
Quantum Computing in the RAN with Qu4Fec: Closing Gaps Towards Quantum-based FEC Processors

Nikolaos Apostolakis
IMDEA Networks Institute
Universidad Carlos III de Madrid

Marta Sierra-Obea
Marco Gramaglia
Universidad Carlos III de Madrid

Jose A. Ayala-Romero
Andres Garcia-Saavedra
NEC Laboratories Europe

Marco Fiore
IMDEA Networks Institute

Albert Banchs
IMDEA Networks Institute
Universidad Carlos III de Madrid

Xavier Costa-Perez
i2CAT, NEC Laboratories Europe and ICREA

Abstract

In mobile communication systems, the increasing densification of radio access networks is creating unprecedented computational stress for baseband processing, threatening the industry's sustainability, and new computing paradigms are urgently needed to improve the efficiency of wireless processors. Quantum computing promises to revolutionize many computing-intensive tasks across diverse fields and therefore may be the key to realizing ultra-dense next-generation mobile systems that remain economically and environmentally viable. This paper investigates the potential of Quantum computing to accelerate Forward Error Correction (FEC), the most compute-heavy component of wireless processors. We first propose Qu4Fec, a novel solution for decoding Low-Density Parity Check (LDPC) codes on Quantum Processing Units (QPUs), which we show to outperform state-of-the-art approaches, by reducing the Block Error Rate (BLER) by nearly an order of magnitude in simulation. We then implement Qu4Fec on a real-world QPU platform to study its practical viability and performance. Our experiments reveal that current cutting-edge QPU architectures curb the capabilities of FEC and expose the underlying factors, including long qubit chains, scaling, and quantization. Based on these insights, we suggest original blueprints for future QPUs that can better support Quantum-based wireless processors. Overall, this paper provides a reliable reality check for the feasibility of wireless processing on Quantum annealers: as QPUs start to be considered part of a possible 6G landscape, our work may open new research paths towards the design of FEC methods for Quantum-powered wireless processors.

1 Introduction

Quantum computing is rapidly shifting from a laboratory-only technology to the market Google (2024), and experiments in production-ready environments are starting to showcase its potential as a revolutionary technology across many domains. This includes sectors such as logistics and supply chain optimization Phillipson (2024), finance Kerenidis et al. (2019), materials and chemistry Kandala et al. (2017), key distribution Lucamarini et al. (2018) and communication networks Wehner et al.

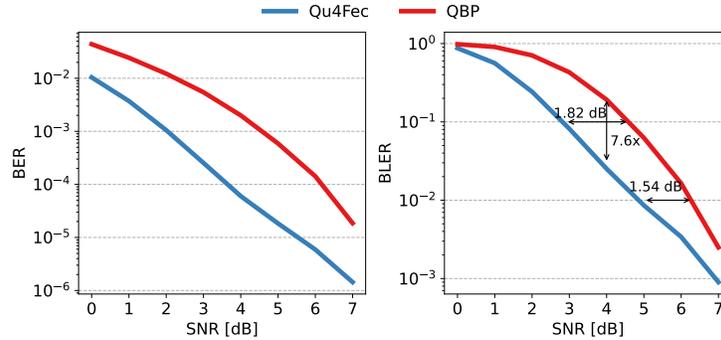


Figure 1: BER/BLER curves of Qu4Fec and QBP Kasi and Jamieson (2020) for a (2,3,420) LDPC code.

(2018). Among the many applications above, Quantum processors may also become a key enabler in the context of next-generation mobile wireless networks. With 6G systems expected to establish a new era of ultra-fast data throughput and sub-millisecond round-trip latency, the integration of advanced and intelligent processing units, such as Quantum processors, becomes critical. These processors can address the challenges posed by the vast number of interconnected devices and the need for rapid and efficient data processing in dense Radio Access Network (RAN) environments. We followingly elaborate on the concept of Quantum computing for RAN.

Sustainable RAN densification. Radio Access Network (RAN) densification Bhushan et al. (2014) is regarded as a fundamental enabler, especially if combined with high-band access technologies Liu et al. (2017) that can provide the required wireless capacity when the site coverage is capillary enough. Despite its promise, RAN densification poses many challenges, largely because of its increased power needs Schiavo et al. (2024), which can be though significantly mitigated by leveraging Quantum technologies Kasi et al. (2023).

Yet, densification alone may fall short of having a real practical impact due to its limited sustainability. Deploying fully-fledged, distributed base stations with high density entails high economic costs associated with the installation (including fiber connectivity to the backhaul) and operation of the pervasive sites. Also, environmental costs risk exploding, making ultra-dense deployments socially unacceptable. Major industrial fora are therefore promoting wireless processing pooling approaches that can substantially mitigate such costs O-RAN Work Group 1 (Use Cases and Overall Architecture) (2024). Indeed, a strategy of installing many Remote Units (RUs), *i.e.*, antennas, while centralizing all wireless baseband processing into shared compute resources yields a potential to cut down energy costs for the operator by orders of magnitude Schiavo et al. (2024). Prototypes AT&T (2022) and initial deployments in operational infrastructures NTT Docomo (2021) are also spearheading the adoption of paradigms for RAN processing pooling in production and at scale.

Gains are mostly due to pooling the physical signal processing blocks at the lower levels of the wireless transmission pipeline I et al. (2020), which are the most critical and compute-intensive functions and directly impact network performance. Forward error correction (FEC) schemes like Low-Density Parity Check (LDPC) codes that are part of the 3GPP 5G New Radio (NR) standards have high data throughput and error correction close to the Shannon limit, yet require parallel processing capabilities and specialized hardware accelerators based on ASICs and FPGAs that dramatically rise power consumption costs if they are dedicated to individual RUs Mavenir (2023). Sharing such accelerators across sets of RUs allows for multiplexing decoding demands, maximizing the utilization of the hardware, and taking advantage of the heterogeneity of available CPUs, ASICs, and FPGAs to seek energy optimization.

Quantum for RAN acceleration. In the scenario above, the more powerful the cloud of accelerators, the larger the number of RUs it can serve, and the higher the gain that baseband resource pooling can achieve. Motivated by this consideration, the quest is open for more capable and power-efficient accelerators, and an emergent technology with unique features, such as Quantum computing, is a prime candidate for investigation. Indeed, Quantum’s very high performance in solving NP-hard problems, such as those commonly encountered in wireless protocols, makes a clear case for Quantum-

based solutions for RAN baseband processing. Quantum Computing Units (QPUs) with many qubits promise to meet the strict budgets and hard execution deadlines that characterize, *e.g.*, FEC decoding operations, even in massive Multiple Input Multiple Output (MIMO) configurations and without leading to increased power consumption or operational costs Kasi et al. (2023).

Seminal studies have already started exploring applications of Quantum computing to baseband processing. Proposals like IoT-ResQ Kim et al. (2022), X-ResQ Kim et al. (2024), and QuAMax Kim et al. (2019) tackle the problem of MIMO detection using Quantum annealers, whereas HyPD Kasi et al. (2024) considers Quantum for polar code decoding. As far as the decoding of actual LDPC codes is concerned, QBP Kasi and Jamieson (2020) is, to date, the first and only solution in the literature. While QBP matches the LDPC code design to the existing Quantum architecture to favor the implementability of the solution, the derived code significantly underperforms existing codes from the literature, such as Gallager Gallager (1962), yielding an overall lower wireless performance. Additionally, QBP’s problem formulation, which also includes prior hyperparameter tuning depending on Signal-to-Noise Ratio (SNR), lacks formal guarantees that the minimum energy solution is the maximum likelihood solution of the fundamental decoding problem.

Further variants of QBP have focused on complementary aspects. In Das Sarma et al. (2023), the authors post-process the Quantum computer results by discarding invalid codewords and identifying the one closest to the received vector. The study in Guo et al. (2024), evaluates different post-processing techniques, including the minimum energy solution or the solution with the highest frequency. In contrast, the authors in Majumder et al. (2022), consider the effects of Rayleigh channel, unlike the Additive White Gaussian Noise (AWGN) channel, which was used in previous works. However, all works in Das Sarma et al. (2023); Guo et al. (2024); Majumder et al. (2022) do not address the fundamental limitations in QBP and share the same weaknesses highlighted above, that is, they employ an underperforming problem formulation for FEC decoding and utilize the same code design algorithm.

Our contributions. In this paper, we present Qu4Fec, an improved LDPC decoder powered by Quantum computing, which outperforms the state-of-the-art QBP by almost one order of magnitude in terms of Bit Error Rate (BER) and Block Error Rate (BLER). As a representative example, Fig. 1 shows how Qu4Fec can achieve 10%, which is set by 3GPP as the channel reliability threshold for enhanced mobile broadband services 3GPP (2023c), and 1% BLER with 1.82 db and 1.54 dB lower SNR respectively and can attain a $7.6\times$ BLER improvement at 4 dB, for a (2,3,420) LDPC code. The design and evaluation of Qu4Fec set forth several original contributions, as follows.

- We demonstrate that tailoring the LDPC code parity check matrix to fit the layout of a specific Quantum architecture as proposed by QBP undermines the code’s error-correcting capabilities. Qu4Fec, instead, builds on top of commercial Quantum computers (like D-Wave’s), regardless of their qubit layout structure, and considers well-studied LDPC codes, such as Gallager.
- We present a novel formulation of the LDPC decoding problem as a Quadratic Unconstrained Binary Optimization (QUBO) task. We re-engineer the constituent terms and directly derive them from the fundamental maximum-likelihood decoding formulation to achieve better accuracy and efficiency, removing the need for prior hyperparameter optimization.
- We perform experiments with a real-world cutting-edge Quantum Processing Unit (QPU) and identify clear limitations of the current and upcoming Quantum architectures in supporting LDPC decoding operations.

Ultimately, our work advances the body of knowledge about Quantum-powered FEC and points in several directions for the future development of practical Quantum RAN accelerators.

A caveat on LDPC and Quantum. Before proceeding further, we define the scope of our work, positioning it with respect to two completely different settings where coding and Quantum come together in the current scientific literature.

On the one hand, retrieving Quantum information has been challenging due to its noisy nature compared to classical bits. The noisiness of the Quantum platforms is ascribed to various factors: *i*) qubits are still not perfectly isolated from the environment and minimal interaction with the platform causes *decoherence*, degrading the Quantum state; *ii*) external disturbances can invert a qubit from $|0\rangle$ to $|1\rangle$ and vice versa; *iii*) additionally, in systems where qubits are entangled, an error in one qubit will propagate to others due to their strong correlation, exacerbating that effect Wilen et al. (2021).

In this context, significant efforts have been put into developing error correction codes that increase reliability when interacting with qubits Breuckmann and Eberhardt (2021). These investigations can be intended as exploring FEC *for* Quantum and also use LDPC as an error-correction approach. Quantum LDPC codes are essential for building fault-tolerant Quantum systems, needed to scale up Quantum computers since they provide a high rate of error detection and correction. Our study does *not* belong to this class of works.

