

Quantum Annealing for Large MIMO Downlink Vector Perturbation Precoding

Srikanth Kasi*
Student Member, IEEE
 Princeton University

Abhishek Kumar Singh*
Student Member, IEEE
 Princeton University

Davide Venturelli
 Universities Space
 Research Association

Kyle Jamieson
Senior Member, IEEE
 Princeton University

Abstract—In a multi-user system with multiple antennas at the base station, precoding techniques in the downlink broadcast channel allow users to detect their respective data in a non-cooperative manner. Vector Perturbation Precoding (VPP) is a non-linear variant of transmit-side channel inversion that perturbs user data to achieve full diversity order. While promising, finding an optimal perturbation in VPP is known to be an NP-hard problem, demanding heavy computational support at the base station and limiting the feasibility of the approach to small MIMO systems. This work proposes a radically different processing architecture for the downlink VPP problem, one based on Quantum Annealing (QA), to enable the applicability of VPP to large MIMO systems. Our design reduces VPP to a quadratic polynomial form amenable to QA, then refines the problem coefficients to mitigate the adverse effects of QA hardware noise. We evaluate our proposed QA based VPP (QAVP) technique on a real Quantum Annealing device over a variety of design and machine parameter settings. With existing hardware, QAVP can achieve a BER of 10^{-4} with $100\mu\text{s}$ compute time, for a 6×6 MIMO system using 64 QAM modulation at 32 dB SNR.

Index Terms—Vector Perturbation, Downlink Precoding, Quantum Computation, Quantum Annealing, Optimization

I. INTRODUCTION

Modern wireless networks are experiencing tremendous growth in traffic loads at base stations, and hence to meet the resulting computational and latency requirements, designers continue to investigate new architectures and hardware for today's 5G and tomorrow's 6G networks. A large component of cellular baseband processing comprises of downlink data traffic due to a significant rise in the popularity and usage of video streaming platforms (*e.g.*, Netflix). To meet the ever-growing user demand, it is critical for the base stations to enhance the quality of downlink data streams in terms of throughput, error rate, and latency.

In a multi-user multiple-input multiple-output (MIMO) downlink data transmission, *precoding* techniques can be used to eliminate the effect of inter-user interference and allow users to detect their respective data non-cooperatively, minimizing error-rate and maximizing throughput. In this work, we focus on *Vector Perturbation Precoding* (VPP) [1]. VPP is a widely studied non-linear precoding technique that performs transmit-side channel inversion over a perturbed user data vector to reduce the transmit power scaling. Although VPP has been shown to achieve better error performance compared

to other precoding techniques (*e.g.*, zero-forcing, Tomlinson-Harashima Precoding [2]), finding an optimal perturbation for user data in VPP is known to be NP-hard, making its implementation in massive/large MIMO systems to be infeasible.

A promising and cost-effective architecture to address the increased computational burden of wireless networks is a *Centralized Radio Access Network (C-RAN)* [3] architecture, which aggregates the computationally-demanding processing at many wireless base stations. Since baseband physical-layer processing is highly time-critical and the required stream of baseband signal samples has a high data-rate, this type of C-RAN deployment imposes both latency and bandwidth requirements on the interconnect between each base station and the data center. Related studies to this end are investigating quantum computation for solving networking problems in the uplink: such as Channel coding [4] and ML detection [5]. However, investigating quantum computation for problems in the downlink remains unexplored to the best of our knowledge.

In this paper, we propose a radically different processing to the NP-hard VPP problem in the downlink: *Quantum Annealing based Vector Perturbation Precoding (QAVP)*. Our proposed technique leverages recent advances in quantum computational devices and applies them to the problem of VPP. QAVP's approach is to represent the VPP problem as an optimization problem over a quadratic polynomial with binary variables *i.e.*, *Quadratic Unconstrained Binary Optimization* (QUBO), which is an optimization form that a *Quantum Annealer (QA)* machine takes as input, then refine the QUBO coefficients to mitigate the adverse effects of QA hardware noise and the process of mapping the polynomial onto the physical QA qubit hardware topology. After these *preprocessing* and *embedding* steps, QAVP uses a real Quantum Annealing machine to solve the resultant QUBO problem and then constructs the VPP perturbation from the solutions returned by the QA machine. Our results show that, from the standpoint of computation time, QAVP can outperform popular encoding algorithms for large MIMO systems.

II. VECTOR PERTURBATION PRECODING

We consider a general MIMO downlink scenario where a base station equipped with N_t transmit antennas communicate data streams with N_r ($\leq N_t$) single-antenna non-cooperative users independently and simultaneously.

* Co-primary authors, appearing in randomly chosen order.

