

X-ResQ: Reverse Annealing for Quantum MIMO Detection with Flexible Parallelism

Minsung Kim^{*,◊}, Abhishek Kumar Singh[◊], Davide Venturelli^{*}, John Kaewell[†], Kyle Jamieson[◊]

^{*}Yale University, [◊]Princeton University, [†]InterDigital, ^{*}USRA Research Institute for Advanced Computer Science

Abstract

Quantum Annealing (QA)-accelerated MIMO detection is an emerging research approach in the context of NextG wireless networks. The opportunity is to enable large MIMO systems and thus improve wireless performance. The approach aims to leverage QA to expedite the computation required for theoretically optimal but computationally-demanding Maximum Likelihood detection to overcome the limitations of the currently deployed linear detectors. This paper presents **X-ResQ**, a QA-based MIMO detector system featuring fine-grained quantum task parallelism that is uniquely enabled by the Reverse Annealing (RA) protocol. Unlike prior designs, X-ResQ has many desirable system properties for a parallel QA detector and has effectively improved detection performance as more qubits are assigned. In our evaluations on a state-of-the-art quantum annealer, fully parallel X-ResQ achieves near-optimal throughput (over 10 bits/s/Hz) for 4×6 MIMO with 16-QAM using six levels of parallelism with 240 qubits and 220 μ s QA compute time, achieving 2.5–5 \times gains compared against other tested detectors. For more comprehensive evaluations, we implement and evaluate X-ResQ in the non-quantum digital setting. This non-quantum X-ResQ demonstration showcases the potential to realize ultra-large 1024×1024 MIMO, significantly outperforming other MIMO detectors, including the state-of-the-art RA detector classically implemented in the same way.

1 Introduction

To satisfy ever-increasing demand on mobile traffic, the *multi-user Multiple-Input Multiple-Output* (MU-MIMO) technique with *spatial multiplexing* becomes an essential building block in modern wireless standards. To enable parallel data streams and thus achieve capacity gains in MU-MIMO systems, a signal processing technique called *MU-MIMO detection* is required to demultiplex mutually-interfering streams at the base station receiver into each user signal. Optimal *Maximum Likelihood* (ML) detectors ensure the best possible performance. Nonetheless, current MIMO systems make use of sub-optimal detectors, sacrificing tremendous potential gains. This is because the computational complexity of the ML detectors increases at an exponential rate with more users. To resolve the computational bottleneck in ML MIMO detection, researchers have started to explore non-traditional heuristic optimization processing [27, 43, 45, 65, 80]. *Physics-Inspired*

Computing (PIC) is a computing paradigm defined by computational principles that imitate laws of physics. For example, PIC algorithms such as *Quantum Annealing* (QA) [9, 36] and *Simulated Annealing* (SA) [60] mimic the convergence nature of thermal annealing to find the global optimum of discrete optimization problems.

Need of Parallelization Strategies in MIMO Detection. Unlike most optimization problems, there exist extremely tight *computation time deadlines* in MIMO detection, at most a few hundred microseconds, regardless of the difficulties of the problems. However, for difficult MIMO scenarios (*e.g.*, large user counts), the use of (near-)optimal detectors is encouraging [64, 98], but their required computations cannot be completed within the deadlines. Regarding this, parallel architecture-based detectors with *detection task parallelism* are promising in that large computation amounts can be split into multiple compute tasks to reduce overall compute times by solving them in parallel at the expense of more computation resources [34, 62]. While transistors' clock speed has stopped increasing rapidly with Moore's Law, the number of embedded transistors per chip keeps increasing nearly exponentially. With this trend, hardware that contains massive *processing elements* (PEs) including GPUs and FPGAs are actively explored in this direction with a variety of parallel MIMO detection algorithms [8, 28, 30, 34, 63, 74, 96]. In this regard, ParaMax [43], a recent classical PIC MIMO detector, parallelizes its PIC heuristics to collect more candidate solutions simultaneously (*i.e.*, sample parallelism) to reduce its overall latency without loss of performance.

We believe the same motivation holds for QA MIMO detectors, and thus designs of highly efficient and scalable parallelization strategies for QA MIMO detection need to be investigated in light of the expected large-scale qubit processors. Since 2011, every new generation of D-Wave quantum annealers has presented exponentially-increasing qubit counts as well as improved hardware connectivity: 128 qubits in 2011 (D-Wave One System) and 5000+ qubits in 2020 (Advantage System). Based on the extrapolation of historical records, over ten thousand qubits are expected in 2030 and hundreds of thousands in 2035 on a single QA machine [41] (likely in an interconnected multi-core setting), which are sufficient enough to cover both data and task parallelism in the physical layer processing. Therefore, parallelization strategies in QA MIMO detectors will become more essential. This paper investigates *quantum MIMO detection (optimization) task*

parallelism. Prior QA MIMO detector design QuAMax [45] discusses sample parallelism like ParaMax, but the strategy is not effective when optimization problems become difficult (§4.2). IoT-ResQ [47] applies decomposition-based parallelism to make problems easier with further parallelization by decomposing a problem into simpler subproblems, but it cannot support flexible parallelism due to its coarse granularity.

In this paper, we introduce *X-ResQ* (§4), a parallel QA MIMO detector system that supports flexible quantum task parallelism that is uniquely enabled by *Reverse Annealing* (RA) (§2.3), a method called *multi-seed parallel ensemble RA*. Unlike standard QA, RA starts its optimization operation from a classical state where its QA search tends to be localized around that state. *X-ResQ* applies multiple independent RA runs initiated from different initial classical states for its QA parallelization. Unlike prior designs, *X-ResQ* has many desirable features for a parallel QA system including fine-grained parallelism (Table 2), and has effectively improved performance as more qubits are assigned, providing an efficient trade-off between qubits and compute time. We describe its design principles based on somewhat unexpected experimental RA results (§4.3). We also present a split-detection scheme (§4.5) to mitigate the error floor phenomenon at high SNRs leveraging QA’s exceptional performance with B/QPSK and analyze its effect both theoretically and experimentally.

We implement and evaluate *X-ResQ* on the state-of-the-art D-Wave quantum annealer. In our evaluations, *X-ResQ* achieves near 10^{-4} uncoded BER performance for 4×6 MIMO with 16-QAM at SNR 20 dB using six levels of parallelism (240 qubits) and $220 \mu\text{s}$ QA compute time, resulting in near-optimal throughput for 1,500-byte packets (over 10 bits/s/Hz). The other parallel QA detectors tested on the same machine with the same qubits either still perform quite poorly or cannot be programmed on the hardware due to inflexible parallelism. Despite the relatively small sizes of enabled MIMO, we believe this is an important step in the direction in that enabling high-order modulations for QA MIMO detection has been constantly identified as an important challenge. For more comprehensive evaluations beyond QA hardware, we classically implement *X-ResQ* using a generic PIC algorithm, *parallel tempering* (§5). We observe the classical *X-ResQ* can potentially enable previously impossible ultra-large 1024×1024 MIMO, achieving BER below 10^{-7} with BPSK. For 256×256 MIMO with QPSK at SNR 14 dB, *X-ResQ* obtains around 10^{-7} BER, showing four to six orders of magnitude better BER than the other tested detectors, with $\approx 333 \mu\text{s}$.

2 Background

This section introduces background knowledge to understand the paper. Section 2.1 explains MIMO detection and Section 2.2 introduces QA and Section 2.3 describes its Reverse

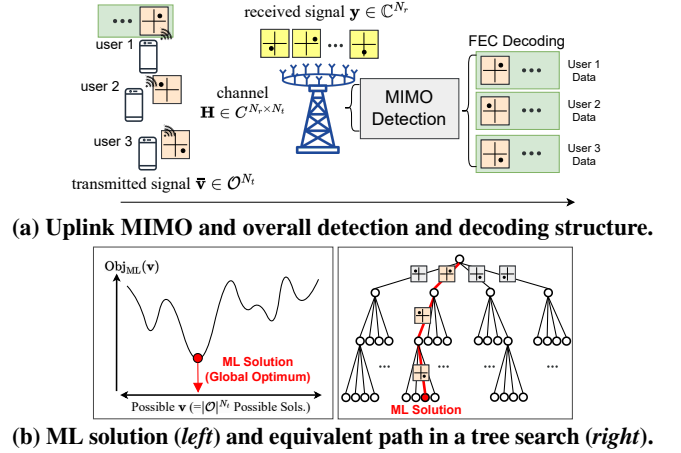


Figure 1: MIMO model and optimal ML solution.

Annealing protocol.

2.1 MIMO Detection

Figure 1(a) shows the uplink MU-MIMO model and overall detection and decoding structure, where N_t single-antenna mobile users are simultaneously sending their signals to a base station (BS) with N_r antennas ($N_r \geq N_t$). Each user sends out a wireless symbol $v \in \mathcal{O}$ that represents M bits (out of Forward Error Correction (FEC)-coded bits) based on a constellation \mathcal{O} with $|\mathcal{O}| = 2^M$ being the modulation size. Then, the received signal at the BS can be expressed as $\mathbf{y} = \mathbf{H}\mathbf{\hat{v}} + \mathbf{n} \in \mathbb{C}^{N_r}$, where $\mathbf{\hat{v}} \in \mathcal{O}^{N_t}$ is the transmitted signal vector, $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is wireless channel matrix, and $\mathbf{n} \in \mathbb{C}^{N_r}$ is a noise vector. We denote this model as $N_t \times N_r$ MIMO and all OFDM subcarriers hold the same model independently.

MIMO Detection. MU-MIMO detection is a signal processing through which the receiver generates an estimated signal $\hat{\mathbf{v}}$ (without \mathbf{n} , $\hat{\mathbf{v}} = \mathbf{\hat{v}}$) based on the received signal and estimated channel [11].¹ After the detection, the detected symbol vector $\hat{\mathbf{v}}$ is demapped into corresponding M bits, and then blocks of those bits across subcarriers are decoded by a channel decoder to reconstruct *error-free* data bits (per user), according to the applied FEC scheme (e.g., LDPC, convolutional coding).

Optimal ML Detectors. Optimal *maximum likelihood* (ML) detectors (e.g., Sphere Decoders [3, 16, 91]) can ensure the *best-possible* detection performance. The ML method detects $\hat{\mathbf{v}}$ among all possible $\mathbf{v} \in \mathcal{O}^{N_t}$ that minimizes the objective function $\text{Obj}_{\text{ML}}(\mathbf{v}) = \|\mathbf{y} - \mathbf{H}\mathbf{v}\|^2$ (i.e., $\hat{\mathbf{v}}_{\text{ML}}$ is global optimum). The ML detection problem can be translated into an equivalent tree search problem that aims to find the best path (from the root node to a leaf node) with the minimum cumulative metric among $|\mathcal{O}|^{N_t}$ possible paths in a perfect tree as shown in Figure 1(b). Each level corresponds to a user, while each

¹In downlink (from BS to users), MIMO precoding is a dual problem of MIMO detection.

branch decision is a symbol decision per user. While optimal, an ML detector is not practically feasible today, because of its exponentially increasing search size with higher N_t and/or $|O|$ and limited available processing time in wireless standards ($\approx 0.5 - 4$ ms) [18, 24, 100].

2.2 Quantum Annealing

Quantum annealers are analog devices designed to find the (near-)optimal solution of combinatorial optimization problems using quantum mechanical fluctuations with tremendous speedup potentials compared to classical computational resources and methods [9, 17, 49]. For the state-of-the-art D-Wave quantum annealers, input optimization problems are *Ising spin models* whose Ising energy (cost) functions E are:

$$E(\mathbf{s}) = \sum_i f_i s_i + \sum_{i,j} g_{ij} s_i s_j \quad (1)$$

where real-value f_i and g_{ij} represent the optimization problem we want to solve. The machines aim to find a spin *configuration* (or state) $\mathbf{s} = \{s_1, s_2, \dots, s_{N_V}\}$ that makes the Ising energy $E(\mathbf{s})$ minimized, where \mathbf{s} consists of N_V spins, with each spin variable s_i being either 1 or -1. The configuration that corresponds to the minimum Ising energy is called a *ground state* (global optimum). For an Ising model of ML MIMO detection, $N_V = N_t \log_2 |O|$ spins are required (§4.4).

QA Heuristics. The time evolution of *Hamiltonian* \mathcal{H} in a system enables QA heuristics to solve the input optimization problem, where (simplified) \mathcal{H} can be defined as:

$$\mathcal{H}(\tau) = A(\tau) \cdot \mathcal{H}_{\text{superposition}} + B(\tau) \cdot \mathcal{H}_{\text{problem } E}, \quad (2)$$

where τ is a time-dependent function ($\tau \in [0, 1]$). $A(\tau)$ represents the transverse energy that controls the strength of the *quantum fluctuations* (i.e., quantum-coherent exploration of the search space), while $B(\tau)$ is the energy that is meant to correlate these fluctuations to the energy function of the problem (Eq. 1). How to control τ as a function of wall-clock *anneal duration* (T_a) determines QA algorithms.

In the standard *Forward Annealing* or FA, the system initially applies strong quantum fluctuations and prepares a *quantum superposition state* that holds every possible configuration information (i.e., $A(\tau) \gg B(\tau)$ at $\tau = 0$). Then it drives the changes until the effect of the initial $\mathcal{H}_{\text{superposition}}$ diminishes (i.e., $A(\tau) \ll B(\tau)$ at $\tau = 1$), hoping that the driven non-equilibrium quantum thermodynamics process (which follows phenomenology from the adiabatic and quasistatic evolution [57]) will lead to a low-energy state (the ground state in ideal cases). At the end of the anneal protocol, the system can interpret the result as a spin configuration \mathbf{s} , being a *sample*. Since QA is a probabilistic technique, the end result is non-deterministic. Thus, multiple anneal iterations (N_a or *anneal counts*) are typically applied, leading to N_a samples. Among them, the best sample \mathbf{s} (i.e., $\min E(\mathbf{s})$) is filtered as the final optimization solution.

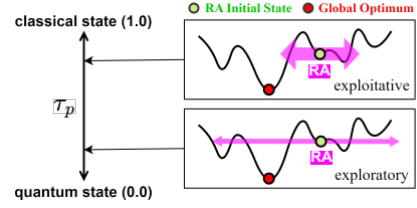


Figure 2: Optimization search of Reverse Annealing (RA) initiated from a classical state (cf. from a superposition state in FA).

2.3 Reverse Annealing (RA)

Reverse Annealing or RA [66] (X-ResQ’s optimization core) is a variation of QA whose initialization process is reversed (R) compared to the standard FA. For RA, the annealer initially programs “a classical state” on the hardware ($\tau = 1$) and then introduces quantum fluctuations by reducing τ to the *switching point* τ_p ($0 \leq \tau_p \leq 1$ e.g., $\tau_p = 0.4$ [46] or 0.6 [69]). When the process reaches τ_p , a superposition state is prepared with relatively weak fluctuations, which enables it to still hold the initial classical state information. After the optional pause (P), the rest of T_a follows the incremental increase of τ , like FA (F). Accordingly, during T_a in the RA protocol, the state of $\mathcal{H}(\tau)$ is changed as following, compared to FA:

- FA: quantum ($\tau = 0$) \xrightarrow{F} classical ($\tau = 1$)
- RA: initial classical ($\tau = 1$) \xrightarrow{R} initial intermediate ($\tau = \tau_p$) \xrightarrow{P} stable intermediate ($\tau = \tau_p$) \xrightarrow{F} final classical ($\tau = 1$).

