Physics-Inspired Heuristics for Soft MIMO Detection in 5G New Radio and Beyond

MINSUNG KIM*,†,*, SALVATORE MANDRÀ$^{\dagger,\top}$, DAVIDE VENTURELLI*, KYLE JAMIESON*, Princeton University*, NASA Ames Research Center, QuAIL†, USRA Research Institute for Advanced Computer Science*, KBR, Inc.$^{\top}$

Overcoming the conventional trade-off between throughput and bit error rate (BER) performance, versus computational complexity is a long-term challenge for uplink Multiple-Input Multiple-Output (MIMO) detection in base station design for the cellular 5G New Radio roadmap, as well as in next generation wireless local area networks. In this work, we present **ParaMax**, a MIMO detector architecture that for the first time brings to bear physics-inspired parallel tempering algorithmic techniques [28, 50, 67] on this class of problems. ParaMax can achieve near optimal maximum-likelihood (ML) throughput performance in the Large MIMO regime, Massive MIMO systems where the base station has additional RF chains, to approach the number of base station antennas, in order to support even more parallel spatial streams. ParaMax is able to achieve a near ML-BER performance up to 160 × 160 and 80 × 80 Large MIMO for low-order modulations such as BPSK and QPSK, respectively, only requiring less than tens of processing elements. With respect to Massive MIMO systems, in 12 × 24 MIMO with 16-QAM at SNR 16 dB, ParaMax achieves 330 Mbits/s near-optimal system throughput with 4-8 processing elements per subcarrier, which is approximately 1.4× throughput than linear detector-based Massive MIMO systems.

CCS Concepts: • **Networks → Wireless access points, base stations and infrastructure**.

Additional Key Words and Phrases: Parallel Tempering, Massive MIMO, MU-MIMO Detection, 5G

1 INTRODUCTION

Multi-User Multiple-Input Multiple-Output (MU-MIMO) has proven an essential technique to maximize capacity in many different kinds of wireless systems such as 802.11 wireless LAN and 5G New Radio cellular networks. In MU-MIMO, the uplink receiver (*i.e.*, an *access point*—AP—in a wireless LAN, or a *base station*—BS—in a cellular network) with multiple antennas supports many users simultaneously by striping data over parallel streams (a technique known as *spatial multiplexing*), and thus enables significantly higher data capacities. In an ideal world, the number of parallel streams that MU-MIMO can support would be the lesser of the number of mobile users and the number of radios at the base station, and overall system capacity would increase proportionally to the number of spatial streams.

In practice, however, the *channel hardening* phenomenon complicates this situation, in the following way. MU-MIMO requires signal processing to disentangle the spatial streams from each other, a technique called *MIMO detection*. For a base station with as many antennas as radio front ends, when the number of users approaches the number of base station antennas, MIMO detection becomes extremely difficult resulting in poor performance for conventional linear detection algorithms [76]: this is the *Large MIMO* regime that lies along the points where the number of users N_t equals the number of base station antennas N_r, as depicted in Figure 1.[1] For Large MIMO, there exist *maximum-likelihood* (ML) *exact* solvers, that can achieve the lowest possible bit error rate and, therefore, restore a high throughput. Unfortunately, these detection algorithms come at the expense of an exponential increase of the required computational resources as MIMO size increases, eventually becoming infeasible for many users because of the processing time limits in wireless

[1] For simplicity, we call $N_t \times N_r$ MIMO regimes, "Large MIMO" when $N_t = N_r$, while "Massive MIMO" when $N_t < N_r$, regardless of N_t size.

Author's address: Minsung Kim*,†,*, Salvatore Mandrà$^{\dagger,\top}$, Davide Venturelli*, Kyle Jamieson*, Princeton University*, NASA Ames Research Center, QuAIL†, USRA Research Institute for Advanced Computer Science*, KBR, Inc.$^{\top}$.

2021. Manuscript submitted to ACM

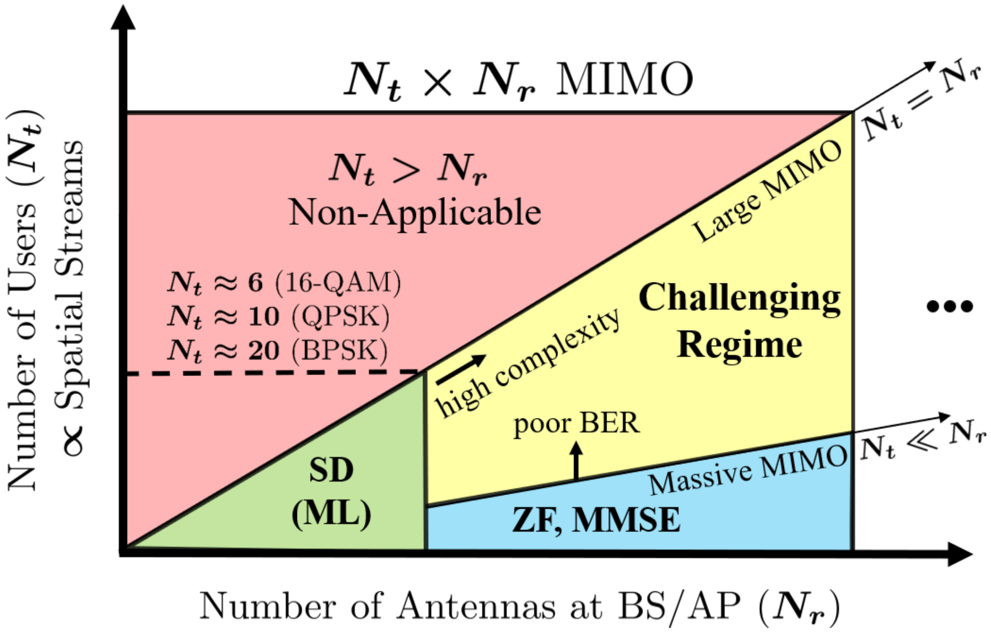


Fig. 1. Fundamental MIMO regimes in 5G New Radio and next generation local-area networks, and approximate feasibility of various detection approaches.

systems. For example, at most three milliseconds of BS's computation are available for both the 4G LTE uplink and downlink [15, 76].