In VPP, the user data symbol vector $\mathbf{u} \in \mathbb{C}^{N_r \times 1}$ is perturbed by an integer vector $\mathbf{v} \in \mathbb{G}^{N_r \times 1}$. This maps the data symbols to a wider constellation space, forming a perturbed transmit vector, $\mathbf{d} = \mathbf{u} + \tau\mathbf{v}$. Here, \mathbf{v} is a vector of Gaussian integers¹ and $\tau = 2(|c_{max}| + \Delta/2)$ is a constant chosen to provide symmetric decoding regions around the constellation points. $|c_{max}|$ is the magnitude of the largest constellation symbol and Δ is the spacing between the constellation symbols [6]. The perturbed vector \mathbf{d} is then precoded with a precoder matrix ($\mathbf{P} \in \mathbb{C}^{N_t \times N_r}$), where the choice of \mathbf{P} inverts the wireless channel $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ and reduces the effect of wide range of eigenvalues of the channel coefficients [6]. The precoder matrix $\mathbf{P} = \mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}$ is used in Zero Forcing (ZF) precoding. The received symbol vector \mathbf{y} corresponding to the transmitted symbol vector $\mathbf{x} = \mathbf{P}\mathbf{d}/\sqrt{P_t}$ is given by,

$$\mathbf{y} = \frac{1}{\sqrt{P_t}}(\mathbf{H}\mathbf{P}\mathbf{d}) + \mathbf{n} \quad (1)$$

where the scalar $P_t = \|\mathbf{P}\mathbf{d}\|^2$ is the transmission power scaling factor, which is assumed to be known at the receiver, and \mathbf{n} is wireless channel noise. The receiver decodes by applying a modulo τ operation to the received signal \mathbf{y} . The transmission power scaling (P_t) causes noise amplification during the decoding process. In order to minimize the transmission power scaling, an optimal choice of the perturbation vector (\mathbf{v}^*) that leads to the smallest P_t is computed as in [6]:

$$\mathbf{v}^* = \arg \min_{\mathbf{v}} \left\| \mathbf{H}^H(\mathbf{H}\mathbf{H}^H)^{-1}(\mathbf{u} + \tau\mathbf{v}) \right\|^2. \quad (2)$$

III. RELATED WORK

The Sphere encoder [6] builds on the Fincke and Pohst algorithm [7] by expressing the precoding matrix \mathbf{P} as $\mathbf{P} = \mathbf{Q}\mathbf{R}$ by QR decomposition, where \mathbf{Q} is unitary and \mathbf{R} is upper triangular. It then performs a tree search, utilizing the upper triangular structure and limiting the search to the points within a hyper-sphere of a suitably chosen radius, and hence avoids exhaustive search over all possibilities. This algorithm is very similar to the Sphere decoder used for the MIMO receiver. While Sphere encoder is much better than exhaustive search, its expected complexity is still exponential. Park *et al.* [8] provide an approximation to the VPP problem by minimizing the real and imaginary parts of the VPP cost function separately. This reduces the complexity of VPP computation at the cost of error performance. The Thresholded Sphere Search algorithm [9] imposes an additional stopping criteria to Sphere encoder algorithm [6] based on an SNR dependent threshold heuristic. Another scheme for reducing the search space of Sphere encoder is discussed in [10]. It restricts the values of each perturbation to four possible values (two for real and two for imaginary) and hence reducing the search space of the Sphere encoder. Despite improvements in computation costs over Sphere encoder, the search space of all these methods is still exponential with respect to the MIMO size.

¹The set of Gaussian integers $\mathbb{G} = \mathbb{Z} + j\mathbb{Z}$ consists of all complex numbers whose real and imaginary parts are integers.

Several approximations for VPP (with polynomial complexity) exist in literature. The Fixed Complexity Sphere Encoder (FSE), adapts the Sphere encoder to have a lower and fixed polynomial complexity by pruning a large number of branches during the tree search but leads to degradation in error performance [11]. Degree-2 Sparse Vector Perturbation (D2VP), presented in [12], is a low complexity algorithm for vector perturbation precoding. It reduces the complexity of finding the VPP solution by assuming that only 2 elements of the perturbation vector can be non-zero and then improves the solution over multiple iterations.