This RA protocol promises to improve QA’s optimization performance by limiting quantum fluctuations around the initial state [13]. In doing so, RA works like a *refined local search*, where optimization search tends to be localized around the initial state based on the quasi-(non)local search [50], providing a chance of exploring a trade-off in searching between exploration and exploitation as shown in Figure 2.² In a rough way, FA can be considered a *wide* and *shallow* search (entire search space w/ scattered search power), whereas RA is a *narrow* and *deep* search (focused search space w/ concentrated search power). Therefore, providing a good initial classical state to RA is a key factor for optimization performance, which will be further discussed later in the paper (§4.3). It has been reported that RA could outperform FA for various applications [23, 46, 50, 69, 89].

3 Motivation and Related Work

3.1 MIMO Systems with Linear Detector

Linear detectors, such as *Zero-Forcing* (ZF) and *Minimum Mean Square Error* (MMSE) methods, feature simple processing, thus being commonly used in experimental wireless

²When τ_p is too far from the initial $\tau = 1$ (i.e., τ_p close to 0), the information related to the initial state would be wiped out with strong fluctuations (i.e., no effects of using the initial state). On the other hand, when τ_p is too close to the initial $\tau = 1$, the strength of quantum fluctuations is not enough to trigger sufficient superposition (i.e., no effects of quantum search).

Table 1: Experimental Massive ($N_t \times N_r$) MIMO systems.

MIMO System	MIMO Detector	MIMO Config.	N_r/N_t (BS ant. / user)	System Imple.
Agora [18]	linear ZF	16×64	4	software
Argos [79]	linear ZF	16×64	4	hardware
BigStation [100]	linear ZF	16×128	8	software
Hydra [24]	linear ZF	32×150	4.69	software
LuMaMi [55]	linear ZF	12×100	8.33	hardware

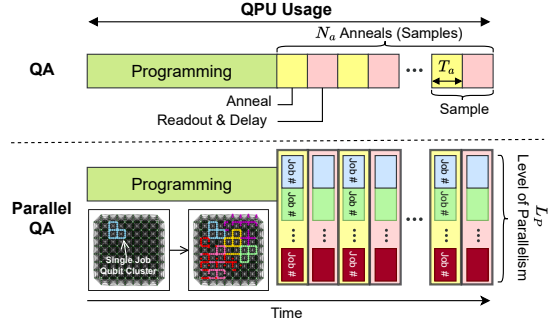
systems as well as practical cellular systems. However, when the concurrently-served user count (N_t) approaches the given BS antenna count (N_r), their detection suffers from large *noise amplification* (Appendix A), resulting in poor detection performance [64]. For this reason, many systems use a *massive MIMO* regime where the BS with massive 64–128 antennas supports only a few users at a time (*i.e.*, $N_r \gg N_t$); Table 1 shows examples of state-of-the-art Massive MIMO systems with their N_r/N_t . However, the gain in spectral efficiency and MIMO capacity is proportional to N_t , and blindly increasing N_t under the given N_r (relying on FEC) is not effective, since the *fundamental BER performance* of linear detectors will be an eventual bottleneck, resulting in significant errors and thus tremendous time and signaling overheads. To keep scaling up N_t towards $N_r/N_t = 1$, a regime called *Large MIMO* (cf. Massive MIMO) for the ideal gains, (near-)optimal ML performance is necessitated [34], which is the case for (small N_r) WLAN scenarios as well. However, enabling ML detectors for large-scale MIMO has a complexity issue (see §2.1).

3.2 PIC for Wireless Processing

From the perspective of non-traditional PIC algorithms, various specialized devices have also emerged such as digital annealers [5], analog quantum annealers [17, 49], bifurcation machines [88], digital quantum gate-model computers [6, 61], memristor and spintronic Ising machines [10, 25, 84], and oscillator-based, photonic, and optical coherent Ising machines [29, 35, 59, 72, 94, 99], targeting the ability to accelerate computing of solving combinatorial optimization problems, leading to new areas to explore. In wireless networks, despite its infancy, the PIC approach has already shown great potential in many different modules such as MIMO precoding [40, 97], MIMO beam selection and satellite beam placement [19, 33], error control coding [38, 39, 78], intelligent meta-surfaces and discrete phase shifting [44, 53, 76], and scheduling [90]. Particularly for MIMO detection, extensive research efforts have been made to expedite the computation required for optimal ML MIMO detection processing [14, 15, 43, 56, 65, 80, 82, 83, 87].

3.3 Challenges in QA MIMO Detction

Through prior studies [20, 45, 47, 56, 86], the promise of QA MIMO detectors has been experimentally demonstrated with great potential of accelerating ML processing. However, their challenges have been identified as well, which are rather


Figure 3: Machine QPU operation for QA and parallel QA.

different from the ones in conventional MIMO detectors.³

(1) Severe performance degradation with high-order modulations such as 16/64-QAM is commonly observed in PIC/QA-based MIMO detection. With low-order modulations such as BPSK and QPSK, large MIMO (like 60×60) with near-optimal performance has been successfully reported within a few hundred microseconds of QA compute time, whereas 16-QAM MIMO problems even with only a few users (like 3×3 MIMO) are quite challenging, requiring several milliseconds.

(2) A BER *error floor* is observed at high SNRs; BER flattens despite increasing SNRs. When detection BER at the point of the error floor is not low enough, FEC decoding could result in errors. Then, data retransmission is required (overheads).

To mitigate these challenges, X-ResQ tweaks QA optimization using a *parallelization* strategy uniquely enabled by RA.

3.4 Parallel QA Optimization

Parallelization of QA is an efficient approach to further accelerate QA optimization or boost the overall QA system performance by using more qubits [32, 70, 71]. Figure 3 shows the machine operation for (non-)parallel QA, consisting of hardware programming, annealing, and readout [1]. In parallel QA, each anneal handles L_p independent (same or different) jobs/tasks simultaneously on different qubit clusters, where L_p is the *level of parallelism*. With *fully parallel* QA that does not require any iterative QPU operations (cf. anneal iterations), the system can avoid multiple occurrences of inevitable overheads related to the considerable programming time.⁴ Thus, parallel QA will become essential for latency-intensive applications like MIMO detection; for example, multi-stage approaches [37, 43, 81] do not work for QA MIMO detectors due to the aforementioned tight deadline.

³These challenges are present due to multiple reasons having to do with fundamental spin-glass physics bottlenecks [52], finite temperature fluctuations [4], and the effect of analog parameter misspecification [68]).

⁴The programming time on the current machine is several milliseconds, while the readout with delay is around $25 - 150 \mu s$ per sample. However, they are being reduced exponentially every generation, and specific techniques to make them within a few (tens of) microseconds have been already identified [26, 73, 92, 95]. We discuss X-ResQ's potential practicality in §7.

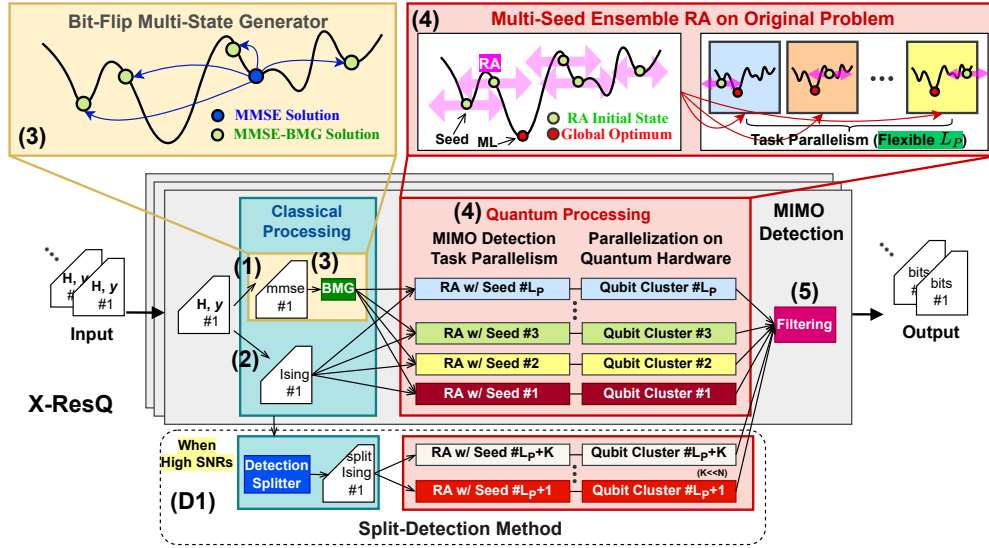


Figure 4: System Architecture of X-ResQ (multi-seed parallel ensemble RA). X-ResQ is based on the currently-deployed MMSE detector, requiring only simple preprocessing. Unlike IoT-ResQ, X-ResQ can support flexible parallelism with fine granularity for any L_P , where all the tasks can converge to the ML solution, thus increasing the ML fidelity through sample parallelism.

While data parallelism is quite straightforward (*e.g.*, across subcarriers) [20], in this paper we are interested in *task parallelism* to further accelerate detection, which has been barely studied so far in the context of PIC methods like QA.

A straightforward parallel QA method is *sample parallelism* that programs/embeds the “same” problem multiple times on QA hardware [32, 70]. This approach aims to collect more samples per anneal (total sample counts = $L_P \cdot N_a$), thus achieving the (linear) speedup across used qubits to obtain the target sample count. QuAMax [45] (and classical ParaMax [43]) applies this technique with FA to solve Ising spin models of ML MIMO problems based on the available qubit counts (without actual implementations). Parallel ensemble QA [7] has also been studied, where multiple Ising models are generated by adding controlled noises to the original one to overcome the system bias, but the method also uses FA and is not tested by actual parallel QA runs. *Decomposition-based parallelism* is a task parallelization strategy commonly used in both classical [8] and QA [71] optimization. Its typical idea is to decompose an original problem into multiple “easier” subproblems (by prefixing some hard variables via *full search*) and process the resulting subproblems in parallel.

IoT-ResQ [47] is a decomposition-based parallel QA MIMO detector that consists of Fixed-complexity Sphere Decoder (FSD) [8] and RA. For detection, IoT-ResQ first transforms an ML problem into an equivalent tree search (Figure 1(b)) and conducts the full search for the initial N_{fs} levels in the tree (*i.e.*, symbol full expansion for N_{fs} users), resulting in $|O|^{N_{fs}}$ subproblems (subtrees with reduced $N_t - N_{fs}$ levels). Then, decomposed subproblems are solved by a greedy search

(so far, FSD), followed by RA optimization initialized by the corresponding greedy search solutions, all in parallel. The overall architecture of IoT-ResQ is available in Appendix F. We discuss limitations of QuAMax and IoT-ResQ in §4.2.

4 Design

In this section, we describe X-ResQ. Section 4.1 introduces X-ResQ’s overall architecture and operation. In Section 4.2, we explain its properties, highlighting the differences against those of other QA MIMO detectors. Section 4.3 discusses the design rationale, and Section 4.4 explains preprocessing for multi-seed parallel RA. Section 4.5 introduces the split-detection method to mitigate the error floor at high SNRs.

Quantum-based C-RAN. Envisioned scenarios for X-ResQ (and other QA MIMO detectors) assume *Centralized Radio Access Networks* (C-RAN) with QA processors co-placed in a cloud or edge *data center* along with existing classical processors, separating their computing roles [48]. The quantum-based C-RAN resolves many practical issues (*e.g.*, on cryogenic temperatures for operations and machine access overheads). More discussions on QA feasibility in wireless cellular networks are available in Section 7.

4.1 X-ResQ Architecture Overview

X-ResQ is a parallel QA-accelerated MIMO detection system that utilizes multi-seed ensemble RA as its task parallelism. Its parallelization strategy is designed to maximize RA’s advantages and thus further expedite quantum ML detection efficiently with more qubits. Figure 4 shows the overall system architecture and operation of X-ResQ. For MIMO detection, X-ResQ’s structure has the following stages:

(1) Initial MMSE Detection. X-ResQ initially applies MMSE detection based on the received signal and estimated channel. This stage has very low complexity as the MMSE detector computes its equalizer only once per channel coherence time (remaining same for several milliseconds) and the detection operation for instances (*i.e.*, channel uses) involves just matrix-vector multiplications.

(2) ML Ising Model Generation for Parallel Ensemble QA. At the same time, the system prepares an Ising model consisting of multiple ML Ising forms for parallel QA (§4.4). During the channel coherence time, only f_i (in Eq. 1) needs to be updated per channel use based on generalized forms [45], requiring also only insignificant matrix-vector multiplications.

(3) Multi-Seed Generation. Based on the MMSE solution, a *bit-flip multi-state generator* (BMG) brings out multiple L_P different solutions/states by flipping a single bit on it (§4.4).

(4) Multi-Seed Parallel Ensemble RA. Then, the system makes use of them as multi-seed initialization states, each of which becomes an initial state for each RA run. X-ResQ applies RA to all L_P runs in parallel (all with the original Ising form). We call this structure and operation, *multi-seed ensemble RA*. Unlike prior designs, with this simple parallelization strategy, X-ResQ features many desirable properties for parallel QA systems, as described in the next subsection.

(D1) Split-Detection (for high SNRs). When SNRs are high, X-ResQ generates an additional Ising form based on the *split-detection approach* described in §4.5, which is designed to mitigate QA detectors’ error floor at high SNRs (§3.3). This new Ising form consists of two parts: *quadrant search* and *position search* in the quadrant. While these two are jointly considered in the original Ising form, they are considered *separately* in X-ResQ’s split-detection Ising form by transforming the original problem into simpler B/QPSK problems.

(5) Filtering as Post-Processing: X-ResQ accumulates all $L_P \cdot N_a$ samples generated by all RA runs, and filters the best one that has the minimum energy (Eq.1) as the final solution, which will be translated into detected bits.

Table 2: Comparison of QA MIMO detectors (Section 4.2)

QA MIMO Detectors	QA Algorithm	Flexible Parallelism	Classical Pre-Process	Sample Parallelism	Easier Sub-problems
QuAMax	FA	yes	light	yes	no
IoT-ResQ	RA	no	heavy	no	yes
X-ResQ	RA	yes	light	yes	yes

4.2 X-ResQ System Characteristics

X-ResQ features many desirable properties for a parallel QA MIMO detector, also satisfying all the requirements for *ideal* parallel MIMO detectors [62].⁵ The comparison of QA

⁵X-ResQ does *not* satisfy “being transparent to the implementation technology” (fifth requirement in [62]) due to QA. However, note that we also demonstrate non-quantum X-ResQ (§5) that is based on a generic PIC algorithm that can be implemented on any platform.

Table 3: The required number of qubits for 16-user fully parallel QA MIMO detection operations with QPSK ($|O| = 4$) on Zephyr-topology QA hardware (§5) are shown.

	Available L_P	Required Qubits
IoT-ResQ	4 / 16 / 64 / 256 / 1024 / ...	480 / 1,792 / 6,656 / 18,432 / 67,584 / ...
X-ResQ, QuAMax	1 / 2 / 3 / 4 / 5 / ...	128 / 256 / 384 / 512 / 640 / ...

MIMO detectors (QuAMax, IoT-ResQ, X-ResQ) is summarized in Table 2, and we discuss the characteristics one by one in this subsection.

QA Algorithm. In the context of MIMO detection, unlike FA (used in QuAMax), RA (§2.3) can provide opportunities for classical-quantum *hybrid* optimization, where RA heuristically corrects the initial classical detector’s (non-ML) solutions into the optimal ML solutions. This hybrid structure (also known as *warm-started* quantum optimization)⁶ provides not only improved QA detection performance over FA [46, 47], but also a chance of *opportunistic* quantum optimization. Specifically, in the case of X-ResQ comprised of linear MMSE and RA, RA quantum optimization can be skipped for MIMO scenarios where MMSE performs well, and be applied only when MMSE often fails to find the ML solution (*e.g.*, user/traffic peak times). For these reasons, we argue that RA is a more pragmatic QA algorithm than FA in MIMO detection. Furthermore, X-ResQ’s design does not have to significantly change the current wireless systems, since it relies on the currently deployed MMSE detector as an initial classical detector (cf. non-linear FSD in IoT-ResQ).

