Invited Paper: Heuristic Quantum Optimization for 6G Wireless Communications

Minsung Kim* (*Student Member, IEEE*), Srikar Kasi* (*Student Member, IEEE*), P. Aaron Lott, Davide Venturelli, John Kaewell (*Senior Member, IEEE*), and Kyle Jamieson (*Senior Member, IEEE*)

Abstract

6G technologies such as Massive MIMO, dense cells, innovative air interface multiplexing techniques, and ultra-high frequencies stand to significantly benefit from significantly-increased amounts of computation at the base station. This position paper surveys recent work the authors have undertaken to realize this vision on today's Noisy Intermediate Scale Quantum devices, illustrating possible system architectures to leverage the power of quantum devices for wireless networks. We sketch the state of the art in quantum processing devices, offering insights into their current and future evolution, and updating our recent experimental results with the most recent such devices, to give the reader a sense of the trajectory of performance improvements over the past 12-24 months.

Index Terms

Quantum Computation, NISQ, Wireless Networks, Massive MIMO, Channel Coding, 5G New Radio, 6G

I. Introduction

As wireless networks evolve through 5G, scaling up spectral density through the use of millimeter-wave frequency bands, Massive MIMO, and dense cells, network designers are looking ahead to the 6G roadmap, anticipating an even more data driven society where wireless brain-computer interfaces, extended reality, and connected robotics drive 6G networks to handle data rates 10–1,000× greater than 5G [1]. To scale up spectral efficiency, designers will consider implementing techniques such as ultra-massive MIMO arrays, innovative air interface multiplexing techniques, more robust forward error correction coding, and even higher-density network deployments in wider bandwidths at higher carrier frequencies. As spectral efficiency is scaled up, 6G system designers will strive to improve *key performance indicators* (KPIs) such as latency, reliability, and energy efficiency of terminals and base-stations while also trying to not compromise one KPI to achieve another. The implementation of algorithms for 6G to optimize data throughput, spectral efficiency, user density, reliability, and latency, operating in wider bandwidths will lead to exponentially more computation than current 5G systems.

In the base station and cellular infrastructure, 5G RF modem signal processing is based on classical compute concepts which are typically implemented in ASIC, FPGA and GPU/CPU fabrics. However, improvements in classical compute performance are not advancing at an exponential rate as in past years, but are plateauing due to transistors reaching atomic limits [2]. Since the design of efficient and fast computational structures is now competing with wireless communication as the most significant challenge for many high-capacity wireless communication systems, it is doubtful that silicon will be able to implement the high spectral performance, low latency, and high reliability optimization algorithms needed to achieve 6G's KPIs.

Quantum computing is a potentially valuable tool to address future tradeoffs between performance, latency, and reliability as the 6G roadmap evolves. If quantum computing enables optimal algorithms for heavy optimization problems that currently bottleneck achievable network throughput, spectral efficiency could then benefit. The numerous hardware platforms capable of quantum information processing could then combine with other scalable technologies such as millimeter wave and small cells, further increasing spectral efficiency. Due to the linearity of quantum mechanics, quantum computing is fundamentally constrained to rely on reversible operations, which dissipate no heat, except in the initialization and readout phase of the computation. While noisy quantum computation has elements of irreversibility, in the long term quantum computation can in principle reach arbitrarily low power consumption for computations that would be power-hungry if performed classically.

Over the last few years, due to advances in nanotechnology and engineering, real-world quantum computers have become commercially available. For wireless networks, recent studies first made use of a quantum annealer, a certain type of analog quantum computing processor, and showed promising results for a quantum-based Multiple Input Multiple Output (MIMO) detector in centralized radio access networks (C-RAN) [3] and for a quantum-based Low Density Parity Check (LDPC) error control decoding [4], providing guidance of how to make use of the machine and baseline performance metrics. In wireless networks, there are representative optimization problems, including but not limited to the previously studied applications, which suffer from well-known conventional trade-offs between throughput and complexity, where optimal solvers are known but very difficult to practically implement considering available hardware and processing time limitations. We expect that overcoming

*Kasi and Kim are co-primary authors, appearing in randomly chosen order. Kyle Jamieson, Srikar Kasi, and Minsung Kim are with the Department of Computer Science, Princeton University; John Kaewell is with InterDigital, Inc.; Aaron Lott and Davide Venturelli are with USRA Research Institute for Advanced Computer Science.

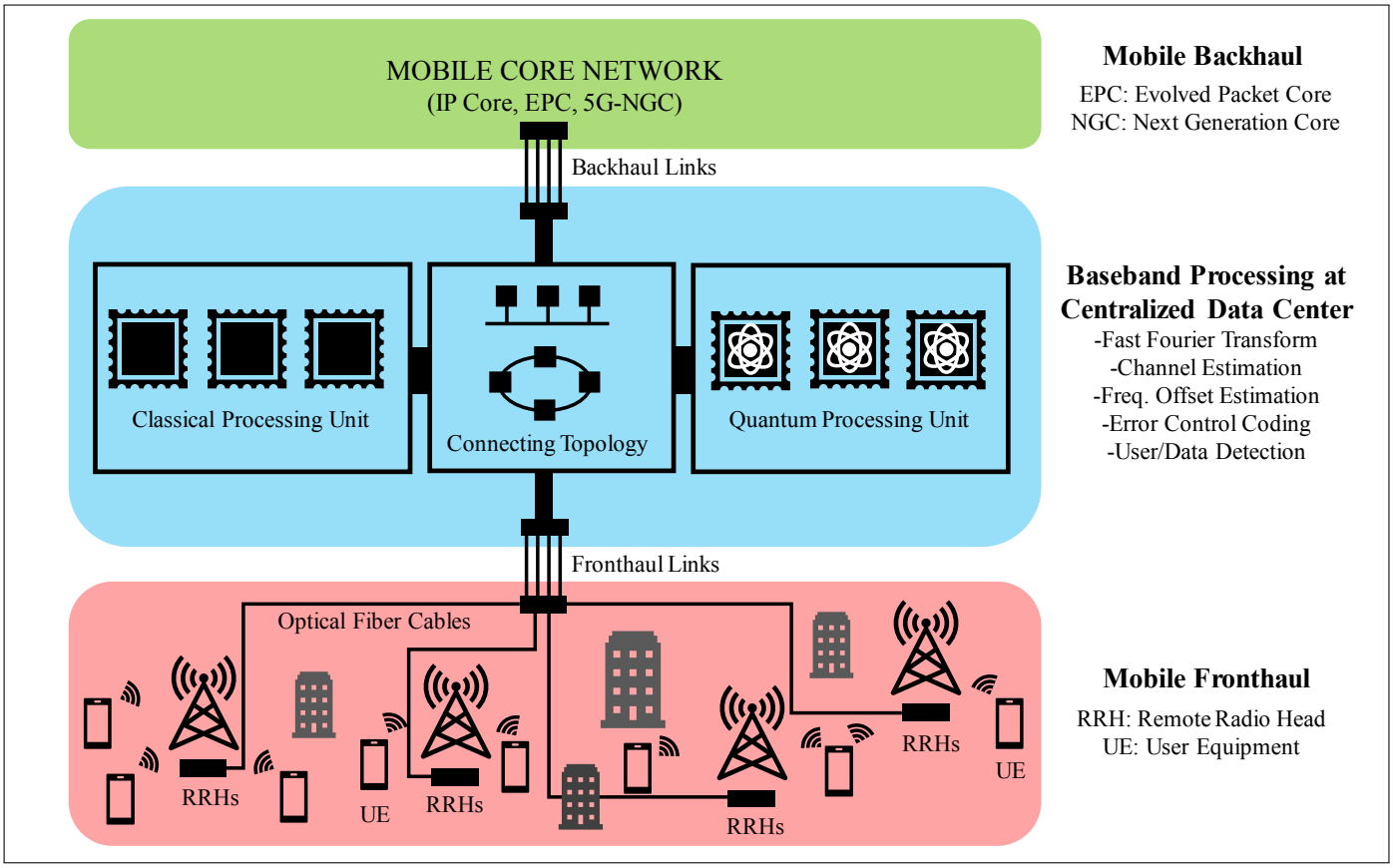

Fig. 1: A quantum compute-enabled system architecture for next-generation 6G wireless networks.