IV. QUANTUM ANNEALING

Quantum Annealing (QA) is a heuristic approach that implements in hardware a quantum computing algorithm inspired by the Adiabatic Theorem of quantum mechanics [13]. The method aims to find the lowest energy *spin configuration* (solution) of the class of *quadratic unconstrained binary optimization (QUBO)* problems in their equivalent *Ising* specification. The Ising form is described by:

$$E = \sum_i h_i s_i + \sum_{i < j} J_{ij} s_i s_j \quad (3)$$

where E is the *energy*, s_i is a solution variable, h_i and J_{ij} are problem parameters called *bias* and *coupler strength* respectively. QUBO form is obtained from Eq.3 by a simple variable transformation ($s_i \rightarrow 2q_i - 1$), where s_i and q_i represent Ising and QUBO form variables respectively. Ising form solution variables (s_i) take on values in $\{-1, +1\}$, and QUBO form solution variables (q_i) take on values in $\{0, 1\}$. The equivalent QUBO form of Eq. 3 is described by:

$$E = \sum_i f_i q_i + \sum_{i < j} g_{ij} q_i q_j \quad (4)$$

A. Primer: Quantum Annealers.

Quantum Annealers are specialized quantum computers, essentially comprised of two types of resources: *qubits* (quantum bits) and *couplers*, whose regular connectivity structure is organized in *unit cells*. Fig. 1 shows the hardware structure of the QA we adapt in this study, the D-Wave 2000Q (DW2Q) quantum annealing machine. The QA device programs the linear (*i.e.*, biases) and quadratic (*i.e.*, coupler strengths) coefficients of Eq.3 onto qubits and couplers using controllable inductive elements in proximity of superconducting Josephson Junctions present on the chip [4].

Embedding. The process of mapping the problem at hand onto the physical QA hardware topology follows the graph-theory problem of graph minor *embedding*. We demonstrate here the embedding process with an example Ising problem:

$$E = J_{12}s_1s_2 + J_{13}s_1s_3 + J_{23}s_2s_3 \quad (5)$$

The direct connectivity of this example problem is shown in Fig.2(a). However a three-node complete connectivity graph structure does not exist in the QA hardware (*cf.* Fig.1). Hence the minor-embedding approach is to map one of the problem variables in Fig.2(a) (*e.g.*, s_3) onto two physical

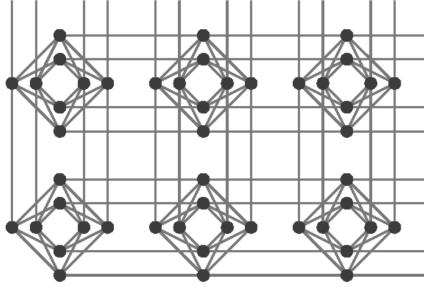


Figure 1: The figure shows *qubits* (nodes) and *couplers* (edges) in DW2Q QA hardware. Each group of eight qubits is called *unit cell*.

qubits (e.g., s_{3a} and s_{3b}) as Fig.2(b) shows, such that the resulting connectivity can be realized on the QA hardware. The couplers between the qubits s_{3a} and s_{3b} are tasked to enforce these physical qubits to be correlated in order to end up with the same value at the end of the annealing process. This is implemented through a strong ferromagnetic interaction energy called *chain strength* J_F (see in Fig.2(b)).

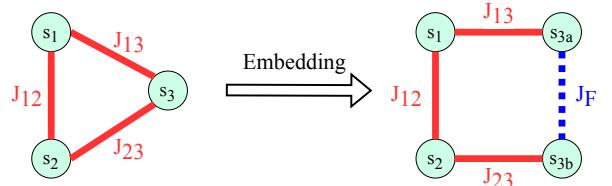
Coefficient considerations. Today's QA devices provide support for bias values in $[-2, +2]$ and coupler strength values in $[-2, +1]$ with a bit-precision guaranteed to 4–5 bits only. Further, the QA introduces an analog machine noise distinct from communication channel noise called *intrinsic control errors* or *ICE*. ICE noise, a collection of errors caused by qubit flux noise, susceptibility, among others [14], essentially alters the problem coefficients ($h_i \rightarrow h_i \pm \delta h_i$, $J_{ij} \rightarrow J_{ij} \pm \delta J_{ij}$). Although the errors δh_i and δJ_{ij} are currently on the order of 10^{-2} , these may degrade the solution quality of some problems in scenarios where ICE noise erases significant information from the ground state of the input problem.

