IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 11, 2021, accepted September 15, 2021, date of publication September 22, 2021,
date of current version September 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3114543

Evaluation of Quantum Annealer Performance via the Massive MIMO Problem

ZSOLT I. TABI[1,2], ÁDÁM MAROSITS[1,3], ZSÓFIA KALLUS[1], PÉTER VADERNA[1], ISTVÁN GÓDOR[1], (Senior Member, IEEE), AND ZOLTÁN ZIMBORÁS[4,5]

[1]Ericsson Research, 1117 Budapest, Hungary
[2]Department of Programming Languages and Compilers, Eötvös Loránd University, 1117 Budapest, Hungary
[3]Department of Broadband Infocommunications and Electromagnetic Theory, Budapest University of Technology and Economics, 1111 Budapest, Hungary
[4]Department of Analysis, Budapest University of Technology and Economics, 1111 Budapest, Hungary
[5]Quantum Computing and Information Group, Wigner Research Centre for Physics, 1121 Budapest, Hungary

Corresponding authors: Zsolt I. Tabi (zsolt.tabi@ericsson.com), István Gódor (istvan.godor@ericsson.com), and Zoltán Zimborás (zimboras.zoltan@wigner.hu)

This work was supported in part by the Hungarian Quantum Technology National Excellence Program, and in part by the Quantum Information National Laboratory of Hungary. The work of Zsolt I. Tabi and Zoltán Zimborás was supported in part by the Hungarian Quantum Technology National Excellence Program under Project 2017-1.2.1-NKP-2017-00001; and in part by the Hungarian National Research, Development and Innovation Office (NKFIH) within the Quantum Information National Laboratory of Hungary under Grant FK 135220, Grant K124176, and Grant KH129601.

ABSTRACT Quantum annealing offers an appealing route to handle large-scale optimization problems. Existing Quantum Annealing processing units are readily available via cloud platform access for solving Quadratic Unconstrained Binary Optimization (QUBO) problems. In particular, the novel D-Wave Advantage device has been recently released. Its performance is expected to improve upon the previous state-of-the-art D-Wave 2000Q annealer, due to higher number of qubits and the Pegasus topology. Here, we present a comparative study via an ensemble of Maximum Likelihood (ML) Channel Decoder problems for MIMO scenarios in Centralized Radio Access Network (C-RAN) architectures. The main challenge for exact optimization of ML decoders with ever-increasing demand for higher data rates is the exponential increase of the solution space with problem sizes. Since current 5G solutions mainly use approximate methodologies, Kim *et al.* leveraged Quantum Annealing for large MIMO problems with Phase Shift Keying and Quadrature Amplitude Modulation scenarios. Here, we extend their work and analyze experiments for more complex modulations and larger MIMO antenna array sizes. By implementing the extended QUBO formulae on the novel annealer architecture, we uncover the limits of state-of-the-art quantum optimization for the massive MIMO ML decoder. We report on the improvements and discuss the uncovered limiting factors learned from the 64-QAM extension. We include the enhanced evaluation of raw annealer sampling via implementation of post-processing methods in the comparative analysis between D-Wave 2000Q and the D-Wave Advantage system.

INDEX TERMS Channel decoding, graph embedding, massive MIMO, NP-hard optimization, quantum annealing, quantum computing, telecommunication.

I. INTRODUCTION

Quantum Computers can harness the processing capabilities of quantum mechanics to speed up calculations for complex mathematical problems [2]. Although we are yet to achieve universal large-scale quantum computation, today's Noisy Intermediate-Scale Quantum (NISQ) devices can already be used in medium-sized experimental setups. A Quantum Annealer (QA) [3]–[5], one of the promising heuristic devices in this NISQ era, is capable of solving complex optimization problems using thousands of noisy qubits. In this paper, we study the performance of state-of-the-art Quantum Annealers for the telecommunication problem of decoding wireless physical channel transmission using large and massive Multiple Input Multiple Output (MIMO) [6] antenna arrays (see illustration in Fig. 1).

To support high transmission rates, modern wireless access points use spatial multiplexing with multiple antennas to transmit more than one data stream at once. In 5G networks, the application of MIMO antenna arrays is indispensable, however, as we increase the number of antennas, we also need to increase the computational power to be able to decode

The associate editor coordinating the review of this manuscript and approving it for publication was Luyu Zhao.

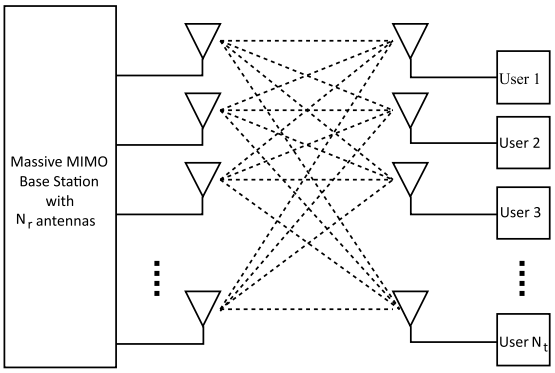


FIGURE 1. Illustration of a Multiple Input Multiple Output (MIMO) antenna array with $N_r \times N_t$ antenna setup.

transmissions at the receiver [7]. This is also due to the complex modulation techniques employed at the transmitter. Maximum Likelihood (ML) is the optimal decoding of received symbols in a MIMO channel, as it is capable of minimizing the probability of bit errors, but it is also known to be NP-hard [8], [9]. Today's commercial massive MIMO antenna systems already contain antenna arrays large enough to face complex decoding problem. E.g., Ericsson's currently available antenna systems have 128 antenna elements (64T64R) in a 2D layout, integrated with up to 64 radio chains and capable of 256-QAM modulation scheme [10]. Furthermore, the size of the future antenna systems is expected to increase, especially when extremely large aperture arrays or holographic massive MIMO will take place [11].

In order to enable practical applicability of ML decoding for large antenna arrays, Kim *et al.* [1] explored the possibility of placing a Quantum Annealer within the data center of a Centralized Radio Access Network (C-RAN) [12] to provide solution to the NP-hard problem while still maintaining high throughput of the ML decoder. This requires formulating the ML decoding problem of the received symbols as a Quadratic Unconstrained Binary Optimization (QUBO) problem [13], [14] making it suitable for a QA. The resulting solution to the optimization problem can simply be mapped back to a bit string according to the constellation diagram of the receiver.

Our first goal is to use an extended methodology derived from [1] and a set of advanced modulation schemes, relevant for high-performance telecommunication scenarios. Hence, the formulae to convert 64-QAM modulated symbols to the Ising spin glass form, presented in [15], is implemented as the highest complexity problem class. Next, a set of experiments with increasing complexity are presented, defined for a comparative analysis between the D-Wave 2000Q and the recently released D-Wave Advantage platform. For the Advantage Quantum Processing Unit (QPU) [16] the number of qubits has increased to 5000, from 2000 of the previous generation, allowing for larger problem mapping. Since the new D-Wave Advantage QPU has not only more qubits but also a topology of significantly higher connectivity, the expectation is that it offers higher quality solutions to more complex QUBO problems as well. Both the increase in problem size to much higher user counts and the progression in problem complexity to more advanced modulation schemes can be tested to reveal the capacity gain limits of the new QPU in complex scenarios.

The structure of this work is as follows. Sec. II presents the theoretical background of Quantum Annealing and the Maximum Likelihood decoding methods. In Sec. III, we present the QUBO formulation of the MIMO channel decoding and its extension to 64-QAM modulation from [15].

In Sec. IV, we investigate methods for embedding QUBO formulated MIMO decoding problems. Next, in Sec. V, we present our experimental results for solving these problems on both the D-Wave 2000Q and the Advantage system. We also compare these results to conventional MIMO decoding techniques. Finally, in Sec. VI, we summarize our results and provide an outlook for potential future work.

II. THEORETICAL BACKGROUND

A. ISING AND QUBO MODELS

The D-Wave Quantum Annealer can solve problems formulated as a QUBO or an Ising spin model. The Ising model describes a physical system of N binary spin variables $s_i \in \{-1, 1\}$ over the following configuration space: $\Omega_N := \{-1, +1\}^{\times N} = \{(s_1, \ldots, s_N) : s_k = \pm 1\}$.

The Ising Hamiltonian defines the energy of the system in a given spin configuration $\mathbf{s} \in \Omega$ via a sum of interaction and local energy terms in the following form:

$$H(\mathbf{s}) = -\frac{1}{2}\sum_{i,j=1}^{N} J_{i,j}s_i s_j - \sum_i h_i s_i \ , \tag{1}$$

where h_i is the *bias* of the ith spin variable and $J_{i,j}$ is the *coupling strength* between variables s_i and s_j. A reformulation of the above energy minimum search is used in [1] with the following notation:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\Omega}\left(\frac{1}{2}\sum_{i,j=1}^{N} g_{i,j}s_i s_j + \sum_i f_i s_i\right) \ , \tag{2}$$

where f_i and $g_{i,j}$ are the Ising model parameters and $\hat{\mathbf{s}}$ is the minimum energy configuration.