On the other hand, Quantum technologies have been recently proposed as a processing tool to solve FEC problems Kasi and Jamieson (2020) already present in the telecommunication area. That is, the idea is to leverage Quantum systems as an alternative to traditional computing solutions to tackle LDPC decoding, which is ubiquitous in the latest WiFi 6E (2022) and 5G NR standards 3GPP (2023b). Thus, these works solve LDPC *with* Quantum. Our study and all the contributions listed above fall in this category.

2 Background

In this section, we provide a brief background on LDPC codes in Sec. 2.1, where we describe the basic principles of code design (Sec. 2.1.1), and the encoding process (Sec. 2.1.2), and the decoding process (Sec. 2.1.3). In Sec. 2.2, we go deeper into the internals of Quantum annealing and its potential energy and cost-saving benefits, when deployed in the RAN (Sec. 2.3), followed by defining LDPC decoding as a binary optimization problem in Sec. 2.4.

2.1 LDPC Codes

An LDPC Gallager (1962) code is a binary linear FEC code that is characterized by a sparse generator matrix $G \in \{0, 1\}^{k \times n}$ and parity-check matrix $H \in \{0, 1\}^{m \times n}$, where k , n is the number of message and codeword bits respectively, and m is the row count of H . A codeword \hat{c} is valid if and only if it satisfies $H \cdot \hat{c}^T = 0$ in the binary domain, or equivalently, $h_i \cdot \hat{c}^T = 0$, also known as check constraints, where h_i with $i=1, \dots, m$ is the row vector i of H . A (d_v, d_c) -regular LDPC code involves exactly d_c codeword bits in each constraint, while each codeword bit is present in d_v constraints. We denote by (d_v, d_c, n) the (d_v, d_c) -regular LDPC code of length n . An LDPC code can be visually represented by the Tanner graph Tanner (1981), a bipartite graph consisting of two sets of nodes: check nodes and variable nodes. Check nodes are the rows of the parity-check matrix, *i.e.*, match the constraints, while the variable nodes represent columns, *i.e.*, the codeword bits. In the top part of Fig. 2, we illustrate the Tanner graph extraction from a parity check matrix, where $H_{ij} = 1$ denotes the existence of a link between check node i and variable node j .

2.1.1 Code Design

The error-correcting capability of a code is strongly correlated with the cycle length distribution of the Tanner graph, as the bit corrections rely on the independent message passing between the graph nodes Zhang and Moura (2003); Lu and Moura (2006). Shorter cycles break this independence earlier in the decoding process, leading to decreased performance. Since an LDPC code always contains cycles in practical scenarios with finite code size n , the challenge is designing the parity check matrix so that the corresponding Tanner graph has cycle lengths as large as possible. Several efficient regular LDPC code design methods are available, such as Gallager Gallager (1962) and PEG Hu et al. (2005). We next discuss the encoding and decoding process.

2.1.2 Encoding

The code design eventually yields a parity check matrix H , with properties that assure the code's error correction capabilities. To convert the matrix H into a generator matrix G , we use the following steps. This procedure assumes that H is a full-rank matrix, and if not, Gaussian elimination method needs to first be performed.

1. **Convert H into a systematic form:** $H = [P|I_m]$, where P is an $m \times (n - m)$ matrix and I_m is the $m \times m$ identity matrix.
2. **Construct the generator Matix:** $G = [I_k|P^T]$, where P^T is the transpose of the matrix P .

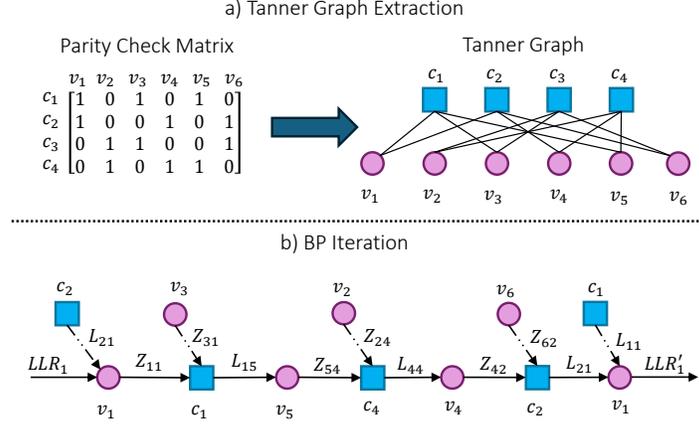


Figure 2: *a)* Tanner graph extraction from a parity check matrix. *b)* A BP update for LLR_1 . The cycle length is 6 since the update traverses 6 nodes before returning to v_1 ($v_1 \rightarrow c_1 \rightarrow v_5 \rightarrow c_4 \rightarrow v_4 \rightarrow c_2 \rightarrow v_1$).

The given an original message a of k bits and the generator matrix G , a codeword is given as $c = aG$. The codewords generated by G satisfy the parity check conditions and it must hold $HG^T = 0$, confirming G generates valid codewords for H .

2.1.3 Decoding

The LDPC maximum likelihood (ML) decoder outputs the codeword $x \in \mathcal{C}$ that maximizes the a-posteriori probability $P(x|y)$, where \mathcal{C} is the valid codeword set and y is the received channel value vector. Expressing this mathematically, we have

$$\hat{x} = \arg \max_{x \in \mathcal{C}} P(x|y). \quad (1)$$

Since the cardinality of \mathcal{C} grows exponentially with k , exhaustive search is not viable for practical applications. Instead, different efficient algorithms have been proposed to solve (1).

Belief propagation (BP) Chen and Fossorier (2002) is a heuristic algorithm that attempts to solve the decoding problem via iterative message passing between the check and variable nodes of the Tanner graph. The messages improve the extrinsic information that is received in one variable node from the rest of the variable nodes, which eventually strengthens the belief of a certain bit being 0 or 1. The input of the algorithm is the log-likelihood ratio (LLR) vector: $LLR_i = \log \frac{Pr(b_i=0|y)}{Pr(b_i=1|y)}$, where $Pr(b_i = k|y)$, $k \in \{0, 1\}$, $i = 0..n-1$ is the probability for bit $b_i = k$ given received vector y . The calculation of LLR_i depends on the modulation scheme used and the noise variance of the channel. The algorithm ends when $H \cdot \hat{c}^T = 0$ or a maximum number of iterations is reached.

We now describe the BP algorithm. Let the set of check nodes connected to bit node n as $\mathcal{M}(n)$ and the set of bit nodes connected to check node m as $\mathcal{N}(m)$. Also denote the message from check node m to variable node n as Z_{mn} and the variable n to check node m message as L_{nm} . The BP decoding algorithm comprises the following steps Chen and Fossorier (2002):

1. **Check to variable message update:** For each m :

$$L_{nm} = 2 \tanh^{-1} \left(\prod_{n' \in \mathcal{N}(m) \setminus n} \left(\tanh \left(\frac{Z_{mn'}}{2} \right) \right) \right)$$
2. **Variable to check message update:** For each n :

$$Z_{mn} = LLR_n + \sum_{m' \in \mathcal{M}(n) \setminus m} (L_{nm'})$$
3. **Belief update** For each n :

$$LLR_n := LLR_n + \sum_{m' \in \mathcal{M}(n)} (L_{nm'})$$
4. **Hard decision:** The candidate codeword $\hat{c} = [\hat{c}_0, \hat{c}_1, \dots, \hat{c}_{n-1}]$ is derived, setting $\hat{c}_i = 1$ if $LLR_i \leq 0$ else 0. The decoding exits if $H \cdot \hat{c}^T = 0$ or a maximum number of iterations is reached. If not, the algorithm proceeds to the next iteration in Step 1.

Different variations of BP have been studied in order to simplify the computationally expensive Step 1, *e.g.*, Min-sum BP Chen et al. (2005), or to speed up the algorithm’s convergence, *e.g.*, Layered BP Hocevar (2004).

In the bottom part of Fig. 2, we display the LLR_1 update across one BP iteration. The message originating from v_1 traverses 6 nodes before concluding to v_1 yielding a cycle length of 6. For BP to function optimally, the messages need to be independent. However, in the presence of unavoidable cycles in practical codes of finite length, this property is invalidated, and the performance of BP deteriorates, as we explain in Sec. 3.1.

2.2 Quantum Annealers

Quantum annealing Morita and Nishimori (2008) is a Quantum computing approach designed to solve optimization problems by finding the global minimum of a given function. It harnesses Quantum phenomena occurring in specialized superconducting loops at extremely low temperatures near absolute zero, including qubit superposition, tunneling, and entanglement. This process is carried out by Quantum annealing machines, known as Quantum Annealers (QA), which share conceptual similarities with Simulated Annealing (SA), a metaheuristic that can be coded and executed on a classical processor. The QA system is configured such that its lowest energy state coincides with the solution to the problem under consideration. Both techniques start with a high-energy state and seek lower-energy states by cooling the system. During the high-temperature annealing phase, the system explores various possible solutions to avoid getting stuck in local minima. As temperature decreases, the system moves to lower energy states, lowering the probability of escaping them and ultimately seeking the global optimum.

Quantum annealing, like simulated annealing, is suitable for solving objective functions formulated as Quadratic Unconstrained Binary Optimization (QUBO) problems Punnen (2022). A QUBO model aims to find the vector of binary variables that minimizes an objective function with polynomial factors up to quadratic terms. Formally

$$E(x) = \sum_i h_i x_i + \sum_{i < j} J_{ij} x_i x_j, \quad (2)$$

where $x = x_i$ for $i = 1, 2, \dots, n$ is the vector of binary decision variables, h_i is the linear term of x_i , and J_{ij} denotes the quadratic term between variables x_i and x_j .

The basic unit of Quantum information is the qubit, analogous to the bit in classical computing. Unlike bits, which can only represent a definitive state of 0 or 1, qubits can be in a superposition of states, representing multiple values simultaneously due to the principles of Quantum mechanics. This unique characteristic allows Quantum annealers to tackle complex optimization problems with an approach that goes beyond classical methods. Qubits must be connected to create entanglement. This is achieved through couplers, which are essentially superconducting loops. When QA is programmed to solve a QUBO, the *bias* of a qubit q_i is set to h_i , and the *strength* of the coupler between q_i and q_j is set to J_{ij} .

In the context of QA, qubits are used to encode potential solutions to optimization problems. In the annealing process, the qubits are initially placed in the superposition state, with an equal probability of being either in 0 or 1 state. While the annealing process evolves, the Quantum phenomena of tunneling and entanglement take place in this low-temperature environment, with the system converging slowly to a minimum of the QUBO model, which can be either a local or a global one. At the end of the anneal, each qubit has a classical state of 0 or 1 and is the solution to the QUBO problem.

2.3 Quantum RAN Power Analysis

Performing wireless tasks such as FEC decoding in a Quantum computer has the potential to substantially impact the overall energy consumption footprint of the mobile network when compared with the traditional CMOS-based computing systems. Google’s Sycamore Quantum computer reportedly has a power consumption of 15 kW Villalonga et al. (2020), primarily attributed to its refrigeration and cooling unit (10 kW) and classical electronics (5 kW). D-Wave’s Advantage system reports a maximum power consumption of 25 kW, with cooling accounting for 15 kW Inc. (2022).