B. Annealing process

QA processors simulate systems in the transverse field Ising model described by the time-dependent energy functional (Hamiltonian):

$$H = -A(s) \sum_i \sigma_i^x + B(s) \left\{ \sum_i h_i \sigma_i^z + \sum_{i < j} J_{ij} \sigma_i^z \sigma_j^z \right\} \quad (6)$$

where $\sigma_i^{x,z}$ are spin operators (Pauli matrices) acting on the i^{th} qubit, h_i and J_{ij} are problem parameters, $s (= t/t_a)$ is called *annealing schedule* where t is the time and t_a is the *annealing time*. $A(s)$ and $B(s)$ are two monotonic scaling signals in the annealer such that at time $t = 0$, $A(0) \gg B(0) \approx 0$ and at time $t = t_a$, $B(1) \gg A(1) \approx 0$. The annealing algorithm initializes the system in the ground state of $\sum_i \sigma_i^x$ where each qubit is in a *superposition* state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, then adiabatically evolves this Hamiltonian from time $t = 0$ until $t = t_a$ (i.e., decreasing $A(s)$, increasing $B(s)$). The time-dependent evolution described by the Schrödinger Equation driven by these signals A and B is essentially the annealing algorithm. While in real processor the dynamics is not ideal but it is dominated by dissipative noise instead [15], the expectations of the model still hold for the most part. The



(a) Problem Connectivity

(b) Physical QA Connectivity

Figure 2: The figure demonstrates the embedding process of Eq. 5. In the figure, **(a)** shows the direct problem connectivity of Eq. 5 and **(b)** shows its physical connectivity on QA hardware.

process of optimizing a problem in the QA is called an *anneal*, while the time taken for an anneal is called *annealing time*.

V. DESIGN

Quantum Annealing based Vector Perturbation (QAVP) envisions a scenario where a Quantum Annealer (QA) machine is co-located with a centralized data center for computational processing in the C-RAN architecture [3] which allows for low latency communication between base stations and the QA machine. QAVP converts a VPP problem to a Quadratic Unconstrained Binary Optimization (QUBO) problem, then fine-tunes the QUBO problem coefficients to mitigate the adverse effects of QA hardware noise and increase the probability of finding the correct solution with QA. The fine-tuned QUBO is next embedded onto the physical QA hardware (see Fig.1) for running the problem. We conduct multiple anneals for a given QUBO problem, where each anneal generates a candidate solution bit-string for VPP. The QUBO solutions are converted to perturbation vectors by inverting the transform described in Eq. 7. Among these returned solutions, the solution that minimizes the VPP objective function is chosen by the base station. In scenarios where QAVP fails to find a solution with lower transmit power scaling than Zero-Forcing (ZF) precoding, the base station discards the QAVP solution and uses ZF precoding instead of VPP.

The steps involved in QAVP computation, excluding embedding and annealing, can be executed by the physical layer of base station or can be offloaded to a server in close proximity to QA machine. Fig. 3 illustrates a typical deployment scenario for QAVP, where N_r users are receiving downlink data streams from a base station with N_t antennas. We next demonstrate QAVP's QUBO formulation for the VPP problem, QAVP's pre-processing, and embedding considerations.

A. QAVP's QUBO Formulation

The downlink VPP problem is to find an optimal perturbation vector \mathbf{v}^* that minimizes the transmit power at the base station, whose NP-hard objective function is represented in Eq. 2. As the search objective in Eq.2 is over the vector $\mathbf{v} \in \mathbb{G}^{N_r \times 1}$, we construct each entry in \mathbf{v} by a linear combination of binary variables as follows.

Let q_i denote the i^{th} QUBO form solution variable, and let $a_k + jb_k$ be the k^{th} entry in \mathbf{v} , where a_k and b_k are integers.

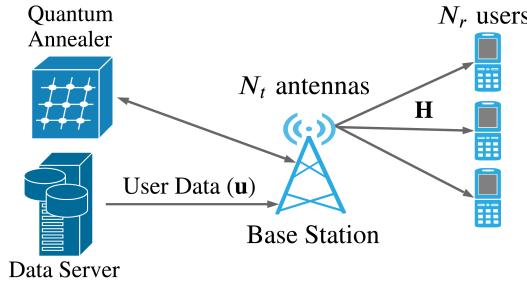


Figure 3: A typical deployment scenario of QAVP. N_r users are receiving downlink data streams from a base station with N_t antennas.

We design each a_k and $b_k \forall k$ in an identical fashion using $t + 1$ distinct solution variables as:

$$a_k / b_k = \sum_{m=1}^t 2^{m-1} \cdot q_m - 2^t \cdot q_{t+1} \quad (7)$$

Since all the solution variables (q_i) are binary, this formulation allows a_k and b_k to take all integer values in the range $[-2^t, 2^t - 1]$, where each integer corresponds to a unique configuration of solution variables. The value of t determines the search range of \mathbf{v} . Yuen *et al.* [16] show that perturbation values are most likely to be $\{-1, 0, 1\}$, and hence $t = 1$ which allows perturbation values in the range $[-2, 1]$, is usually sufficient. We substitute the entries of \mathbf{v} with their corresponding solution variables into Eq.2 to obtain the QAVP's QUBO:

$$\arg \min_{\forall q} \left\{ \sum_{\forall i} f_i(\mathbf{H}, \mathbf{u}) q_i + \sum_{\forall i, j} g_{ij}(\mathbf{H}, \mathbf{u}) q_i q_j \right\} \quad (8)$$

where f and g are functions of MIMO channel matrix and user data vector. The linear and quadratic coefficients obtained from Eq. 8 can be programmed into the QA processor. The computational complexities of the proposed QA technique and the optimal case Sphere Encoder are $O(e^{\sqrt{N_u}})$ [17] and $O(e^{N_u})$ [18] respectively, where N_u is the number of users. We next present our QUBO pre-processing considerations.