We can also refer to optimizations on Quantum Annealers as QUBO problems, as they are trivially equivalent to the Ising model in (1):

$$\hat{\mathbf{q}} = \arg\min_{\{q_1,\ldots,q_N\}} \frac{1}{2}\sum_{i,j=1}^{N} Q_{i,j}q_i q_j \ , \tag{3}$$

where the symmetric matrix Q holds the coefficients of the binary decision variables ($q_i \in \{0, 1\}$), having the useful property: $q_i^2 = q_i$ and $\hat{\mathbf{q}}$ is the solution bit string. Replacing s_i in (1) with $(q_i \cdot 2) - 1$, we arrive at the same problem description.

B. QUANTUM ANNEALING

Quantum Annealing algorithms are a set of heuristic methods for finding a global minimum of a given objective function,

using quantum mechanical evolution. The objective function is usually given in the form of an Ising Hamiltonian that encodes a combinatorial optimization problem [13].

In the now standard QA devices, the solution is obtained by first initializing the system in a superposition of all possible computational basis states with equal amplitudes that is stabilized by a transverse field.

Then the system evolves according to the time-dependent Schrödinger equation while the amplitudes keep changing as the problem Ising Hamiltonian is slowly introduced and the transverse field is slowly turned down. The still remaining transverse field enables the system to tunnel through the energy barriers so that it can reach lower energy states. Indeed, according to the Adiabatic Theorem, if the change of the coupling strengths of the Ising Hamiltonian and the transverse field is slow enough the system remains in the ground state of the momentary Hamiltonian throughout the annealing. Thus, for such a perfect annealing the resulting configuration of the system (when the transverse field is set to zero) is a minimal energy state of the Ising Hamiltonian, in case of imperfect annealing process the system can also end up in an excited state.

The D-Wave QPUs [17] implement an imperfect version of this process using a physical lattice of qubits and couplers referred to as the Chimera and Pegasus architecture (described in the Sec. IV-A). These systems perform thousands of anneals for each problem in a quick manner, which means they often leave the ground state. However, the hope is, that some of the samples obtained this way will reflect the minimal energy configuration of the problem's Ising Hamiltonian. Recent results of algorithmic benchmarking of QA architectures have been reported in [18]–[22], highlighting diverse application areas.

C. MAXIMUM LIKELIHOOD OPTIMIZATION FOR MIMO CHANNEL DECODING

We refer to a setup of N_t users with single-antenna transmitters and an Access Point (AP) capable of receiving N_r transmission symbols simultaneously, as a MIMO scenario of size $N_t \times N_r$. Each user can send multiple bits with only one symbol using digital modulation techniques, such as Phase Shift Keying or Quadrature Amplitude Modulation, hence the transmitted symbols can be represented by complex vectors [23]. The set of possible values of the transmitted symbols is called the *constellation* $\mathcal{O}$. The size of such a constellation grows exponentially as we increase the complexity (the number of transmitted bits per symbol) of the corresponding modulation scheme. We denote the vector of transmitted symbols from the user antennas as $\bar{\mathbf{v}} \in \mathbb{C}^{N_t}$, and the vector of received symbols at the AP as $\mathbf{y} \in \mathbb{C}^{N_r}$. The channel matrix is denoted as $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$. We get the received symbols by letting the channel matrix effect the transmitted symbols and adding Gaussian white noise (GWN): $\mathbf{n} \in \mathbb{C}^{N_r}$ ($\mathbf{y} = \mathbf{H}\bar{\mathbf{v}} + \mathbf{n}$). Introducing $\mathbf{v}$, the variable for possible transmitted symbols, and $\hat{\mathbf{v}}$, the vector of decoded symbols, the ML decoding [24] at the AP is:

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v} \in \mathcal{O}^{N_t}} \|\mathbf{y} - \mathbf{H}\mathbf{v}\|^2 \ , \tag{4}$$

which is a search in a space of $|\mathcal{O}|^{N_t}$. To regain the original binary message, one can use the constellation to get the decoded bit vector $\hat{\mathbf{b}}$. The parameters f, g in (2) will be derived from the channel matrix and the received symbol vector, as detailed in Appendix VI. For the rest of this discussion, we shall only consider scenarios of $N_t = N_r$, since the qubit requirement only depends on N_t, see Sec. III-A for more details. However, we note that, in general, $N_t \neq N_r$ might be the case, and the methodologies described here work well with such setups.

Although ML decoding would provide optimal detection of the transmission, in practice – due to the computational complexity – only approximations (e.g., *Zero Forcing* [25]) and heuristic methods (e.g., *Sphere Decoding* [26]) are used. As ML decoding can maximize throughput, successfully applying Quantum Annealing to speed up computation would undoubtedly yield an important real-world application of quantum computing.

III. OVERVIEW OF QUBO FORMALISM FOR LARGE AND MASSIVE MIMO OPTIMIZATION

A. QUBO FORMULATION OF THE MIMO ML DECODING

QUBO formulation of the MIMO ML decoding (*QuAMax transform* [1]) requires assigning QUBO variable(s) to each symbol of the transmitted symbol vector. First, using the variable-to-symbol transform function $\mathbf{T}$, one has to map each value in the constellation to logical qubits of the QUBO equation. Next, by expanding (4) via substitution of symbol vectors with the derived qubit equation, one can write the final QUBO form of the ML optimization problem.

For linear $\mathbf{T}$, this process will provide at most quadratic terms, however, in case of Gray-coded transmission,[1] nonlinear transformation is necessary. Here, the method introduced by Kim *et al.* [1] is based on the linear mapping for the Ising transformation, but with an additional post-translation of the decoded bits of $\hat{\mathbf{b}}$ to finally restore the original, Gray-coded message.

In case of higher-order modulations, more qubits are needed to encode one symbol. In particular, an N-QAM modulation requires exactly $\log_2 N$ variables per symbol for the linear mapping $\mathbf{T}$. The total qubit requirement of encoding a symmetric MIMO setup of $N_t \times N_r$ is $N_t \log_2 N$, giving us an estimate on the problem sizes that can be tackled by current Quantum Annealing hardware.

In the following, we give a brief overview of the basic modulations already covered in [1]. In addition, we include the extension of the QuAMax transform to the 64-QAM modulated symbol vectors presented in [15].

[1] Gray code is an encoding technique where each subsequent symbol is encoded by a bit pattern that only differs in one bit in order to make error correction more robust.

B. BPSK, QPSK AND 16-QAM

For the modulation scheme of Binary Phase Shift Keying (BPSK), the mapping **T** is a trivial conversion, as one can simply map each possible symbol $v_i \in \{-1, 1\}$ to $2q_i - 1$.

Higher order modulations have complex numbers as symbols, with exponentially increasing constellation size: $v_i = v_i^I + jv_i^Q$. For Quadrature Phase Shift Keying (QPSK) modulation, we have $v_i^I, v_i^Q \in \{\pm 1\}$, i.e., each dimension can encode one bit. Therefore, mapping **T** requires two qubit variables:

$$v_i = (2q_{2i-1} - 1) + j(2q_{2i} - 1). \tag{5}$$

A more complex encoding is introduced by Quadrature Amplitude Modulation (QAM). For 16-QAM, the number of transmitted bits per symbol is $M = 4$ and the scheme requires a constellation of size 16 (2^M), with usual values of: $v_i^I, v_i^Q \in \{\pm 1, \pm 3\}$. Since both v_i^I and v_i^Q can encode 2 bits, we require 2 qubits per dimension for the mapping **T**:

$$v_i = (4q_{4i-3} + 2q_{4i-2} - 3) + j(4q_{4i-1} + 2q_{4i} - 3). \tag{6}$$

From these equations, substituting back to (4) gives us the expanded formulae of encoding the ML problem in QUBO form. In [1], the expanded QUBO coefficients are written explicitly for BPSK, QPSK and 16-QAM modulations.

C. THE 64-QAM MODULATION

Although 256-QAM modulation is already available in commercial mobile systems targeting Gbit per second data rates, such a complex modulation scheme requires very good radio conditions, i.e., high signal-to-noise-ratio (SNR); furthermore, it is often limited to mobile devices served at the very center of the mobile cells. Hence, mobile devices still utilize lower-complexity modulation schemes, as well. Although implementing the highest commercially available scheme would be desirable, we have to align with the limitations of current QPU hardware architectures, therefore we only increase the complexity to the next level compared to Kim *et al.* [1] via 64-QAM. This allows for a fair comparison with the already investigated implementations of BPSK, QPSK and 16-QAM by [1].

Here we extend our study to include the corresponding QUBO formalism presented in [15]. The usual values of 64-QAM modulated symbols are: $v_i^I, v_i^Q \in \{\pm 1, \pm 3, \pm 5, \pm 7\}$, which requires constellation of size 64 (see Fig. 9 in Appendix VI). A straightforward (and linear) variable-to-symbol transform is thus:

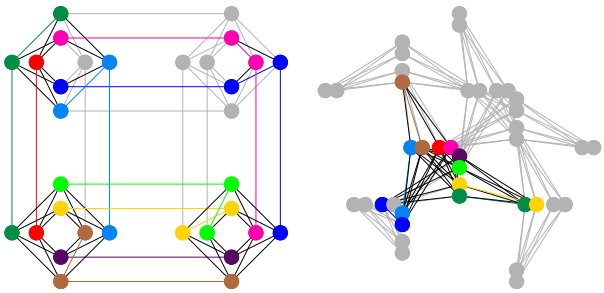
$$\begin{aligned} v_i = {} & (8q_{6i-5} + 4q_{6i-4} + 2q_{6i-3} - 7) \\ & + j(8q_{6i-2} + 4q_{6i-1} + 2q_{6i} - 7). \end{aligned} \tag{7}$$

The linearity comes at the price of disparity between the Gray-code and QuAMax, requiring additional transformation steps. The details of how this post-translation technique works can be found in Sec. 3.2 of [1]. The full expansion of (4) with 64-QAM is given in Appendix VI.