practical issues of these optimization problems existing on current algorithms and/or computer architectures will lead to drastic improvements in wireless communications and networks, achieving the optimal performance within short processing time and thus enabling Ultra-Reliable Low-Latency Communications (URLLC) visioned in 5G wireless networks and beyond [5], making investigation of quantum computing as a potential accelerator a compelling research objective. In our envisioned scenario shown in Fig. 1, quantum processors will be co-located with C-RAN computational resources in a data center, partitioning the compute roles for hundreds of base stations or Remote Radio Heads (RRH) connected via low-latency optical fiber, where quantum computing will take care of heavy optimization processing that bottlenecks achievable throughput and latency, while classical processors will take care of otherwise tractable processing. We believe that the C-RAN embedded quantum processors will have optimized interfaces and information processing architecture to maximize wireless system performance.

In the rest of this article we provide a brief tutorial on using currently available quantum computing technology, sometimes referred as Noisy-Intermediate-Scale Quantum (NISQ) technology [6], to solve optimization problems in wireless networks, present two case studies and relevant performance results from each, and share critical lessons learned, including challenges and future directions for network designers and engineers.

II. NISQ Architectures and Optimization Algorithms

There are several models of quantum computation that can be implemented via an array of technologies. Classical computers store and manipulate bits, whereas quantum computers use *qubits*, physical devices that can encode a combination of 0 and 1 bits simultaneously via quantum dynamics. A useful dichotomy to frame the current landscape is fault-tolerant approaches to quantum computing vs. NISQ computing (digital or analog). Fault tolerant approaches require a level of control of the quantum resources that is still far away from what technology can deliver in the next few years. For this reason, much of the applied work in quantum computing has focused on co-design of hardware and software that, while not necessarily scalable, could work in specific devices to deliver a quantum advantage in specific problems. NISQ processors that could be used to address optimization problems today can be further classified into gate or annealing architectures.

A. Gate Model Processors

The design and implementation of current and near-term gate-based NISQ architectures are guided by the fault-tolerant theoretical models that offer general computational functionality (*universality*) using programmable logic gates acting on

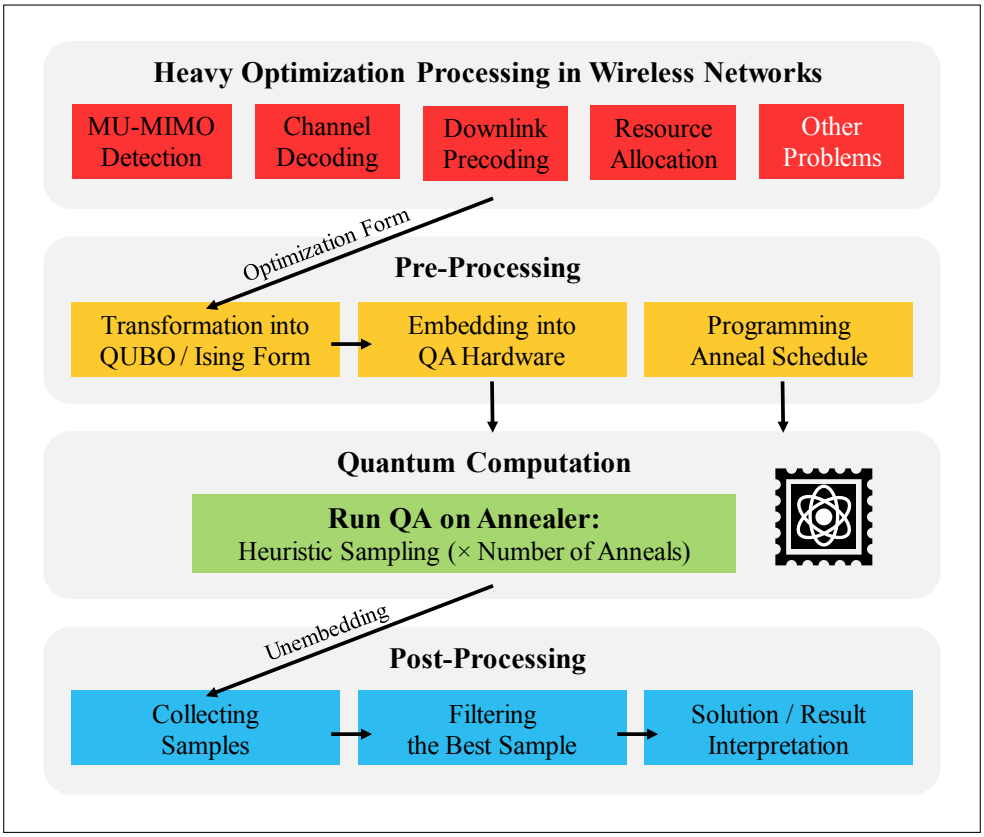

Fig. 2: The general workflow of QA-based optimization in wireless networks.

qubits—in analogy with classical digital architectures. These devices implement several programmable gate-sets that can represent a variety of algorithms. The most advanced platforms include superconducting, solid state chips by Google, IBM, Intel, Rigetti Computing, IQM, Alibaba and atom-based systems by IonQ, Honeywell, ColdQuanta, AQT, QuEra, and Pasqal. Calibrating these gates to enable controllable quantum dynamics with high-fidelity, requires close alignment of material properties. Noise sources reduce the number of qubits that can be calibrated to perform gate operations. For instance, current commercial superconducting architectures are limited by coherence times in the 10s of μs regime, allowing only tens of gate operations on up to 10–50 qubits. Public design targets based on improvements in coherent materials, fabrication processing and circuit design could enable thousands of gate operations on the order of 100–1000 qubits within the next 5 years. The Quantum Alternating Operator Ansatz (QAOA) [7] algorithm is designed to leverage these advances to solve optimization problems. This heuristic procedure consists of two alternating phases: an exploration step (*mixing*) and an exploitation step (*phase separation*) where the operations of two phases depend on parameters that are set by leveraging statistics obtained in real-time by operating the device as a neural network in the training phase. Analysis and modifications of the original QAOA algorithm have been developed in order to leverage properties of current and near-term NISQ architectures. However, the most advanced tests have been able to solve the MaxCut problem up to 23 nodes [8], a benchmark still far away from applications of practical value.

B. Quantum annealers and Ising Machines

While the gate-model is motivated by abstractions from theoretical computer science, *quantum annealing* (QA) is inspired by the adiabatic principle of quantum mechanics, a useful means to search for configurations in a high-dimensional energy landscape that correspond to low energy states. To solve problems via annealers, computational problems are mapped on to an optimization form in which the solution corresponds to a configuration of variables that specifies the location of the minimum in a high-dimensional energy landscape. In current hardware, there are two equivalent mathematical forms to describe the optimization problem, a *Quadratic Unconstrained Binary Optimization* (QUBO) problem or *Ising model*, depending on whether the variables are encoded as bits $\in \{0, 1\}$ or *spins* $\in \{-1, +1\}$ respectively.[1] Unlike the gate-based approach, the computation is performed via a continuous process. QA hardware implements annealing algorithms for minimizing Ising models. Problems are specified by prescribing a real-valued matrix of values where diagonal entries define the *bias*, denoted

[1]Since QUBO and Ising forms are equivalent and can be easily transformed into each other, we use both terms interchangeably in this article.