Consequently, cooling is the primary source of the power consumption in Quantum systems Parker and Vermeer (2023), yet it is not expected to scale with the number of contained qubits Villalonga et al. (2020); Parker and Vermeer (2023). In contrast, CMOS 5G base stations consume between 35 and 250 kW Kasi et al. (2023), depending on factors such as MIMO degree, number of receive antennas and transmission bandwidth. These figures are significantly higher than those expected for Quantum systems, indicating the potential for substantial energy and OPEX reductions of up to 1,500% that can be attained by deploying Quantum processors for RAN.

2.4 QUBO Formulation for the Decoding Problem

In order to use Quantum computing to solve the fundamental problem of LDPC decoding in noisy communication systems, a challenge arises in that classical iterative solutions such as BP cannot be executed on QA platforms. As anticipated in Sec. 1, QBP Kasi and Jamieson (2020) is, to date, the only solution in the literature to propose a QUBO formulation for LDPC codes that is suitable for subsequent solution by the QA. In the formulation proposed by QBP, the variables used in the QUBO problem are split into two types: *i*) the code variables $[q_0, q_1, \dots, q_{n-1}]$ that denote the output decision of the decoder, and *ii*) the ancillary variables that describe the modulo-2 constraints enforced by the LDPC as an optimization objective.

The QUBO formulation for the LDPC decoding comprises two terms: *i*) the LDPC constraint term that weights the possible solution, penalizing not valid (*i.e.*, with no valid check constraints) codewords, denoted by Q_{LDPC} , and *ii*) a correlation function that associates the problem solution to the received channel values, denoted by Q_C . This latter term steers the QA solver toward solutions that are more related to the initial set of received values after the demodulation.

The final QUBO formulation implemented by QBP is a weighted summation of the LDPC constraint function and the correlation function

$$Q = a \cdot Q_{LDPC} + Q_C \quad (3)$$

where a is a positive weight factor, which steers the balance between the two terms. QA can find the solution that minimizes both terms, yielding to a correctly decoded codeword.

Considering an (m, n) parity check matrix, we have

$$Q_{LDPC} = \sum_{j=1}^m Q_{LDPC_j}$$

with $Q_{LDPC_j} = \sum_{i=0}^{n-1} (H_{ji} \cdot q_i - 2 \cdot L_j)^2$ being the minimization function objective related to the j -th constraint and L_j is a function of the ancillary qubits, which depends on the degree (number of 1's) of that constraint. The utility of L_j is to enforce that the modulo-2 (*i.e.*, XOR in binary) summation of the j -th constraint's constituent variable nodes sums to 0 and is expressed by

$$L_j = \sum_{s=1}^t (2^{s-1} \cdot q_{e_j,s}^{j,s})$$

where t is a function of the check node degree and $q_{e_j,s}$ is the s -th ancillary qubit of the j -th constraint.

The second term of the QUBO is the correlation function. This term incorporates the channel output values and aims to associate them with the problem solution. Let $y = [y_0, y_1, \dots, y_{n-1}]$ be the received channel vector. The authors in Kasi and Jamieson (2020), set Q_C to minimize the distance

$$Q_C = \sum_{i=0}^{n-1} (q_i - Pr(q_i = 1|y_i))^2$$

Finally, in Kasi and Jamieson (2020) a is set based on the transmission SNR, performing a hyperparameter search.

3 Decoding in the Quantum Environment

We now introduce our original design for an LDPC code suitable for a QA, named Qu4Fec. Developing such a code requires finding solutions for two aspects: *i*) a code design and *ii*) a QUBO

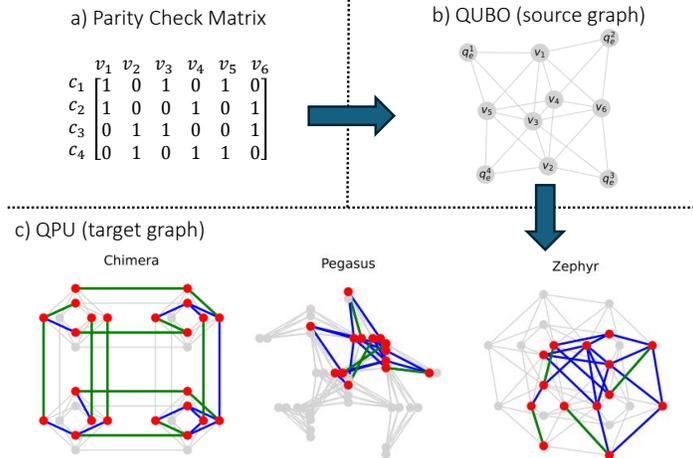


Figure 3: Steps from designing an LDPC code for Quantum decoding. *a)* Design the parity check matrix. *b)* Extract the QUBO formulation (source graph). *c)* Embed it to the QPU architecture (target graph). In red, we show the occupied qubits; in blue, the edges of the source QUBO; and in green, the edges of the formatted chains. Qu4Fec uses the Gallager method to generate regular codes (step *a*) and D-Wave’s MM to embed to QPU (*b* \rightarrow *c*).

formulation of the LDPC decoding problem. We discuss these steps using as a reference QBP Kasi and Jamieson (2020), which is the state-of-the-art Quantum-friendly LDPC decoder. In the following Sec. 3.1, we discuss the code design part of the problem: we point out the detrimental effect that custom-designed codes that are primarily aimed at fitting the Quantum hardware, such as those utilized by QBP, have on the error correcting capabilities and, in contrast, we propose to use well-established code design methods. In Sec. 3.2, we formulate an alternative QUBO expression that formally verifies the fundamental LDPC decoding problem in Eq. 1, which ultimately yields stronger guarantees and higher performance than that introduced by QBP in Sec. 2.4.

3.1 Code Design Strategies

FEC codes are carefully conceived to improve error-correcting capabilities in digital communications, and the error rate performance is the main metric that needs to be optimized in the development of an FEC code. Yet, in a QA, FEC code design needs to also consider the specificity of the execution platform (*i.e.*, the QPU) that will be used. In Fig. 3, we illustrate the natural order for this process: from (a) designing the LDPC code to (b) the extraction of the QUBO formulation (which produces a so-called source graph) till (c) the embedding on the QPU target architecture (such as commercial QPUs like Chimera, Pegasus or Zephyr manufactured by D-Wave¹) used for decoding.

The flow from (a) to (c) is not straightforward since the target graphs of cutting-edge QA platforms impose severe constraints that reflect on the QUBO and parity check matrix. The problem is so binding that QBP devises an LDPC code specifically tailored to D-Wave’s Chimera QA framework by reversing the logical process: the design starts from the constraints in (c) and goes backward to a code design in (a). This turnaround has, however, the effect that the choices of QUBO formulations and LDPC design are critically curbed.

We prove this point with a practical example. Using the QBP embedding method Kasi and Jamieson (2020), we reconstructed the parity check matrix of a (2,3,420) code. For comparison, we also constructed a code with the same properties using the Gallager Gallager (1962) method. For each of the codes, we generated codewords, passed them through an AWGN channel, and decoded them using BP. We measured the performance in terms of BER/BLER, which is a fair comparison since the two codes add the same number of redundancy bits to the wireless channel transmission.

The comparison of both BER and BLER, depicted in Fig. 4, shows a noticeable gap between the two strategies, with the Gallager code design outperforming that of QBP by almost an order of magnitude.

¹<https://www.dwavesys.com/>

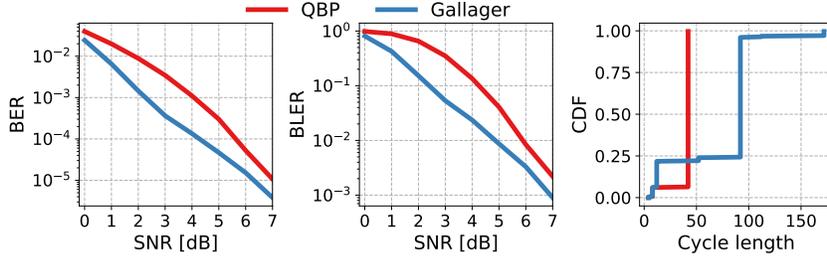


Figure 4: Comparison of QBP Kasi and Jamieson (2020) and *Gallager* design method of (2,3,420) LDPC code. a) BER, b) BLER, c) Cycle length distribution CDF.

Although the manually designed code in QBP benefits the target QA by occupying qubits as efficiently as possible, it basically disrupts the cycle length property discussed in Sec. 2.1.1 and results in inoperable performance.

The effect is revealed in the right plot of Fig. 4, where we measured the minimum cycle length of each variable node in the Tanner graph and plotted the CDF for the two cases considered, *i.e.*, QBP and Gallager. The median cycle length of QBP is half that of Gallager. As shorter cycles introduce higher correlations between the bits in the sequence to be decoded and limit the inherent capabilities of the code itself, the result clearly highlights how the manual design of QBP inherently and substantially curbs the final decoding performance.

In light of these observations, we propose a novel design for FEC codes in Quantum settings that are primarily aiming at preserving decoding capabilities. To this end, we abide by a traditional, sensible pipeline that starts from efficient codes created with the Gallager method.

3.2 QUBO Formulation in Qu4Fec

As analyzed in Sec. 2.4, the QUBO formulation proposed by QBP Kasi and Jamieson (2020) exhibits one main drawback: the factor a must be computed and optimized based on prior hyperparameter tuning. This makes QBP SNR-dependent, relying on accurate SNR estimations, which are challenging to derive in operational settings Albatineh et al. (2020). Even then, however, it is not formally guaranteed that the minimum energy solution of the Quantum annealer coincides with the maximum likelihood solution of the fundamental decoding problem. We thus propose Qu4Fec, a formulation that provides formal proof without the need for prior hyperparameter optimization as it works by only taking the received channel values as input.