IV. EMBEDDING ONTO D-Wave QPUs

A. D-Wave ARCHITECTURES

There are two types of publicly available QA architectures in D-Wave's current portfolio. D-Wave 2000Q is a model with up to 2048 physical qubits, accessible in a Chimera topology. The novel D-Wave Advantage architecture presents up to 5640 physical qubits in a Pegasus topology. As an illustration, Fig. 2 depicts two embeddings of a complete graph into first, a Chimera C_{16} subgraph in Fig. 2a of four unit cells, and second, a Pegasus P_{16} unit cell in Fig. 2b. While Chimera C_{16} topology is composed of $K_{4,4}$ graphs in a 16×16 lattice, the recently released Advantage QPU has a Pegasus P_{16} topology, which is more connected with nodes of degree 15 via programmable qubit couplers as opposed to previously available degree of 6.



(a) Chimera embedding (b) Pegasus embedding

FIGURE 2. Embedding of a complete graph of 9 nodes to two QPU topologies. Each logical qubit is represented by a different color, inactive nodes and couplers of the graph are represented by gray in (a) Chimera C_{16} subgraph of four unit cells, and (b) Pegasus P_{16} subgraph of single unit cell. The higher connectivity results in lower number of physical qubits and shorter chains representing the logical qubits.

B. MINOR-EMBEDDING METHODS WITH EXTENDED HEURISTICS

QUBO models can have arbitrarily high connectivity, hence mapping them directly to a QPU hardware topology is rarely possible. Instead, one needs to find optimal chains of physical qubits created using large negative couplings to represent the logical qubits. The process of finding such a problem mapping is called *minor-embedding* [27]. Finding the graph minor is NP-hard and since one needs to find it on a classical machine as part of pre-processing QUBO problems, one must employ a heuristic algorithm. One such heuristic method is the MinorMiner (*MM*) algorithm [28] developed for finding arbitrary graph minors. It can be used for finding minors of C_{16} and P_{16} as implemented in D-Wave's open-source framework, the *Ocean SDK* [29]. An example of embedding the same graph into C_{16} and P_{16} is illustrated in Fig. 2.

The quality of embedding is measured in the length of resulting chains and the uniformity of the chain length distribution. As MM is a heuristic algorithm, it has guarantees for neither of these properties. To ensure near-uniform chain lengths, Native Clique Embedding (CLIQUE) [30] can be used. It is a specialized algorithm for quickly finding embeddings of cliques onto C_{16} and P_{16} which one can use to embed an arbitrary graph since any graph can be mapped

TABLE 1. Qubit requirements of different ML encoded MIMO configurations with Native Clique Embedding: logical qubits (physical qubits on C_{16}/P_{16}).

Config.	BPSK	QPSK	16-QAM	64-QAM
10×10	10 (40/16)	20 (120/60)	40 (440/192)	60 ($1K$/402)
20×20	20 (120/60)	40 (440/192)	80 ($2K$/670)	120 ($4K$/$1K$)
30×30	30 (270/114)	60 ($1K$/402)	120 ($4K$/$1K$)	180 ($8K$/$3K$)
40×40	40 (440/192)	80 ($2K$/670)	160 ($7K$/$2K$)	240 ($15K$/$5K$)
60×60	60 ($1K$/402)	120 ($4K$/$1K$)	240 ($15K$/$5K$)	360 ($33K$/$11K$)
80×80	80 ($2K$/670)	160 ($7K$/$2K$)	320 ($26K$/$9K$)	480 ($58K$/$20K$)
100×100	100 ($3K$/$1K$)	200 ($10K$/$4K$)	400 ($40K$/$14K$)	600 ($91K$/$31K$)
120×120	120 ($4K$/$1K$)	240 ($15K$/$5K$)	480 ($58K$/$20K$)	720 ($130K$/$45K$)
140×140	140 ($5K$/$2K$)	280 ($20K$/$7K$)	560 ($79K$/$27K$)	840 ($177K$/$60K$)
160×160	160 ($7K$/$2K$)	320 ($26K$/$9K$)	640 ($103K$/$35K$)	960 ($231K$/$79K$)
180×180	180 ($8K$/$3K$)	360 ($33K$/$11K$)	720 ($130K$/$45K$)	1080 ($293K$/$99K$)

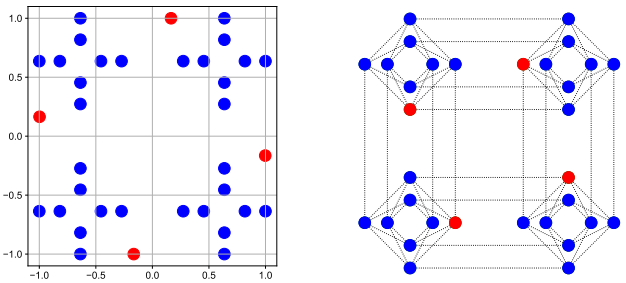
to a complete graph. However, CLIQUE has a fixed upper limit for the maximal clique sizes that it can handle, which is 64 and 180 for C_{16} and P_{16}, respectively [31].

We note that despite the high number of available qubits, many factors can lead to high ratio of inactive nodes after an embedding. First, the special QPU topologies are hard to be fully utilized due to their sparse connectivity. Furthermore, since the actual hardware graphs can have manufacturing imperfections some graphs that would be embeddable into perfect C_{16} or P_{16} will not be embeddable in practice [32].

In order to embed larger QUBO problems that supersede the limitations of CLIQUE, we use two heuristic approaches that aim to yield higher qubit utilization. Clique-Based MinorMiner (*CLMM*) and Spring-Based MinorMiner (*SPMM*) [31] work by finding initial chains of qubits which can be passed to MM as a parameter in hopes of finding better final embedding using these as starting points. CLMM finds initial chains using CLIQUE, which results in the near uniformity of chains. As MM is able to shorten these chains, CLMM is capable of finding embeddings to even larger problems, exceeding the capability of pure MM. SPMM lays out both the hardware topology graph and the problem graph on a $[-1, 1] \times [-1, 1]$ plane and matches each problem node to the nearest hardware node in Euclidean distance. Hence, the initial "chains" in case of SPMM all have length of one, therefore this method is more suitable for sparser problem graphs (Fig. 3). We present test results for both methods and use the found limiting cases of embeddable MIMO ML decoding scenarios of maximum problem size.

C. MINOR-EMBEDDING OF MIMO ML DECODING

In this section, we present the found hard limits of MIMO ML decoding as QUBO problem regarding size and modulation complexity embeddable into Pegasus P_{16} and Chimera C_{16} graphs. As any N-QAM $N_t \times N_r$ MIMO ML setup is (almost) equivalent to a $K_{N_t \log_2 N}$ complete graph [1], using CLIQUE, one can derive the largest embeddable problem sizes. For Chimera C_{16}, it is known that the largest native clique is K_{64}, whereas for Pegasus P_{16} it is K_{180}.



(a) Layout (b) Initial Chains

FIGURE 3. SPMM initial chain mapping on Chimera. Red dots are the problem nodes and the blue ones represent physical qubits of the Chimera hardware topology.

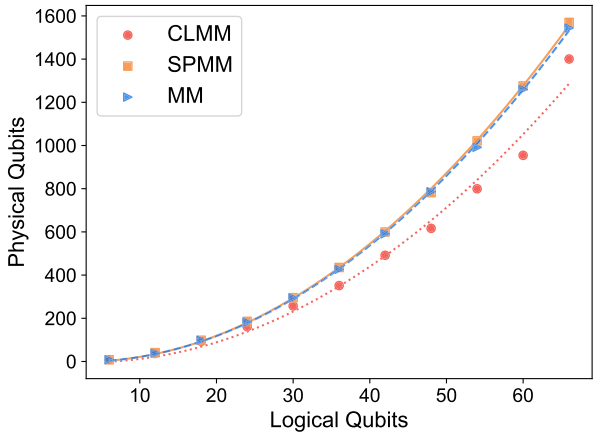
From these assumptions, one can trivially derive the limits of each problem complexity. However, for completeness, we have compiled the logical and physical qubit requirements in Table 1 for a set of large and massive MIMO scenarios that are relevant in practice. The green cells indicate feasible Native Clique Embedding on both QPU architectures, yellow cells are only embeddable to Pegasus P_{16} topology and red cells indicate non-feasibility on both the 2000Q and the Advantage system.