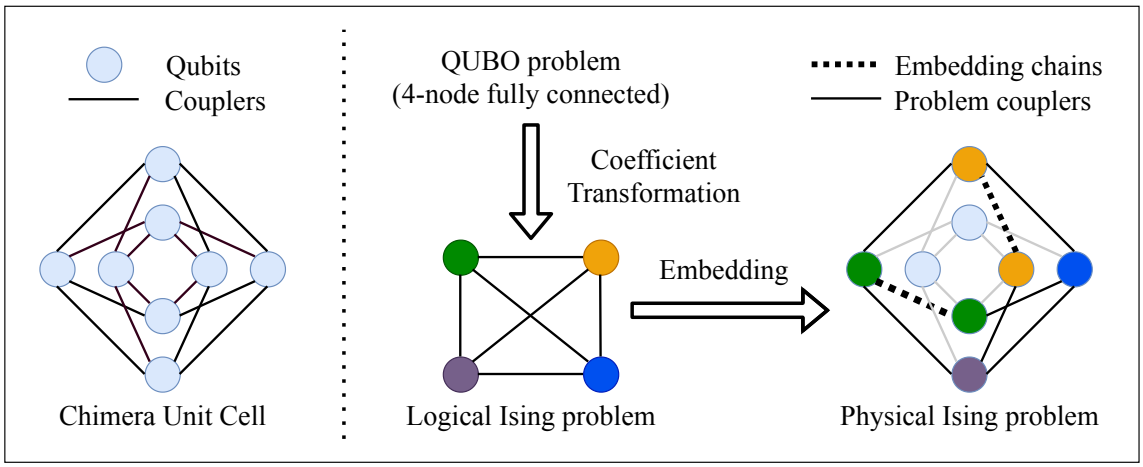

Fig. 3: Mapping process of a QUBO problem onto the physical Chimera unit cell architecture featured in DW_2000Q.

by h_i, for each variable configuration, and the off-diagonals define *coupling strengths*, denoted by J_{ij}, correlating pairs of variables. Each spin is encoded into a qubit and the biases and coupler strengths are programmed into the QA device using on-chip control circuitry and the minimal energy configuration of spins is determined by evolving a *transverse field Ising model Hamiltonian*, where the *transverse field energy* and *problem energy* together define the *schedule* of the annealer, which represents annealing algorithms. Starting with the transverse field energy much greater than the problem energy, the annealing algorithm initializes the system in a ground state of an initial Hamiltonian where each qubit is in a *superposition state*, then gradually evolves this Hamiltonian by decreasing the transverse field energy and increasing the problem energy, such that at the end of the anneal the problem energy is much greater than the transverse field energy. By driving the Hamiltonian changes slowly enough, the annealing algorithm reaches the lowest energy configuration of spins corresponding to the input problem with high probability. However, physical chips operate in noisy environments, noise sources such as thermal bath, high-energy photons and electromagnetic cross talk, causing the performance to escape numerical or analytical predictions and be determined in practice by a heuristic parameter setting procedure.

Companies such as D-Wave Systems and large, publicly funded federal, Japanese or EU consortia have developed annealer machines. Moreover, the quantum annealer design has inspired similar hardware architectures that either exploit quantum mechanical fluctuations as a minor resource or are purely inspired by the quantum analog dynamics. These *Ising machines* include Optically Coherent Ising Machines (CIM) from NTT, digital annealers from Fujitsu, and simulated bifurcation machines from Toshiba. The performance is determined by very different physics-based principles, but from an end-user perspective, while the programming model is the same (QUBO), each architecture balances trade-offs between precision of inputs, graph connectivity, and total number of variables. Some physics-inspired algorithms like simulated annealing can be immediately implemented on classical computing platforms such as CPUs and GPUs [9].

Currently D-Wave provides around 5000 sparsely connected qubits whereas digital annealers operate with up to 8192 fully connected variables at 16-bit precision and 4096 variables at 64-bit precision. A typical workflow of QA-based optimization with some application examples in wireless networks is shown in Fig. 2. Most problems that are not natively defined as a QUBO can be mapped to a QUBO via a *penalty method*, often requires high precision and several ancillary qubits to represent the penalties as logical binary variables in a QUBO. Logical binary variables in the QUBO may need to be mapped to multiple qubits to express the connections of the logical variables. If an architecture does not provide a fully-connected graph topology, the logical QUBO must be mapped onto a physical architecture through a graph minor *embedding*. In Fig. 3 we demonstrate the process of embedding a QUBO problem into the Chimera topology (C16) of the D-Wave 2000Q quantum annealer (DW_2000Q). To embed a dense problem into a sparse architecture graph, one must apply additional penalties to *chain* qubits together via strong ferromagnetic couplings causing qubits to be strongly correlated (see dotted lines in Fig. 3). The number of qubits that represent a logical variable is called the *chain length* of that variable. For instance, in Fig. 3 note that the chain length of orange variable is two. In general, this embedding process can require many qubits and high precision to couple them effectively, reducing the problem size and available precision to represent the coefficient matrix. For example, DW_2000Q features 2,048 qubits and six couplers per qubit, and can implement at maximum a fully-connected graph of 64 variables. Embedding these maximum size cliques involve very long chains of qubits causing solution quality to suffer since these long chains require high precision to couple and longer range coherent dynamics to co-tunnel.

Recent updates to QA architectures include a lower noise QA architecture (DW_2000Q_6) and a more connected architecture *Advantage_system1.1* that implements a Pegasus graph topology. Pegasus P(M) contains an overall $24M(M-1)$ qubits and 15 couplers per qubit, with native K_4 and $K_{6,6}$ subgraphs. With P16 being the current hardware scale, Advantage_system1.1 consists a total of 5,760 qubits. Practical challenges with today's QA devices include embedding, precision, and additional

overheads, which are well beyond pure computing time, along the lines of pre-processing (30–50 ms), programming (8–9 ms), and readout times per anneal (100–300 μs).

III. Quantum Computing for Wireless Networks

In this section, we present two case studies, focusing on QA technologies due to the problem sizes and architectures available today. While the general descriptions of processing on the QA hardware (Fig. 2) are introduced in the previous section, here we investigate application-specific blocks with two wireless applications: MIMO detection and LDPC decoding. We review the recent study and share critical lessons learned on important aspects of applying QA to each application including QUBO formulation and embedding. For fault-tolerant approaches that require more advanced quantum processors that will be available in the next few years, we refer the reader to the review [10].

A. Review: QA-enabled Multi-User MIMO Detection

Multi-User MIMO (MU-MIMO) employs spatial multiplexing to enable parallel spatial streams, and is considered one of the most promising ways to increase wireless capacity. Thus MU-MIMO has featured in nearly every networking standard, including cellular and local area networks. In MU-MIMO systems, the receiver must demultiplex mutually-interfering streams in order to detect a signal for each user (*MU-MIMO detection*). Linear methods such as Zero-Forcing (ZF) and Minimum Mean Square Error (MMSE) are commonly used, featuring low computational complexity. However, the detection performance rapidly degrades as the number of user antennas approaches the number of receiver antennas where the wireless channel becomes poorly-conditioned, which is critical since the MU-MIMO system generally needs to support more users at a time for higher throughput. The Sphere Decoder (SD), which is an optimal Maximum Likelihood (ML) solver, improves detection performance in these situations, but requires an exponentially-increasing amount of computation as the user number increases and thus the SD cannot always satisfy processing time requirements.