Flexible Parallelism with Fine-Granularity. Available levels of parallelism (L_P) in X-ResQ are *any* number (until hardware limits). Thanks to this flexibility, X-ResQ can achieve detection performance gains *elastically* with fine-grained parallel QA, thus requiring fewer L_P (*i.e.*, fewer qubits) to obtain (near-)optimal performance. IoT-ResQ cannot support this flexible parallelism; because of the symbol full expansion (§3.4), available L_P is limited to the power of the modulation size (*i.e.*, $L_P = |O|^1, |O|^2, \dots$). For example, with 16-QAM, IoT-ResQ cannot achieve any gains between $L_P = 16^1$ and 16^2 , which implies coarse-grained parallelism with inflexible and inefficient qubit usage (see Table 3 for comparisons).

Classical QA Preprocessing. Classical preprocessing is light in X-ResQ because of the linear MMSE. Furthermore, MMSE detection and Ising formulation can be processed in parallel, and the time required for the bit flip-based BMG is negligible. Also, X-ResQ requires only quantum parallelism, while IoT-ResQ requires parallelism in both classical and quantum processing parts. This implies classical resources can be used for data parallelism (*e.g.*, subcarriers) in X-ResQ.

Sample Parallelism. Due to the use of the original Ising form

⁶This hybridization also applies to gate-model quantum processors [21].

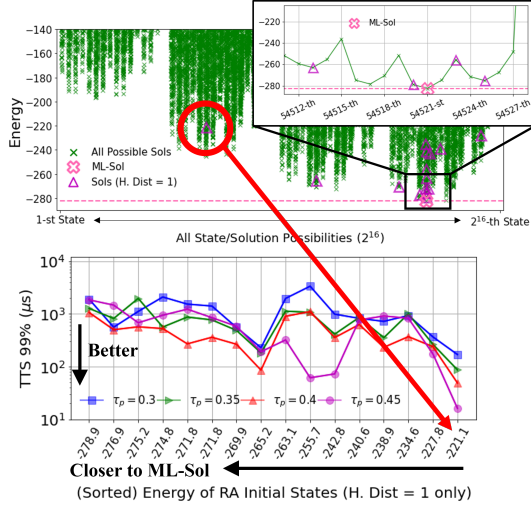


Figure 5: TTS Analysis of RA that is initialized from H. Dist=1 initial states using a 4×4 MIMO detection instance ($N_V = 16$) at SNR 20 dB. It demonstrates initial states that have lower Ising energies do not necessarily result in better RA results.

for parallel tasks, all the jobs can converge to the ML solution in X-ResQ (*i.e.*, sample parallelism in §3.4). In terms of resource usage, this is a desirable feature in that there are no qubits used on searches that never reach the ML solution. Furthermore, given that QA is a probabilistic technique, sample parallelism not only accelerates optimization convergence but also increases the fidelity of getting the ML solution by collecting more samples: the optimum probability is $1 - (1 - P_G)^{\text{sample count}}$, where P_G is the probability of finding the ML solution per sample. In the case of IoT-ResQ, sample parallelism is not supported, since all parallel QA jobs solve different subproblems because of the decomposition nature, where among all the subproblems, *only one* entails the ML solution (*i.e.*, $P_G = 0$ for the others). While this is inevitable for the approach, its optimum probability could be low, despite slightly increased \hat{P}_G in the correct subproblem⁷ with the reduced search space ($2^{N_V} \rightarrow 2^{N_V - N_{fs} \log_2(|O|)}$).

Making Problems Easier with Parallelization. As discussed, IoT-ResQ makes a detection problem easier by decomposing it into simpler subproblems (*i.e.*, impact on P_G and thus TTS (§4.3) across L_P). QuAMax’s sample parallelism does not have the capability because of the same Ising form used for parallel tasks (*i.e.*, same P_G across tasks); when detection scenarios become challenging where most of the anneal trials fail ($P_G \approx 0$), it achieves nearly no effects. On the other hand, X-ResQ has an impact on P_G across L_P , even with

⁷To satisfy $1 - (1 - P_G)^{N_a \cdot |O|} < 1 - (1 - \hat{P}_G)^{N_a}$, the gain (\hat{P}_G/P_G) has to be high enough (*e.g.*, over 8 for $P_G = 0.1$ w/ 16-QAM). However, the (FSD-based) decomposition method has not been quite effective for QA MIMO detection [42], and surprisingly we experimentally found that it often results in adverse effects, particularly for $N \times N$ Large MIMO (see Appendix D).

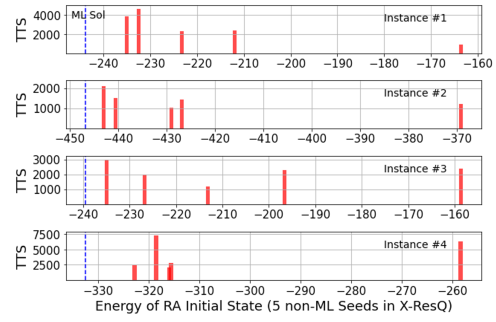


Figure 6: TTS 99% (μs) of fully parallel X-ResQ ($L_P = 5$) for 4×4 MIMO detection at SNR 20 dB. Each task out of the five tasks is parsed for TTS analysis from parallel X-ResQ results. A similar insight as Figure 5 is observed for fully parallel runs.

the same Ising form. In X-ResQ, RA is initiated by a different state per task. In doing so, a similar effect of solving a different-difficulty optimization problem is caused, since RA’s optimization search tends to be localized around the initial state (quasi-nonlocal search [50]), thus achieving better performance with higher L_P . We discuss this further in the next subsection with the rationale of X-ResQ’s strategy, showing its experimental validation.

4.3 Rationale of Parallelization Strategy

In this subsection, we explain the rationale for the X-ResQ’s parallelization strategy (*i.e.*, multi-seed ensemble RA).

QA Benchmark Metric: Time-to-Solution. We use a commonly accepted metric for QA performance microbenchmarks, *time-to-solution* (TTS) [75]. TTS indicates the estimated required time (μs) to find the global optimum in an optimization problem (*i.e.*, ML solution in MIMO detection) with *target confidence* (typically 99%): $\text{TTS}(99\%) = T_a \cdot \log(1 - 0.99) / \log(1 - P_G)$, where P_G is empirically obtained (out of 5,000 samples in this paper). Lower TTS typically implies better QA optimization performance.

Recall that RA optimization performance depends on its classical initial state (§2.3). Thus, we delve into RA to identify high-quality initial states that trigger better RA optimization (lower TTS). There have been some attempts to correlate the quality to initial states’ Hamming distance (H. Dist) or energies against the global optimum [23, 46, 67]. Since H. Dist is unknown information in MIMO detection, we analyze the impact of energies of initial states.⁸

Figure 5 shows all possible states and the 16 possible states that have H. Dist 1 against the ML solution with their corresponding energies for a single, illustrative 16 spin-variable detection instance at SNR 20 dB (*upper*). Then, we plot their TTS performance of RA optimization (*lower*) with these

⁸In [47], the correlation between H. Dist and Ising energies of possible states is observed to some extent, but it is only limited to low-order modulations.

H.Dist-1 states being an RA initial state, for different switching points τ_p (§2.3). **We observe that the states of lower energies do not necessarily result in better RA optimization, regardless of τ_p choices** (for QPSK as well). Except for $\tau_p = 0.45$, two states are considered good initial states (one at $E = -265.2$ and the other at $E = -221.1$). Interestingly, the latter is the state with the highest energy among them. We verify this unexpected outcome with *fully parallel* X-ResQ runs as well (with states generated by BMG). Figure 6 plots TTS of X-ResQ’s RA with four random instances, for each of which we parse the parallel run results into each task and analyze its TTS. It is also observed that initial states that are closer to the global optimum (ML Sol) do not necessarily result in better RA performance. For all the instances, *unpredictable* states turn out to result in the best TTS (among them), sometimes showing even large TTS gaps against the others.

These surprising results led us to three following conclusions that X-ResQ’s design derives from: (1) There are currently unknown (but existing) factors that determine the quality of initial states in RA. (2) Leveraging multiple initial states could improve the overall RA performance due to this ambiguity. (3) Thus, for parallel tasks, assigning qubits to multiple independent RA runs that are initialized by different states could be more efficient than focusing all resources on an RA run with a single initial state (unless we can precisely define the factors and hence better initial states). Intuitively, by avoiding using a single state, X-ResQ can prevent its optimization search from being stuck at the same local minima that is hard to escape from and make better use of the effect of RA’s quasi-(non)local search with distributed initial points in parallel.

4.4 Generating Ising Models and Initialization States for Parallel Ensemble RA

For QA MIMO detection, Ising models of the ML problems need to be prepared. Then, a QA solver applies a QA algorithm to solve the generated Ising models whose ground states correspond to optimal ML solutions.

Ising Spin Models of ML Problems. ML problems can be transformed into equivalent Ising models with *linear mapping* between possible wireless symbols and spin variables [45]. Per instance, $N_V = N_t \log_2(|O|)$ spins are required, and the resulting ML Ising models are (nearly) fully-connected (*i.e.*, non-zero g_{ij} in Eq. 1). The objective function of ML (*i.e.*, $\text{Obj}_{\text{ML}}(\mathbf{v})$ on p. 2) is expressed as an Ising energy function (Eq. 1) by one-by-one mapping, where $E(\mathbf{s})$ corresponds to $\text{Obj}_{\text{ML}}(\mathbf{v})$ with $\mathbf{s} \leftrightarrow \mathbf{v}$ (Figure 1(b)). Based on generalized forms, all f and g values can be obtained independently in parallel for any MIMO sizes and modulations.

Combined Model for Parallel QA. In QA, only a single Hamiltonian \mathcal{H} (Eq. 2) can be run (with a single Quantum Machine Instruction or QMI) [1]. Thus, for parallel QA, we

need to prepare \mathcal{H} with a single $\mathcal{H}_{\text{problem } E}$ that contains all parallelization information. In X-ResQ, the same Ising model $E(\mathbf{s})$ is used for parallel QA. Simply adding the same one to the end of one another leads to a single larger Ising model. As an example, we consider three levels of parallelism for a N_V -variable Ising model: *combined* $E = E(\{s_1, \dots, s_{N_V}\}) + E(\{s_{N_V+1}, \dots, s_{N_V+N_V}\}) + E(\{s_{2N_V+1}, \dots, s_{2N_V+N_V}\})$, requiring $3N_V$ variables, whose ground state is the same as the three-time repeated ground state of the original E . Despite the combined Ising spin model, different qubit clusters (near or far) on QA hardware can be assigned to each of them.

Bit-Flip Multi-State Generator (BMG). Based on the MMSE solution, BMG generates multiple states as seeds for RA initial states by flipping one bit (in default). While there are many possible variations in selecting which bit to flip, we use a random (but non-overlapped) index for the minimum overhead. Note that this randomness is a reasonable choice according to our design principle (§4.3). The generated multiple states are prepared as a single appended array, since a single initialization state can be run for RA, like the Hamiltonian.

4.5 Split-Detection Method

X-ResQ’s split detection scheme is designed to mitigate the error floor phenomenon with high-order modulations at high SNRs by exploiting the exceptional performance of QA MIMO detectors with low-order modulations such as BPSK and QPSK (§3.3). The method transforms the original ML Ising problem with high-order modulations into multiple simpler B/QPSK problems based on the initial MMSE solution.

For 16-QAM ML Ising models, four spin variables are required to represent an ML variable: two spins are related to the quadrant decision, while the other two spins are related to the position decision. Using these separate detection roles of the spin variables along with the MMSE solution, in the split-detection we generate an Ising form of N_V spins (*same size* as the original form) that contains two independent QPSK Ising ML problems. In one problem, we decide the quadrant (based on the position of the MMSE solution), and in the other, we decide the position (based on the quadrant of the MMSE solution). In the original Ising form, these two are *jointly* considered, but in the split detection method, they are considered *separately* as two independent *simpler* problems. The scheme can be easily generalized for any modulation size, requiring the bare minimum additional forms and overheads (*e.g.*, only two for 16-QAM, three for 64-QAM, and so on). Further, multi-seed parallel ensemble RA can be applied to them together with the baseline X-ResQ’s Ising representation (§4.4) without requiring any iterative QPU runs.

Through multiple applications of Cauchy-Schwartz inequality to find an upper bound on the effective noise, we also theoretically prove that the method can effectively improve the performance at high SNRs. However, note that the scheme

does not work well at low SNRs where both MMSE's quadrant and position are often wrong. The analytical proof of the method with MMSE analysis is available in Appendix B.

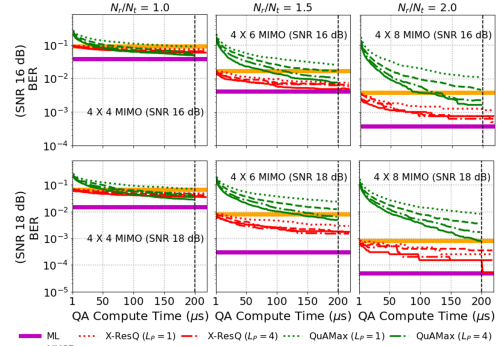
5 Implementation

Quantum Annealers. We implement X-ResQ on the state-of-the-art D-Wave Advantage Annealers. The *prototype* of Advantage2 System with Zephyr topology with 20 couplers per qubit [2] (1.1ver. released in 2022) is used because of the observed QA improvement for 16-QAM performance over the previous Pegasus-topology systems [47]. Despite the most advanced topology, the prototype machine has only 576 qubits (cf. previous Pegasus Advantage_6.1 has 5,760 qubits with 15 couplers), and has been very recently upgraded to 1,200 qubits (2.2ver. released in 2024). A 7,000 qubits generation is planned to be released by the end of 2025. All the QA experiments for comparisons among different QA detectors are conducted on the exactly same machines and Ising forms; We also implement fully parallel QuAMax using the same physical qubits for the same L_P as X-ResQ. More details on QA hardware programming and implementations including embedding benchmarks are in Appendix C. Note that X-ResQ is one of the first reports of fully parallel RA.

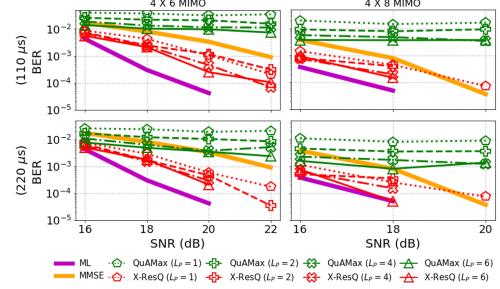
Classical Implementation with Parallel Tempering. To test large-scale parallelism and more comprehensive MIMO scenarios beyond QA hardware, we also implement classical variants of X-ResQ and IoT-ResQ using *Parallel Tempering* (PT) [85], a generic PIC algorithm that the state-of-the-art classical PIC MIMO detector, ParaMax [43], utilizes. Inspired by quantum RA (§2.3), we initiate PT from a given state (instead of a random state) for classical IoT-ResQ and X-ResQ, and apply different parallelization and initialization strategies corresponding to ParaMax, IoT-ResQ, and X-ResQ. They feature the same compute time for their PIC optimization processing due to the same PT engine, except for their preprocessing including their initial detectors (*i.e.*, FSD in IoT-ResQ and MMSE in X-ResQ). Unlike the QA implementations, each parallel task collects only one sample for the minimum latency here, so L_P is equal to collected sample counts. Classical experiments are executed on an Intel i9-9820X.