The novel QUBO formulation of Qu4Fec derives directly from the ML objective detailed in (1) and avoids the hyperparameter tuning step. We consider BPSK modulation $y_i = 1 - 2 \cdot x_i$ (where x_i is the bit to transmit, y_i is the transmitted symbol) over an AWGN memoryless channel. By expanding (1) we have

$$\begin{aligned}
 \hat{x} &= \arg \max_{x \in \mathcal{C}} \prod_{k=0}^{n-1} P(y_k | x_k). \\
 \hat{x} &= \arg \max_{x \in \mathcal{C}} \sum_{k=0}^{n-1} \ln f_{Y|X}(y_k | x_k), \\
 \hat{x} &= \arg \max_{x \in \mathcal{C}} \sum_{k=0}^{n-1} y_k (1 - 2x_k). \tag{4}
 \end{aligned}$$

Based on this, we developed a novel QUBO formulation for the LDPC decoding problem P_1 . We use the same structure as (3) and re-define

$$P_1 = a' \cdot Q_{LDPC} + P_C \tag{5}$$

by re-designing the correlation function as the minimization function of the negative ML function of (4) as

$$P_C = \sum_{k=0}^{n-1} -y_k (1 - 2 \cdot q_k)$$

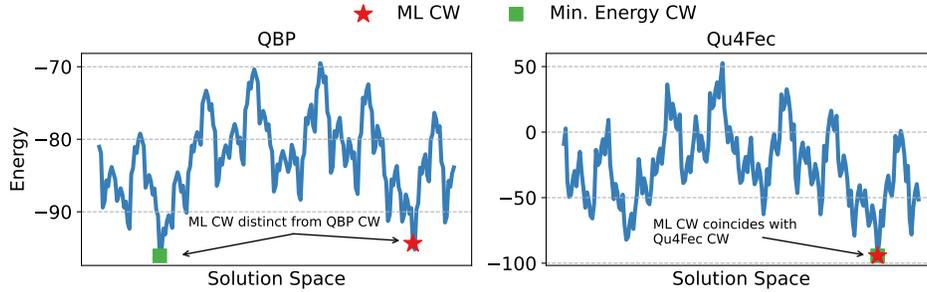


Figure 5: Sample comparison of QBP Kasi and Jamieson (2020) and Qu4Fec energy levels. QBP formulation does not guarantee that the QUBO’s minimum energy codeword (CW) coincides with the Maximum Likelihood (ML) codeword. Qu4Fec formulation always reassures that its minimum energy codeword is the ML codeword.

Similar derivations can be made for higher-order modulations. For QPSK modulation following the 5G NR mapping 3GPP (2023b), input bits x_{2k}, x_{2k+1} will be mapped to symbol $s_k = (1 - 2x_{2k}) + j(1 - 2x_{2k+1})$ where $k = 0.. \frac{n}{2} - 1$. The Eq. 4 will now become for QPSK:

$$\hat{x} = \arg \max_{x \in \mathcal{C}} \sum_{k=0}^{\frac{n}{2}-1} \text{Re}(y_k)(1 - 2x_{2k}) + \text{Im}(y_k)(1 - x_{2k+1})$$

where $\text{Re}(\cdot), \text{Im}(\cdot)$ denote the real and imaginary part of the complex y_k . For the rest of the analysis and without loss of generality, we consider BPSK modulation.

Our design objective is to determine a sufficiently large a' that prioritizes constraint fulfillment (Q_{LDPC}) over correlation function optimization (P_C). A heavy weight on P_C and hence basing the decoding process largely on the channel values may be counterproductive in many conditions.

Thus, we follow a similar approach to Tawada et al. (2020) and consider the correlation function directed by the ML decoder instead of the Euclidean distance. Consider a valid codeword solution vector \hat{q} and an invalid solution \tilde{q} . It holds that $P_C^{max} = \sum_{k=0}^{n-1} |y_k|$, $P_C^{min} = \sum_{k=0}^{n-1} -|y_k|$, and that for any solution vector q : $P_C^{min} \leq P_C(q) \leq P_C^{max}$. Thus, we have

$$\begin{aligned} a' \cdot Q_{LDPC}(\hat{q}) + P_C^{min} &\leq P_1(\hat{q}) \leq a' \cdot Q_{LDPC}(\hat{q}) + P_C^{max} \\ a' \cdot Q_{LDPC}(\tilde{q}) + P_C^{min} &\leq P_1(\tilde{q}) \leq a' \cdot Q_{LDPC}(\tilde{q}) + P_C^{max}. \end{aligned}$$

Since \hat{q} is a valid codeword: $Q_{LDPC}(\hat{q}) = 0$ and \tilde{q} is invalid: $Q_{LDPC}(\tilde{q}) \geq 1$. We also want the maximum energy of any valid codeword to be lower than the minimum of any invalid one. Thus, we obtain

$$P_C^{max} \leq a' + P_C^{min} \Rightarrow a'_{min} = P_C^{max} - P_C^{min} = \sum_{k=0}^{n-1} 2 \cdot |y_k| \quad (6)$$

These derivations yield that the ML solution \hat{q}_{ML} , *i.e.*, the valid codeword that minimizes P_C , is also the one that minimizes P_1 since

$$P_1(\hat{q}_{ML}) = P_C(\hat{q}_{ML}) \leq P_C(\hat{q}) = P_1(\hat{q}).$$

Unlike the Q formulation used in QBP Kasi and Jamieson (2020), our proposed P_1 formulation ensures that the solution that yields minimum energy is the ML codeword and removes the need for an offline optimization of a . To showcase that with an example, we consider a (2, 3, 420) LDPC code and a channel of SNR 2 dB. Given the transmitted maximum-likelihood (ML) codeword (CW), we generate the 256 solution space by freezing the ML CW’s first 412 bits and considering any binary combination of the last 8 bits. For each of these generated possible solutions, we evaluate the energy level according to the Q (QBP) and P_1 (Qu4Fec) QUBO formulations and we compute the minimum energy solution. In Fig. 5, we illustrate the energy level for the two formulations across the solution space. We notice that QBP’s minimum energy CW may be distinct from the ML CW, which will lead to a higher BLER, unlike Qu4Fec, whose formulation guarantees this property.

3.3 Improving Stability in Qu4Fec

We observe that any codeword error increases the output QUBO energy by at least a' , which also depends on n . This creates a big energy gap between invalid and valid codewords, leading to instability in the annealing process, where the probability of avoiding local optima depends on the energy differential between the previous and the current state.

Consequently, the annealing process interprets any decrease in the energy as progress towards the global solution, with a very low probability of escaping these suboptimal states due to the energy decrease caused by a' . This issue inspired the derivation of a new formulation P_2 , which restricts the energy differential between invalid, valid, and ML codewords.

The basic observation we made is that each row in the parity check matrix functions as a parity check code itself (with just one row). We apply (5) for the j -th constraint

$$P_1^j = a'_j \cdot Q_{LDPC_j} + P_C^j,$$

with $P_C^j = \sum_{k=0}^{n-1} -H_{jk} \cdot y_k \cdot (1 - 2 \cdot q_k)$ being the correlation function considering only the variable nodes of that constraint and a'_j weighting individually each LDPC constraint Q_{LDPC_j} . Following the same derivations as in (6), we derive that

$$a_j^{min} = \sum_{k=0}^{n-1} 2 \cdot H_{jk} \cdot |y_k|.$$

Since we want collectively to optimize for all constraints, the Qu4Fec QUBO formulation is

$$P_2 = \sum_{j=1}^m P_1^j. \quad (7)$$

P_2 achieves better numerical stability. For a single error in the achieved codeword at constraint j , the energy differential is a'_j , which considers the channel values only of that constraint. The previous formulation, P_1 , used a' , which considered the whole n values. From our experiments, P_2 led to better stability and performance closer to BP, as we demonstrate later.

4 Simulated and Experimental Evaluations

We study the practical performance of Qu4Fec, using QBP as a reference where appropriate, in two different settings, *i.e.*, via simulated annealing and with a real-world Quantum computer, as follows.

- First, in Sec. 4.1, we reproduce the Quantum process via Simulated Annealing (SA) Bertsimas and Tsitsiklis (1993), a stochastic optimization technique for approaching the global maximum of a QUBO. SA initializes a random solution to the problem, and in every step, picks a state close to the previous one. The system starts at a high-temperature state, where the probability of accepting a worse solution (a solution that increases the global energy) is high, facilitating the exploration of the energy landscape.

As the annealing process proceeds, the temperature cools down, this probability decreases, and the process slowly approaches the global minimum. Commercial QA platforms, such as the ones used in this paper, implement this algorithm using a physical system. Thus, we consider SA the ideal annealer.

This perspective disregards any physical limitations and implementation imperfections of the underlying analog computing platform, focusing solely on the interactions between the qubits and couplers. Since the annealing process is inherently stochastic, both SA and QA perform a certain number of anneals for each submitted QUBO. Each anneal generates a solution, and the objective function is then evaluated for each solution to determine the achieved *energy*.

- Then, in Sec. 4.2–4.5, we implement Qu4Fec into a real-world QPU. We use the D-Wave Advantage McGeoch and Farré (2022) 6.1 platform on top of a hardware QPU with the Pegasus layout portrayed in Fig. 3. We extensively test its performance with a range of

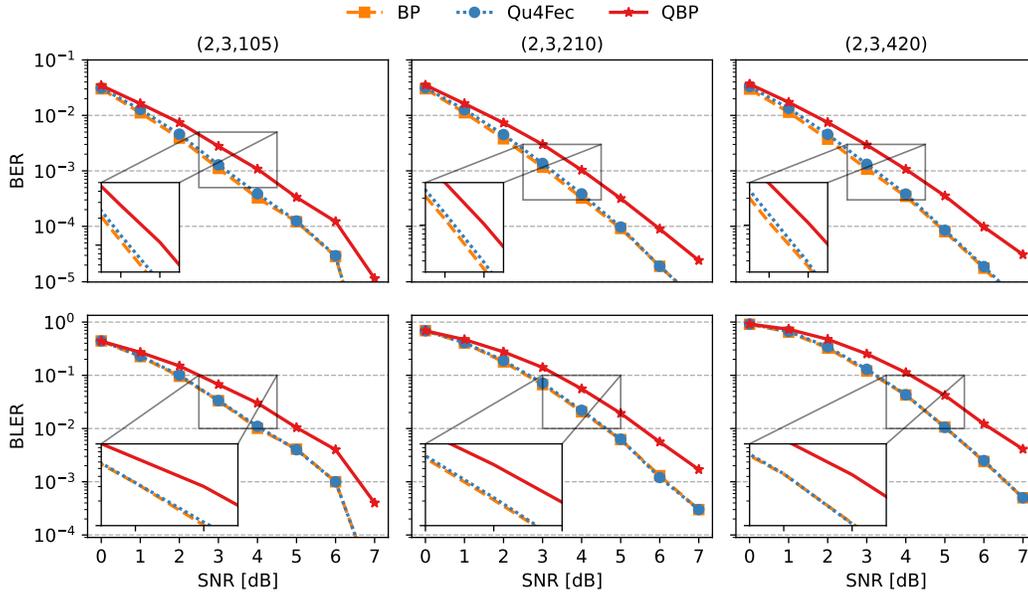


Figure 6: BER/BLER curves for BP, Qu4Fec, QBP and different code lengths.

LDPC workloads and Quantum annealing parameters (*e.g.*, annealing time, number of samples per anneal, chain strength Le et al. (2023), among others). The results show, in fact, a significant discrepancy compared to the promising performance presented in Sec. 4.1 under SA. In particular, BER and BLER surged to almost 100% for code lengths greater than 60 bits, regardless of the SNR level.

Motivated by these QPU results, we investigate the root causes of the decoding performance drop, exploring the effect of the embedding algorithm and that of scaling and quantization. The analysis lets us explain some crucial steps that need to be taken before Quantum annealing takes place, the restrictions they impose, and their effect on decoding quality.