Already by using Pegasus P_{16} the results surpassed previous limits published by [1], more than doubling the largest embeddable problem sizes. However, additional heuristics further improved these results. In addition, a comparison of physical qubit usage of MM, SPMM and CLMM is depicted in Fig. 4. Since the embedded problems are represented by near complete graphs, CLMM is the best performer, whereas SPMM has inferior performance – as expected based on [31]. The rate of growth is fast but sub-exponential for all three algorithms. Still, none of the embeddings could leverage all available qubits due to, at least in part, the connectivity limitations of the QPU topologies.

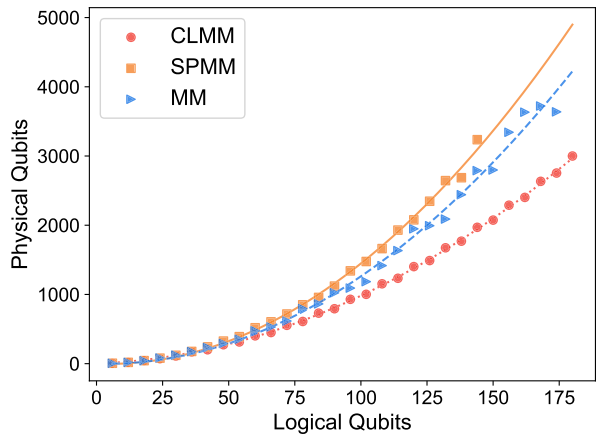
The upper limits for each modulation and for the two architectures are summarized in Table 2 comparing the Native Clique Embedding to best-performing CLMM. Note that CLMM produced near-uniform chain lengths, whereas MM and SPMM did not. This gives CLMM an advantage as it is

TABLE 2. Minor-embedding limits of QUBO-form MIMO ML decoding problems with Native Clique Embedding (CLIQUE) and heuristics (CLMM).

Arch	Method	BPSK	QPSK	16-QAM	64-QAM
C_{16}	CLIQUE	64×64	32×32	16×16	10×10
	CLMM	65×65	33×33	16×16	11×11
P_{16}	CLIQUE	180×180	90×90	45×45	30×30
	CLMM	182×182	91×91	45×45	30×30



(a) Chimera



(b) Pegasus

FIGURE 4. Comparison of embedding methods via QUBO graphs of a series of 64-QAM MIMO decoding scenarios. Heuristic algorithms of CLMM, SPMM and MM were each tested in (4a) Chimera C_{16} and (4b) Pegasus P_{16} for 10 embedding runs. The average number of required physical qubits is depicted per problem size and embedding method.

capable of producing both shorter and more uniform chains, the two main factors affecting the optimization quality of Quantum Annealing solutions.

V. OPTIMIZATION OF MIMO ML DECODING WITH QUANTUM ANNEALING DEVICES

A. EXPERIMENTS FOR COMPARATIVE ANALYSIS

We tested each modulation with many different symmetric user setups ($N_t \times N_r$, $N_t = N_r = 2^p$) that are relevant for commercial telecommunication installations. For each specific problem, we generated 10 different noiseless random

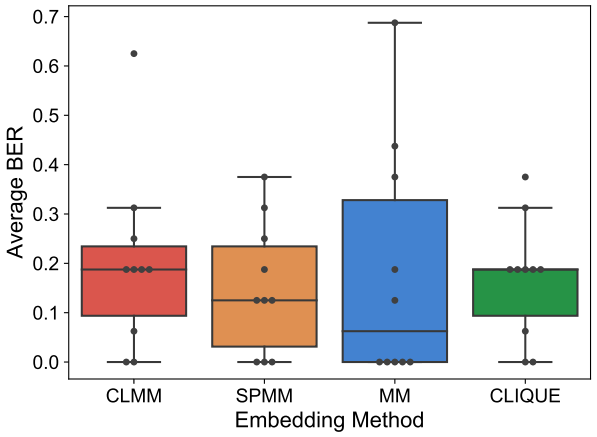


FIGURE 5. Comparison of embedding methods with respect to solution quality. 10 random 4 × 4 16-QAM problems were generated and solved using the D-Wave Advantage system. Each embedding method was tested on each random instance and evaluated using bit error rate as performance metric.

instances (channel matrices, transmitted bits), each of which was run 5 times.

Our main performance metric was the bit error rate (*BER*), which indicates the ratio of unsuccessfully decoded bits, i.e.: $\mathrm{BER}(\hat{\mathbf{b}}) = \frac{\|\mathbf{b}-\hat{\mathbf{b}}\|}{n}$, where n is the length of the sent bit string $\mathbf{b}$.

The natural expectation is that increasing problem complexity leads to increased BER. Moreover, the hope is that the more advanced QPU will not only handle higher problem sizes, but also improve upon performance for the smaller problems running on both architectures. Both of these expectations were proven by the experiments presented in the next subsection.

For most of the runs, we used Native Clique Embedding as it produced uniform chains, leading to more stable results as opposed to heuristic embedding methods. However, the largest instances on the Pegasus architecture were embedded using the base MinorMinor method, since CLIQUE would require more qubits than available in the current hardware topology. Fig. 5 shows that CLIQUE is the most stable one of the presented embedding methods when solving a 4×4 16-QAM problem multiple times on the D-Wave Advantage system.

B. EVALUATION OF EXPERIMENTS ON D-Wave QPUs

The experiments were run using the D-Wave DW_2000Q_6 and the Advantage_system_1.1 QPU solvers [33], [34]. The QPU solvers have several parameters available for controlling the annealing process. Annealing time (T_a) and the number of requested samples (N_a) are the most basic annealing process parameters, and also the most important ones for the quality of solution. T_a sets the duration (in μsec) of each annealing cycle, and N_a tells the QPU how many sampling cycles should be executed. As both of these parameters have high impact on the optimization outcome one often needs to empirically search the whole parameter space for each particular problem to find the most promising combination. We tested a minimal set of parameter pairs that proved to be well-performing

during our earlier research [35]. We concluded that setting T_a to 15 and N_a to 500 is a good choice overall as it gives stable results across all problem complexities (BPSK, QPSK, 16-QAM, 64-QAM) and MIMO sizes ($N_t \in \{2, 4, \ldots, 128\}$). We should note however, that this setting does not necessarily meet the practical real-time computing requirements of the MIMO ML detection problems.

The other important parameter was the *chain strength* that governs the cohesion of physical qubit chains. Setting this parameter too high can lead to lost precision of the QUBO coefficients, while setting it too low can cause broken chains. We conducted the experiments with chain strengths of 0.3 and 0.8. From our experiments, the lower complexity problems require smaller chain strength and the QAM modulations benefit from stronger chains.

The results of the performance tests are shown in Fig. 6, where BER statistics for each problem's solution on both QPUs are depicted. We can see that increasing problem complexity indeed leads to increased BER. Besides the more advanced Advantage QPU being able to handle higher problem sizes, the results also show performance improvement for smaller problem solutions as compared to the 2000Q architecture solutions. Hence, our expectations were proven. Although the improvements per problem instance are not substantial, still, the problem size extension is significant.

For the dimension of complexity of modulation types, we can see a separation between Phase Shift Keying and Quadrature Amplitude Modulations. First, for the BPSK and QPSK modulations, even the larger instances could be solved via both QPUs with near zero median BER, furthermore, for most of these problems the QPUs could produce at least one proper solution. In Fig. 6, we can observe that the median BER increases significantly when we are reaching the limits of the largest investigated size of the problems. This is not unexpected, as larger cliques require embedding with longer chains leading to the aforementioned phenomenon of losing precision of QUBO coefficients. Second, for the 16-QAM and 64-QAM cases, the results are not satisfactory. For these complex modulation schemes, even small-sized problems could incorporate some bit errors and the largest problems rarely had proper solutions.

For the dimension of the transmitter numbers, there is a separation only in case of the more complex modulation schemes. First, for the 2×2 and 4×4 instances of the 16-QAM cases, there exist perfect solutions, but the high bit error rate dominates in the majority of the samples; for 64-QAM, we could only produce proper solutions for the 2×2 instance. Second, in case of the large instances of higher-order modulations, further parameter fine-tuning would be required to produce near-optimum solutions.

C. POST-PROCESSING OF QUANTUM ANNEALING SAMPLES

Single Qubit Correction (SQC) is a post-processing technique introduced by [36]. It aims to reduce energies of Quantum Annealing samples by iteratively changing each solution bit depending on its energy impact. It is a heuristic approach, i.e., there is no guarantee that the resulting samples will have lower energies than the original ones (but it is guaranteed that it does not increase the energies).

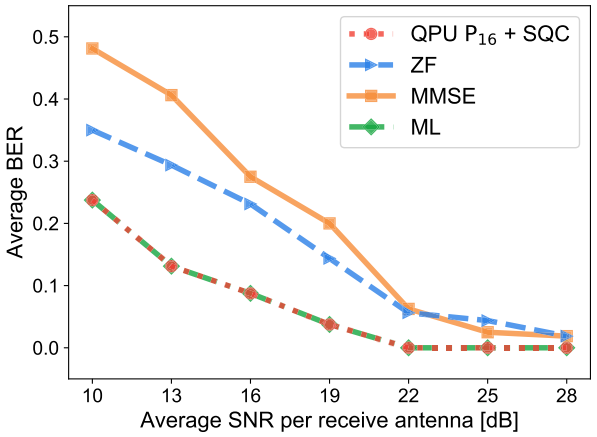


FIGURE 7. Comparing conventional ML decoders to the QA-based method. With different levels of SNR, 10 random 4 × 4 16-QAM problems were generated and solved using the D-Wave Advantage system with SQC post-processing and the following conventional ML decoding techniques: Zero Forcing (ZF), Minimum Mean Squared Error (MMSE) and Sphere Decoding (ML). The performance was evaluated using bit error rate as metric.