Using QA to perform MIMO detection was initially studied on the DW_2000Q [3]. The work provided guidance of how to apply QA for optimal ML MIMO detection, including a reduction method of the ML MIMO detection into QUBO formulation and an embedding method. We summarize important observations and results with three different aspects:

- **QUBO formulation.** The key idea of the QUBO reduction introduced in the work is to find a linear mapping between possible symbols and binary variables, and replace the symbols with this mapping in the ML objective function. Then the norm expansion in the objective function results in a QUBO form. For the mapping, each symbol per user requires $\log_2 |\mathcal{O}|$ logical binary variables, where $|\mathcal{O}|$ is the size of modulation, and thus $N_\mathrm{v} = N \cdot \log_2 |\mathcal{O}|$ logical variable count is required to support N users at a time. The generated QUBO does not include any constraint terms, so generalized forms can be easily obtained, given $|\mathcal{O}|$ and N. Once the receiver estimates the wireless channel and receives the signal, input coefficients can be immediately generated, since the required computation time and resources for the QUBO reduction are insignificant. The QUBO form of the ML MIMO detection was also tested with classical heuristics on CPUs and GPUs [9].
- **Embedding.** The generated QUBO of the ML MIMO detection is nearly fully connected; most QUBO coefficients are non-zero. While the minor embedding itself is another NP problem, in the case of fully connected problems very efficient direct embeddings called *clique embedding* [11] are known for many different architectures including Chimera and Pegasus. This embedding method is straightforward and easily extended to advanced architectures with more connectivity. In the case of Chimera structure, MIMO problems with up to $N_\mathrm{v} = 64$ can be embedded on C16 for the ML detection by this method, requiring $N_\mathrm{v}(\lceil N_\mathrm{v}/4 \rceil + 1)$ qubits and $\lceil N_\mathrm{v}/4 \rceil + 1$ chain lengths.
- **Detection performance.** For low-order modulations such as BPSK ($|\mathcal{O}| = 2$) and QPSK ($|\mathcal{O}| = 4$), this QA-based MIMO detector achieves promising detection performance, enabling Large MIMO with over 10 users, even assuming the same number of users and receiver antennas (i.e., $N \times N$ MIMO). However, for 16-QAM ($|\mathcal{O}| = 16$) or higher-modulations, the detector does not perform well even with small number of users. The quality of sampling depends on the distribution of possible values of the objective function, due to the presence of analog noise. Smaller gaps between the values, especially between the ground state and second best candidate, result in worse sampling performance, and $|\mathcal{O}|$ affects that distribution more critically than N. Furthermore, wireless channel noise makes the gaps smaller in general, so higher channel noise leads to a worse sampling quality, and thus longer processing time, for the same detection performance.

B. Review: QA-enabled Decoding of Error Control Codes

Modern communication standards are increasingly utilizing LDPC codes for correcting bit-errors in data transmissions because of their capacity-approaching error performance. With the immense progress in LDPC code construction, today's research is directed towards designing efficient hardware implementations for the computationally complex processing demands of the decoder. LDPC codes are traditionally decoded via the iterative belief propagation (BP) algorithm. This approach is highly attractive, allowing network designers to select several custom design parameters such as likelihood bit-precision, iteration limit, and decoder parallelism, but requires critical trade-offs between decoder accuracy, throughput, and flexibility.

As a result, practical LDPC decoders today are typically realized using partially-parallel architectures, with limited calculation precision, implying that current silicon technology often does not fully exploit the potential of LDPC codes in practice.

One may circumvent these trade-offs altogether with the representation of the LDPC decoder as a QUBO problem. Flexible decoders allow the communication system to dynamically adapt parity check matrices (PCMs) to time-varying transmission conditions, such as decreasing the coding rate in high noise environments. As PCM considerations vary with time, a QUBO decoding requires adjustments only in the values of the QUBO coefficients, implying that a QUBO decoding may be more flexible than BP decoding which requires hardware reconfiguration to target different code structures. Furthermore, recent QA implementations of LDPC decoders have shown to quantitatively outperform BP-based silicon FPGA implementations under typical likelihood bit-precision and iteration limits [4], while further similar QA-based studies have found correct solutions for LDPC decoding problems for which the BP algorithm does not converge for 1000 iterations [12]. Resembling a fully parallel decoder, QUBO decoding approaches eschew the sequentially iterative nature of the BP algorithm, opening the door to potentially accelerate decoding throughput. We next summarize the core ideas and future directions with QA-based LDPC decoding [4]:

- **QUBO formulation.** To represent the LDPC decoder as a QUBO problem, the key idea is to construct two types of cost penalty functions that: 1) Ensure the LDPC encoding conditions are satisfied (i.e., zero checksum). This is obtained by encoding integer sum of bits (variables) in check constraints into even integers. 2) Ensure the decoder to select the codeword with closest proximity to the received information (with wireless channel noise).
  - **Future directions.** The encoding of the variable sum in the LDPC check constraints to even integers can take many possible forms, with unary/binary encoding minimizing/maximizing the coefficient values and maximizing/minimizing the number of additional QUBO variables respectively. The trade-off between the number of variables and coefficient values determines the encoding form, and it can be chosen to best fit the available QA hardware in hand.
- **Embedding.** The graph of the LDPC decoder QUBO is sparse, and consists a particular repeating connectivity pattern that depends on the check bit degrees of the LDPC code. Although heuristic embedding methods and clique embedding can be used to map generic problem graphs onto QA hardware, it is possible for sparse problems to find more efficient embedding designs. For instance, a custom embedding design for (2,3)-regular LDPC codes presented in an earlier study makes use of the entire QA hardware [4]. The regular variable connectivity in the LDPC QUBO graph allows for flexible extension of embedding patterns to different LDPC code block lengths.
  - **Future directions.** A promising direction is to find the regularly repeating embedding patterns for check bit degrees employed in practical protocol standards, to investigate the code block lengths the available QA hardware supports. This will allow network designers to keep on track with the potential of QA hardware advances, and may further motivate the designers of future quantum devices to tailor the hardware to the problem of interest.
- **Directions for higher-order optimization.** It is also possible to design LDPC decoder as a higher order optimization (*Polynomial Unconstrained Binary Optimization* or *PUBO*) problem for decoding with minimal number of problem variables, by mapping bits in LDPC check constraints to Ising problem variables ($0 \longrightarrow -1$ and $1 \longrightarrow +1$). The sign of the product of the Ising variables (in a given check constraint) determines if the LDPC check-sum encoding condition is satisfied. Although practical machines that optimize PUBO problems are currently not available, a future possibility may arise, while PUBO forms can be reduced to QUBO forms by existing quadratization penalty methods.

IV. Next Steps for QA for Wireless Networks

Along with the previously seen practical challenges, many open problems remain as well for network designers, directly related to pure QA computation performance. In wireless networks, there exist many potential applications that could achieve benefits from the use of QA computation. In this regard, their QUBO formulations and embedding techniques need to be optimized, considering the impact of noises and sparse connectivity on the device, which is a general open problem in the field. While we introduce some possible examples with two specific case studies in this article, more applications (e.g., downlink precoding [13]) and/or further advanced QUBO formulations and embedding methods should be considered and studied towards expected performances in 6G. Furthermore, potential next steps in the area include studies that continue and elaborate the head-to-head error-rate and throughput comparison of QA against silicon implementations [4] under similar wireless network parameters. While the impact of silicon hardware parameters such as bit-precision, clock frequency, and routing designs have been well investigated, we here discuss free QA parameters and advanced annealing techniques to boost performance.