6 Evaluation

In this section, we evaluate X-ResQ's MIMO detection performance. In Section 6.1, the baseline X-ResQ is tested and compared against QuAMax (FA), MMSE (linear), and ML detector (optimal). Pure QA compute times for detection are calculated as anneal counts multiplied by each anneal duration (QuAMax: $N_a \times 2 \mu s$, X-ResQ: $N_a \times 2.2 \mu s$). In Section 6.2, we conduct more comprehensive evaluations with various classical and PIC-based detectors. X-ResQ and IoT-ResQ are implemented classically using a PT technique (§5), allowing for experiments with large-scale parallelism and various



(a) BER across QA time at SNR 16 dB (upper) and 18 dB (lower).



(b) BER across SNRs at QA time 110 μs (upper) & 220 μs (lower).

Figure 7: BER performance of fully parallel X-ResQ (requiring approx. $40 \cdot L_P$ qubits) for 4-user MIMO varying L_P , BS antenna counts N_r , and SNRs. IoT-ResQ that requires 768 qubits even for its minimum parallelism ($L_P = 16$) cannot be programmed on the state-of-the-art Zephyr-topology QA machine.

MIMO scenarios that current QA hardware cannot support.

Wireless Channel and Ising spin models. Our default channel setting is (i.i.d) Gaussian channels. We also use NVIDIA's open-source library Sionna [31]. Sionna generates OFDMA channel traces (Ray tracing channel) corresponding to 3GPP Urban Macrocell (UMa) channel model (link-level simulations instead of a stochastic model). We use a single-sector topology with users moving around at 3 m/s and a base station operating at 2.4 GHz with horizontally polarized antennas arranged in a linear array. Using the channels with random data and noises, we formulate up to 100,000 ML Ising models per scenario (N_t , N_r , SNRs, and modulations); relatively fewer (thousands of) instances are tested for QA experiments.

6.1 Detection Performance of X-ResQ

For QA MIMO detection evaluations, we test Ising ML detection instances with 100 anneals to make compute time at most few hundred microseconds (up to 220 μs). Due to the limited QA hardware, we consider a 4-user MIMO X-ResQ for WLAN scenarios ($N_r = 4, 6, 8$) with convolutional FEC coding, while for cellular network scenarios ($N_r = 16, 64$), we use cumulative sequential QA to estimate fully parallel performance.⁹ We report BER of MIMO detection (fundamental

⁹In Appendix C, we empirically verify that cumulative QA runs can estimate the performance of a fully parallel QA run (as an upper bound performance)

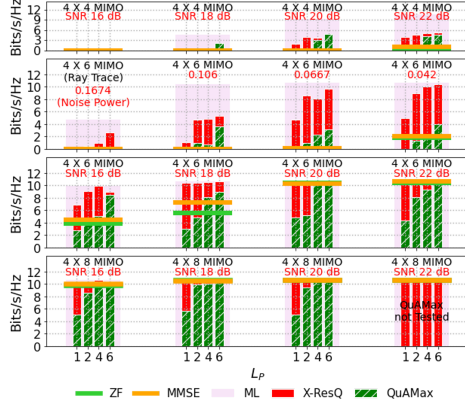


Figure 8: Throughput performance (bits/s/Hz) of fully parallel X-ResQ (220 μ s) with adaptive convolutional FEC (w/ 1/3, 1/2, 2/3 rate) for 1500-byte packets. QA detectors are with $N_a = 100$.

metric) and system throughput in bits/s/Hz (or empirically obtained spectral efficiency) using an adaptive coding scheme that selects the best code rate among $\frac{1}{3}$, $\frac{1}{2}$ and $\frac{2}{3}$, given SNRs with the convolutional code for 1500-byte packets. In this subsection, IoT-ResQ is not reported, since IoT-ResQ even with its minimum L_p cannot be programmed on the hardware. More comprehensive evaluations (including direct comparisons against IoT-ResQ) are in the next subsection.

First, we compare *fully parallel* X-ResQ (without the split detection method) against QuAMax. For fair comparisons, we reimplement QuAMax on the same Zephyr-topology prototype machine used for X-ResQ. Figure 7 shows BER for 4-user MIMO with different receiver antenna counts (N_r) and varying applied levels of parallelism ($L_p : 1, 2, 4, 6$).¹⁰ Figure 7(a) plots BER as a function of compute time (by translating N_a into time using each anneal duration) at SNR 16 dB (*upper*) and 18 dB (*lower*). Both QuAMax and X-ResQ have generally better BER as higher L_p is applied. As N_r increases, it is clearly observed that X-ResQ can outperform QuAMax and MMSE, reporting over an order of magnitude better BER for the same L_p . For example, for 4×8 MIMO at SNR 18 dB, while X-ResQ reports below 10^{-4} (optimal) BER with $L_p = 6$ around 200 μ s QA compute time, while QuAMax and MMSE obtain about 10^{-3} BER. Note that X-ResQ with $N_a = 50$ (110 μ s) works better than QuAMax with $N_a = 100$ (200 μ s) in most scenarios. Figure 7(b) plots BER across SNRs at fixed computing time with $N_a = 50$ and 100, *i.e.*, 110 and 220 μ s for X-ResQ, while 100 and 200 μ s for QuAMax. We also observe that X-ResQ outperforms QuAMax (and MMSE) and tends to achieve lower BER as more

without significant performance gaps for these levels of parallelism.

¹⁰Rarely, BER becomes higher in *better* scenarios (*e.g.*, higher SNRs, L_p , and/or compute times). This is because QA is a probabilistic heuristic solver and lower-energy states do not always correspond to lower-bit-error states, particularly when the energies of the best solutions are high (*i.e.*, bad quality).

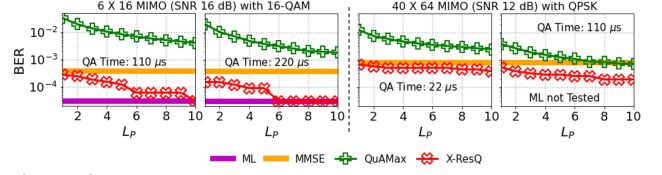


Figure 9: BER of cumulative sequential X-ResQ (cf. fully parallel X-ResQ) across L_p for 6×16 MIMO w/ 16-QAM and 40×64 MIMO w/ QPSK, whose required qubit counts for fully parallel QA would be $96 \cdot L_p$ (Zephyr) and $564 \cdot L_p$ (Pegasus), respectively.

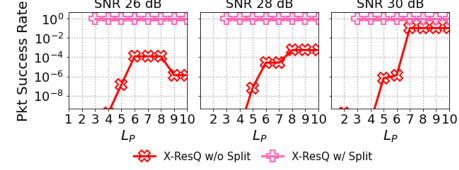


Figure 10: Impact of the split-detection method on the packet success rate of X-ResQ (220 μ s) at high SNRs for 4×4 MIMO with 16-QAM, where $L_p = 2$ is assigned to the split scheme.

qubits are assigned; X-ResQ with $L_p \geq 4$ obtain *no bit errors* for 4×6 MIMO at SNR 22 dB (out of 28,800 tested bits) and 4×8 MIMO at SNR 20 dB (out of 27,040 tested bits).

In Figure 8, we show system throughput performance with 16-QAM, using 100 random packets varying L_p , SNRs, and N_r . Note that linear detectors (ZF, MMSE) are not designed for parallelism, so their detection performance is the same, regardless of L_p . As expected, they rely heavily on both N_r/N_t and SNRs. As N_r/N_t or SNR decreases (from *bottom* to *top*, from *right* to *left*, respectively), their performance rapidly degrades; for 4×4 MIMO ($N_r/N_t = 1$), their throughput becomes almost zero. In the case of the QA MIMO detectors (X-ResQ, QuAMax), while they both generally achieve higher throughput with higher L_p (*i.e.*, more qubits) for the tested scenarios, X-ResQ shows more efficient parallelism, requiring less L_p to reach the (near-)optimal throughput. Interestingly, X-ResQ greatly outperforms the others, particularly with the Ray Trace channel model. For example, with noise power of 0.042 (far right on the second from top), X-ResQ with $L_p = 6$ reaches near-optimal performance (over 10 bits/s/Hz), achieving 2.5–5x throughput compared to the other comparison schemes including QuAMax with the same L_p .

Next, we test the detectors for scenarios with relatively larger MIMO sizes. For this, we estimate parallel X-ResQ performance through cumulative sequential results due to the problem sizes. Figure 9 shows the BER performance as a function of L_p , where X-ResQ keeps outperforming QuAMax for the same L_p for both 16-QAM (*two on the left*) and QPSK (*two on the right*) experiments. For 40-user QPSK MIMO, we implement X-ResQ and QuAMax on the Pegasus-based Advantage system (cf. Zephyr-based Advantage2). For 6×16 MIMO with 16-QAM, X-ResQ requires $L_p = 10$ (960 qubits) to achieve optimal BER with 110 μ s QA time ($N_a =$

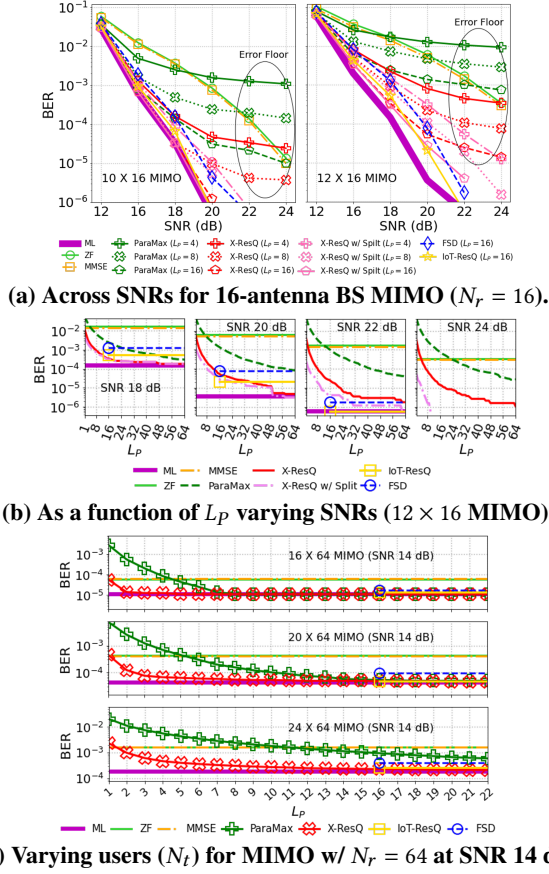
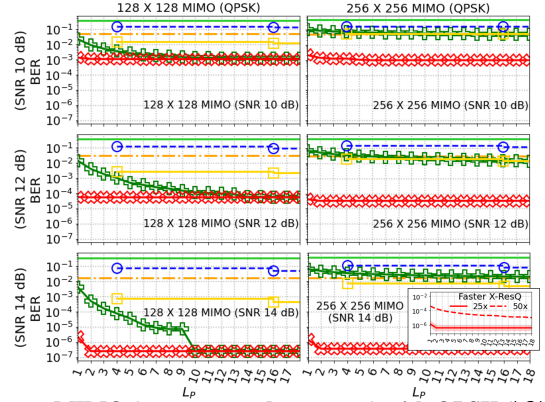


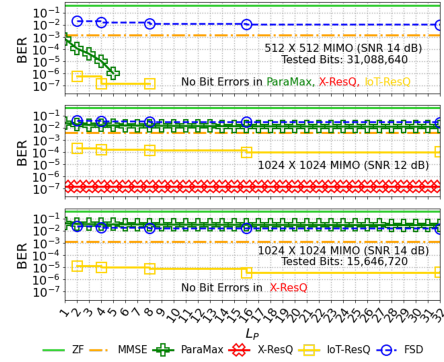
Figure 11: BER performance of various MIMO detectors including classically-implemented X-ResQ and IoT-ResQ for Massive MIMO with 16-QAM. Recall that FSD and IoT-ResQ require the minimum $L_p = 16$ for fully parallel processing, and current MIMO systems use linear detectors (ZF, MMSE) (§3.1).

50), while with $220 \mu s$ ($N_a = 100$) it requires $L_p = 6$ (574 qubits), showing trade-off between L_p (or qubit usage) and QA compute times. Similar patterns are observed for QPSK, though ML detection is not reported due to its large N_t .

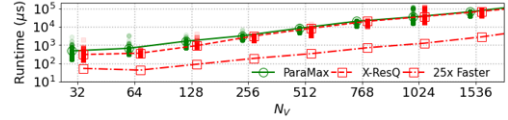
To experimentally evaluate the split-detection scheme (§4.5) on QA hardware, we show the packet success rate (based on the uncoded BER for 1500-byte packets) of sequential cumulative X-ResQ *with* and *without* the method at arbitrarily high SNRs in Figure 10 (note that *high* SNRs where the error floor occurs are dependent on L_p , N_t , and N_r/N_t). At the tested SNRs, method-assisted X-ResQ effectively improves the performance, resulting in 1.0 packet success rates without errors; X-ResQ without the scheme has several orders of magnitude lower rates, and the MMSE detector achieves ≈ 0.0 success rates. However, recall that QA experiments in this work are tested with a limited number of instances, and every optimal solution happens to obtain no errors for this experiment. In general, high-SNR detection experiments need testing more instances to capture the detectors' performance precisely, but



(a) Large MIMO (128 x 128 and 256 x 256) with QPSK ($|O| = 4$).



(b) Large MIMO (512 x 512 and 1024 x 1024) with BPSK ($|O| = 2$).



(c) Compute latency of classical X-ResQ across spin counts (N_V).

Figure 12: BER performance for Ultra-Large MIMO ($N_r/N_t = 1$) with low-order modulations (BPSK and QPSK) and 99.5-th percentile compute latency of classical X-ResQ across N_V for $N_V/\log_2(|O|)$ -user MIMO (e.g., $N_V/2$ -user MIMO with QPSK).

conducting QA experiments with a great number of instances is costly today. For this reason, the evaluation is somewhat limited, and thus the method will be further discussed in the next subsection, relying on classical experiments.

6.2 More Comprehensive Evaluations with Classical Implementations

With classical implementations of X-ResQ and IoT-ResQ (§5), we benchmark their performance with more various MIMO scenarios that current QA hardware cannot support.

Figure 11 compares the BER performance of various detectors for MIMO scenarios with 16-QAM varying SNRs, N_t , N_r , and L_p . Figure 11(a) shows 12-user (*left*) and 16-user (*right*) MIMO scenarios with $N_r = 16$. As one can see, the performance of linear detectors (ZF, MMSE) is poor in general, showing relatively high BER compared to the other detectors.

In the case of ParaMax and X-ResQ, they achieve better BER as L_P increases, but the error floor phenomenon is observed for them. So, we test X-ResQ with the split-detection method, where half of L_P is used for the split-Ising form. We observe X-ResQ with the split-detection method mitigates the error floor, achieving nearly two orders of magnitude better BER with $L_P = 8$ against the baseline one at SNR 24 dB. Interestingly, FSD and IoT-ResQ with their minimum $L_P = 16$ outperform the others including X-ResQ with the same L_P .