The method works by iteratively changing the sign of bits in every sample in order to reduce the sample's energy. Let s_i denote the ith bit of a sample obtained via Quantum Annealing. Given the matrix of QUBO coefficients Q and the set of indices of non-zero quadratic biases $C = \{i, j | Q_{i,j} \neq 0\}$, the *influence* of s_i is:

$$I_i = Q_{i,i} + \sum_{i,j \in C} Q_{i,j} s_j. \tag{8}$$

If I_i and s_i has the same sign, we can flip the sign of s_i to reduce the energy of the complete sample by $2I_i$. Here, s_i runs over the entire set of QUBO variables. After every bit has been checked and potentially flipped, the whole process can be repeated until no more change is possible.

With SQC, we were able to achieve an average BER reduction of 0.067, with a maximum reduction of 0.25. The method achieved the most significant improvements for 16-QAM and 64-QAM modulations with averages of 0.158 and 0.118, and with the maximums reaching 0.25 and 0.229 reduction in BER, respectively. Fig. 6 shows the results achieved via post-processing of P_{16} QPU samples as compared to the raw samples from C_{16} and P_{16} QPUs for the same set of random instances.

D. COMPARISON TO CONVENTIONAL METHODS

We compared the performance of the QA-based ML decoding to conventional decoding schemes [37], [38]. As a baseline for linear decoding methods, we used Zero Forcing [25] and Minimum Mean Squared Error [39]. Furthermore, we also included the Sphere Decoding [40], [41] method with infinite lattice that implements Maximum Likelihood estimation, which is capable of finding the best possible solution to the

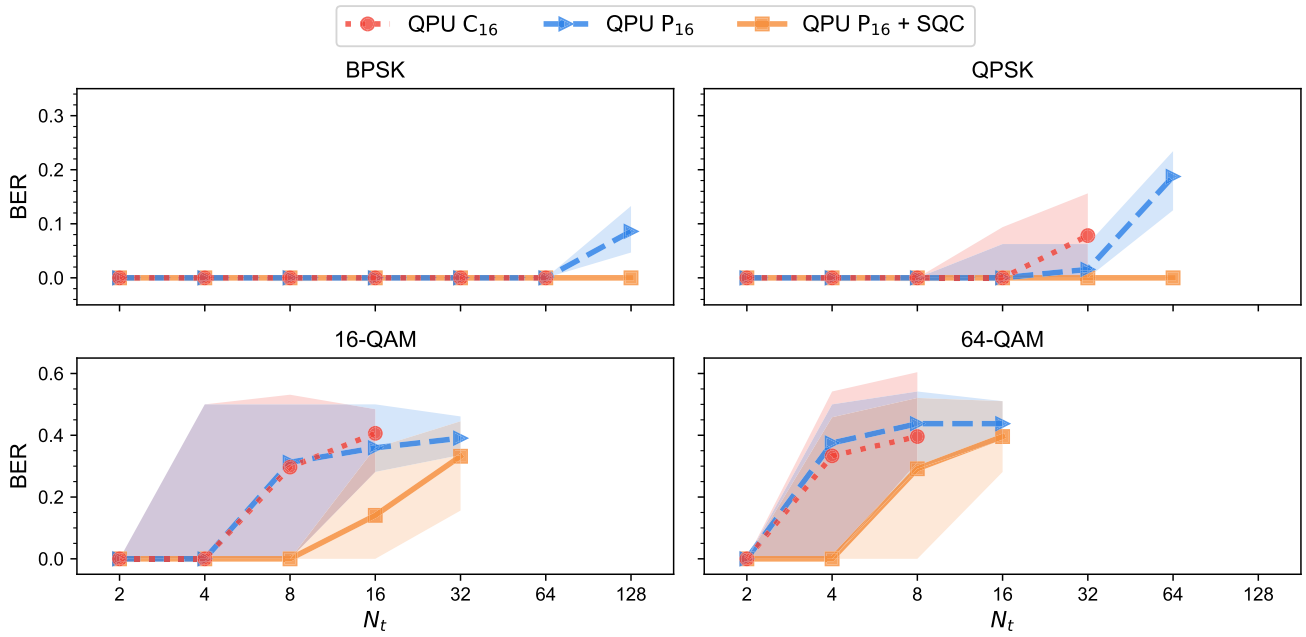


FIGURE 6. Solution quality measured in bit error rate (BER) of MIMO ML decoding problems solved by the D-Wave 2000Q and Advantage system. Statistics of modulation types BPSK, QPSK, 16-QAM and 64-QAM are shown for C_{16} QPU raw sampling (red dot), P_{16} QPU raw sampling (blue triangle) and P_{16} QPU SQC-enhanced sampling (yellow square). Per problem size and per modulation type, 50 experimental runs were performed by 10 random instances optimized 5 times each. The lines represent the median values at the given problem size with 6 to 94 percentile intervals colored around them.

decoding problem. We have compared these methods with the Quantum Annealing based technique, in particular the P_{16} QPU + SQC method with CLMM embedding. We note that improvements to the mentioned basic classical techniques already exist [42]–[44], which can outperform these basic versions either in accuracy, or in complexity.

Without noise, the conventional methods can solve any decoding problem, as they can calculate the exact solution due to their mathematical framework. However, the same cannot be said about the QA-based decoder as it needs to work on a modified (embedded) version of the problem. Furthermore, an analog QPU device also suffers from environmental noise and low floating-point precision. To make the problem more realistic, we added noise to the received symbols, which was done by specifying the signal-to-noise ratio (SNR), measured in dB.

Each of the decoders solved 4×4 16-QAM problems with 7 different SNR values with 10 randomly generated instances per SNR value. SNR analysis of BPSK and QPSK comparing quantum and classical methods was already presented in Kim *et al.* [1] . We chose 16-QAM to extend upon this work toward practical use cases. Fig. 7 depicts the result of the comparison measured in BER averaged over the 10 random instances. From this, we can conclude that the QA-based method is capable of providing the same solution quality as the conventional decoding methods in noisy channels. In particular, it produced the same results as the Sphere Decoding ML method, which is the best possible outcome.

The simulations were done using a modified version of [45], a massive MIMO simulator with multiple available decoders.

E. TIME STATISTICS OF QPU SAMPLING

The process of QA sampling on D-Wave's QPUs can be deconstructed into a sum of processing time intervals, each measured in μs. On the high level, *QPU Access Time* can be split into *QPU Programming Time*, *QPU Sampling Time*, and *QPU Access Overhead Time*. During Programming, the biases and couplers are initialized according to the QUBO model, Sampling Time is used to collect N_a samples, whereas Access Overhead accounts for post-processing of the batch of samples last to be collected (the other batches are post-processed concurrently with the annealing run of the consecutive batch). For more details on D-Wave QPU timing, see [46].

QPU Sampling Time can be further decomposed into timing requirements of each individual sampling. Producing one sample includes *QPU Annealing Time Per Sample* (T_a), *QPU Read Out Time Per Sample* (R_a) and *QPU Delay Time Per Sample*. Although Sampling Time is usually referred to as $N_a \cdot T_a$, we wanted to highlight that Read Out Times take up a large portion of Sampling Time, which must be considered for the real-time requirements of the MIMO ML decoding problem.

Fig. 8 shows the breakdown of the total Sampling Time for a single QPU run grouped by modulation, QPU architecture

