(1), 2909–2913 (2016). https://doi.org/10.5555/2946645.3007036
31. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983). https://doi.org/10.1126/science.220.4598.671
32. Intel Xeon Gold 6134 Processor - Product Specification. [Online] https://ark.intel.com/content/www/us/en/ark/products/120493/intel-xeon-gold-6134-processor-24-75m-cache-3-20-ghz.html. Accessed 25 Oct 2021
33. 3gpp tr 38.901 v16.1.0 "3rd generation partnership project; technical specification group radio access network; study on channel model for frequencies from 0.5 to 100 ghz (release 16) (2019). [Online] https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/16.01.00_60/tr_138901v160100p.pdf. Accessed 4 Dec 2022
34. Boros, E., Hammer, P.L.: Pseudo-boolean optimization. Discret. Appl. Math. **123**(1–3), 155–225 (2002). https://doi.org/10.1016/S0166-218X(01)00341-9
35. Boros, E., Hammer, P.L., Tavares, G.: Preprocessing of unconstrained quadratic binary optimization. Technical report, Technical Report RRR 10-2006, RUTCOR (2006)
36. Verma, A., Lewis, M.: Penalty and partitioning techniques to improve performance of qubo solvers. Discret. Optim. **44**, 100594 (2022). https://doi.org/10.1016/j.disopt.2020.100594
37. Ayodele, M.: Penalty weights in qubo formulations: Permutation problems. In: European Conference on Evolutionary Computation in Combinatorial Optimization (Part of EvoStar), pp. 159–174. Springer (2022). https://doi.org/10.1007/978-3-031-04148-8_11
38. Tsukamoto, S., Takatsu, M., Matsubara, S., Tamura, H.: An accelerator architecture for combinatorial optimization problems. Fujitsu Sci. Tech. J. **53**(5), 8–13 (2017)
39. Karloff, H.: Linear Programming. Springer (2008)
40. Dantzig, G.B.: Linear programming. Oper. Res. **50**(1), 42–47 (2002)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Deborah Volpe[1] · Giovanni Amedeo Cirillo[1] · Roberto Fantini[2] · Andrea Boella[2] · Giovanni Mondo[2] · Mariagrazia Graziano[3] · Giovanna Turvani[1]

✉ Deborah Volpe
deborah.volpe@polito.it

Giovanni Amedeo Cirillo
giovanniamedeo.cirillo@st.com

Roberto Fantini
roberto.fantini@telecomitalia.it

Andrea Boella
andrea.boella@telecomitalia.it

Giovanni Mondo
giovanni.mondo@telecomitalia.it

Mariagrazia Graziano
mariagrazia.graziano@polito.it

Giovanna Turvani
giovanna.turvani@polito.it

[1] Department of Electronics and Telecommunications, Politecnico di Torino, Corso Duca degli Abruzzi, Turin, Italy

[2] Research and Development, TIM S.p.A, Via Reiss Romoli Guglielmo, 10148 Turin, Italy

[3] Department of Applied Science and Technology, Politecnico di Torino, Corso Duca degli Abruzzi, 10129 Turin, Italy