To further analyze this, we plot BER as a function of L_P in Figure 11(b). It is observed that even though IoT-ResQ works best at the point of $L_P = 16$, X-ResQ can converge to ML performance with less L_P , since IoT-ResQ (and FSD) cannot achieve gains until $L_P = 256$. Precisely, X-ResQ reaches the optimal BER ($\approx 5 \cdot 10^{-6}$) at SNR 20 dB with around $L_P = 50$, flexibly improving the performance with fine-grained parallelism. However, the BER of IoT-ResQ and FSD remains over 10^{-5} , despite increasing L_P due to their inflexibility. In this figure, we also observe that the split-detection X-ResQ gradually works better as SNRs increase, leading to bigger gaps against the baseline one. Next, we present BER as a function of L_P for different N_t for 64- N_r MIMO scenarios at SNR 14 dB in Figure 11(c). For all the tested N_t , X-ResQ reaches near-optimal BER with less L_P , even though both FSD and IoT-ResQ can reach it immediately with $L_P = 16$.¹¹

Ultra-Large MIMO with Low-Order Modulations. Now, we show the BER performance of X-ResQ for extremely large MIMO (over 100×100) with $N_r/N_t = 1$ and low-order modulations, varying SNRs in Figure 12. This is the regime where PIC MIMO detectors show exceptionally better performance compared to other (conventional) detectors. Here, we cannot report ML performance because of the large N_t .

First, Figure 12(a) presents 128×128 (*left*) and 256×256 (*right*) MIMO with QPSK. X-ResQ outperforms the others, converging to a certain BER with a few L_P , which we assume is the optimal performance, given that both the converged BERs of ParaMax and X-ResQ are the same in 128×128 MIMO. However, for 256×256 MIMO, only X-ResQ performs well and even ParaMax obtains poor detection performance. At SNR 14 dB, X-ResQ obtains five orders of magnitude lower BER than the others, requiring only a few L_P . This performance gap is very surprising and interesting, considering that there exists only a simple design difference between ParaMax and X-ResQ (*i.e.*, X-ResQ's initialization). Further, X-ResQ can converge to a BER with quite a few L_P . Thus, we also test *faster* X-ResQ by reducing its engine *sweep iteration* count (from original $N_{SW} = 50$) that largely affects its computation time [43]. While 50 \times faster version ($N_{SW} = 1$)

shows some performance trade-offs for the same L_P , 25 \times faster X-ResQ ($N_{SW} = 2$) is still able to achieve near-optimal BER even with similar L_P , which implies the evidence of speedup potential. The compute latency of classical X-ResQ also depends on spin variable counts ($N_V = N_t \cdot \log_2 |O|$) as shown in Figure 12(c), where 25 \times faster X-ResQ operates with $N_{SW} = 2$ (instead of $N_{SW} = 50$).

Lastly, we test 512×512 and 1024×1024 MIMO with BPSK. Similar results as QPSK are observed in Figure 12(b), where ParaMax, IoT-ResQ, and X-ResQ perform well for 512×512 MIMO (*top*), resulting in no bit errors among 10s of million tested bits after a few L_P . For 1024×1024 MIMO (*mid, bottom*), X-ResQ significantly outperforms the others (while IoT-ResQ still works decently). Note that X-ResQ even with $L_P = 1$ achieves error-free results at SNR 14 dB, showing great promise of enabling ultra-large MIMO. To our best knowledge, this is the largest spatial multiplexing MIMO ever reported with (assumably) near-optimal detection BER. It has been discussed that this extreme MIMO regime is particularly beneficial for massive IoT connectivity [47].

7 Discussion

Feasibility of X-ResQ in Cellular Networks. Apart from current insufficient qubit counts, X-ResQ is not immediately available largely due to the existing milliseconds of overheads in QPU usage other than pure QA time (§3.4). However, many studies have already identified specific techniques to reduce them effectively and their expected improvements [95]. Collectively, for 10 M-qubit quantum annealers (ca. 2040) [41], the total projected QPU time of X-ResQ with 100 anneals including all the overheads is around 250–500 μ s, consisting of approx. 50 μ s for programming and thermalization [73], $100 \times 2 \mu$ s for readout time and delay [26, 92], and $100 \times T_a$ for pure QA time, *e.g.*, 100×88 ns assuming the same RA schedule but with the potential minimum 40 ns¹² (48 ns back-and-forth anneal w/ $\tau_p = 0.4$ and 40 ns pause). This implies the stringent 5/6G latency requirements can be potentially satisfied, supporting 1000s of subcarriers with data and task parallelism. Such annealers also bring both cost and power advantages over 1.5 nm CMOS hardware to C-RAN [41].

Gate-Model Quantum Processors for MIMO Detection.

It should be noted that the porting of MU-MIMO detection to superconducting gate-model paradigms with *Quantum Approximate Optimization Algorithm* (QAOA) is an active field

¹¹However, we observe X-ResQ performs poorly with “64-QAM modulation” even for Massive MIMO scenarios, which remains an important challenge in PIC MIMO detectors. 64-QAM negative results are in Appendix E.

¹²A recent study has identified that QA with the nanosecond-scale anneal duration can follow coherent quantum theory [49], and indeed, the minimum anneal duration (T_a) that real-world quantum annealers support also tends to lower (currently 0.5–1 μ s). X-ResQ already utilizes the minimum $T_a \approx 2 \mu$ s that the current machine supports for the RA schedule with $\tau_p = 0.4$ (cf. $\approx 15 \mu$ s w/ FA [20, 86]), and thus is expected to keep following the trend.

of research [15, 27]. However, current gate-model processors can support the optimization algorithm targeting fully-connected Ising models with only a few tens of variables. Momentous progress is being made, but current relevant experiments [54, 58] still exhibit lower performance with respect to annealers. Furthermore, it has been discussed that QAOA even on future *fault-tolerant* gate processors has several critical challenges to overcome that do not apply to QA [22, 77, 93].

Rethinking Forward Error Correction. When (near-)optimal performance becomes achievable in MIMO detection, we should rethink forward error correction (FEC), since much simpler schemes than current LDPC (or very high code rate) could be applied instead. For example, some X-ResQ’s detection (uncoded) BERs observed are getting close to the target BER that FEC decoding aims to achieve (*e.g.*, Figure 12 w/ $N_r/N_t = 1$), which implies slightly increasing N_r/N_t can make detection nearly error-free. Currently, LDPC decoding has been identified as the most computationally heavy part of the physical layer whose computational amounts also scale with users [18]. Leveraging a sophisticated MIMO detector for ML performance (*e.g.*, X-ResQ) while exploiting a simple FEC decoder might be a more efficient design, which is the opposite of the current architecture. Moreover, ultra-large MIMO potentially enabled by (near-)optimal detectors (*e.g.*, 1024×1024 MIMO with X-ResQ) could not only improve spectral efficiency and device connectivity but also potentially simplify MAC layer designs. For instance, the expected increased capacity can make resource scheduling much simpler and faster. Overall, we believe realizing scalable optimal MIMO detectors could open up a research area that allows us to explore new system architectures in both mobile devices and base station (or access point in WLAN) systems.

End-to-End Systems. Despite the impressive performance, the current QA and PIC-based MIMO detectors have met with only limited success in that the studied systems, including this work, are still far from end-to-end (E2E) system implementations and evaluations, leaving their full potential and practicality as part of real wireless systems questionable. Due to the immaturity of technology, building quantum-based E2E wireless systems is many years away. However, classical generic PIC detectors (*e.g.*, classical X-ResQ) can be embedded into an E2E system and evaluated. Integrating them into the recent softwarization outputs of the MIMO physical layer such as Agora [18] and Hydra [24] would be a good initiative for the direction. To do so, significant system research efforts will be required for the efficient pipeline and parallelism design for a real-time E2E system with the newly added dimensions of parallelism. We leave this as our future work, where *system automation* will be also considered for intelligence and elasticity in the physical layer (*e.g.*, opportunistic RA, *adaptive* split-detection method, N_r/N_t , and level of parallelism).

8 Conclusion

The paper introduces X-ResQ, a QA-based MIMO detector that uses multi-seed ensemble RA as its parallelization strategy. We show that X-ResQ’s design is simple but effective, trading off between qubits and compute time. Given that nearly all QA components such as qubit counts, overheads, and noise control are being improved drastically, the significantly enhanced MIMO detection performance achieved by X-ResQ over prior designs on the current machine is a good indication of the potential general use of QA MIMO detectors with a long-term vision. Furthermore, the paper shows the viability of enabling ultra-large MIMO by classical X-ResQ, which itself is another PIC MIMO detector candidate.

Acknowledgement

We thank the Princeton Advanced Wireless Systems (PAWS) Group and the Yale Efficient Computing Lab (ECL) for their extensive technical feedback and helpful discussion. This research is based upon work supported by InterDigital Communications, and National Science Foundation (NSF) Award No. CNS-1824357 and CCF-1918549. QA experiments on the D-Wave quantum annealers have been supported by InterDigital. This work does not raise any ethical issues.

References

- [1] Technical description of the d-wave quantum processing unit. *D-Wave Technical Report*, 2021.
- [2] Zephyr topology of d-wave quantum processors. *D-Wave Technical Report*, 2021.
- [3] E. Agrell, T. Eriksson, A. Vardy, K. Zeger. Closest point search in lattices. *IEEE Trans. Inf. Theory*, **48**(8), 2201–2214, 2002.
- [4] T. Albash, V. Martin-Mayor, I. Hen. Temperature scaling law for quantum annealing optimizers. *Physical review letters*, **119**(11), 110502, 2017.
- [5] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, H. Katzgrabeer. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Frontiers in Physics*, **7**, 48, 2019.
- [6] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.* Quantum supremacy using a programmable superconducting processor. *Nature*, **574**(7779), 505–510, 2019.
- [7] R. Ayanzadeh, P. Das, S. Tannu, M. Qureshi. Equal: Improving the fidelity of quantum annealers by injecting controlled perturbations. *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 516–527. IEEE, 2022.
- [8] L. Barbero, J. Thompson. Fixing the complexity of the sphere decoder for MIMO detection. *IEEE Transactions on Wireless Communications*, **7**(6), 2131–2142, 2008.
- [9] S. Boixo, V. N. Smelyanskiy, A. Shabani, S. V. Isakov, M. Dykman, V. S. Denchev, M. H. Amin, A. Y. Smirnov, M. Mohseni, H. Neven. Computational multiqubit tunnelling in programmable quantum annealers. *Nature Communications*, **7**, 2016.
- [10] F. Cai, S. Kumar, T. Van Vaerenbergh, X. Sheng, R. Liu, C. Li, Z. Liu, M. Foltin, S. Yu, Q. Xia, *et al.* Power-efficient combinatorial optimization using intrinsic noise in memristor hopfield neural networks. *Nature Electronics*, **3**(7), 409–418, 2020.
- [11] O. Castaneda, T. Goldstein, C. Studer. Data detection in large multi-antenna wireless systems via approximate semidefinite relaxation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, **63**(12), 2334–2346, 2016.
- [12] V. Choi. Minor-embedding in adiabatic quantum computation: I. the parameter setting problem. *Quantum Information Processing*, **7**(5), 193–209, 2008.
- [13] E. Crosson, D. Lidar. Prospects for quantum enhancement with diabatic quantum annealing. *Nature Reviews Physics*, **3**(7), 466–489, 2021.
- [14] J. Cui, G. L. Long, L. Hanzo. General hamiltonian representation of ml detection relying on the quantum approximate optimization algorithm. *arXiv preprint arXiv:2204.05126*, 2022.
- [15] J. Cui, Y. Xiong, S. X. Ng, L. Hanzo. Quantum approximate optimization algorithm based maximum likelihood detection. *IEEE Transactions on Communications*, **70**(8), 5386–5400, 2022.
- [16] M. Damen, H. El Gamal, G. Caire. On maximum likelihood detection and the search for the closest lattice point. *IEEE Trans. Inf. Theory*, **49**(10), 2389–402, 2003.
- [17] V. Denchev, S. Boixo, S. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, H. Neven. What is the computational value of finite range tunneling? *Physical Review X*, **6**, 031015, 2016.
- [18] J. Ding, R. Doost-Mohammady, A. Kalia, L. Zhong. Agora: Real-time massive mimo baseband processing in software. *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, 232–244, 2020.
- [19] T. Q. Dinh, S. H. Dau, E. Lagunas, S. Chatzinotas. Efficient hamiltonian reduction for quantum annealing on satcom beam placement problem. *ICC 2023-IEEE International Conference on Communications*, 2668–2673. IEEE, 2023.
- [20] J. D. L. Ducoing, K. Nikitopoulos. Quantum annealing for next-generation mu-mimo detection: Evaluation and challenges. *ICC 2022-IEEE International Conference on Communications*, 637–642. IEEE, 2022.
- [21] D. J. Egger, J. Mareček, S. Woerner. Warm-starting quantum optimization. *Quantum*, **5**, 479, 2021.
- [22] E. Farhi, D. Gamarnik, S. Gutmann. The quantum approximate optimization algorithm needs to see the whole graph: Worst case examples. *arXiv preprint arXiv:2005.08747*, 2020.
- [23] J. Golden, D. O’Malley. Reverse annealing for nonnegative/binary matrix factorization. *Plos one*, **16**(1), e0244026, 2021.
- [24] J. Gong, A. Kalia, M. Yu. Scalable distributed massive mimo baseband processing. *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, 405–417, 2023.
- [25] J. Grollier, D. Querlioz, K. Camsari, K. Everschor-Sitte, S. Fukami, M. D. Stiles. Neuromorphic spintronics. *Nature electronics*, **3**(7), 360–370, 2020.
- [26] J. A. Grover, J. I. Basham, A. Marakov, S. M. Disseler, R. T. Hinkey, M. Khalil, Z. A. Stegen, T. Chamberlin, W. DeGottardi, D. J. Clarke, *et al.* Fast, lifetime-preserving readout for high-coherence quantum annealers. *PRX Quantum*, **1**(2), 020314, 2020.
- [27] B. Gulbahar. Maximum-likelihood detection with qaoa for massive mimo and sherrington-kirkpatrick model with local field at infinite size. *Authorea Preprints*, 2023.
- [28] Z. Guo, P. Nilsson. Algorithm and implementation of the k-best sphere decoding for mimo detection. *IEEE Journal on selected areas in communications*, **24**(3), 491–503, 2006.
- [29] Y. Haribara, S. Utsunomiya, Y. Yamamoto. Computational Principle and Performance Evaluation of Coherent Ising Machine Based on Degenerate Optical Parametric Oscillator Network. *Entropy*, 2016. ISSN 1099-4300. doi:10.3390/e18040151.
- [30] G. He, X. Zhang, Z. Liang. Algorithm and architecture of an efficient mimo detector with cross-level parallel tree-search. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **28**(2), 467–479, 2019.
- [31] J. Hoydis, S. Cammerer, F. Ait Aoudia, A. Vem, N. Binder, G. Marcus, A. Keller. Sionna: An open-source library for next-generation physical layer research. *arXiv preprint*, 2022.
- [32] T. Huang, Y. Zhu, R. S. M. Goh, T. Luo. When quantum annealing meets multitasking: Potentials, challenges and opportunities. *Array*, 100282, 2023.
- [33] Y. Huang, W. Li, C. Pan, S. Hou, X. Lu, C. Cui, J. Wen, J. Xu, C. Cao, Y. Ma, *et al.* Quantum computing for mimo beam selection problem: Model and optical experimental solution. *arXiv preprint arXiv:2310.12389*, 2023.
- [34] C. Husmann, G. Georgis, K. Nikitopoulos, K. Jamieson. FlexCore: Massively parallel and flexible processing for large MIMO access points. *Proc. of the USENIX NSDI Symp.*, 2017.
- [35] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu, *et al.* A coherent ising machine for 2000-node optimization problems. *Science*, **354**(6312), 603–606, 2016.
- [36] T. Kadowaki, H. Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, **58**(5), 5355, 1998.
- [37] H. Karimi, G. Rosenberg. Boosting quantum annealer performance via sample persistence. *Quantum Information Processing*, **16**(7), 166, 2017.
- [38] S. Kasi, K. Jamieson. Towards quantum belief propagation for ldpc decoding in wireless networks. *ACM MobiCom*, 1–14, 2020.
- [39] S. Kasi, J. Kaewelh, K. Jamieson. A quantum annealer-enabled