B. QAVP's Pre-processing

Our pre-processing scheme aims to mitigate the adverse effects of QA hardware ICE noise (§IV) by scaling and eliminating minor QUBO coefficients. We note that a generic QUBO form of Eq.4 can be equivalently expressed as:

$$E = \sum_i f_i q_i + \sum_{i < j} g_{ij} q_i q_j = \mathbf{q}^T \mathbf{Q} \mathbf{q} \quad (9)$$

where $\mathbf{q} = [q_1, q_2, \dots]^T$, and \mathbf{Q} is an upper triangular matrix with $Q_{ij} = f_i$ ($i = j$), $Q_{ij} = g_{ij}$ ($i < j$), $Q_{ij} = 0$ ($i > j$). Let $Q_{max} = \max_{i,j} |Q_{ij}|$ be the maximum QUBO coefficient value, T_{high} and T_{low} be our chosen upper and lower bounds for the QUBO coefficient values. We define a *scaleFactor* as:

$$scaleFactor = \begin{cases} 1 & Q_{max} \leq T_{high} \\ \frac{T_{high}}{Q_{max}} & Q_{max} > T_{high} \end{cases}, \quad (10)$$

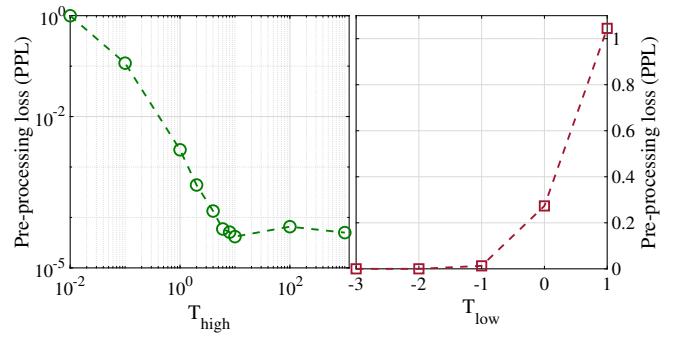


Figure 4: QAVP's pre-processing loss for different choices of T_{high} and T_{low} , showing that $T_{high} = 6$ is near-optimal and $T_{low} = -2$ leads to negligible PPL.

Our approach is to first scale each entry in \mathbf{Q} with the scale-factor, then eliminate coefficients below a chosen threshold of $10^{T_{low}}$. This scaling process is summarized below.

Step 1: Set $\mathbf{Q} \leftarrow \mathbf{Q} * scaleFactor$.

Step 2: For every $i \leq j$, if $Q_{ij} < 10^{T_{low}}$ then set $Q_{ij} \leftarrow 0$.

Let \mathbf{Q}_{pre} correspond to the QUBO obtained after this pre-processing step. Let $\mathbf{q}_{pre}^* = \arg \min_{\mathbf{q}} \mathbf{q}^T \mathbf{Q}_{pre} \mathbf{q}$, and $\mathbf{q}^* = \arg \min_{\mathbf{q}} \mathbf{q}^T \mathbf{Q} \mathbf{q}$. We define a *pre-processing loss (PPL)* as:

$$PPL = \frac{|(\mathbf{q}_{pre}^{*T} \mathbf{Q} \mathbf{q}_{pre}^* - \mathbf{q}^{*T} \mathbf{Q} \mathbf{q}^*)|}{|(\mathbf{q}^{*T} \mathbf{Q} \mathbf{q}^*)|} \quad (11)$$

PPL is used to quantify deterioration in the optimal value of the VPP problem due to the modification of the original QUBO problem. We see from Fig. 4 that PPL increases with an increase in T_{low} as a higher T_{low} causes more coefficients to be reduced to zero. We further observe that PPL reduces initially with T_{high} , and then becomes constant. While having a high negative value for T_{low} leads to low PPL, it makes it more susceptible to QA ICE noise. The optimal choices for T_{high} and T_{low} depend on the distribution of QUBO coefficients, magnitude of QA ICE noise, and problem embedding. Our empirical studies find that a $T_{high} = 6$ and a $T_{low} = -2$ obtains good solutions with the DW2Q QA.