4.1 Benchmarking Qu4Fec via Simulated Annealing

Based on P_2 and Q formulations we demonstrate the superiority of our solution Qu4Fec over QBP. Using the Gallager method, we generate different LDPC codes and compare the BER/BLER curves on SA. Additionally, we evaluate the BP performance (which we consider the ground truth for traditional computing decoders) to quantify its discrepancy from Qu4Fec. For BP, we limit it to a maximum of 10 iterations, as further iterations provide only negligible improvements in BLER. For SA, we set the number of samples per submitted QUBO to 100 and select the one with minimum energy, as an ideal annealing-based decoder would do.

For BP, Qu4Fec, and QBP, we compute BER and BLER directly from the input and output codewords². For each SNR level, we generate 10,000 message blocks, encode, modulate, and pass them through an AWGN channel. We generate 3 different $(2, 3, n)$ LDPC codes where $n \in \{105, 210, 420\}$. Authors in Kasi and Jamieson (2020) used 420 bits as the fixed codeword size, so we used a fraction as the baseline size in our benchmarks.

As illustrated in Fig. 6, the BLER discrepancy between BP and Qu4Fec is negligible across all code lengths. This demonstrates that our revised Qu4Fec QUBO formulation achieves comparable performance to BP³. However, as previously discussed in Sec. 2.3, this similar BLER performance can be achieved with lower energy consumption compared to BP, which operates on conventional

²Note that this contrasts with the approach in Kasi and Jamieson (2020), which uses an *a posteriori* model based on the samples' distribution gathered from several executions on the same instance to estimate these metrics and has a harder applicability to a real-world decoder.

³We also evaluated the performance of commercial solvers, such as CPLEX with varying parameter settings, in solving the QUBO. However, their performance was substantially inferior to that of SA. For instance, at 6

silicon. This makes Qu4Fec a promising alternative for carrier-grade baseband processing in the Quantum setting, enabling resource pooling as proposed in I et al. (2020).

We also observe that QBP performance diverges, especially in high SNR scenarios. This divergence arises from the fundamental approach of Qu4Fec QUBO formulation, where each invalid codeword yields higher energy than any valid one. QBP frequently fails in this regard, favoring invalid solutions that minimize the correlation function without giving the appropriate weight to the LDPC constraint satisfier component.

4.2 Embedding QUBO Problems in QPUs

When moving from a simulated annealing environment to a real-world Quantum processor, after the QUBO is formulated, it must be correctly programmed on the QPU before initiating the annealing process. To achieve this, each logical variable is mapped to a qubit, while the interactions between variables are mapped to couplers, the circuits that interconnect the qubits. The QUBO linear and quadratic terms set the qubit biases and coupler strengths, respectively. This process is called *embedding*.

In an ideal, fully connected QPU, each logical variable could be mapped to any qubit. QPUs are continually improving in terms of qubit connectivity; for instance, the qubit out-degree grew from 6 in Chimera to 15 in Pegasus and 20 in Zephyr Boothby et al. (2021). Yet, the resulting QPU graphs are still sparsely connected due to the inherently technical complexity of providing complex qubit fabrics. Consequently, having only one-to-one mapping between a variable in the QUBO formulation variables and a qubit is impossible, and *chains* must be created.

Chains are sets of qubits that represent a single logical variable, ensuring the proper representation of the interactions in the initial QUBO. That is, neighboring variables in the QUBO formulation must also be so in the QPUs, either directly or through a chain. When the QA process terminates, the state of the logical variable is determined by the majority vote of its chain’s qubits’ classical states. Intuitively, longer chains introduce larger sources of error in the annealing process, as more qubits and couplers (along with their inherent noise) must be used to solve the problem. In Fig. 3, we illustrate the embedding for the three different D-Wave target QPU architectures; in green, we denote the couplers used for formatting chains, and in blue, the couplers as imposed by the original QUBO quadratic interactions.

The process of formatting chains to map a source graph S (i.e., a QUBO graph in our case) to a target graph T (i.e., the QPU graph) can be formulated as a *minor extraction problem*. In graph theory, S is a minor of T if S can be obtained by deleting edges and vertices or by contracting edges in T . Minor extraction is, in general, an NP-hard problem Lobe and Lutz (2024). MinorMiner (MM) Cai et al. (2014); Choi (2008, 2011) is an iterative heuristic algorithm that solves the dual problem; determining which set of vertices (chain) in T corresponds to each vertex in S by taking into account the chain length of each source vertex. Once this mapping is obtained, the QUBO can be executed on the QPU, as the qubits and couplers can be programmed appropriately. In Fig. 3, we illustrate the embedding process with the green lines representing the chain links.

4.3 Effects of Non-ideal Embeddings

The embedding process is a crucial step of Quantum annealing on real platforms since it affects the total number of qubits participating in the process. It has been shown that a few dominant chains produced by MM hold significantly more qubits than the average chain in practical cases Ayanzadeh and Qureshi (2023). In the Noisy Intermediate-scale Quantum (NISQ) era, each qubit is inherently noisy, so when more than one qubits are coupled together into a chain, the problem is exacerbated.

These imperfections in Quantum annealers arise from their analog nature, requiring operation at temperatures near absolute zero to enable Quantum phenomena like tunneling and entanglement. Controlling complex phenomena in extreme conditions is challenging and can lead to deviations from the intended representation of a given problem. These deviations are collectively referred to as Integrated Control Errors (ICE).

dB, CPLEX resulted in a 10% higher BLER. This is attributed to CPLEX’s limitations in solving non-convex problems with a large number of variables such as our QUBO.

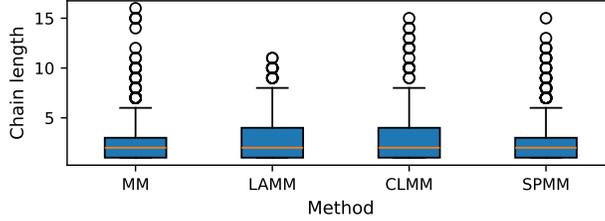


Figure 7: CDF of qubits' chain length using different embedding methods.

To better illustrate the issue, let us consider a problem with N binary variables, where h_i and $J_{i,j}$ represent the linear and quadratic terms respectively, with $i, j = 1, \dots, N$ and $i < j$. ICE can be modeled as errors δh_i and $\delta J_{i,j}$ that affect the precision of these coefficients, thereby altering the original problem definition. Consequently, these errors can modify the objective function the QPU is attempting to optimize.

To counter this behavior, the authors in Kasi and Jamieson (2020) resorted to a manual embedding solution supported by the relatively easy (hence non-ideal) structure of the LDPC code they designed for QBP. The LDPC parity check matrix perfectly embeds on the D-Wave 2000Q QA, shipped with the Chimera architecture. This method allows QBP to fit larger codewords with fewer qubits due to the limited chain length.

As discussed in Sec. 3.1, this comes with a high price in terms of maximum cycle lengths and overall decoding performance. However, manual embedding faces more problems. The first is the generality of the solution: each generation of QA has a short lifetime, with frequent changes in the underlying architecture that are not backward-compatible and render previous manual embeddings inapplicable and their associated LDPC code design obsolete. Indeed, newer QPU platforms are not necessarily superset of previous platforms: for example, D-Wave 2000Q is based on the Chimera C_{16} lattice, while D-Wave Advantage's Pegasus architecture is a super graph of Chimera C_{15} . We attempted to fit the $(2,3,420)$ code of Kasi and Jamieson (2020) onto the newest architecture using the QBP manual embedding, but the attempt was unsuccessful due to the bigger chain lengths that were created. Finally, embedding optimization for a specific code compromises generality, limiting its applicability to codes with different parameters.

Qu4Fec offloads the embedding task to the standardized and well-evolved MM, effectively decoupling the code design phase from the graph embedding. This yields many advantages, including versatility in accommodating diverse (and efficient, *e.g.*, Gallager) parity check matrices and invariance to the target graph's structural properties.

In our Qu4Fec implementation, we used D-Wave's open-source MM D-Wave Systems (2024). We also investigated variants of this algorithm, such as LAMM Pinilla and Wilton (2019), SPMM, and CLMM Zbinden et al. (2020), which aim to optimize the mapping process by exploiting the structural characteristics of the source graph. Fig. 7 shows that the different embedding methodologies do not consistently improve the chain length distribution for a reference $(2,3,420)$ code mapped onto the Pegasus layout. The reason is that the QUBO source graph of an LDPC decoding problem does not present a specific layout that those variants could leverage to render the embedding more efficient. In the rest of the paper, we thus consider MM to be the embedding method.

4.4 Scaling and Quantization

The qubits and couplers in a QPU operate within specific ranges. For instance, the qubit bias range for the latest Advantage QPU is between -4 and 4 , and the coupler strength ranges from -1 to 1 . Since the coefficients of an objective function can theoretically span from negative to positive infinity, they must be scaled down to fit within these hardware constraints. Before programming the QPU, the coefficients are adjusted with a scaling factor to ensure they do not exceed the platform's acceptable range, leading to higher inaccuracies when coefficients have a larger span.

Following the programming of biases and couplers, a quantization process with finite resolution is applied. This process inherently introduces error, as closely valued terms might be quantized to the same value, potentially leading to significant distortions in the solution quality. In some instances,

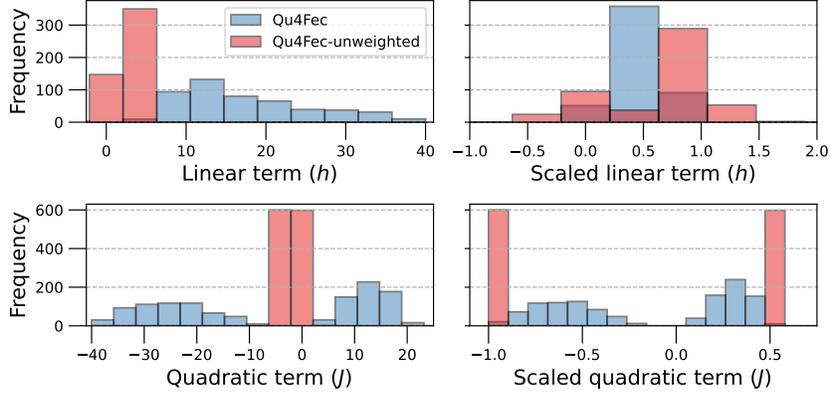


Figure 8: Original, scaled linear, and quadratic term histogram for the Qu4Fec and Qu4Fec-unweighted QUBO formulations.