- decoder and hardware topology for nextg wireless polar codes. *IEEE Transactions on Wireless Communications*, 2023.
- [40] S. Kasi, A. K. Singh, D. Venturelli, K. Jamieson. Quantum annealing for large mimo downlink vector perturbation precoding. *ICC 2021-IEEE International Conference on Communications*, 1–6. IEEE, 2021.
- [41] S. Kasi, P. Warburton, J. Kaewell, K. Jamieson. Challenge: A cost and power feasibility analysis of quantum annealing for nextg cellular wireless networks. *arXiv preprint arXiv:2109.01465*, 2021.
- [42] M. Kim, K. Jamieson. Finer-grained decomposition for parallel quantum mimo processing. *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE, 2023.
- [43] M. Kim, S. Mandrà, D. Venturelli, K. Jamieson. Physics-inspired heuristics for soft mimo detection in 5g new radio and beyond. *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 42–55, 2021.
- [44] M. Kim, A. Stockley, K. Briggs, K. Jamieson. Physics-inspired discrete-phase optimization for 3d beamforming with pin-diode extra-large antenna arrays. *arXiv preprint arXiv:2311.16128*, 2023.
- [45] M. Kim, D. Venturelli, K. Jamieson. Leveraging quantum annealing for large mimo processing in centralized radio access networks. *Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM'19)*, 241–255. ACM, 2019.
- [46] —. Towards hybrid classical-quantum computation structures in wirelessly-networked systems. *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, 110–116, 2020.
- [47] M. Kim, D. Venturelli, J. Kaewell, K. Jamieson. Warm-started quantum sphere decoding via reverse annealing for massive iot connectivity. *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, 1–14, 2022.
- [48] M. Kim, *et al.* Heuristic quantum optimization for 6g wireless communications. *IEEE Network*, **35**(4), 8–15, 2021.
- [49] A. D. King, J. Raymond, T. Lanting, R. Harris, A. Zucca, F. Altomare, A. J. Berkley, K. Boothby, S. Ejtemaee, C. Enderud, *et al.* Quantum critical dynamics in a 5,000-qubit programmable spin glass. *Nature*, 1–6, 2023.
- [50] J. King, M. Mohseni, W. Bernoudy, A. Fréchet, H. Sadeghi, S. V. Isakov, H. Neven, M. H. Amin. Quantum-assisted genetic algorithm. *arXiv preprint arXiv:1907.00707*, 2019.
- [51] C. Klymko, B. D. Sullivan, T. S. Humble. Adiabatic quantum programming: minor embedding with hard faults. *Quantum information processing*, **13**(3), 709–729, 2014.
- [52] S. Knysh. Zero-temperature quantum annealing bottlenecks in the spin-glass phase. *Nature communications*, **7**(1), 1–9, 2016.
- [53] Q. J. Lim, C. Ross, A. Ghosh, F. Vook, G. Gradoni, Z. Peng. Quantum-assisted combinatorial optimization for reconfigurable intelligent surfaces in smart electromagnetic environments. *IEEE Transactions on Antennas and Propagation*, 2023.
- [54] F. B. Maciejewski, S. Hadfield, B. Hall, M. Hodson, M. Dupont, B. Evert, J. Sud, M. S. Alam, Z. Wang, S. Jeffrey, *et al.* Design and execution of quantum circuits using tens of superconducting qubits and thousands of gates for dense ising optimization problems. *arXiv preprint arXiv:2308.12423*, 2023.
- [55] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. C. Wong, F. Tufvesson, V. Öwall, O. Edfors. The world's first real-time testbed for massive mimo: Design, implementation, and validation. *IEEE Access*, **5**, 9073–9088, 2017.
- [56] Á. Marosits, Z. Tabi, Z. Kallus, P. Vadera, I. Gódor, Z. Zimborás. Exploring embeddings for mimo channel decoding on quantum annealers. *Infocommunications Journal*, **13**(1), 11–17, 2021.
- [57] J. Marshall, D. Venturelli, I. Hen, E. Rieffel. The power of pausing: advancing understanding of thermalization in experimental quantum annealers. *arXiv:1810.05881*, 2018.
- [58] A. Mazumder, A. Sen, U. Sen. Benchmarking metaheuristic-integrated quantum approximate optimisation algorithm against quantum annealing for quadratic unconstrained binary optimization problems. *arXiv preprint arXiv:2309.16796*, 2023.
- [59] P. L. McMahon, A. Marandi, *et al.* A fully programmable 100-spin coherent Ising machine with all-to-all connections. *Science*, 2016.
- [60] N. Metropolis, S. Ulam. The monte carlo method. *Journal of the American statistical association*, **44**(247), 335–341, 1949.
- [61] C. Monroe, J. Kim. Scaling the ion trap quantum processor. *Science*, **339**(6124), 1164–1169, 2013.
- [62] K. Nikitopoulos. Massively parallel, nonlinear processing for 6g: Potential gains and further research challenges. *IEEE Communications Magazine*, **60**(1), 81–87, 2022.
- [63] K. Nikitopoulos, G. Georgis, C. Jayawardena, D. Chatzipanagiotis, R. Tafazolli. Massively parallel tree search for high-dimensional sphere decoders. *IEEE Transactions on Parallel and Distributed Systems*, **30**(10), 2309–2325, 2018.
- [64] K. Nikitopoulos, J. Zhou, B. Congdon, K. Jamieson. Geosphere: Consistently turning MIMO capacity into throughput. *Proc. of the ACM SIGCOMM Conf.*, 631–642, 2014.
- [65] M. Norimoto, R. Mori, N. Ishikawa. Quantum algorithm for higher-order unconstrained binary optimization and mimo maximum likelihood detection. *IEEE Transactions on Communications*, **71**(4), 1926–1939, 2023.
- [66] M. Ohkuwa, H. Nishimori, D. A. Lidar. Reverse annealing for the fully connected p-spin model. *Physical Review A*, **98**(2), 022,314, 2018.
- [67] G. Passarelli, P. Lucignano. Counterdiabatic reverse annealing. *Physical Review A*, **107**(2), 022,607, 2023.
- [68] A. Pearson, A. Mishra, I. Hen, D. A. Lidar. Analog errors in quantum annealing: doom and hope. *npj Quantum Information*, **5**(1), 1–9, 2019.
- [69] E. Pelofske, G. Hahn, H. Djidjev. Initial state encoding via reverse quantum annealing and h-gain features. *arXiv preprint arXiv:2303.13748*, 2023.
- [70] E. Pelofske, G. Hahn, H. N. Djidjev. Parallel quantum annealing. *Scientific Reports*, **12**(1), 1–11, 2022.
- [71] —. Solving larger optimization problems using parallel quantum annealing. *arXiv preprint arXiv:2205.12165*, 2022.
- [72] D. Pierangeli, G. Marcucci, C. Conti. Large-scale photonic ising machine by spatial light modulation. *Physical review letters*, **122**(21), 213,902, 2019.
- [73] M. D. Reed, B. R. Johnson, A. A. Houck, L. DiCarlo, J. M. Chow, D. I. Schuster, L. Frunzio, R. J. Schoelkopf. Fast reset and suppressing spontaneous emission of a superconducting qubit. *Applied Physics Letters*, **96**(20), 2010.
- [74] S. Roger, C. Ramiro, A. Gonzalez, V. Almenar, A. M. Vidal. Fully parallel gpu implementation of a fixed-complexity soft-output mimo detector. *IEEE Transactions on Vehicular Technology*, **61**(8), 3796–3800, 2012.
- [75] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, M. Troyer. Defining and detecting quantum speedup. *Science*, **345**(6195), 420–424, 2014.
- [76] C. Ross, *et al.* Engineering reflective metasurfaces with ising hamiltonian and quantum annealing. *IEEE Transactions on Antennas and Propagation*, **70**(4), 2841–2854, 2021.
- [77] Y. R. Sanders, D. W. Berry, P. C. Costa, L. W. Tessler, N. Wiebe, C. Gidney, H. Neven, R. Babbush. Compilation of fault-tolerant quantum heuristics for combinatorial optimization. *PRX Quantum*, **1**(2), 020,312, 2020.

- [78] A. D. Sarma, *et al.* On quantum-assisted ldpc decoding augmented with classical post-processing. *arXiv preprint arXiv:2204.09940*, 2022.
- [79] C. Shepard, H. Yu, N. Anand, L. Li, T. Marzetta, R. Yang, L. Zhong. Argos: Practical many-antenna base stations. *Proc. of the ACM MobiCom Conf.*, 2012.
- [80] A. K. Singh, K. Jamieson, P. L. McMahon, D. Venturelli. Ising machines' dynamics and regularization for near-optimal mimo detection. *IEEE Transactions on Wireless Communications*, **21**(12), 11,080–11,094, 2022.
- [81] A. K. Singh, A. Kapelyan, D. Venturelli, K. Jamieson. Uplink mimo detection using ising machines: A multi-stage ising approach. *arXiv preprint arXiv:2304.12830*, 2023.
- [82] A. K. Singh, D. Venturelli, K. Jamieson. Perturbation-based formulation of maximum likelihood mimo detection for coherent ising machines. *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2523–2528, 2022.
- [83] S. Sreedhara, J. Roychowdhury, J. Wabnig, P. K. Srinath. Mu-mimo detection using oscillator ising machines. *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 1–9. IEEE, 2023.
- [84] B. Sutton, K. Y. Camsari, B. Behin-Aein, S. Datta. Intrinsic optimization using stochastic nanomagnets. *Scientific reports*, **7**(1), 44,370, 2017.
- [85] R. H. Swendsen, J.-S. Wang. Replica monte carlo simulation of spin-glasses. *Physical review letters*, **57**(21), 2607, 1986.
- [86] Z. I. Tabi, Á. Marosits, Z. Kallus, P. Vaderna, I. Gódor, Z. Zimborás. Evaluation of quantum annealer performance via the massive mimo problem. *IEEE Access*, **9**, 131,658–131,671, 2021.
- [87] S. Takabe. Deep unfolded simulated bifurcation for massive mimo signal detection. *arXiv preprint arXiv:2306.16264*, 2023.
- [88] K. Tatsumura, M. Yamasaki, H. Goto. Scaling out ising machines using a multi-chip architecture for simulated bifurcation. *Nature Electronics*, **4**(3), 208–217, 2021.
- [89] D. Venturelli, A. Kondratyev. Reverse quantum annealing approach to portfolio optimization problems. *arXiv:1810.08584*, 2018.
- [90] F. Vista, G. Iacovelli, L. A. Grieco. Hybrid quantum-classical scheduling optimization in uav-enabled iot networks. *Quantum Information Processing*, **22**(1), 47, 2023.
- [91] E. Viterbo, J. Boutros. A universal lattice code decoder for fading channels. *IEEE Trans. Inf. Theory*, **45**(5), 1639–1642, 1999.
- [92] T. Walter, P. Kurpiers, S. Gasparinetti, P. Magnard, A. Potočnik, Y. Salathé, M. Pechal, M. Mondal, M. Oppliger, C. Eichler, *et al.* Rapid high-fidelity single-shot dispersive readout of superconducting qubits. *Physical Review Applied*, **7**(5), 054,020, 2017.
- [93] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, P. J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature communications*, **12**(1), 6961, 2021.
- [94] T. Wang, J. Roychowdhury. OIM: Oscillator-Based Ising Machines for Solving Combinatorial Optimisation Problems. I. McQuillan, S. Seki, eds., *Unconventional Computation and Natural Computation*, 232–256. Springer International Publishing, Cham, 2019. ISBN 978-3-030-19311-9.
- [95] S. Weber, J. Cummings, J. Miloshi, K. Thompson, J. Rokosz, D. Holtman, D. Conway, A. Kerman, W. Oliver. High-density i/o for next-generation quantum annealing: Part 1-cryogenic wiring. *APS March Meeting Abstracts*, vol. 2021, M30–008, 2021.
- [96] M. Wenk, M. Zellweger, A. Burg, N. Felber, W. Fichtner. K-best MIMO detection VLSI architectures achieving up to 424 Mbps. *IEEE International Symposium on Circuits and Systems*, 4 pp.–1154, 2006.
- [97] S. Winter, Y. Zhang, G. Zheng, L. Hanzo. A lattice-reduction aided vector perturbation precoder relying on quantum annealing. *arXiv preprint arXiv:2402.07643*, 2024.
- [98] K. Xu, C. Gong, B. Liang, Y. Wu, B. Di, L. Song, C. Xu. Low-latency visible light backscatter networking with retrumumimo. *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 448–461, 2022.
- [99] M. Yang, Z. Zhong, M. Ghobadi. On-fiber photonic computing. *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*, 263–271, 2023.
- [100] Q. Yang, X. Li, H. Yao, J. Fang, K. Tan, W. Hu, J. Zhang, Y. Zhang. BigStation: Enabling scalable real-time signal processing in large MU-MIMO systems. *ACM SIGCOMM Computer Communication Review*, **43**(4), 399–410, 2013.

A Linear MIMO Detectors

Zero-Forcing (ZF) linear method makes use of the pseudo-inverse of \mathbf{H} , $\mathbf{H}^\dagger = (\mathbf{H}^* \mathbf{H})^{-1} \mathbf{H}^*$ with \mathbf{H}^* Hermitian transpose, for detection, by multiplying \mathbf{H}^\dagger with \mathbf{y} :

$$\mathbf{H}^\dagger \mathbf{y} = (\mathbf{H}^* \mathbf{H})^{-1} \mathbf{H}^* \mathbf{H} \bar{\mathbf{v}} + (\mathbf{H}^* \mathbf{H})^{-1} \mathbf{H}^* \mathbf{n} = \bar{\mathbf{v}} + \mathbf{n}' \quad (3)$$

with \mathbf{n}' the noise vector affected by the amplification factor. For each t^{th} user, the symbol \hat{v}_t among $v \in \mathcal{O}$ with the minimum Euclidean distance against $\bar{v}_t + \mathbf{n}'_t$ (i.e., the closest member of constellation to t^{th} element in $\mathbf{H}^\dagger \mathbf{y}$) is detected, forming detected symbol vector $\hat{\mathbf{v}}$. As one can see, ZF is sensitive to both noise vector \mathbf{n} and channel \mathbf{H} that decide \mathbf{n}' . When the noise and/or amplification factor becomes large (i.e., low SNRs and/or low N_r/N_t), the detection performance is severely degraded.

Minimum Mean Square Error (MMSE) linear detector multiplies \mathbf{G} with \mathbf{y} (instead of \mathbf{H}^\dagger in ZF) where $\mathbf{G} = \text{SNR} \cdot (\mathbf{I} + \text{SNR} \cdot \mathbf{H}^* \mathbf{H})^{-1} \mathbf{H}^*$, with \mathbf{I} being an identity matrix and SNR a signal-to-noise ratio, which satisfies the minimum $\|\bar{\mathbf{v}} - \mathbf{G}\mathbf{y}\|^2$. Unlike ZF, MMSE considers \mathbf{n} in addition to \mathbf{H} for regularization, thus outperforming ZF.