C. QAVP's Embedding and Parallelism.

We use a D-Wave's heuristic algorithm described in Ref. [19] and implemented in D-Wave MinorMiner software libraries for mapping of QAVP's QUBO design onto the QA hardware. It is to note that multiple QUBOs or multiple instances of a QUBO can be parallelly processed by mapping different problems onto distinct physical locations in the QA chip. For instance, QAVP for 7×7 MIMO system requires approximately 200 qubits, implying that 9–10 such problems can be simultaneously solved on the DW2Q QA with 2,048 qubits. The number of qubits in the QA hardware has been steadily doubling each year and this trend is expected to continue [4], with the latest 5,000 qubit chip just released

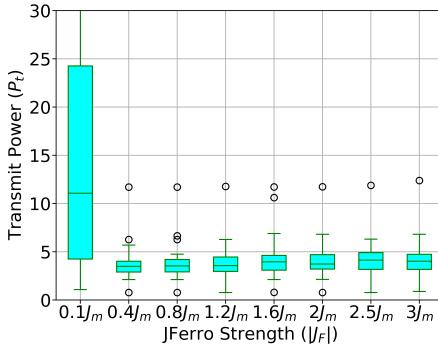


Figure 5: Evaluation of the overall transmission power (P_t) amplitudes over various choices of $|J_F|$ at $T_a = 100 \mu\text{s}$ and 10 dB channel SNR. The plot shows that QAVP's performance variance is not significant for J_F values above $0.4J_m$.

at time of writing. While the evolution of QA technology is currently at an early stage (2011-), future QA processors with higher qubit counts will potentially increase the supported parallelism for QAVP.

VI. EVALUATION

In this section, we first present microbenchmarks for QA machine parameters. We then investigate QAVP's end-to-end error performance, comparing head-to-head against Zero Forcing (ZF) and Fixed Complexity Sphere Encoder (FSE) algorithms. The DW2Q QA system performance is majorly affected by the choice of *annealing time* (T_a), number of anneals (N_a), and *chain strengths* (J_F). Recall that number of anneals (N_a) refers to the number of times annealing process is repeated for each QUBO problem. Our evaluation methodology is as follows: we obtain the VPP solutions from the QA (DW2Q), and use these solutions to simulate an end-to-end downlink data transmission on a trace-driven MIMO simulator (implemented in MATLAB). We simulate data transmission over Rayleigh fading wireless channels and empirically measure the BER corresponding to different users.

A. Effect of Chain Strength (J_F)

We evaluate QAVP's performance over an 10×8 MIMO system with 20 QAVP problem instances. Our channel matrix \mathbf{H} is random Rayleigh fading, modulation scheme is BPSK, and wireless channel noise is Gaussian. Let J_m be the maximum coupler strength value of our QUBO problem at hand.

In Fig. 5 we investigate the sensitivity of chain strengths $|J_F|$ over the overall transmission power P_t . For this evaluation, we set a high annealing time of $100 \mu\text{s}$ and a moderate SNR of 10 dB to ensure minimal disturbance from the time limit and the channel SNR respectively. We observe that the performance of QAVP varies with J_F and that the sensitivity of QAVP performance beyond $|J_F| = 0.4J_m$ is not significant. The average P_t is much higher for $|J_F| < 0.4J_m$, and the variance in performance of QAVP is barely distinguishable for higher values of $|J_F| > 0.4J_m$. While the optimal value of J_F depends on the parameters of the system (e.g., modulation and MIMO size), we empirically determine that $|J_F| = 1.2J_m$

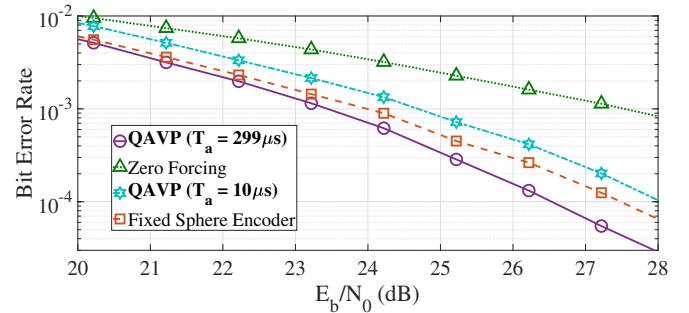


Figure 6: Downlink multi-user 6×6 MIMO: BER for various QAVP anneal times T_a . We compare error performance of QAVP for various values of anneal time (T_a) while keeping the number of anneals ($N_a = 10^4$) fixed. We set $|J_F| = 1.2J_m$.