- **Pre-processing of coefficients.** QA devices introduce an analog machine noise called *ICE* or *intrinsic control error* into the input problem. ICE noise may degrade the solution quality of problems with narrow energy gaps, nevertheless pre-processing and tuning the coefficient values typically help mitigate the adverse effects ICE noise. Particular to wireless networks, optimization problems involving wireless channel matrices generally result in wide spread of coefficient values, with very low and very high values. Eliminating or pruning such extreme coefficients after the problem embedding potentially mitigates the effect of ICE noise and tailors the coefficients into supported limits.

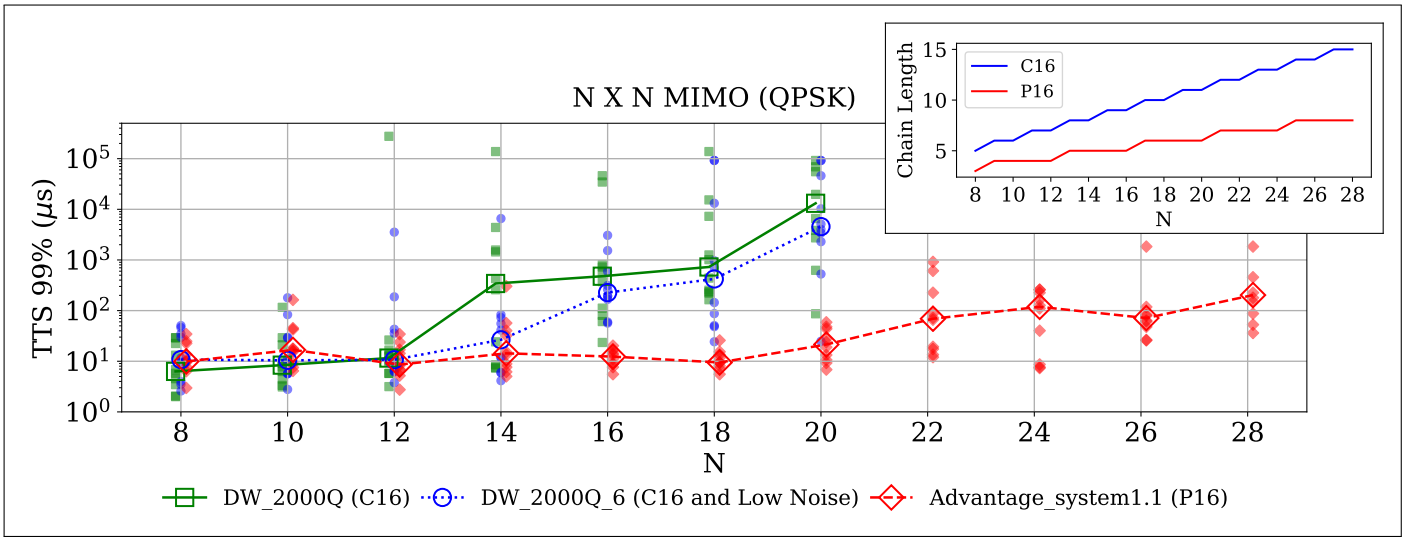

Fig. 4: TTS comparison as a function of MIMO size (N) for three different quantum annealers. Both DW_2000Q-based devices use a Chimera architecture (C16), while the Advantage_system1.1 uses the Pegasus architecture (P16). In the graph insert, we plot the chain length for both architectures as a function of N.

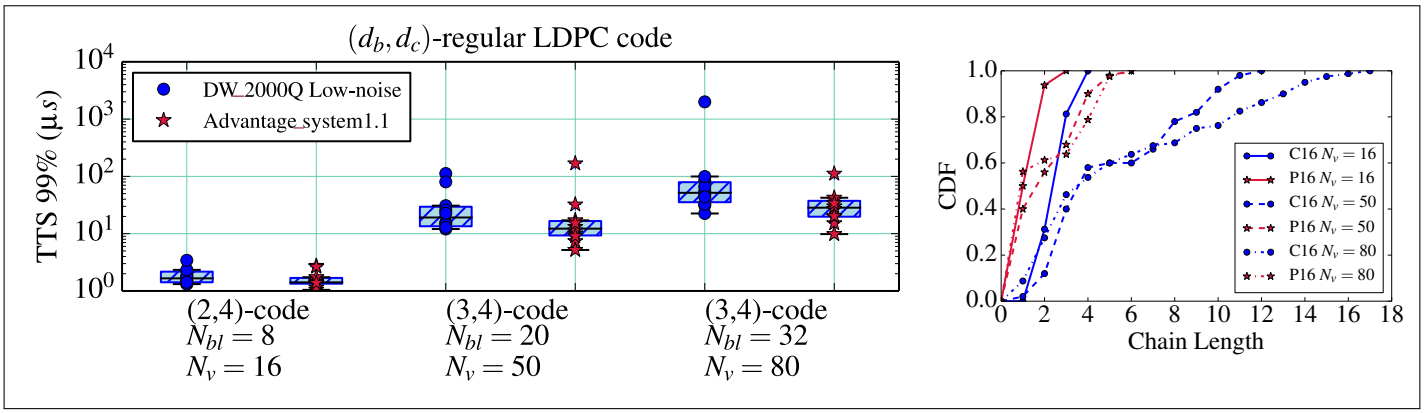

Fig. 5: TTS comparison of different quantum annealers for various LDPC decoding problems. In the figure, N_{bl} is the LDPC code block length, and N_v is the number of problem variables in QUBO formulation of the respective code. The distribution in figure *right* is across the chain lengths of the variables present in figure *left* in their respective embeddings.

- **Hybrid computation.** There has been a surge of interest in hybrid classical-quantum computation structures, which may improve performance by leveraging the advantages of both classical and quantum computing resources [14]. For example, *Reverse Annealing* (RA), a variation of QA, naturally provides an opportunity of hybridization, where it begins annealing from a given classical configuration for refined search. Earlier studies evaluated a hybrid prototype based on RA to ascertain feasibility and compare against fully-quantum designs for QA-based MIMO detection [15]. The key result obtained is that RA initiated from the solution of classical solvers is able to improve performance over solely QA computation in terms of processing time, even with the simplest classical solver that frequently fails to obtain the ground state, but takes nearly negligible compute time.
- **Parameter setting.** Many interdependent free parameters related to embedding and annealing schedule influence the QA sampling performance and the processing time. However, it is challenging to find the best parameter settings, since there is no theoretical guidance that applies to noisy QA machines. Several orders of magnitude performance gap is observed between the optimized and median best settings for all the instances tested by QA-based MIMO detection and LDPC decoding, implying that the choice of machine parameters is critical to performance. Efforts such as applying neural networks to estimate the best parameter setting are currently underway, but to our best knowledge, no significant progress has been reported so far.

TABLE I: The MIMO sizes and LDPC code block lengths up to which the respective QA hardware can support. P24 and P30 are extended versions of the Pegasus topology.