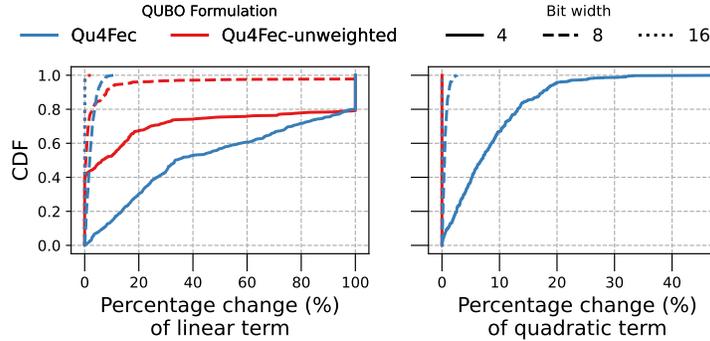


Figure 9: Percentage change of linear and quadratic terms due to scaling and quantization effect for Qu4Fec and Qu4Fec-unweighted formulations. Considered 3 different bit widths of 4, 8, and 16 bit.

these distortions can dramatically impact the energy landscape of the objective function, with even a slight change in value or sign altering the optimal solution.

While quantization errors occur independently of scaling, experiments indicate that scaling before quantization amplifies these errors. This suggests that when terms are first scaled, the subsequent quantization can lead to greater inconsistencies, further complicating the problem-solving process on a Quantum annealer.

To demonstrate this effect, we introduce an unweighted version of the Qu4Fec QUBO formulation (Qu4Fec-unweighted) where $a'_j = 1$. Setting $a'_j = 1$ effectively reduces the heterogeneity and range of the coefficients, simplifying the problems linked to scaling and quantization since they will now face similar modifications. However, this worsens the decoding performance as it does not optimize as per P_2 .

In Fig. 8, we show the histograms of the linear and quadratic coefficients for both QUBO formulations when injected into the Quantum platform within the above-mentioned range $-4 \dots 4$ and $-1 \dots 1$ for linear and quadratic terms respectively. The variability of the unscaled linear and quadratic terms of the Qu4Fec formulation is disrupted when a scaling factor is applied, hence changing the factors of the problem.

While we could compute the effect of the scaling process, we do not have access to the actual QPU hardware implementation to understand how the real-valued variables are quantized into the couplers. Thus, we consider three different bit resolutions (4, 8, and 16 bits) and show in Fig. 9 the cumulative distribution function (CDF) of the percentage change of both the linear and quadratic terms after quantization has taken place.

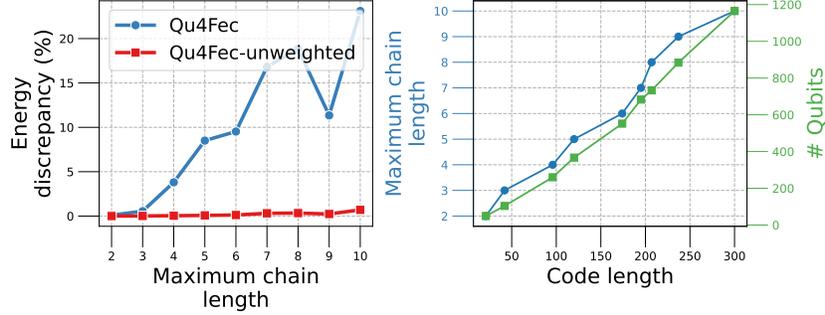


Figure 10: Left plot: Percentage energy discrepancy between QA and SA for Qu4Fec and Qu4Fec-unweighted QUBO formulations. Right plot: Maximum chain length and number of occupied qubits achieved by MM for various code lengths.

The quadratic terms (which exist only in the LDPC satisfier term) of the unweighted version incurs a median error of 8% for 4-bit resolution and near negligible error for higher resolutions, as these terms are not multiplied by the real numbers a'_j . However, we notice a much larger discrepancy in the linear terms. For the lowest resolution, some terms may even suffer a 100% error, and only at very high resolutions is the accuracy high enough to avoid errors, indicating the high sensitivity of the linear terms to the scaling and quantization effect.

4.5 Effects on the QA process

The errors introduced by embedding, scaling, and quantization affect the overall quality of the annealing process when executed on a physical platform. To quantify this effect, we compare Qu4Fec with its unweighted version on SA and QA and compute the discrepancy between the two processes in the left plot of Fig. 10. We compute the relative energy discrepancy as the energy difference between the solutions found by QA and SA divided by the ones found by SA. We can observe that an overall easier problem such as Qu4Fec-unweighted always has a very low discrepancy between the lowest energy solutions, indicating how the inaccuracies introduced by quantization on the QA platform have a minor impact. As we consider SA an *ideal* annealer, higher discrepancies such as the ones shown for larger codeword sizes are likely due to the combined effect of quantization, scaling, and embedding.

Especially the latter has a noticeable effect, which we capture in the right plot of Fig. 10, where we measure the maximum chain lengths and number of occupied qubits experienced by the QPU for various codelengths. When the chain lengths are limited, even the more complex Qu4Fec still attains comparable results to SA when executed with QA.

Throughout our analysis, we found that current QA platforms still have room for improvement before they can give reliable results when employed as wireless processors due to the two error sources analyzed before: *i*) the embedding process, which affects the total number of qubits that will be involved in the annealing procedure, and *ii*) the resolution of the coefficients of the QUBO when they are programmed on the actual QPU.

5 Towards Quantum-based Baseband Processors

By executing Qu4Fec on a commercial platform, we discussed how state-of-the-art QPUs are still far from being used as baseband processors, mostly due to the difficulties in the embedding process. In this section, we explore qubit fabric structures to accommodate LDPC codes discussing promising design trends for a Quantum wireless processor design.

5.1 Embedding Algorithms Analysis

Embedding algorithms such as MM take the LDPC source graph (*i.e.*, the one resulting from the code design) as input and map it onto the target one, resulting from the hardware design of the QPU. While

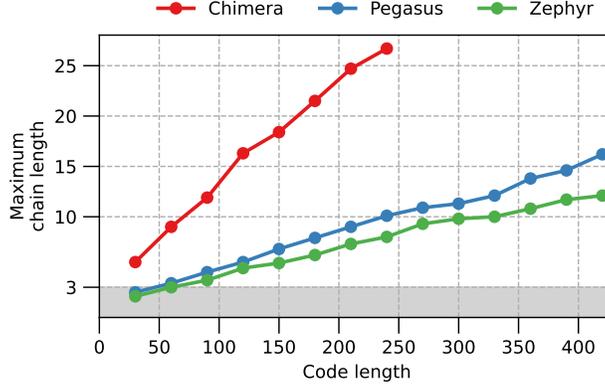


Figure 11: Maximum chain length achieved by MM for different code word lengths and target QPU architecture.

in our experimental evaluation we used the Pegasus architecture, we now analyze the impact of the Chimera (past) and Zephyr (future) D-Wave architectures on the embedding process by measuring the resulting maximum chain length in Fig. 11. Subsequent QPU generations do improve the embedding process in our results. However, only very short codeword source graphs can be embedded with a maximum chain length that is sufficiently short to avoid energy discrepancy with Qu4Fec. According to the left plot of Fig. 10, the maximum chain length should not exceed a value 3, outlining the gray region in Fig. 11, and codewords must be shorter than 40 bits to allow so even with the most recent Zephyr QPU architecture. More realistic 3GPP (2023a) codeword lengths start to suffer a linear increase, with the newest generation only marginally (17% on average) improving the embedding process.

5.2 Optimal QPU Graph Structure

A tailored QPU architecture for LDPC decoding guarantees that no chains are employed at all during the embedding. This requires understanding the characteristics of the source graph. To embed any LDPC code with a source graph of n total nodes and l ancillary nodes, a full mesh of $n + l$ nodes is required. Even a full mesh of just n nodes is already impractical. For instance, to perfectly embed any (2,3)-regular codeword of up to length n as the ones we study in this paper, the target graph should have *i*) $l = \frac{2n}{3}$ for ancillary variables, which are completely unconnected between them, and *ii*) n nodes for the code variables, forming a full mesh to accommodate any possible parity check matrix structure. These two sets of nodes are interconnected as a complete semi-bipartite graph with $\frac{n(7n-3)}{6}$ edges. However, our empirical findings show that typically less than 1% of these edges are employed, indicating an extremely inefficient utilization.

5.3 Zephyr Layout Parameterization

QPU architectures are designed to facilitate QUBO problem embeddings while being practical and extensible. For instance, D-Wave’s next-generation Zephyr QPU follows this principle by creating a qubit fabric characterized by a concatenation of tiles so that QPU target graphs can be effectively represented by two parameters: the tile replication factor m and the internal tile pattern t . In a nutshell, m controls the size of the QPU, while t matches the available edges in a single tile. As a reference, in the former-generation Chimera architecture, the unit cell consisted of a bipartite graph of two shores of nodes, each of size $t = 4$. Pegasus and Zephyr retained this parameter but added sparse connections between nodes on the same shore, known as *odd couplers*.

To understand the capabilities of the Zephyr architecture, we computed the maximum chain length distribution achieved by MM under different values of m and t . The current Zephyr topology implementation from D-Wave offers $t = 4$, so extending this parameter allows us to infer performance for soon-available QPUs. We start by fixing t and varying m in the left plot of Fig. 12. The result shows that if m is large enough, then the advantages brought by extending the size of the architecture are null, hinting at the internal structure of the tile as the most important parameter. We corroborate

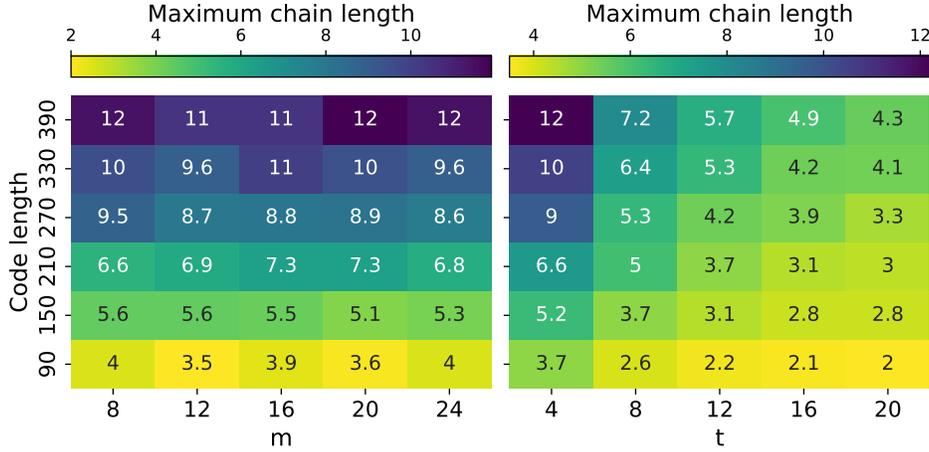


Figure 12: Left: Effect of m on the maximum chain length, with a fixed $t = 4$. Right: Effect of t on the maximum chain length with a fixed $m = 8$.

this in the right plot of Fig. 12, which shows how increasing t effectively reduces the maximum chain lengths as more edges between nodes are available. Still, the gap between current implementations and those that should offer dependable performance (*i.e.*, $t \geq 20$) is large.