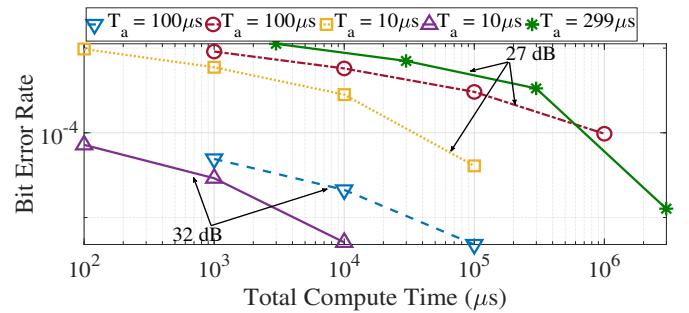


Figure 7: Downlink multi-user 6×6 MIMO: Bit Error Rate vs Total Computation Time ($N_a \times T_a$). We demonstrate the BER performance, at two values of SNR, over different combinations of anneal time (T_a) and number of anneals (N_a). We set $|J_F| = 1.2J_m$.

is the optimal setting for 6×6 and 12×12 MIMO systems with 64 QAM modulation scheme.

B. Effect of Anneal Time (T_a)

Another critical QA parameter that affects the QAVP performance is the anneal time (T_a), along with its associated number of anneals (N_a). We heretofore quantify the QA performance with the total compute time: $T_a \times N_a$. Fig. 6 compares the BER performance of FSE, ZF and QAVP, and Fig. 7 reports the BER performance of QAVP at different QA compute times, for a $N_t = 6$, $N_r = 6$ MIMO system with 64 QAM modulation, for various choices of T_a and N_a .

We first observe in Fig. 6 that a higher anneal time leads to better BER performance from QA. Although a higher anneal time allows QA to return better quality solutions, it increases the communication latency (due to a higher computation time) as Fig. 7 depicts. The tradeoff between BER performance and anneal time represents a fundamental design problem in QAVP. We observe from Figs. 6 and 7 that QAVP achieves an acceptable BER with a low anneal time of $T_a = 10 \mu\text{s}$, while keeping the computation time lower by an order of magnitude.

C. End-to-end BER performance

In this section, we consider a large 12×12 MIMO system with Rayleigh fading channel, White Gaussian noise, and 64 QAM modulation. Fig. 8 reports results (with $|J_F| = 1.2J_m$,

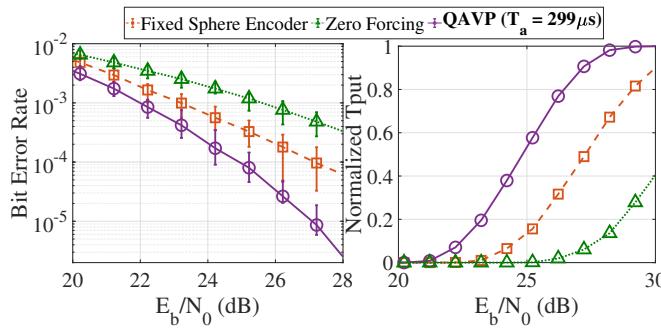


Figure 8: Downlink multi-user 12×12 MIMO: BER and throughput performance. We compare the BER performance of QAVP against FSE and ZF. We set $|J_F| = 1.2J_m$, $T_a = 299\mu\text{s}$ and $N_a = 10^4$.

$T_a = 299 \mu\text{s}$, $N_a = 10^4$) for mean BER and normalized throughput (ratio of achieved to maximum possible throughput), with packets size of 1500 bytes, for uncoded data transmission. We observe from Fig. 8 that QAVP outperforms ZF and FSE by achieving a 1–2 orders of magnitude lower BER at E_b/N_0 of 27dB. We see from the throughput curves that QAVP achieves a 3 dB gain over FSE and around 6 dB gain over ZF. It is to note that a 12×12 MIMO system with VPP, using theoretically optimal Sphere Encoder, is practically infeasible due to its high computational complexity.

VII. CONCLUSION

In this paper we propose QAVP, a novel technique that performs Vector Perturbation Precoding using Quantum Annealing. We evaluate QAVP on a real DW2Q QA device over a variety of machine parameters. Our studies show that for large MIMO systems, if the analysis is restricted to computation time, QAVP can outperform practically feasible state-of-the-art techniques like FSE and ZF algorithms for VPP. While prior work investigates QA technology for problems in the uplink [4], [5], our studies investigate its potential in the downlink. Our analysis disregards the engineering and system integration overheads of currently available commercial QA systems (e.g. latency, programming time and thermalization times between the runs), since they can be optimized heavily if the system is built for a specific application deployment. Nevertheless, the techniques we propose in this work, in the more distant future, may enable the applicability of VPP to large MIMO systems by exploiting stronger QA devices and parallelism.