B Split-Detection: Theoretical Analysis

As discussed in the main paper, the split-detection method transforms a 16-QAM MIMO ML problem into solving two independent QPSK problems. This method can be generalized for any modulations. For a square QAM modulation, the optimization variable \mathbf{v} in ML problem (§2.1) can be expressed using $n_q = \log_2(\lceil \sqrt{|\mathcal{O}|} \rceil)$ QPSK variables, as $\mathbf{v} = \sum_{i=1}^{n_q} 2^{i-1} \mathbf{q}_i$, where $\mathbf{q}_i \in \{-1-j, -1+j, 1-j, 1+j\}^{N_t}$ consists of QPSK symbols. We fix the values of $(n_q - 1)$ variables in this representation using the MMSE solution and reduce the problem to effectively have an effective search space of $\{-1-j, -1+j, 1-j, 1+j\}^{N_t}$, which is equivalent to searching with a QPSK modulation. If the modulation is not a square constellation, the procedure remains the same, except that $\mathbf{b}_{n_b} \in \{-1, 1\}^{N_t}$ and the corresponding reduced problem is equivalent to BPSK, instead of QPSK.

Noise Analysis. Let us analytically look at the noise in the split-detection with 16-QAM to understand it better, considering an $N \times N$ MIMO system (i.e., $N_t = N_r$):

$$\mathbf{y} = \mathbf{H} \bar{\mathbf{v}} + \mathbf{n}, \quad (4)$$

where \mathbf{n} is white Gaussian noise with $E[|\mathbf{n}|] = 0$ and $E[|\mathbf{n}|^2] = \sigma^2$. Note that the channel is assumed to be Rayleigh fading (i.e., each element H_{ij} is drawn from a complex normal distribution). For 16-QAM, the transmit vector can be expressed as $\mathbf{v} = 2\mathbf{q}_1 + \mathbf{q}_2 \in \mathcal{O}^N$, where $\mathbf{q}_1, \mathbf{q}_2 \in \{-1-i, -1+i, 1-i, 1+i\}^N$, each of which can be expressed with two spin variables: $q = s_1 + js_2$. Note that, \mathbf{q}_1 is the quadrant of the transmit symbol and \mathbf{q}_2 expresses the position within the

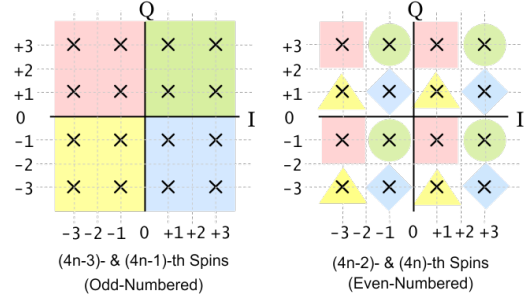


Figure 13: Decision impact of spin variables in ML Ising models. In 16-QAM ($O = 16$), for n -th user's symbol, $(4n-3)$ - and $(4n-1)$ -th (odd-numbered) spins are related to the quadrant decision (i.e., $2q_1$ in \mathbf{v}), while $(4n-2)$ - and $(4n)$ -th spins (even-numbered spins) to the position decision (i.e., q_2 in \mathbf{v}). This can be generalized with higher-order modulations.

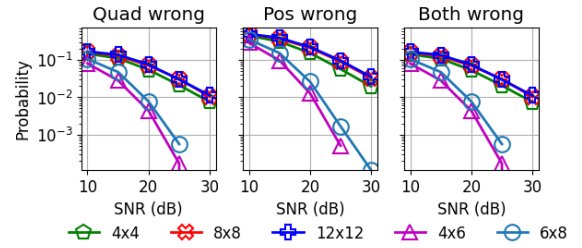


Figure 14: Probability of MMSE solution wrongly predicting the quadrant and/or position of the ML solution. We see that the MMSE solution becomes increasingly accurate with an increase in SNR; hence, the split-detection-based Ising form tends to be equivalent to the original ML problem as SNR increases.

quadrant as shown in Figure 13.

When we perform split detection assisted by the MMSE solution $\mathbf{v}_m = 2\mathbf{q}_{m1} + \mathbf{q}_{m2}$, and we try to decode \mathbf{q}_1 , after subtracting $\mathbf{H}\mathbf{m}_2$ to cancel interference of \mathbf{q}_2 , the equivalent system is

$$\frac{\mathbf{y}}{2} = \mathbf{H}\mathbf{q}_1 + 0.5(\mathbf{n} + \mathbf{H}(\mathbf{q}_2 - \mathbf{q}_{m2})) \quad (5)$$

let us define $\delta = \mathbf{q}_2 - \mathbf{q}_{m2}$, where for user i , $|\delta_i|^2 = 0$ if MMSE estimate is correct, $|\delta_i|^2 = 4$ if only real or imaginary part of MMSE estimate is correct, and $|\delta_i|^2 = 8$ if MMSE estimate is completely wrong.

Then the *effective noise power* is given by,

$$0.25(E[|\mathbf{n} + \mathbf{H}\delta|^2]) \quad (6)$$

and therefore by triangle inequality

$$\leq 0.25(E[|\mathbf{n}|^2] + E[|\mathbf{H}\delta|^2] + 2E[|\mathbf{n}| \cdot |\mathbf{H}\delta|]). \quad (7)$$

The Cauchy-Swartz inequality for two random variables X and Y states,

$$|E[XY]| \leq \sqrt{E[X^2]E[Y^2]} \quad (8)$$

by Cauchy-Schwartz inequality,

$$\leq 0.25(E[||\mathbf{n}||^2] + E[||\mathbf{H}\delta||^2] + 2\sqrt{E[||\mathbf{n}||^2] \cdot E[||\mathbf{H}\delta||^2]}) \quad (9)$$

$$= 0.25(\sigma^2 + E[||\mathbf{H}\delta||^2] + 2\sigma\sqrt{E[||\mathbf{H}\delta||^2]}). \quad (10)$$

Let us look at the random variable $\mathbf{H}\delta$,

$$E[||\mathbf{H}\delta||^2] = E[\delta^\dagger \mathbf{H}^\dagger \mathbf{H} \delta]. \quad (11)$$

Let $\mathbf{D} = \mathbf{H}^\dagger \mathbf{H}$,

$$E[||\mathbf{H}\delta||^2] = E[\delta^\dagger \mathbf{D} \delta] = E\left[\sum_i \sum_j D_{ij} \delta_i \delta_j^*\right] \quad (12)$$

As $||\mathbf{H}\delta||^2$ is real,

$$\begin{aligned} &= E\left[\sum_i \sum_j \text{Re}\{D_{ij} \delta_i \delta_j^*\}\right] \\ &= \sum_i \sum_j E[\text{Re}\{D_{ij}\} \text{Re}\{\delta_i\} \text{Re}\{\delta_j\}] \\ &\quad - E[\text{Re}\{D_{ij}\} \text{Im}\{\delta_i\} \text{Im}\{\delta_j\}] \\ &\quad - E[\text{Im}\{D_{ij}\} \text{Re}\{\delta_i\} \text{Im}\{\delta_j\}] \\ &\quad - E[\text{Im}\{D_{ij}\} \text{Im}\{\delta_i\} \text{Re}\{\delta_j\}]. \end{aligned} \quad (13)$$

Let us look at the term $|E[\text{Re}\{D_{ij}\} \text{Re}\{\delta_i\} \text{Re}\{\delta_j\}]|$, by Cauchy-Schwartz inequality,

$$\leq \sum_i \sum_j \sqrt{E[\text{Re}\{D_{ij}\}^2]} \sqrt{E[\text{Re}\{\delta_i\}^2 \text{Re}\{\delta_j\}^2]}. \quad (14)$$

Now,

$$E[\text{Re}\{\delta_i\}^2 \text{Re}\{\delta_j\}^2] \leq \sqrt{E[\text{Re}\{\delta_i\}^4]} \sqrt{E[\text{Re}\{\delta_j\}^4]}. \quad (15)$$

Note that $\text{Re}\{\delta_i\}$ can only take values $\{0, -2, 2\}$

$$E[\text{Re}\{\delta_i\}^4] = \mathcal{P}(\text{Re}\{\delta_i\} \neq 0) \cdot 16 \quad (16)$$

which implies

$$E[\text{Re}\{\delta_i\}^2 \text{Re}\{\delta_j\}^2] \leq 16 \cdot \sqrt{\mathcal{P}(\text{Re}\{\delta_i\} \neq 0) \mathcal{P}(\text{Re}\{\delta_j\} \neq 0)}.$$

Similarly,

$$E[\text{Im}\{\delta_i\}^2 \text{Im}\{\delta_j\}^2] \leq 16 \cdot \sqrt{\mathcal{P}(\text{Im}\{\delta_i\} \neq 0) \mathcal{P}(\text{Im}\{\delta_j\} \neq 0)}$$

$$E[\text{Re}\{\delta_i\}^2 \text{Im}\{\delta_j\}^2] \leq 16 \cdot \sqrt{\mathcal{P}(\text{Re}\{\delta_i\} \neq 0) \mathcal{P}(\text{Im}\{\delta_j\} \neq 0)}$$

$$E[\text{Im}\{\delta_i\}^2 \text{Re}\{\delta_j\}^2] \leq 16 \cdot \sqrt{\mathcal{P}(\text{Im}\{\delta_i\} \neq 0) \mathcal{P}(\text{Re}\{\delta_j\} \neq 0)}.$$

Let us look at $E[\text{Re}\{D_{ij}\}^2]$

$$= E\left[\left(\sum_k \text{Re}\{H_{ki}\} \text{Re}\{H_{kj}\} + \text{Im}\{H_{ki}\} \text{Im}\{H_{kj}\}\right)^2\right].$$

Note that the real and imaginary parts of channel coefficients are I.I.D Gaussian random variables with zero mean and

variance 0.5.

For $i \neq j$,

$$\begin{aligned} E[\text{Re}\{D_{ij}\}^2] &= \sum_k E[\text{Re}\{H_{ki}\}^2] E[\text{Re}\{H_{kj}\}^2] + \\ &\quad E[\text{Im}\{H_{ki}\}^2] E[\text{Im}\{H_{kj}\}^2] \\ &= 0.5N. \end{aligned}$$

For $i = j$,

$$\begin{aligned} E[\text{Re}\{D_{ii}\}^2] &= E\left[\left(\sum_k \text{Re}\{H_{ki}\}^2 + \text{Im}\{H_{ki}\}^2\right)^2\right] \\ &= \sum_k E[\text{Re}\{H_{ki}\}^4] + E[\text{Im}\{H_{ki}\}^4] + \\ &\quad 2 \cdot \sum_{j,k,j \neq k} E[\text{Re}\{H_{ki}\}^2 \text{Im}\{H_{kj}\}^2] \\ &= 6N \cdot (1/2)^4 + (1/2)^4 \cdot N(N-1). \end{aligned}$$

Next, $E[\text{Im}\{D_{ij}\}^2]$

$$= E\left[\left(\sum_k \text{Re}\{H_{ki}\} \text{Im}\{H_{kj}\} - \text{Im}\{H_{ki}\} \text{Re}\{H_{kj}\}\right)^2\right]$$

for $i \neq j$,

$$= \sum_k E[\text{Re}\{H_{ki}\}^2] E[\text{Im}\{H_{kj}\}^2]$$

$$+ E[\text{Im}\{H_{ki}\}^2] E[\text{Re}\{H_{kj}\}^2] = 0.5N$$

and zero otherwise.

Therefore, for $i \neq j$, $E[\text{Re}\{D_{ij}\} \text{Re}\{\delta_i\} \text{Re}\{\delta_j\}]$

$$\leq \sqrt{0.5N} \cdot \sqrt{16 \mathcal{P}(\text{Re}\{\delta_i\} \neq 0) \mathcal{P}(\text{Re}\{\delta_j\} \neq 0)}$$

and for $i = j$,

$$\leq 4 \sqrt{[6N \cdot (1/2)^4 + (1/2)^4 \cdot N(N-1)] \mathcal{P}(\text{Re}\{\delta_i\} \neq 0)}.$$

Putting everything together, $E[||\mathbf{H}\delta||^2]$

$$\begin{aligned} &\leq \sum_i 4 \sqrt{[6N \cdot (1/2)^4 + (1/2)^4 \cdot N(N-1)]} \\ &\quad \times (\sqrt{\mathcal{P}(\text{Re}\{\delta_i\} \neq 0)} + \sqrt{\mathcal{P}(\text{Im}\{\delta_i\} \neq 0)}) \\ &+ \sum_{i,j,i \neq j} \sqrt{8N} \cdot [(\mathcal{P}(\text{Re}\{\delta_i\} \neq 0) \mathcal{P}(\text{Re}\{\delta_j\} \neq 0))^{\frac{1}{4}} \\ &\quad + (\mathcal{P}(\text{Im}\{\delta_i\} \neq 0) \mathcal{P}(\text{Im}\{\delta_j\} \neq 0))^{\frac{1}{4}} \\ &\quad + (\mathcal{P}(\text{Re}\{\delta_i\} \neq 0) \mathcal{P}(\text{Im}\{\delta_j\} \neq 0))^{\frac{1}{4}} \\ &\quad + (\mathcal{P}(\text{Im}\{\delta_i\} \neq 0) \mathcal{P}(\text{Re}\{\delta_j\} \neq 0))^{\frac{1}{4}}] \end{aligned}$$

We assume $\mathcal{P}(\text{Re}\{\delta_j\} \neq 0) = \mathcal{P}(\text{Re}\{\delta_i\} \neq 0) = \mathcal{P}(\text{Im}\{\delta_j\} \neq 0) = \mathcal{P}(\text{Im}\{\delta_i\} \neq 0) = P^2$, given the symmetry of the problem. Therefore, $E[||\mathbf{H}\delta||^2]$

$$\leq 2P \cdot [N \sqrt{6N + N(N-1)} + N^{\frac{3}{2}}(N-1)]. \quad (17)$$

As wireless noise reduces *i.e.*, $\sigma \rightarrow 0$ (as SNRs increase), MMSE becomes increasingly accurate and will have lesser bit errors $\Rightarrow P \rightarrow 0$. Therefore, according to Eq. 17, $E[||\mathbf{H}\delta||^2] \rightarrow 0$, and the effective noise given by Eq. 10 goes to zero, leading to good performance at high SNR. Note that a similar analysis can be performed when we try to decode \mathbf{q}_2 , resulting in the same conclusion. Further, in Figure 14 we illustrate the accuracy of MMSE solution in predicting the quadrant and position of the ML solution. Similar pre-decision schemes based on collected samples in the preceding run have been discussed in [37, 43], but the methods require iterative runs of PIC optimization which are not allowed in QA MIMO detection due to the programming time (§3.4).

Limitations of a Naïve Alternative Method. A natural alternative to our proposed strategy could be a method that treats either the quadrant or the position as part of noise, rather than fixing them to the MMSE solution. However, such a strategy will not be suitable for X-ResQ as it will significantly increase the effective noise in the problem.