5.4 QPU Fabric Extensions

As not even the evolution of current commercial QPU platforms can reliably embed LDPC QUBO problems, we next investigate how they should be extended towards this goal. As we showed above, the key factor in decreasing the maximum chain length is the QPU’s internal connectivity factor. Increasing connectivity within a tile (as shown in Fig. 12) offers a sublinear improvement on the maximum chain length (*i.e.*, from 12 to 4.3 by making the graph 5 times more complex), hence we propose to add couplers *between tiles*.

In the commercial Zephyr, qubits in a tile connect to the corresponding qubits in the neighboring tiles using *external couplers*. We extend this design with a Zephyr- k topology, meaning that each qubit in a tile connects to all the neighboring tiles up to k , as depicted in the top plot of Fig. 13.

We then show the results of the embedding process for several codeword sizes and $(m, t) = (8, 4)$ in the bottom plot of Fig. 13. Increasing the inter-tile connectivity impacts the maximum chain length and offers an easier target graph for embedding the Qu4Fec QUBO problem. A Zephyr-2 architecture would introduce 8% more edges than the original Zephyr, while the Zephyr-8, which may be effective for LDPC decoding, results in a constant 28% increase in the number of edges.

6 Discussion

The transition from 4G/LTE to 5G/NR saw Turbo codes replaced by LDPC codes, largely due to the high degree of data parallelism offered by parity-check matrices, which aligns well with the computational capabilities of modern CPU, GPU, and FPGA/ASIC architectures Richardson and Kudekar (2018). Protograph-based LDPC codes Divsalar et al. (2005), which are already standardized in 5G/NR, in conjunction with Layered BP (introduced in Sec. 2.1.3), are well-suited to the Simple Instruction Multiple Data (SIMD) programming paradigms supported by the aforementioned platforms. This combination significantly enhances algorithm performance, resulting in higher throughput.

QPUs, however, exhibit heterogeneous computational characteristics that are fundamentally different and often orthogonal to those of conventional computing platforms. In this context, an alternative research direction is being shaped at the intersection of FEC and Quantum computing. To fully harness the potential of QPUs, a novel family of LDPC codes could be designed specifically to exploit the unique optimization capabilities of Quantum annealers, such as their ability to find global minima efficiently in high-dimensional spaces, given a certain qubit fabric layout. For example,

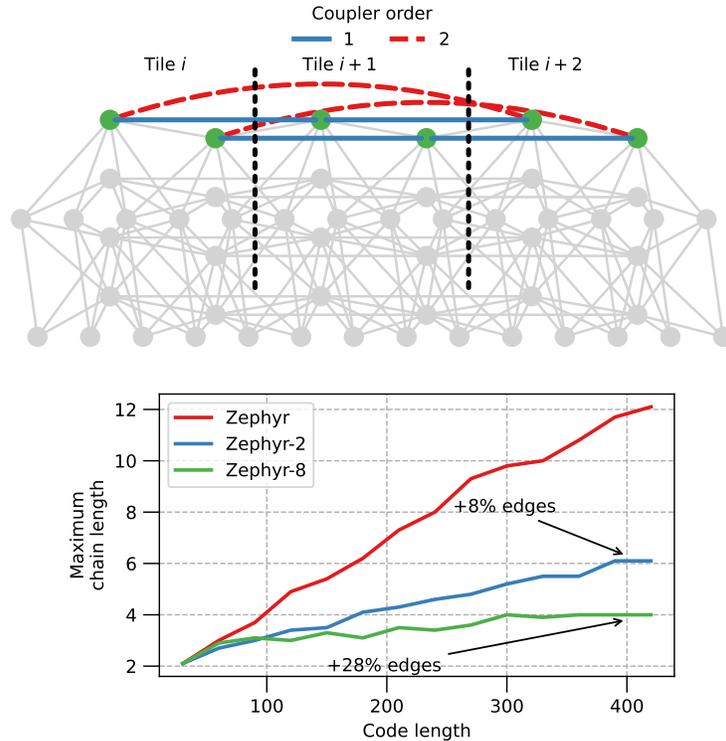


Figure 13: Top: Qubit in tile i connects to qubit in tile $i + 1$ via first-order coupler (original Zephyr) and to qubit in tile $i + 2$ via second-order coupler (our proposal). Bottom: Effect of introducing higher order couplers on the maximum chain length.

jointly co-designing the qubit structure and LDPC code can help minimize the maximum chain length, which is hinted as a significant factor in the low performance in the previous sections. It is important to note that this approach contrasts with QBP, which derived an LDPC code *based* on the qubit fabric rather than co-designing it jointly.

Alternatively, entirely new coding schemes tailored to the specialized characteristics of Quantum platforms could be developed. For instance, Quantum-specific features like superposition and entanglement might inspire coding schemes that exceed the limitations of classical algorithms. These innovations could address challenges such as noise resilience, qubit decoherence, and error propagation in Quantum systems, ensuring more reliable performance in future mobile networks.

7 Conclusion

In this paper, we presented Qu4Fec, a solution for providing LDPC decoding on QPUs. We discussed our guidelines for designing an LDPC decoder on Quantum annealing platforms, formulating it as a QUBO problem. Our decoder outperforms the BLER obtained by the state-of-the-art approaches by almost an order of magnitude and offers carrier-grade performance in simulation. However, when we implemented Qu4Fec in a real QPU, performance dropped, hinting at severe constraints brought in by the hardware platform.

We investigated the causes of such behavior, discovering how scaling, quantization, and embedding the problem on real hardware make dependable wireless decoding currently unfeasible in commercial QPUs. We finally analyzed how extending current platforms would improve performance and identified key internal connectivity characteristics a QPU for wireless baseband processing should have. Our work states the current significant limitations of commercial Quantum computing platforms when employed in a Radio Access Network scenario. Despite the Qu4Fec improved performance,

there are still complex gaps to be filled, both in terms of algorithmic and coding design and the underlying Quantum platform, opening interesting avenues for improvement.

In conclusion, this paper provides a reliable reality check for the feasibility of wireless processing on Quantum annealers: as QPUs are expected to be part of the 6G landscape, this work may open new research paths toward the definition of Quantum-friendly Forward Error Correction Codes for wireless processors.

Acknowledgement

We would like to thank the anonymous reviewers and Nitish Kumar Panigrahy (shepherd) for their valuable feedback that helped us improve this work. This work is supported by the ORIGAMI and TrialsNet projects, which have received funding from the SmartNetworks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101139270 and 101095871. This work is also partially supported by the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 5G I+D 6G-CLARION project.

References

2012. IEEE Standard for Information technology-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11ad-2012* (2012), 1–628. <https://doi.org/10.1109/IEEESTD.2012.6392842>
- 3GPP. 2023a. *NR; Multiplexing and Channel Coding*. Technical Specification (TS) 38.212. 3rd Generation Partnership Project (3GPP). Version 18.0.0.
- 3GPP. 2023b. *NR; Physical Channels and Modulation*. Technical Specification (TS) 38.211. 3rd Generation Partnership Project (3GPP). Version 18.0.0.
- 3GPP. 2023c. *NR; Physical Layer Procedures for Data*. Technical Specification (TS) 38.214. 3rd Generation Partnership Project (3GPP). Version 18.0.0.
- Zaid Albatineh, Khaled Hayajneh, Haythem Bany Salameh, Chinh Dang, and Ahmad Dagmseh. 2020. Robust Massive MIMO Channel Estimation for 5G Networks Using Compressive Sensing Technique. *AEU - International Journal of Electronics and Communications* 120 (2020), 153197. <https://doi.org/10.1016/j.aeue.2020.153197>
- AT&T. 2022. *Cloudifying 5G with an Elastic RAN*. <https://about.att.com/innovationblog/2022/cloudifying-5g-with-elastic-ran.html>
- Ramin Ayanzadeh and Moinuddin Qureshi. 2023. Skipper: Improving the Reach and Fidelity of Quantum Annealers by Skipping Long Chains. arXiv:2312.00264 <https://arxiv.org/abs/2312.00264>
- Dimitris Bertsimas and John Tsitsiklis. 1993. Simulated Annealing. *Statist. Sci.* 8, 1 (1993), 10 – 15. <https://doi.org/10.1214/ss/1177011077>
- Naga Bhushan, Junyi Li, Durga Malladi, Rob Gilmore, Dean Brenner, Aleksandar Damnjanovic, Ravi Teja Sukhavasi, Chirag Patel, and Stefan Geirhofer. 2014. Network Densification: The Dominant Theme for Wireless Evolution into 5G. *IEEE Communications Magazine* 52, 2 (2014), 82–89. <https://doi.org/10.1109/MCOM.2014.6736747>
- K. Boothby, A. D. King, and J. Raymond. 2021. *Zephyr Topology of D-Wave Quantum Processors*. Technical Report 14-1656A-A. D-Wave Technical Report.
- Nikolas P. Breuckmann and Jens Niklas Eberhardt. 2021. Quantum Low-Density Parity-Check Codes. *PRX Quantum* 2, 4 (Oct. 2021). <https://doi.org/10.1103/prxquantum.2.040101>
- Jun Cai, William G. Macready, and Aidan Roy. 2014. A Practical Heuristic for Finding Graph Minors. arXiv:1406.2741