A. Future Work

Our work has several possible improvements along the lines of QUBO pre-processing, problem embedding, and QA machine parameter selection. It is of interest to further refine our pre-processing step to optimize the QUBO representation based on the embedding of the problem. Further, a more sophisticated tuning and selection of QA parameters such as chain strengths and annealing times can potentially improve the QAVP performance. Porting our model to other alternative emerging Ising-model related technologies such as quantum-inspired algorithms, reverse quantum annealing, and quantum-classical hybrid approaches is also a potential work direction.

ACKNOWLEDGEMENTS

This research is supported by National Science Foundation (NSF) Awards CNS-1824357 and CNS-1824470. Support from USRA Cycle 3 Research Opportunity Program allowed machine time on a D-Wave Quantum Annealer machine hosted at NASA Ames Research Center.

REFERENCES

- [1] C. B. Peel, B. M. Hochwald, and A. L. Swindlehurst, “A vector-perturbation technique for near-capacity multiantenna multiuser communication—Part I: channel inversion and regularization,” *IEEE Transactions on Communications*, vol. 53, no. 1, pp. 195–202, 2005.
- [2] A. A. D’Amico, “Tomlinson-Harashima Precoding in MIMO Systems: A Unified Approach to Transceiver Optimization Based on Multiplicative Schur-Convexity,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3662–3677, 2008.
- [3] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, “Cloud RAN for mobile networks—a technology overview,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 405–426, 2014.
- [4] S. Kasi and K. Jamieson, “Towards quantum belief propagation for LDPC decoding in wireless networks,” in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom, Association for Computing Machinery, 2020.
- [5] M. Kim, D. Venturelli, and K. Jamieson, “Leveraging quantum annealing for large MIMO processing in centralized radio access networks,” in *Proceedings of the ACM Special Interest Group on Data Communication*, pp. 241–255, 2019.
- [6] B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst, “A vector-perturbation technique for near-capacity multiantenna multiuser communications—Part II: perturbation,” *IEEE Transactions on Communications*, vol. 53, no. 3, pp. 537–544, 2005.
- [7] U. Fincke and M. Pohst, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, 1985.
- [8] S. Park, H. Han, S. Lee, and I. Lee, “A decoupling approach for low-complexity vector perturbation in multiuser downlink systems,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 6, pp. 1697–1701, 2011.
- [9] C. Masouros, M. Sellathurai, and T. Ratnarajah, “Computationally efficient vector perturbation precoding using thresholded optimization,” *IEEE Transactions on Communications*, vol. 61, no. 5, pp. 1880–1890, 2013.
- [10] H.-S. Han, S.-H. Park, S. Lee, and I. Lee, “Modulo loss reduction for vector perturbation systems,” *IEEE Transactions on Communications*, vol. 58, no. 12, pp. 3392–3396, 2010.
- [11] M. Mohaisen and K. Chang, “Fixed-complexity sphere encoder for multi-user MIMO systems,” *Journal of Communications and Networks*, vol. 13, no. 1, pp. 63–69, 2011.
- [12] Y. Ma, A. Yamanı, N. Yi, and R. Tafazolli, “Low-complexity MU-MIMO nonlinear precoding using degree-2 sparse vector perturbation,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 497–509, 2016.
- [13] T. Albash and D. A. Lidar, “Adiabatic quantum computation,” *Reviews of Modern Physics*, vol. 90, no. 1, p. 015002, 2018.
- [14] D-Wave, “Technical description of the D-Wave Quantum Processing Unit,” *D-Wave Systems User Manual*, pp. 09-1109A-O, 2019.
- [15] J. Marshall, D. Venturelli, I. Hen, and E. G. Rieffel, “Power of pausing: Advancing understanding of thermalization in experimental quantum annealers,” *Physical Review Applied*, vol. 11, no. 4, p. 044083, 2019.
- [16] C. Yuen and B. M. Hochwald, “How to gain 1.5 dB in vector precoding,” in *IEEE Globecom 2006*, pp. 1–5, 2006.
- [17] S. Mukherjee and B. K. Chakrabarti, “Multivariable optimization: Quantum annealing and computation,” *The European Physical Journal Special Topics*, vol. 224, no. 1, pp. 17–24, 2015.
- [18] M. Mohaisen, B. Hui, K. Chang, S. Ji, and J. Joung, “Fixed-complexity vector perturbation with block diagonalization for MU-MIMO systems,” in *IEEE 9th Malaysia International Conference on Communications (MICC)*, pp. 238–243, IEEE, 2009.
- [19] J. Cai, W. G. Macready, and A. Roy, “A practical heuristic for finding graph minors,” *arXiv preprint arXiv:1406.2741*, 2014.