If we try to perform split detection and try to detect \mathbf{q}_1 first then the equivalent system is given by,

$$\frac{\mathbf{y}}{2} = \mathbf{H}\mathbf{q}_1 + 0.5(\mathbf{n} + \mathbf{H}\mathbf{q}_2) \quad (18)$$

and effective noise power is given by,

$$\begin{aligned} & 0.25 \cdot (E[||\mathbf{n}||^2] + E[||\mathbf{H}\mathbf{q}_2||^2]) \\ &= 0.25 \cdot (\sigma^2 + E[\mathbf{q}_2^\dagger \mathbf{H}^\dagger \mathbf{H} \mathbf{q}_2]) \end{aligned}$$

We know that for any two random variables X and Y , $E[XY] = E[X \cdot E[Y|X]]$. Further, given \mathbf{n} , \mathbf{H} , and \mathbf{q}_2 are independent and $E[\mathbf{H}^\dagger \mathbf{H}] = N \cdot \mathbf{I}$ (Rayleigh fading channels)

$$\begin{aligned} &= 0.25 \cdot (\sigma^2 + E[\mathbf{q}_2^\dagger E[\mathbf{H}^\dagger \mathbf{H} \mathbf{q}_2 | \mathbf{q}_2]]) \\ &= 0.25 \cdot (\sigma^2 + E[\mathbf{q}_2^\dagger E[\mathbf{H}^\dagger \mathbf{H}] \mathbf{q}_2]) \\ &= 0.25 \cdot (\sigma^2 + N \cdot E[||\mathbf{q}_2||^2]). \end{aligned}$$

Now, $E[||\mathbf{q}_2||^2] = 2N$

$$= 0.25 \cdot (\sigma^2 + 2N^2) \quad (19)$$

If we try to decode \mathbf{q}_2 instead then the equivalent system is given by,

$$\mathbf{y} = \mathbf{H}\mathbf{q}_2 + (\mathbf{n} + 2\mathbf{H}\mathbf{q}_1) \quad (20)$$

and effective noise power is given by,

$$(E[||\mathbf{n}||^2] + E[||2\mathbf{H}\mathbf{q}_1||^2]) \quad (21)$$

$$= (\sigma^2 + 8N^2) \quad (22)$$

and hence decoding \mathbf{q}_1 will experience lower noise. Unlike before (Eq. 6), δ is now independent of \mathbf{H} and \mathbf{n} . Note that for both these scenarios, there is a significant interference; even when $\sigma \rightarrow 0$, effective noise power doesn't go to zero, leading to a bad performance even at high SNRs.

Summary. It is observed that, unlike the naïve method, the MMSE-assisted split method used in X-ResQ is effective, as MMSE becomes increasingly accurate with an increase in SNR, and the probability that it wrongly predicts both quadrant and position of the ML solution, is very low. Based on the analytical and empirical analysis, we draw the following conclusions:

- At high SNRs, the likelihood of both the quadrant and position of the MMSE solution being wrong is very low. As MMSE more accurately predicts at least one of quadrant or position correctly, the effective noise in the split Ising forms becomes very low. Therefore, the MMSE-assisted split detection is an effective method of transforming the original 16-QAM ML problem into an equivalent QPSK ML problem that is a much easier problem for QA and other PIC methods.
- We demonstrate through our experimental analysis (§6) that the split-detection formulation successfully mitigates the error floor phenomenon observed in QA and PIC MIMO detection with 16-QAM at high SNRs.
- Our split method has much less noise compared to the alternate splitting strategy that considers either the quadrant or position as part of the noise. Also, note that the method can be used along with any PIC algorithm.

C Parallel QA Hardware Programming

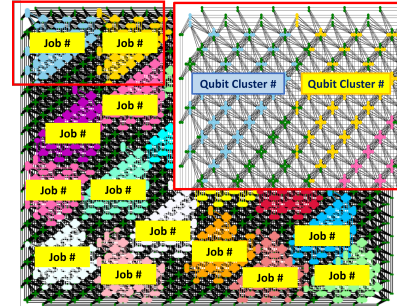


Figure 15: Clique embedding for parallel quantum processing on Pegasus-topology QA hardware using 16-user QPSK MIMO detection. Color coding represents different jobs in parallelism.

Embedding. Unlike Ising spin models, physical qubits on real hardware are rather sparsely connected. Thus, we need extra qubits to compile the Ising problem on the machine. This procedure is called *hardware minor embedding* [12]. We use the *clique embedding* technique that is designed to embed fully connected graph models into sparse graphs since our ML models are (nearly-)fully connected. Since the shape of each qubit cluster in clique embedding can be formed as a triangle (*e.g.*, Figure 15 on Pegasus-connectivity hardware and similarly on Zephyr hardware), it is an efficient way of using qubits for parallel QA as well, forming a rectangular shape

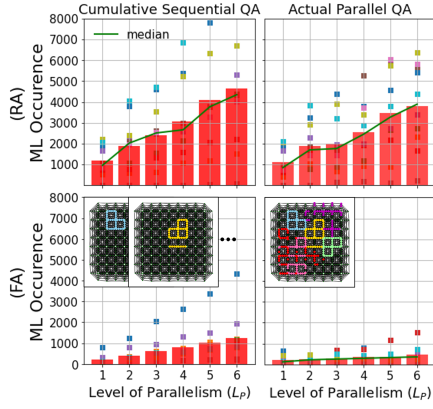


Figure 16: Average ML occurrence across instances (out of $L_p \cdot 5000$ QA samples per instance), comparing cumulative sequential QA runs to mimic parallel QA (left) against actual parallel runs (right). Both X-ResQ w/ RA (upper) and QuAMax w/ FA (lower) are tested. 4×4 16-QAM MIMO instances are used, where each symbol reports each detection instance (color: inst. index).

with two triangular clusters and thus promisingly supporting maximum parallelism. The embedding information can be prepared in a hash table based on the input size (N_V) and level of parallelism (L_p) without causing high overheads. However, since the machine we use for the X-ResQ implementation is a prototype machine with small qubit counts and also faulty qubits on the hardware need to be considered [51], we leave the customized maximum-parallel clique embedding and its hash table as our future work (with the full-size machine). Our parallel embedding used in this work leaves some qubits idle. Some weird shapes of the clique embedding are observed due to the (small-size) hardware geometry and faulty qubits.

Selection of QA Parameters. The choice of QA parameters in X-ResQ is decided by the brute-force exploration, following the same steps as the previous work [45, 46]. Ideally, QA parameter settings can be optimized per instance (also, different initialization states should be considered in RA). However, this is not practical due to the lack of theoretical means to acquire the best setting per instance. Thus, we use the empirically-obtained best-median setting obtained from a few tens of instances through the brute-force exploration: $2.2 \mu\text{s}$ anneal duration (T_a) and $\tau_p = 0.4$ switching point (i.e., $1.2 \mu\text{s}$ back-and-forth anneal with $1.0 \mu\text{s}$ pause). The precise QA schedules used in QuAMax and X-ResQ (and IoT-ResQ) as a form of $[T_a (\mu\text{s}), \tau]$ follows:

- QuAMax (FA with $1 \mu\text{s}$ pause at $\tau_p = 0.3$):
 $[0.0, 0.0] \xrightarrow{F} [0.3, 0.3] \xrightarrow{P} [1.3, 0.3] \xrightarrow{F} [2.0, 1.0],$
- X-ResQ (RA with $1 \mu\text{s}$ pause at $\tau_p = 0.4$):
 $[0.0, 1.0] \xrightarrow{R} [0.6, 0.4] \xrightarrow{P} [1.6, 0.4] \xrightarrow{F} [2.2, 1.0].$

For further details about the system implementations (e.g., embedding penalty term, coupling dynamic range, and post-processing), we refer to Ref. [45].

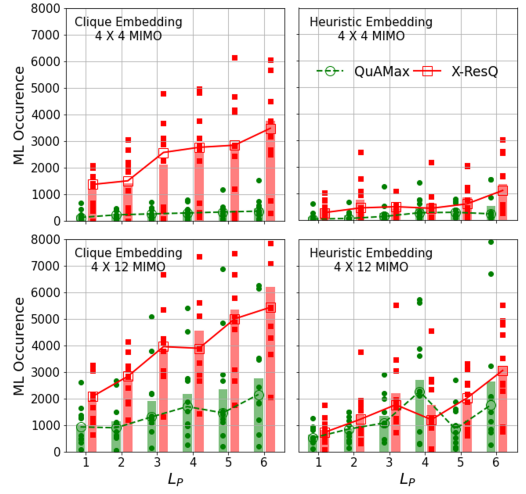


Figure 17: ML occurrence comparison between clique embedding (applied in the implementation) (left) versus heuristic embedding (right) for QuAMax (FA) and X-ResQ (RA) using fully parallel QA with 4-user MIMO at SNR 20 dB (upper: 4×4 , lower: 4×12). Bars report the mean while lines report the median.

Cumulative Sequential QA to Estimate Fully Parallel QA.

Current quantum annealer hardware features a limited number of qubits to test parallel QA, especially on the Advantage2 prototype while it is the state-of-the-art machine with the most advanced topology hardware (see Table 4). To test large parallelism beyond one that the current machine can support, we also estimate parallel QA performance by accumulating separate (sequential) QA results.

We empirically validate this by comparing the cumulative results against the actual fully parallel QA run with small-size instances. For the comparisons, we use the *ML occurrence* out of total anneal samples as a benchmark metric, instead of TTS. This is because parallel QA's motivation is to hit the global optimum more at the cost of more qubits, even though the quality of annealing in each task might get slightly degraded due to the extended Hamiltonian and cross-talk among qubits. Thus, we care about absolute anneal counts that hit the ML solution (global optimum) out of the total collected samples. We plot ML occurrence as a function of levels of parallelism in Figure 16 to compare cumulative sequential QA runs that mimic a parallel QA run against actual parallel run performance tested on the Zephyr Advantage2 machine. As expected, actual parallel runs obtain slightly degraded results than cumulative sequential results for both RA and FA. However, considering the nature of the probabilistic heuristics we believe these gaps are not quite critical (for these levels of parallelism) in that at $L_p = 6$ the median difference between them is less than 3% in both FA and RA, given 30,000 total collected samples ($N_a \cdot L_p$). Furthermore, the gaps are mainly caused by certain instances that work particularly well with

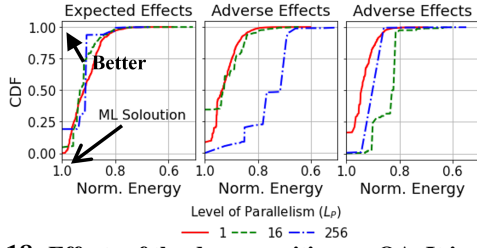


Figure 18: Effects of the decomposition on QA. It is observed that further decomposition could cause harder problems for QA, lowering the probability of finding the ML solution per anneal (P_G), despite the reduced search space.

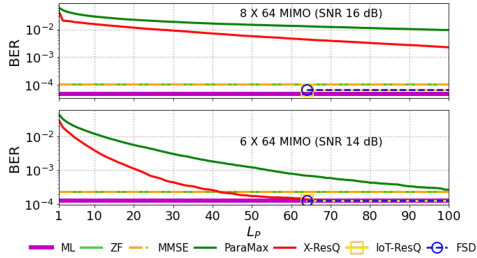


Figure 19: BER performance of classical X-ResQ across L_P for 64-antenna MIMO ($N_r = 64$) with 64-QAM ($|O| = 64$).

sequential runs. Thus, we believe the performance of cumulative sequential QA runs can be used to estimate fully parallel QA performance, to some extent.

Clique versus Heuristic Embedding for Parallel QA. We also compare two embedding methods for parallel QA, clique embedding and heuristic embedding. Figure 17 shows the comparison with 4-user MIMO detection instances. It plots the average ML occurrence out of $L_P \cdot 5000$ samples as a function of L_P for QuAMax and X-ResQ. We observe that while the ML occurrences keep increasing gradually as L_P increases with both the clique embedding and the heuristic embedding, the latter results in relatively smaller ML occurrences. Thus, the clique embedding (X-ResQ opts for) is a more efficient hardware embedding technique for parallel QA with MIMO detection instances.

Table 4: D-Wave quantum annealers and features: D-Wave 2000Q (DW2Q) & Advantage (DWAdv) machine.

	DW2Q	DWAdv	DWAdv2 (prototype)	DWAdv2 (full-size)
Release Year	2017	2020	2024	(exp) 2025
Topology	Chimera	Pegasus	Zephyr	Zephyr
Qubits	2000+	5000+	1000+	7000+
Connectivity	6	15	20	20

D Adverse Effect of Problem Decomposition in QA Optimization

Regarding decomposition-based parallel QA MIMO detectors, we experimentally found that decomposed subproblems could be even harder problems than the original problem for

QA, which does not occur in the case of deterministic classical solvers like the greedy search in FSD. We plot the CDF of the normalized Ising energies of the collected samples out of QA (FA) runs with different levels of parallelism in Figure 18 for three 4×4 MIMO detection instances at SNR 20 dB. Higher levels of parallelism imply further decomposition (L_P is equal to $16^{N_{fs}}$ with $N_{fs} = 0, 1, 2$) and we focus on only a subproblem that contains the global optimum. While the left panel shows the expected impact of the decomposition approach, where further decomposition improves optimization quality, the others show the adverse effects where further decomposition leads to worse results despite the reduced search space. Among 50 tested instances, less than 5–10 instances follow the expected effects; similar phenomena have been also observed for different modulations and RA, especially with $N \times N$ MIMO ($N_t = N_r$). Considering the reduced search space in subproblems compared to the one in the original problem, these are unexpected results (further decomposition makes search space size $2^{16-4N_{fs}}$ exponentially smaller). While it is difficult to explain the clear reason, it may be due to the Ising updates for decomposed subproblems resulting in critical coefficient values to analog-related noise on the machine (§3.3) (e.g., very large coefficient values).

E Negative Results with 64-QAM

In Figure 19, we compare the BER performance of classical X-ResQ against that of other MIMO detectors for 64-QAM massive MIMO systems with 64 receive antennas. While X-ResQ demonstrates promising results for 16-QAM and lower, it is observed that its performance deteriorates for 64-QAM and becomes even worse than linear detectors, ZF and MMSE. We see that in 8×64 MIMO, X-ResQ would require a very high L_P to achieve similar performance as classical conventional detectors. This is surprising in that X-ResQ is based on the MMSE solution and its final solution’s Ising energy is always lower than that of MMSE. In our future work, we will analyze this further as an effort to enable 64-QAM. Nevertheless, recall that X-ResQ’s RA optimization can be opportunistically skipped when MMSE performs well; for these MIMO scenarios, X-ResQ can rely only on the MMSE detector without conducting PIC optimization.

F IoT-ResQ Architecture

The overall architecture of IoT-ResQ is shown in Figure 20 (cf. Figure 4 for X-ResQ).

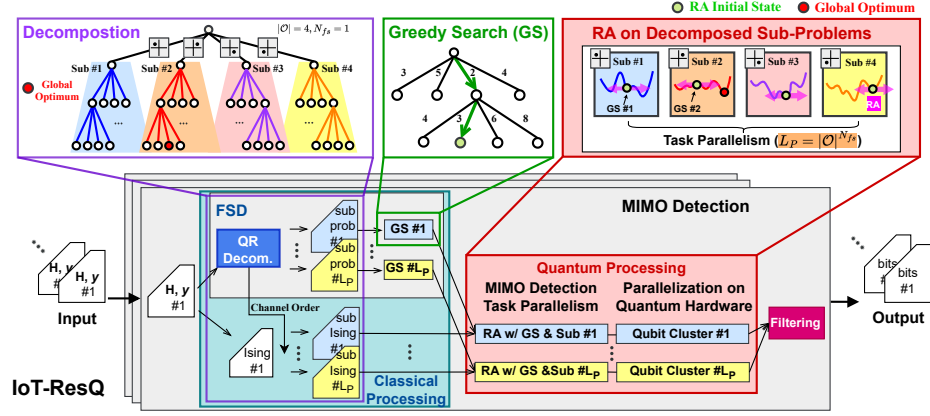


Figure 20: System architecture of IoT-ResQ [47] consisting of FSD and RA (decomposition-based parallel RA). Since IoT-ResQ relies on the (FSD) decomposition approach, only a *single* sub-problem retains the global optimum (sub-Ising model #2 in this example).