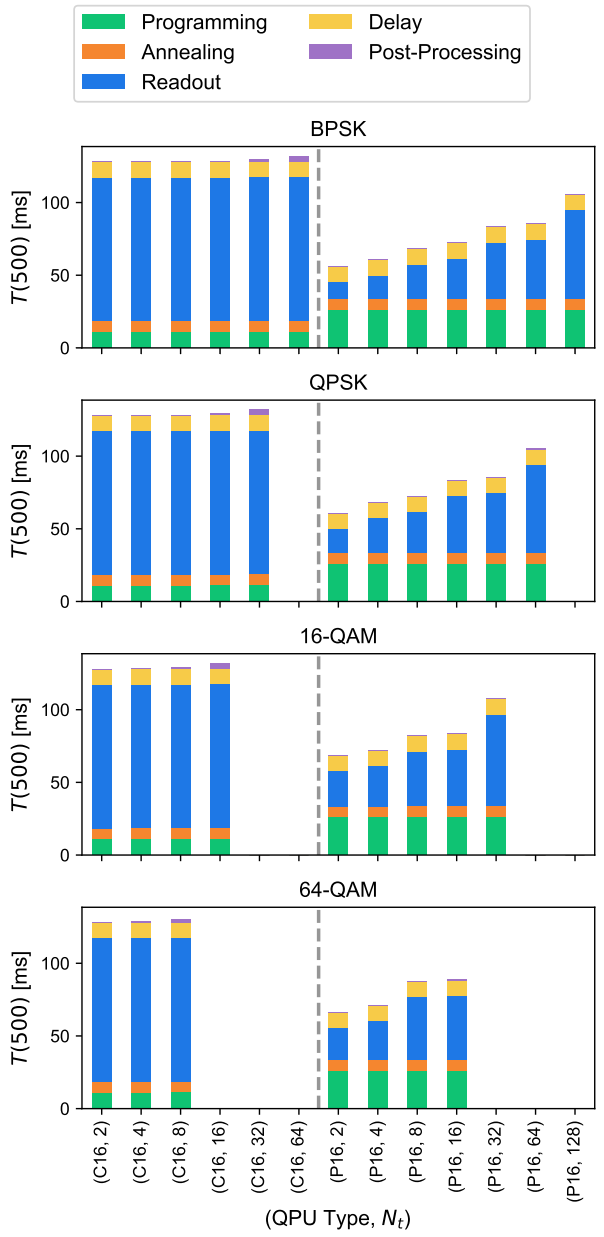


FIGURE 8. QA timing statistics over experiments for 2000Q (C16) and Advantage (P16) architectures. Values show averages over 50 experiments per problem size, with parameters of 500 sample readouts and 15μs annealing time per sample. A decomposition of the total QA sampling process is used with the following elements: Programming (green) initializes the QPUs, Annealing (orange) drives the system to optimal state, Readout (blue) measures the solution states, Delay (yellow) is used for thermalization of the QPU to its initial temperature, and Post-Processing (purple) transforms the last batch of sample measurements before returning the best optimum and additional information of the annealing run.

and N_t. The first step for a QPU run the qubits are initialized (Programming Time). This is a one-time only process, and it is nearly constant for all problem sizes, but it differs on the two QPUs significantly due to the size difference.

The actual annealing process takes $N_a \cdot T_a \cdot R_a$ total time, where N_a and T_a is determined by the API user, whereas R_a is dependent on the hardware implementation. In case of our experiments, $N_a = 500$ and T_a is 15. One could improve upon these results by lowering T_a (down to 1) or requesting fewer samples, however, the optimal values T_a and N_a are often problem dependent and finding them requires thorough search of the parameter space.

D-Wave's post-processing optimization of samples is applied batch-by-batch, in parallel with the annealing runs, and therefore only the last batch's post-processing adds extra time to the overall annealing run. In Fig. 8, we can observe that post-processing time is highly dependent on problem size and has negligible impact on the whole sampling time.

Fig. 8 also tells us that the 2000Q QPU has constant time requirement regarding readouts, whereas the Advantage system demonstrates scaling with the problem size. Furthermore, the overall readout times improved also on the architecture and therefore it is not only capable of solving larger problems, but also achieves speedup over all experiment types and problem sizes.

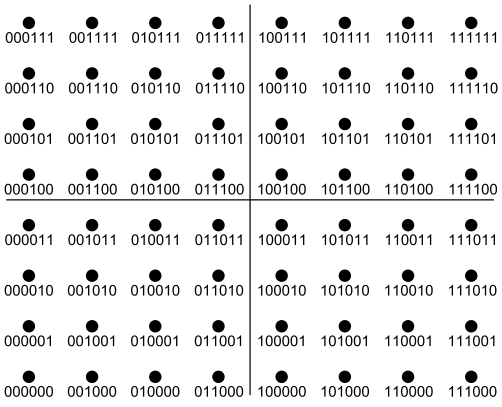
VI. CONCLUSION AND OUTLOOK

In this paper, we presented the experimental evaluation of the MIMO ML decoding tests of multiple modulation schemes on currently available Quantum Annealers. For testing the newly available D-Wave Advantage system, we first reproduced the results of [1] for BPSK, QPSK, 16-QAM on the predecessor hardware D-Wave 2000Q and then extended the range of experiments to the 64-QAM modulation scheme using the novel state-of-the-art QPU.

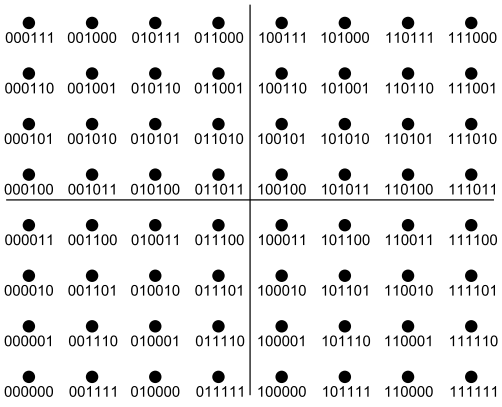
As expected, using the Pegasus P_{16} architecture of the D-Wave Advantage system, the embedding and solution qualities of the MIMO ML decoding problems scale better, for both the dimensions of modulation complexity and transmitter numbers, than in the previously available Chimera C_{16} architecture. Whereas the embeddable problem sizes could be doubled on the new architecture, paving the way for massive MIMO applications, the improvements per problem instance were not substantial for smaller QUBO problems.

We believe that this is due to precision limitations of the analog QPU hardware. With the use of the SQC post-processing technique, the quality of solutions could be further improved, however, only to a certain degree. We believe that additional heuristics and parameter tuning could be tested for further enhancement of the optimization results. Furthermore, fine-tuning for lower values for N_a and T_a could be an additional step towards requirements of latency sensitive optimization tasks. We leave the study of these possibilities to future work.

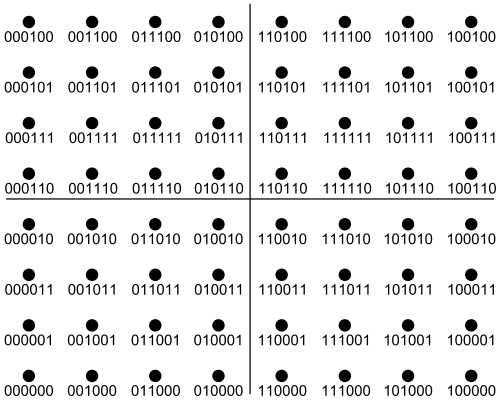
We showed that using SQC post-processing the current QPU is already capable of solving up to 16-QAM problems with the same quality as the classical ML decoders. This is an indication that future QPU architectures (with higher qubit connectivity) will be able to solve MIMO ML decoding problems of any complexity with at least the same quality as classical decoders.



(a) QuAMax transform



(b) Intermediate code



(c) Gray code

FIGURE 9. The process of converting the *QuAMax* encoded bits to the original, Gray-coded transmission with 64-QAM.

While NISQ quantum technologies present a remarkable progress each year, until their maturity there are hybrid and quantum-inspired classical solutions already commercially available – with impressive performance reports for specific optimization tasks. For large-scale problem sizes, D-Wave's new Leap Hybrid service [47], [48] is capable of processing up to 20,000 QUBO variables with fully connected graphs, using an approach of problem partitioning. Digital annealing technologies [49]–[52] also pave the way towards time sensitive optimization scenarios with high-parallelism and high-performance dedicated architectures.

The quality metrics of novel optimization systems would also have to be tested via algorithmic comparative experiments of specific problem types, and we expect to see more diverse set of industry-specific performance studies published in the near future.

APPENDIX A
CONVERSION OF ENCODED BITS WITH 64-QAM

Here we refer to (7) in Sec.III-C, and present a more detailed constellation diagram for 64-QAM modulation scheme. Fig. 9 illustrates the process of converting the QuAMax encoded bits to the original, Gray-coded 64-QAM transmission as follows.

First, the sender transmits Gray-coded message using the constellation of Fig. 9c. At the receiver, the QuAMax constellation of Fig. 9a is used to decode the message, which uses non-Gray-coded bit strings in order to retain linear QUBO transformation. To restore the originally sent bit string, post processing of the minimization result is required using post-translation which flips the bits of even-numbered columns in the constellation resulting in Intermediate code constellation of Fig. 9b. Finally, from the Intermediate code constellation we convert back to the original, Gray-coded bit string via either simple mapping or by differential bit encoding.