- Jinghu Chen and P.M.C. Fossorier. 2002. Density Evolution for BP-based Decoding Algorithms of LDPC Codes and Their Quantized Versions. In *IEEE GLOBECOM '02*, Vol. 2. 1378–1382 vol.2. <https://doi.org/10.1109/GLOCOM.2002.1188424>
- Jinghu Chen, R.M. Tanner, C. Jones, and Yan Li. 2005. Improved Min-sum Decoding Algorithms for Irregular LDPC Codes. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005*. 449–453. <https://doi.org/10.1109/ISIT.2005.1523374>
- Vicky Choi. 2008. Minor-Embedding in Adiabatic Quantum Computation: I. The Parameter Setting Problem. arXiv:0804.4884 <https://arxiv.org/abs/0804.4884>
- Vicky Choi. 2011. Minor-embedding in Adiabatic Quantum Computation: II. Minor-universal Graph Design. *Quantum Information Processing* 10, 3 (jun 2011), 343–353. <https://doi.org/10.1007/s11128-010-0200-3>
- D-Wave Systems. 2024. *Minorminer*. <https://github.com/dwavesystems/minorminer>
- Aditya Das Sarma, Utso Majumder, Vishnu Vaidya, M. Girish Chandra, A. Anil Kumar, and Sayantan Pramanik. 2023. On Quantum-Assisted LDPC Decoding Augmented with Classical Post-processing". In *"Parallel Processing and Applied Mathematics"*, Roman Wyrzykowski, Jack Dongarra, Ewa Deelman, and Konrad Karczewski (Eds.). Springer International Publishing, Cham, 153–164.
- D. Divsalar, C. Jones, S. Dolinar, and J. Thorpe. 2005. Protograph Based LDPC Codes with Minimum Distance Linearly Growing with Block Size. In *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, Vol. 3. 5 pp.–. <https://doi.org/10.1109/GLOCOM.2005.1577834>
- R. Gallager. 1962. Low-density Parity-check Codes. *IRE Transactions on Information Theory* 8, 1 (1962), 21–28. <https://doi.org/10.1109/TIT.1962.1057683>
- Google. 2024. The Willow Quantum Processor: Advancements in Quantum Error Correction and Scalability. <https://www.googlewillow.org/>.
- Yuhang Guo, Han Zeng, Feng Xiong, Tian Luan, Zaichen Zhang, and Xiaojun Wang. 2024. Quantum Annealing with Post-processing of Maximum Likelihood for LDPC Decoding. In *2024 5th Information Communication Technologies Conference (ICTC)*. 168–172. <https://doi.org/10.1109/ICTC61510.2024.10601825>
- D.E. Hocevar. 2004. A Reduced Complexity Decoder Architecture via Layered Decoding of LDPC Codes. In *IEEE Workshop on Signal Processing Systems, 2004. SIPS 2004*. 107–112. <https://doi.org/10.1109/SIPS.2004.1363033>
- Xiao-Yu Hu, E. Eleftheriou, and D.M. Arnold. 2005. Regular and Irregular Progressive Edge-growth Tanner Graphs. *IEEE Transactions on Information Theory* 51, 1 (2005), 386–398. <https://doi.org/10.1109/TIT.2004.839541>
- Chih-Lin I, Shuangfeng Han, and Sen Bian. 2020. Energy-efficient 5G for a Greener Future. *Nature Electronics* 3, 4 (April 2020), 182–184. <https://doi.org/10.1038/s41928-020-0404-1>
- D-Wave Systems Inc. 2022. Advantage Datasheet. https://www.dwavequantum.com/media/htjclcey/advantage_datasheet_v10.pdf
- Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. 2017. Hardware-efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets. *Nature* 549 (2017), 242–246. <https://doi.org/10.1038/nature23879>
- Srikanth Kasi and Kyle Jamieson. 2020. Towards Quantum Belief Propagation for LDPC Decoding in Wireless Networks. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (London, United Kingdom) (MobiCom '20)*. Association for Computing Machinery, New York, NY, USA, Article 50, 14 pages. <https://doi.org/10.1145/3372224.3419207>

- Srikar Kasi, John Kaewell, and Kyle Jamieson. 2024. A Quantum Annealer-Enabled Decoder and Hardware Topology for NextG Wireless Polar Codes. *IEEE Transactions on Wireless Communications* 23, 4 (2024), 3780–3794. <https://doi.org/10.1109/TWC.2023.3311204>
- Srikar Kasi, Paul Warburton, John Kaewell, and Kyle Jamieson. 2023. A Cost and Power Feasibility Analysis of Quantum Annealing for NextG Cellular Wireless Networks. *IEEE Transactions on Quantum Engineering* 4 (2023), 1–17. <https://doi.org/10.1109/TQE.2023.3326469>
- Jordanis Kerenidis, Anupam Prakash, and Dániel Szilágyi. 2019. Quantum Algorithms for Portfolio Optimization. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies (Zurich, Switzerland) (AFT '19)*. Association for Computing Machinery, New York, NY, USA, 147–155. <https://doi.org/10.1145/3318041.3355465>
- Minsung Kim, Abhishek Kumar Singh, Davide Venturelli, John Kaewell, and Kyle Jamieson. 2024. X-ResQ: Reverse Annealing for Quantum MIMO Detection with Flexible Parallelism. arXiv:2402.18778 [cs.NI] <https://arxiv.org/abs/2402.18778>
- Minsung Kim, Davide Venturelli, and Kyle Jamieson. 2019. Leveraging Quantum Annealing for Large MIMO Processing in Centralized Radio Access Networks. In *Proceedings of the ACM Special Interest Group on Data Communication (Beijing, China) (SIGCOMM '19)*. Association for Computing Machinery, New York, NY, USA, 241–255. <https://doi.org/10.1145/3341302.3342072>
- Minsung Kim, Davide Venturelli, John Kaewell, and Kyle Jamieson. 2022. Warm-started Quantum Sphere Decoding via Reverse Annealing for Massive IoT Connectivity. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking (Sydney, NSW, Australia) (MobiCom '22)*. Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3495243.3560516>
- Thinh V. Le, Manh V. Nguyen, Tu N. Nguyen, Thang N. Dinh, Ivan Djordjevic, and Zhi-Li Zhang. 2023. Benchmarking Chain Strength: An Optimal Approach for Quantum Annealing. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 01. 397–406. <https://doi.org/10.1109/QCE57702.2023.00052>
- Junyu Liu, Min Sheng, Lei Liu, and Jiandong Li. 2017. Network Densification in 5G: From the Short-Range Communications Perspective. *IEEE Communications Magazine* 55, 12 (2017), 96–102. <https://doi.org/10.1109/MCOM.2017.1700487>
- Elisabeth Lobe and Annette Lutz. 2024. Minor Embedding in Broken Chimera and Derived Graphs is NP-complete. *Theoretical Computer Science* 989 (March 2024), 114369. <https://doi.org/10.1016/j.tcs.2023.114369>
- J. Lu and J.M.F. Moura. 2006. Structured LDPC codes for High-density Recording: Large Girth and Low Error Floor. *IEEE Transactions on Magnetics* 42, 2 (2006), 208–213. <https://doi.org/10.1109/TMAG.2005.861748>
- Marco Lucamarini, Zhiliang L. Yuan, James F. Dynes, and Andrew J. Shields. 2018. Overcoming the Rate–distance Limit of Quantum Key Distribution Without Quantum Repeaters. *Nature* 557, 7705 (2018), 400–403. <https://doi.org/10.1038/s41586-018-0066-6>
- Utso Majumder, Aditya Das Sarma, Vishnu Vaidya, and M Girish Chandra. 2022. On Quantum-Enhanced LDPC Decoding for Rayleigh Fading Channels. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*. 462–467. <https://doi.org/10.1109/SEC54971.2022.00070>
- Mavenir. 2023. A Holistic Study of Power Consumption and Energy Savings Strategies for Open vRAN Systems. *White Paper* (February 2023).
- Catherine McGeoch and Pau Farré. 2022. *Advantage™ Processor Overview*. Technical Report. D-Wave Systems. https://www.dwavesys.com/media/3xvdipcn/14-1058a-a_advantage_processor_overview.pdf
- Satoshi Morita and Hidetoshi Nishimori. 2008. Mathematical Foundation of Quantum Annealing. *J. Math. Phys.* 49, 12 (12 2008), 125210. <https://doi.org/10.1063/1.2995837>

- NTT Docomo. 2021. 5G Open RAN Ecosystem. *White Paper* (June 2021).
- O-RAN Work Group 1 (Use Cases and Overall Architecture). 2024. *Use Cases Analysis Report*. <https://specifications.o-ran.org/specifications>
- Edward Parker and Michael J. D. Vermeer. 2023. *Estimating the Energy Requirements to Operate a Cryptanalytically Relevant Quantum Computer*. RAND Corporation, Santa Monica, CA. <https://doi.org/10.7249/WRA2427-1>
- Frank Phillipson. 2024. Quantum Computing in Logistics and Supply Chain Management an Overview. arXiv:2402.17520 <https://arxiv.org/abs/2402.17520>
- Jose P. Pinilla and Steven J. E. Wilton. 2019. Layout-Aware Embedding for Quantum Annealing Processors. In *High Performance Computing*, Michèle Weiland, Guido Juckeland, Carsten Trinitis, and Ponnuswamy Sadayappan (Eds.). Springer International Publishing, Cham, 121–139.
- Abraham P. Punnen. 2022. *The Quadratic Unconstrained Binary Optimization Problem: Theory, Algorithms, and Applications*. Springer International Publishing. <https://doi.org/10.1007/978-3-031-04520-2>
- Tom Richardson and Shrinivas Kudekar. 2018. Design of Low-Density Parity Check Codes for 5G New Radio. *IEEE Communications Magazine* 56, 3 (2018), 28–34. <https://doi.org/10.1109/MCOM.2018.1700839>
- Leonardo Lo Schiavo, Gines Garcia-Aviles, Andres Garcia-Saavedra, Marco Gramaglia, Marco Fiore, Albert Banchs, and Xavier Costa-Perez. 2024. CloudRIC: Open Radio Access Network (O-RAN) Virtualization with Shared Heterogeneous Computing. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (Washington D.C., DC, USA) (*ACM MobiCom '24*). Association for Computing Machinery, New York, NY, USA, 558–572. <https://doi.org/10.1145/3636534.3649381>
- R. Tanner. 1981. A Recursive Approach to Low Complexity Codes. *IEEE Transactions on Information Theory* 27, 5 (1981), 533–547. <https://doi.org/10.1109/TIT.1981.1056404>
- Masashi Tawada, Shu Tanaka, and Nozomu Togawa. 2020. A New LDPC Code Decoding Method: Expanding the Scope of Ising Machines. In *2020 IEEE International Conference on Consumer Electronics (ICCE)*. 1–6. <https://doi.org/10.1109/ICCE46568.2020.9043057>
- Benjamin Villalonga, Dmitry Lyakh, Sergio Boixo, Hartmut Neven, Travis S Humble, Rupak Biswas, Eleanor G Rieffel, Alan Ho, and Salvatore Mandrà. 2020. Establishing the Quantum Supremacy Frontier with a 281 Pfp/s Simulation. *Quantum Science and Technology* 5, 3 (apr 2020), 034003. <https://doi.org/10.1088/2058-9565/ab7eeb>
- Stephanie Wehner, David Elkouss, and Ronald Hanson. 2018. Quantum Internet: A Vision For The Road Ahead. *Science* 362, 6412 (2018), eaam9288. <https://doi.org/10.1126/science.aam9288> arXiv:<https://www.science.org/doi/pdf/10.1126/science.aam9288>
- C. D. Wilen, S. Abdullah, N. A. Kurinsky, C. Stanford, L. Cardani, G. D’Imperio, C. Tomei, L. Faoro, L. B. Ioffe, C. H. Liu, A. Opremcak, B. G. Christensen, J. L. DuBois, and R. McDermott. 2021. Correlated Charge Noise and Relaxation Errors in Superconducting Qubits. *Nature* 594, 7863 (2021), 369–373. <https://doi.org/10.1038/s41586-021-03557-5>
- Stefanie Zbinden, Andreas Bäertschi, Hristo Djidjev, and Stephan Eidenbenz. 2020. Embedding Algorithms for Quantum Annealers with Chimera and Pegasus Connection Topologies. In *High Performance Computing*, Ponnuswamy Sadayappan, Bradford L. Chamberlain, Guido Juckeland, and Hatem Ltaief (Eds.). Springer International Publishing, Cham, 187–206.
- Haotian Zhang and J.M.F. Moura. 2003. The Design of Structured Regular LDPC Codes with Large Girth. In *IEEE GLOBECOM '03*, Vol. 7. 4022–4027 vol.7. <https://doi.org/10.1109/GLOCOM.2003.1258984>