QA hardware graph	Qubits	Couplers	MIMO detection (users $N \times N$)				LDPC decoding (block length)
			BPSK	QPSK	16-QAM	64-QAM	(2, 3)-regular [4]
Chimera C16	2048	6016	64	32	16	10	420
Pegasus P16	5640	40,484	180	90	45	30	1175
Pegasus P24	13,064	95,204	276	138	69	46	2720
Pegasus P30	20,648	151,364	348	174	87	58	4300

V. Illustrative Results

In this section, we evaluate QA-based MIMO detection and LDPC decoding problems using three different QA devices: the DW_2000Q (C16 topology), DW_2000Q_6 (C16 topology and low-noise), and Advantage_system1.1 (P16 topology and state-of-the-art). These early results on new platforms consider *Time-to-Solution* (TTS) as the figure of merit for the computation, though our earlier work has measured bit error rate as well. TTS (99 percent) represents the time required to obtain the *ground state* (solution) of input problem with 99 percent probability. To focus on achievable gains with hardware, we conduct our experiments without channel noise present in wireless systems.

A. MIMO Detection

In the case of MIMO detection, the ground state corresponds to the ML solution. Fig. 4 plots TTS performance as a function of MIMO size N with QPSK modulation, where data points report each channel use instance and lines report the median across many instances. It is observed that compared to Chimera-based annealers, the Pegasus-based Advantage_system1.1 achieves results that scale better with the MIMO size, thus improving TTS with smaller variances of performance across instances, obtaining approximate 1000× median gains at $N = 20$. For Chimera-based annealers, while the DW_2000Q_6 is able to achieve 2–10× reduced processing time compared against the baseline DW_2000Q at some points, no dramatic gain is observed in general.

B. LDPC Decoding

In the case of LDPC decoding, the ground state corresponds to the decoded codeword. Our evaluation of the QA LDPC decoder is on the same aforementioned machines. In Fig. 5 *(left)*, we compare TTS performance for multiple randomly chosen problem instances of (d_b, d_c)-regular LDPC codes, where d_b and d_c are bit and check node degrees of the code respectively. The data points in the figure represent individual problem instances, the boxes' lower/upper whiskers and quartiles represent 10^{th}/90^{th} and 25^{th}/75^{th} percentiles respectively, and horizontal lines inside the boxes are medians. We first observe in Fig. 5 *(left)* that TTS is less than 100 μs for most of the problems solved on both the DW_2000Q_6 and Advantage_system1.1, and that gains in TTS increase with increase in problem size, reaching up to a 10× TTS gain for 80-variable problems when they are solved on the Advantage_system1.1 QA hardware.

C. Discussion

In order to understand these performance gains, we investigate how chain lengths are distributed in the embedded problems. The clique embedding used in MIMO detection has typically the same chain lengths for all problem variables as shown in Fig. 4 (*insert*), and the heuristic embedding used in LDPC decoding consists of different chain lengths across variables as in Fig. 5 (*right*). To leverage the advantage of sparseness in the LDPC QUBO (i.e., solving problems with more variables), we opt for heuristic embedding over clique embedding for LDPC decoding. The P16 topology allows for significantly lower chain lengths than those of C16 due to its denser qubit connectivity. This reduction in chain lengths decreases the chance of embedding failures (i.e., broken chains) and reduces the accumulated ICE noise of qubits (due to fewer used qubits), which both point to the performance advantage the P16 topology offers. Indeed, similar performance is achieved for similar chain lengths with fewer variables, for example $N \approx 14$ (Chimera) and $N \approx 28$ (Pegasus) MIMO sizes. Table I outlines the maximum supported MIMO size at different modulations, and the maximum supported LDPC code block lengths, on the previous (C16), current (P16), and predicted future (P24, P30) QA processor topologies.

VI. Conclusion

This work describes several opportunities and challenges in applying quantum computation techniques for problems in wireless networks. While we focus our analysis on existing QA technology in our case studies due to the problem sizes that can be solved on the architectures available today, we have provided an overview of the anticipated future benefits quantum technology may enable for 6G wireless networks, and foreshadow how the computation structures on today's architectures may

generalize to the broader set of quantum platforms that we anticipate will be available in the next five to ten years. To see how advances in QA technology benefit wireless systems, we implement in hardware two computationally demanding wireless uplink problems, MIMO detection and LDPC decoding, on different real-world quantum annealers. Our results show that a state-of-the-art QA machine requires significantly less time to reach the sought solution compared to previous QA processors.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. CNS-1824357 and CNS-1824470, and a Princeton School of Engineering Innovation Fund award. The Princeton Advanced Wireless Systems research group gratefully acknowledges use of D-Wave machine time under the USRA Cycles 3 and 4 Research Opportunity Award Programs.

REFERENCES

[1] P. Yang *et al.*, "6G wireless communications: Vision and potential techniques," *IEEE Network*, vol. 33, no. 4, pp. 70–75, 2019.
[2] ITRS, "International technology roadmap for semiconductors 2.0, executive report (2015)," 2015.
[3] M. Kim, D. Venturelli, and K. Jamieson, "Leveraging quantum annealing for large MIMO processing in centralized radio access networks," in *Proceedings of the ACM SIGCOMM*, 2019, pp. 241–255.
[4] S. Kasi and K. Jamieson, "Towards quantum belief propagation for LDPC decoding in wireless networks," in *Proceedings of the 26th ACM MobiCom*, 2020, pp. 663–676.
[5] J. Wu *et al.*, "Cloud radio access network (C-RAN): a primer," *IEEE Network*, vol. 29, no. 1, pp. 35–41, 2015.
[6] J. Preskill, "Quantum computing in the NISQ era and beyond," *arXiv preprint arXiv:1801.00862*, 2018.
[7] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint 1411.4028*, 2014.
[8] M. P. Harrigan *et al.*, "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor," *Nature Physics*, vol. 17, no. 3, pp. 332–336, 2021.
[9] M. Kim *et al.*, "Physics-inspired heuristics for soft MIMO detection in 5G new radio and beyond," in *Proceedings of the 27th ACM MobiCom*, 2021, p. 42–55.
[10] P. Botsinis *et al.*, "Quantum search algorithms for wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1209–1242, 2018.
[11] T. Boothby, A. D. King, and A. Roy, "Fast clique minor generation in chimera qubit connectivity graphs," *Quantum Information Processing*, vol. 15, no. 1, pp. 495–508, 2016.
[12] Z. Bian *et al.*, "Discrete optimization using quantum annealing on sparse Ising models," *Frontiers in Physics*, vol. 2, p. 56, 2014.
[13] S. Kasi *et al.*, "Quantum annealing for large MIMO downlink vector perturbation precoding," *arXiv preprint arXiv:2102.12540*, 2021.
[14] R. Sweke *et al.*, "Stochastic gradient descent for hybrid quantum-classical optimization," *Quantum*, vol. 4, p. 314, 2020.
[15] M. Kim, D. Venturelli, and K. Jamieson, "Towards hybrid classical-quantum computation structures in wirelessly-networked systems," in *Proceedings of the 19th ACM HotNets*, 2020, pp. 110–116.

Minsung Kim [S] (minsungk@princeton.edu) is a current Ph.D. student in the Department of Computer Science at Princeton University. He received his B.E. (Electrical Engineering, Great Honor, 2016) degree from Korea University and M.A. (Computer Science, 2019) degree from Princeton University. His research interest includes 5G wireless networks and quantum computing.

Srikar Kasi [S] (skasi@princeton.edu) is a Ph.D. student in the Department of Computer Science at Princeton University (PAWS research group). He received his B.Tech degree (Electrical Engineering, 2018) from the Indian Institute of Technology (IIT) Delhi. His research interests include wireless networks, quantum computing, graph theory, and mobile systems.