APPENDIX B
BER IMPROVEMENT USING SQC

Fig. 10 shows the improvement in BER of the Single Qubit Correction post-processing for both QPUs, where $\Delta\text{BER} = \text{BER}_{\text{SQC}} - \text{BER}_{\text{QPU}}$

APPENDIX C
ISING TRANSFORMATION OF 64-QAM MODULATION

The equations for the Ising coefficients of the 64-QAM Modulation follow the same notation as Kim *et al.* [1] , where $\mathbf{H}_{(:,i)}$ denotes the ith column of the channel matrix $\mathbf{H}$ and $\mathbf{y}$ is the vector of received symbols,[2] (10), as shown at the next page.

$$f_i(\mathbf{H},\mathbf{y}) = \begin{cases} -8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^I\right) - 8\left(\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^Q\right), & i = 6n-5 \\ -4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^I\right) - 4\left(\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^Q\right), & i = 6n-4 \\ -2\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^I\right) - 2\left(\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^Q\right), & i = 6n-3 \\ -8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^Q\right) + 8\left(\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^I\right), & i = 6n-2 \\ -4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^Q\right) + 4\left(\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^I\right), & i = 6n-1 \\ -2\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^Q\right) + 2\left(\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\cdot\mathbf{y}^I\right), & i = 6n \end{cases} \tag{9}$$

[2]We note that the correct QUBO model coefficient of the 16-QAM QuAMax description is $g_{ij}(\mathbf{H}) = 2(\mathbf{H}^I_{(:,\lceil i/4\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/4\rceil)}) - 2(\mathbf{H}^I_{(:,\lceil j/4\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/4\rceil)})$ (in case $i = 4n,\ j = 4n' - 2$ in (14)), contrary to the misprinted formula in Appendix C of [1].

$$
g_{ij}(\mathbf{H}) = \begin{cases}
\begin{cases}
+32\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+32\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-5\\
+16\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+16\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-4\\
+8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+8\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-3\\
-32\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)+32\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-2\\
-16\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)+16\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-1\\
-8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)+8\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'
\end{cases} & i=6n-5\\
\begin{cases}
+16\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+16\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-5\\
+8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+8\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-4\\
+4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+4\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-3\\
-16\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)+16\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-2\\
-8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)+8\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-1\\
-4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)+4\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'
\end{cases} & i=6n-4\\
\begin{cases}
+8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+8\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-5\\
+4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+4\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-4\\
+2\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+2\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-3\\
-8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)+8\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-2\\
-4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)+4\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-1\\
-2\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)+2\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'
\end{cases} & i=6n-3\\
\begin{cases}
+32\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)-32\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-5\\
+16\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)-16\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-4\\
+8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)-8\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-3\\
+32\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+32\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-2\\
+16\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+16\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-1\\
+8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+8\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'
\end{cases} & i=6n-2\\
\begin{cases}
+16\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)-16\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-5\\
+8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)-8\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-4\\
+4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)-4\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-3\\
+16\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+16\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-2\\
+8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+8\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-1\\
+4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+4\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'
\end{cases} & i=6n-1\\
\begin{cases}
+8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)-8\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-5\\
+4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)-4\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-4\\
+2\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\right)-2\left(\mathbf{H}^I_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-3\\
+8\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+8\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-2\\
+4\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+4\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right), & j=6n'-1\\
+2\left(\mathbf{H}^I_{(:,\lceil i/6\rceil)}\cdot\mathbf{H}^I_{(:,\lceil j/6\rceil)}\right)+2\left(\mathbf{H}^Q_{(:,\lceil j/6\rceil)}\cdot\mathbf{H}^Q_{(:,\lceil i/6\rceil)}\right) & j=6n'
\end{cases} & i=6n
\end{cases} \tag{10}
$$

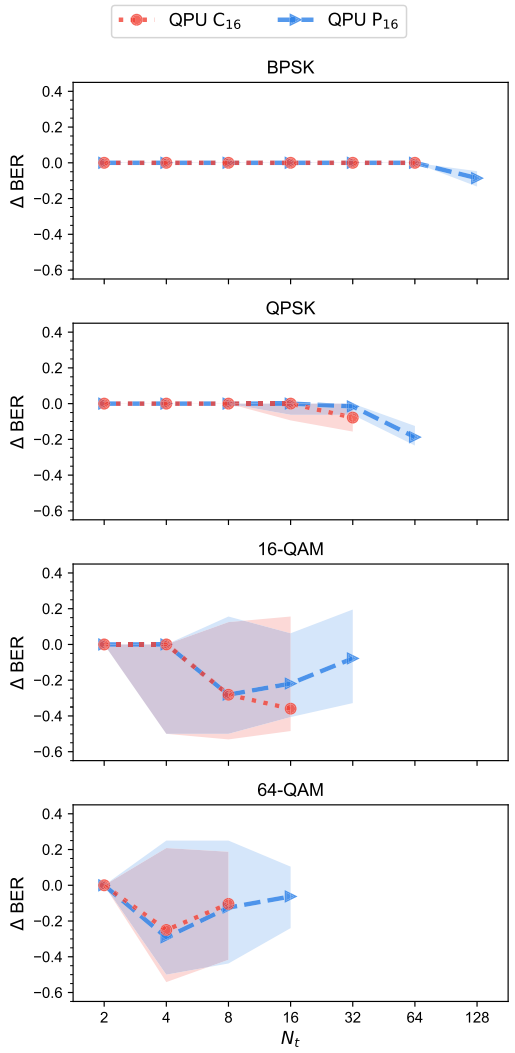


FIGURE 10. Improvements in BER using SQC post-processing by reducing energies of individual samples. Statistics of modulation types BPSK, QPSK, 16-QAM and 64-QAM are shown for C_{16} QPU SQC-enhanced sampling (red dot) and P_{16} QPU SQC-enhanced sampling (blue triangle). A total of 500 experimental runs were post-processed using SQC. The lines represent the median values of $\Delta\text{BER} = \text{BER}_{\text{SQC}} - \text{BER}_{\text{QPU}}$ at the given problem size with 6 to 94 percentile intervals colored around them.

REFERENCES

[1] M. Kim, D. Venturelli, and K. Jamieson, "Leveraging quantum annealing for large MIMO processing in centralized radio access networks," in *Proc. ACM Special Interest Group Data Commun.*, Aug. 2019, pp. 241–255, doi: 10.1145/3341302.3342072.

[2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[3] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 58, no. 5, p. 5355, 1998, doi: 10.1103/PhysRevE.58.5355.

[4] E. Farhi, J. Goldstone, and S. Gutmann, "Quantum adiabatic evolution algorithms versus simulated annealing," 2002, *arXiv:quant-ph/0201031*. [Online]. Available: https://arxiv.org/abs/quant-ph/0201031

[5] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, and E. M. Chapple, "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, 2011, doi: 10.1038/nature10012.

[6] E. Dahlman, S. Parkvall, and J. Sköld, "New 5G radio-access technology," in *4G LTE-Advanced Pro and The Road to 5G*, 3rd ed, E. Dahlman, S. Parkvall, and J. Sköld, Eds. New York, NY, USA: Academic, 2016, pp. 547–573, doi: 10.1016/B978-0-12-804575-6.00024-8.

[7] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014, doi: 10.1109/MCOM.2014.6736761.

[8] E. G. Larsson, "MIMO detection methods: How they work [lecture notes]," *IEEE Signal Process. Mag.*, vol. 26, no. 3, pp. 91–95, May 2009.

[9] B. Trotobas, A. Nafkha, and Y. Louët, "A review to massive MIMO detection algorithms: Theory and implementation," in *Advanced Radio Frequency Antennas for Modern Communication and Medical Systems*. Rijeka, Croatia: IntechOpen, 2020, doi: 10.5772/intechopen.93089.

[10] P. von Butovitsch, D. Astely, C. Friberg, A. Furuskär, B. Göransson, B. Hogan, J. Karlsson, and E. Larsson, "Advanced antenna systems for 5G networks," Ericsson, Stockholm, Sweden, Tech. Rep. GFMC-18:000530, Nov. 2018. [Online]. Available: https://www.ericsson.com/4a8a87/assets/local/reports-papers/white-papers/10201407_wp_advanced_antenna_system_nov18_181115.pdf

[11] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive mimo is a reality—What is next? Five promising research directions for antenna arrays," *Digit. Signal Process.*, vol. 94, pp. 3–20, Nov. 2019.

[12] C. Mobile, "C-RAN: The road towards green RAN," *White Paper, Ver*, vol. 2, no. 5, pp. 15–16, 2011.

[13] A. Lucas, "Ising formulations of many NP problems," *Frontiers Phys.*, vol. 2, p. 5, Feb. 2014.

[14] F. Glover, G. Kochenberger, and Y. Du, "A tutorial on formulating and using QUBO models," 2018, *arXiv:1811.11538*. [Online]. Available: http://arxiv.org/abs/1811.11538

[15] Á. Marosits, Z. Tabi, Z. Kallus, P. Vaderna, I. Gódor, and Z. Zimborás, "Exploring embeddings for MIMO channel decoding on quantum annealers," *Infocommun. J.*, vol. 13, no. 1, pp. 11–17, 2021.

[16] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, "Next-generation topology of D-Wave quantum processors," 2020, *arXiv:2003.00133*. [Online]. Available: http://arxiv.org/abs/2003.00133

[17] D-Wave. (2021). *Technical Description of the D-Wave Quantum Processing Unit, User Manual*. [Online]. Available: https://docs.dwavesys.com/docs/latest/doc_qpu.html

[18] E. Grant, T. S. Humble, and B. Stump, "Benchmarking quantum annealing controls with portfolio optimization," *Phys. Rev. A, Gen. Phys.*, vol. 15, no. 1, Jan. 2021, Art. no. 014012.

[19] C. D. G. Calaza, D. Willsch, and K. Michielsen, "Garden optimization problems for benchmarking quantum annealers," 2021, *arXiv:2101.10827*. [Online]. Available: http://arxiv.org/abs/2101.10827

[20] D. Vert, R. Sirdey, and S. Louise, "Benchmarking quantum annealing against," hard" instances of the bipartite matching problem," *Social Netw. Comput. Sci.*, vol. 2, no. 2, pp. 1–12, 2021.