P. Aaron Lott (plott@usra.edu) is a senior research scientist in the Research Institute of Advanced Computer Science at USRA. He earned his Ph.D. in applied mathematics and scientific computation at the University of Maryland, College Park, and performed post-doctoral research at the National Institute of Standards and Technology and Lawrence Livermore National Laboratory.

Davide Venturelli (dventurelli@usra.edu) is the Associate Director for Quantum Computing of the Research Institute of Advanced Computer Science at USRA, working at the Quantum AI Laboratory (QuAIL) as a senior research scientist under the NASA Academic Mission Service contract. Prior to joining USRA and QuAIL, he obtained is Ph.D. at SISSA in Trieste and University of Grenoble and worked as a postdoc at the Normale School in Pisa.

John Kaewell [SM] (john.kaewell@interdigital.com) joined InterDigital in 1986 where he has developed multiple generations of wireless communication systems. He leads InterDigital's exploration of using quantum computing to solve wireless optimization problems. Mr. Kaewell has been inducted into the Drexel College of Engineering Circle of Distinction. He holds 56 US Patents.

Kyle Jamieson [SM] (kylej@princeton.edu) is Professor of Computer Science and Associated Faculty in Electrical and Computer Engineering at Princeton University. He received the B.S. (Mathematics, Computer Science), M.Eng. (Computer Science and Engineering), and Ph.D. (Computer Science, 2008) degrees from the Massachusetts Institute of Technology. He then received a Starting Investigator fellowship from the European Research Council, a Google Faculty Research Award, and the ACM SIGMOBILE Early Career Award. He served as an Associate Editor of IEEE Transactions on Networking (2018–2020).

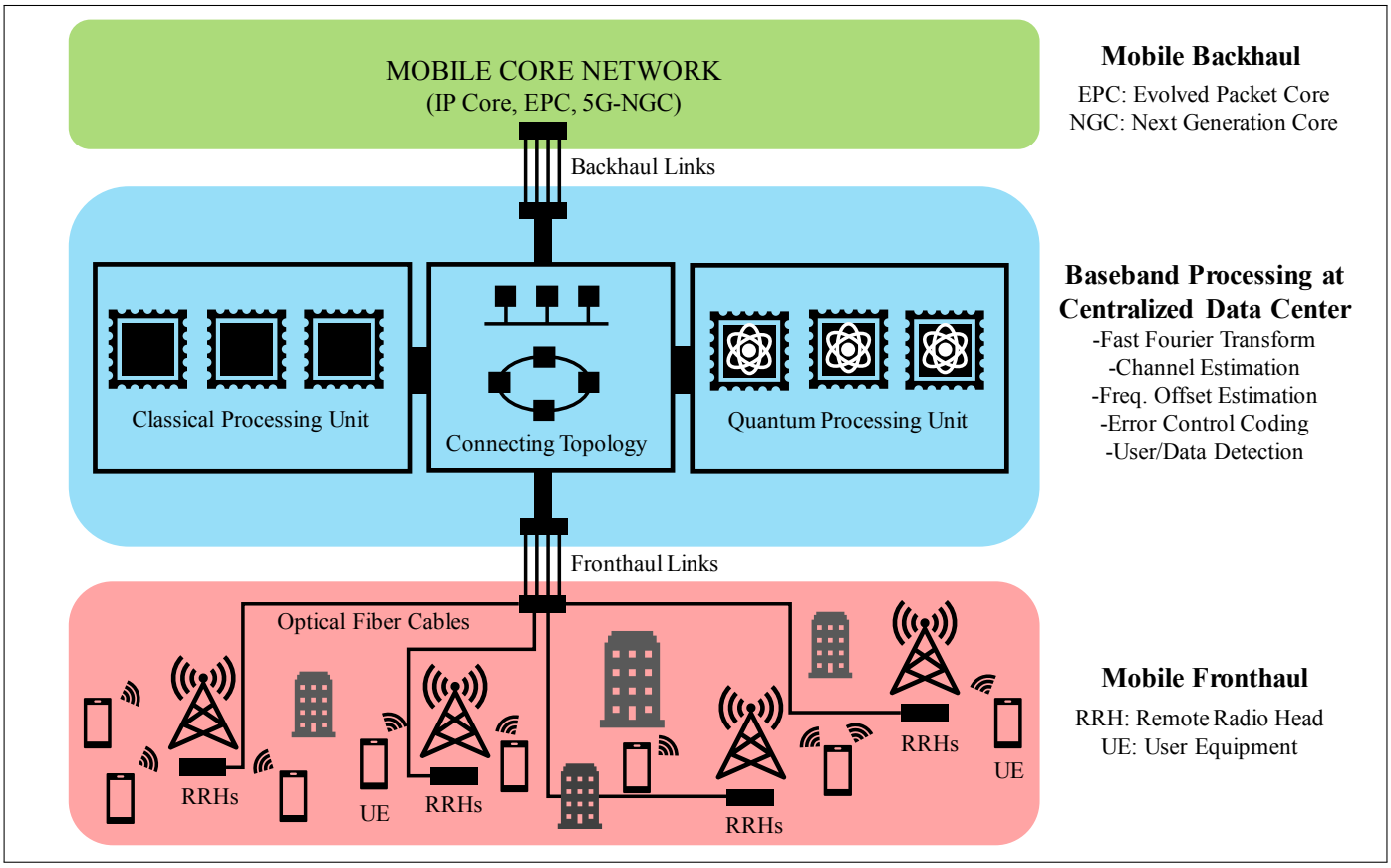

Fig. 1: A quantum compute-enabled system architecture for next-generation 6G wireless networks.

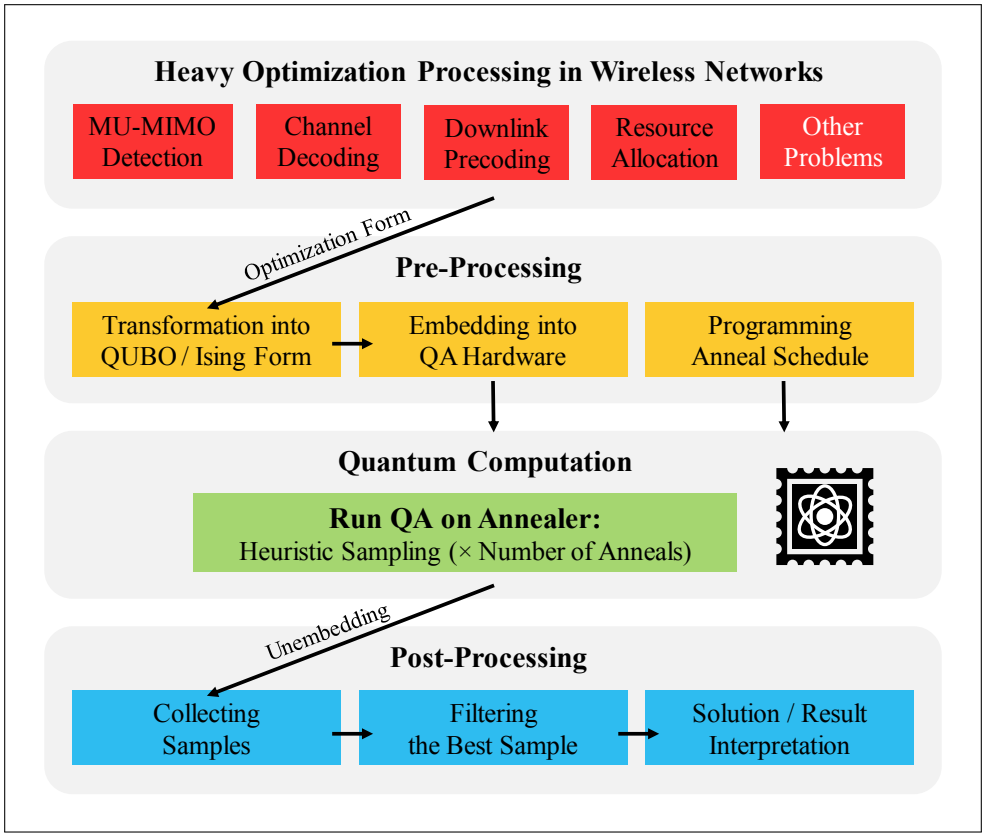

Fig. 2: The general workflow of QA-based optimization in wireless networks.

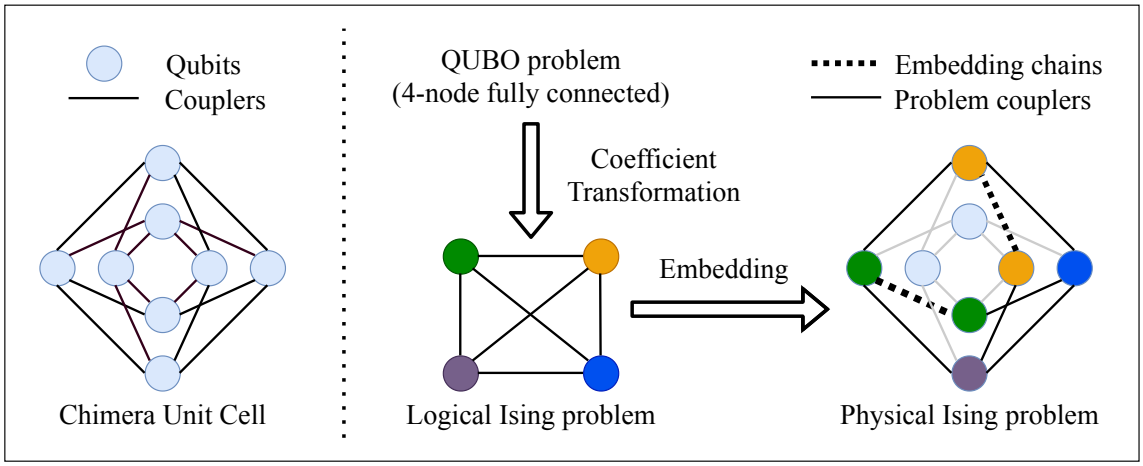

Fig. 3: Mapping process of a QUBO problem onto the physical Chimera unit cell architecture featured in DW_2000Q.

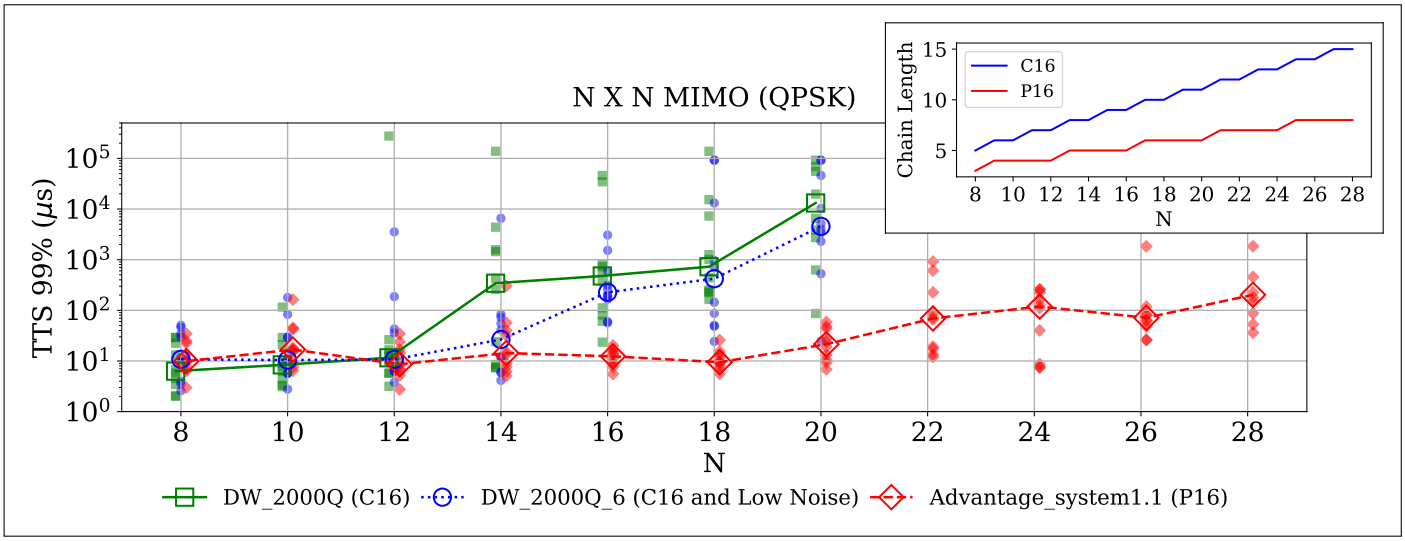

Fig. 4: TTS comparison as a function of MIMO size (N) for three different quantum annealers. Both DW_2000Q-based devices use a Chimera architecture (C16), while the Advantage_system1.1 uses the Pegasus architecture (P16). In the graph insert, we plot the chain length for both architectures as a function of N.

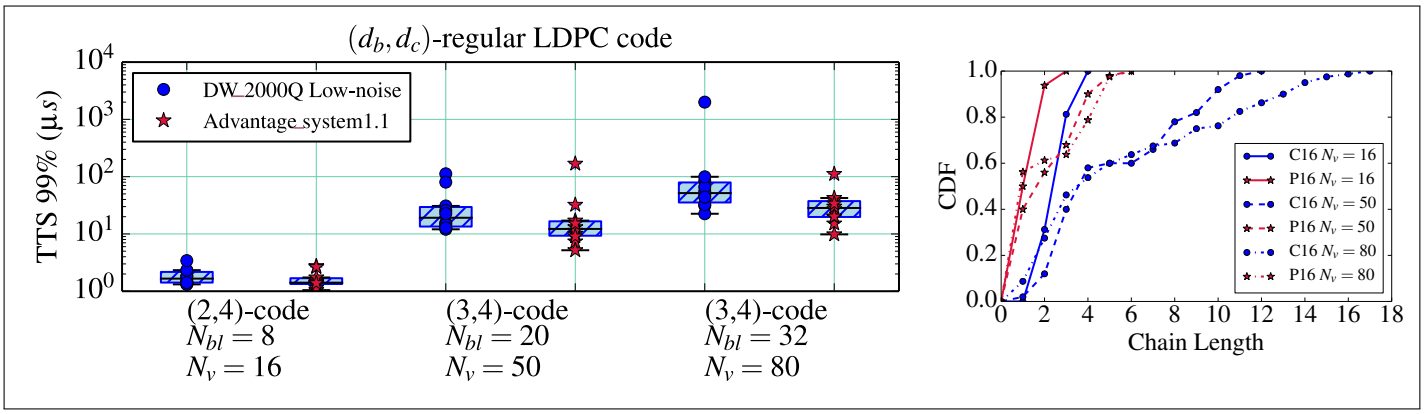

Fig. 5: TTS comparison of different quantum annealers for various LDPC decoding problems. In the figure, N_{bl} is the LDPC code block length, and N_v is the number of problem variables in QUBO formulation of the respective code. The distribution in figure *right* is across the chain lengths of the variables present in figure *left* in their respective embeddings.