[21] G. Pilon, N. Gugole, and N. Massarenti, "Aircraft loading optimization - QUBO models under multiple constraints," 2021, *arXiv:2102.09621*. [Online]. Available: http://arxiv.org/abs/2102.09621

[22] D. Inoue, A. Okada, T. Matsumori, K. Aihara, and H. Yoshida, "Traffic signal optimization on a square lattice with quantum annealing," *Sci. Rep.*, vol. 11, no. 1, pp. 1–12, Dec. 2021.

[23] P. Botsinis, S. X. Ng, and L. Hanzo, "Low-complexity iterative quantum multi-user detection in SDMA systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 5592–5597, doi: 10.1109/icc.2014.6884212.

[24] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003, doi: 10.1109/TIT.2003.817444.

[25] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt, "Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 461–471, Feb. 2004, doi: 10.1109/TSP.2003.821107.

[26] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005, doi: 10.1109/TSP.2005.850352.

[27] L. Lovász, "Graph minor theory," *Bull. Amer. Math. Soc.*, vol. 43, no. 1, pp. 75–86, Jan. 2006, doi: 10.1090/S0273-0979-05-01088-8.

[28] J. Cai, W. G. Macready, and A. Roy, "A practical heuristic for finding graph minors," 2014, *arXiv:1406.2741*. [Online]. Available: http://arxiv.org/abs/1406.2741

[29] *D-Wave OCean Software Documentation*. Accessed: Jan. 30, 2020. [Online]. Available: https://docs.ocean.dwavesys.com/en/stable/
[30] T. Boothby, A. D. King, and A. Roy, "Fast clique minor generation in chimera qubit connectivity graphs," *Quantum Inf. Process.*, vol. 15, no. 1, pp. 495–508, Jan. 2016, doi: 10.1007/s11128-015-1150-6.
[31] S. Zbinden, A. Bärtschi, H. Djidjev, and S. Eidenbenz, "Embedding algorithms for quantum annealers with chimera and pegasus connection topologies," in *Proc. Int. Conf. High Perform. Comput.* Springer, 2020, pp. 187–206, doi: 10.1007/978-3-030-50743-5_10.
[32] E. Lobe, L. Schürmann, and T. Stollenwerk, "Embedding of complete graphs in broken chimera graphs," 2020, *arXiv:2012.12720*. [Online]. Available: http://arxiv.org/abs/2012.12720
[33] J. King, S. Yarkoni, J. Raymond, I. Ozfidan, A. D. King, M. M. Nevisi, J. P. Hilton, and C. C. McGeoch, "Quantum annealing amid local ruggedness and global frustration," *J. Phys. Soc. Jpn.*, vol. 88, no. 6, Jun. 2019, Art. no. 061007.
[34] C. McGeoch and P. Farré, "The D-wave advantage system: An overview," D-Wave Systems Inc., Burnaby, BC, Canada, Tech. Rep. 14-1049A-A, 2020. [Online]. Available: https://www.dwavesys.com/d-wave-advantage-system-overview
[35] Z. Tabi, K. H. El-Safty, Z. Kallus, P. Haga, T. Kozsik, A. Glos, and Z. Zimboras, "Quantum optimization for the graph coloring problem with space-efficient embedding," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Oct. 2020, pp. 56–62.
[36] J. E. Dorband, "Improving the accuracy of an adiabatic quantum computer," 2017, *arXiv:1705.01942*. [Online]. Available: http://arxiv.org/abs/1705.01942
[37] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1941–1988, 4th Quart., 2015.
[38] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO detection techniques: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3109–3132, 4th Quart., 2019.
[39] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Minimum mean squared error interference alignment," in *Proc. Conf. Rec. 43rd Asilomar Conf. Signals, Syst. Comput.*, Nov. 2009, pp. 1106–1110.
[40] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005, doi: 10.1109/JSSC.2005.847505.
[41] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.
[42] O. Castañeda, T. Goldstein, and C. Studer, "Data detection in large multi-antenna wireless systems via approximate semidefinite relaxation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2334–2346, Dec. 2016.
[43] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–5.
[44] C. Jeon, O. Casta neda, and C. Studer, "A 354 Mb/s 0.37 mm^2 151 mW 32-user 256-QAM near-map soft-input soft-output massive MU-MIMO data detector in 28nm CMOS," *IEEE Solid-State Circuits Lett.*, vol. 2, no. 9, pp. 127–130, Sep. 2019.
[45] (Apr. 2021). *IIP-Group/massiveMIMOdetection*. Accessed: Oct. 29, 2020. [Online]. Available: https://github.com/IIP-Group/massiveMIMOdetection
[46] *D-Wave QPU Execution and Timing*. Accessed: Mar. 1, 2021. [Online]. Available: https://docs.dwavesys.com/docs/latest/c_post-processing_2.html
[47] C. McGeoch, P. Farre, and W. Bernoudy. (Sep. 2020). *D-Wave Hybrid Solver Service + Advantage: Technology Update*. [Online]. Available: https://www.dwavesys.com/hybrid-solver-service-advantage-technology-update
[48] D. Whitepaper. (2020). *Hybrid Solver for Discrete Quadratic Modes*. [Online]. Available: https://www.dwavesys.com/sites/default/files/14-1050A-A_Hybrid_Solver_for_Discrete_Quadratic_Models.pdf
[49] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. Katzgrabeer, "Physics-inspired optimization for quadratic unconstrained problems using a digital annealer," *Frontiers Phys.*, vol. 7, p. 48, Apr. 2019, doi: 10.3389/fphy.2019.00048.
[50] H. Goto, K. Tatsumura, and A. R. Dixon, "Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems," *Sci. Adv.*, vol. 5, no. 4, Apr. 2019, Art. no. eaav2372, doi: 10.1126/sciadv.aav2372.
[51] C. Stanek and T. Werner, "Personal protective equipment optimization validation," Res. Facilitation Lab. Army Anal. Group Monterey, Monterey, CA, USA, Tech. Rep. 0704-0188, 2020.
[52] H. Goto, K. Endo, M. Suzuki, Y. Sakai, T. Kanao, Y. Hamakawa, R. Hidaka, M. Yamasaki, and K. Tatsumura, "High-performance combinatorial optimization based on classical mechanics," *Sci. Adv.*, vol. 7, no. 6, Feb. 2021, Art. no. eabe7953, doi: 10.1126/sciadv.abe7953.

ZSOLT I. TABI received the B.Sc. and M.Sc. degrees in computer science from Eötvös Loránd University, Budapest, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in computer science.

From 2016 to 2020, he was a Developer on the MINI-LINK Project with Ericsson Hungary. Since 2020, he has been working with Ericsson Research Team as part of the ETH Quantum Project. He is also teaching programming languages at Eötvös Loránd University. His research interests include programming language design and adiabatic quantum computing.

ÁDÁM MAROSITS received the B.Sc. degree in electrical engineering from Budapest University of Technology and Economics (BME), Budapest, Hungary, where he is currently pursuing the M.Sc. degree in electrical engineering. He has done the internship at Ericsson Research. His research interests include optimization with quantum annealing, furthermore quantum communication and quantum random number generation.

ZSÓFIA KALLUS received the Ph.D. degree in physics from the Doctoral School of Physics, Eötvös Loránd University, Budapest. She studied open quantum systems and the structure and dynamics of large-scale real-world networks. She joined ER in 2014. She currently works as a Master Researcher at the Traffic Analysis and Network Performance Laboratory, Ericsson Research, Hungary. Her current research interests include emerging quantum technologies, quantum computing applications, and trustworthy AI solutions for optimization and automation of high-performance telecommunication systems. She is also interested in sustainability and urban sciences.

PÉTER VADERNA received the Ph.D. degree in physics from the Department of Physics of Complex Systems, Eötvös Loránd University, Budapest, in 2008, in the area of traffic modeling in communication networks. He currently works as a Master Researcher at the Traffic Analysis and Network Performance Laboratory, Ericsson Research, Budapest. His current research interests include AI in network analytics, network management, and network automation. His research interests also include emerging technologies that can potentially be involved in various business areas of telecommunication, such as quantum AI, camera-based positional tracking, and AR/VR.

ISTVÁN GÓDOR (Senior Member, IEEE) received the Ph.D. degree from Budapest University of Technology and Economics. He is currently a Research Leader at the Traffic Analysis and Network Performance Laboratory, Ericsson Research, Hungary. His research interests include network design, traffic analysis and modeling, self-organizing networks, energy efficiency, collective sensing, and the Industrial IoT and smart manufacturing. He is a member of the public body of the Hungarian Academy of Sciences (MTA). He was awarded the 2014 IEEE Communications Society Fred W. Ellersick Prize.

ZOLTÁN ZIMBORÁS received the Ph.D. degree from Eötvös Loránd University, Budapest. Later, he worked at the ISI Torino, University College London, and Freie Universität Berlin before returning to Hungary. He is currently the Head of the Quantum Computing and Information Group, Wigner Research Centre for Physics, Budapest. He is a member of several editorial boards and also a Board Member of QWorld, a non-profit organization that aims to develop open-source software for quantum programming at all levels. His research interests include quantum computing, quantum control, and quantum statistical physics.