Massive MIMO systems such as LuMaMi and Lund (6-12 users, 100-128 BS antennas) [44, 59, 69], Argos (eight users, 96 BS antennas) [62, 64], BigStation [76], Agora [18], and Samsung's 5G base stations (16 users, 64 BS antennas) [61] mitigate channel hardening in the following way. Since linear detectors such as *Zero-Forcing* (ZF) and *Minimum Mean-Squared Error* (MMSE) can achieve near-ML performance when the wireless channel is well-conditioned, systems that use many more base station antennas than users/spatial streams (*i.e.,* $N_r \gg N_t$ for $N_t \times N_r$ MIMO) may offer each base station radio a choice of one out of a number of antennas to use. This largely negates the effect of channel hardening, but requires base station antennas numbering a sufficient factor greater than users (*e.g.* $N_r \geq 10N_t$ [8] or $N_r \geq 4N_t$ for 16 users or below [62], while there is no proven rule-of-thumb of N_r/N_t that maximizes the spectral efficiency [8]), as shown in Figure 1, to achieve the full throughput of N_t spatial streams. In addition, the deployment of larger numbers of antennas eventually becomes challenging from a practical standpoint, most acutely in wireless local area networks, but also in small, densely-deployed 5G base stations where form factors preclude excessive numbers of antennas, and eventually in normal base stations where tower size faces practically limited.

In this paper, we take a complementary approach to Massive MIMO: we begin with a particular Massive MIMO configuration in which the number of base station antennas is practically at its maximum, and then ask the question *how can performance be further improved via additional spatial streams?* The answer lies in a fusion of two preceding ideas: add radio chains at the Massive MIMO base station to equal the number of antennas, and at the same time, utilize near-ML detection algorithms. This pushes us out towards the upper-right corner of the space in Figure 1 and maximizes computational complexity, yet offers the promise of the greatest spatial multiplexing gains, given our practical constraint on base station antenna count.

A shift to Physics-inspired approaches. Over the last few years, there has been a surge of interest in alternative computation approaches to reduce the complexity of current detectors by leveraging algorithms that relate optimization convergence to Physics principles. This interest is further accelerating in view of experimental initiatives featuring hardware-native implementations of these approaches, using both quantum and classical physics-based computations [4, 13, 24, 25, 36, 38, 39]. One common aspect of these algorithms is that they frame the computational problem as an energy minimization problem of a magnetic spin system, also known as the *Ising spin model* [32]. Beside being an important model to understand the physics behind magnetic systems, *any* NP computational problem can be expressed as the energy minimization of an appropriate Ising spin model [43] (that is, the Ising spin model is NP-Complete [74]). In this regard, physics-inspired algorithms can be seen as parametric "black boxes" that accept an Ising spin problem as input, and output the configuration with lowest associated energy. What distinguishes one algorithm from another is the underlying mechanism used to find the global minimum, which corresponds to the ML optimal solution in MIMO detection.

This paper presents the design and implementation of **ParaMax**, a soft MU-MIMO detector system for Large and Massive MIMO networks that uses *parallel tempering*, a physics-inspired heuristic algorithm, on classical platforms. ParaMax operates flexibly in parallel for any number of available processors, supporting fixed latency and highly-scalable parallelism. We design the *ParaMax Ising Solver* (§4.1), a parallel tempering-based solver that is tailored for MIMO detection, implemented as a fully classical algorithm that does not require any specific hardware, and integrate it into the overall design of our system (§4.2). We also introduce a new algorithm (§4.2.2) to generate soft information for heuristic detectors that enables a more reliable detection and decoding. The proposed algorithm utilizes heuristic detection outputs and generates soft information, defined as the bitwise detection confidences that implicitly take channel conditions and noise into consideration. To our best knowledge, this is the first application of parallel tempering to wireless networks, and ParaMax is the first heuristic-based MIMO detector that demonstrates near-ML performance for both very Large and Massive MIMO successfully.

Our experiments show that ParaMax achieves a constantly-increasing performance as the number of processing elements increases. In the case of lower-order BPSK and QPSK modulations, very large MIMO of 160 × 160 and 80 × 80 respectively, can achieve near-ML performance for less than tens of processing elements, as depicted in Figure 2. With respect to Massive MIMO systems, in 12 × 24 MIMO with 16-QAM modulation at SNR 16 dB, ParaMax achieves a 330 Mbits/s near-optimal system throughput with 4-8 processing elements (PEs) per subcarrier, approximately 1.4× better throughput than linear detector-based Massive MIMO systems.

2 BACKGROUND

This section introduces background knowledge, indicating relevant literature. Sections 2.1 and 2.2 respectively explain ParaMax's algorithmic foundations, simulated annealing and parallel tempering. Section 2.3 describes the MU-MIMO model and detection problem.

2.1 Simulated Annealing

Simulated annealing (SA) is a classical heuristic optimization technique typically used to find the state or *configuration* $\mathbf{s}$ with the lowest energy of *Ising spin* problems, where $\mathbf{s}$ is a vector consisting of $\{s_1, s_2, \cdots, s_{N_V}\}$ spins, with each spins s_i assuming the values {-1, +1}. In general, the energy objective function of Ising spin problems (also called *Hamiltonian*)

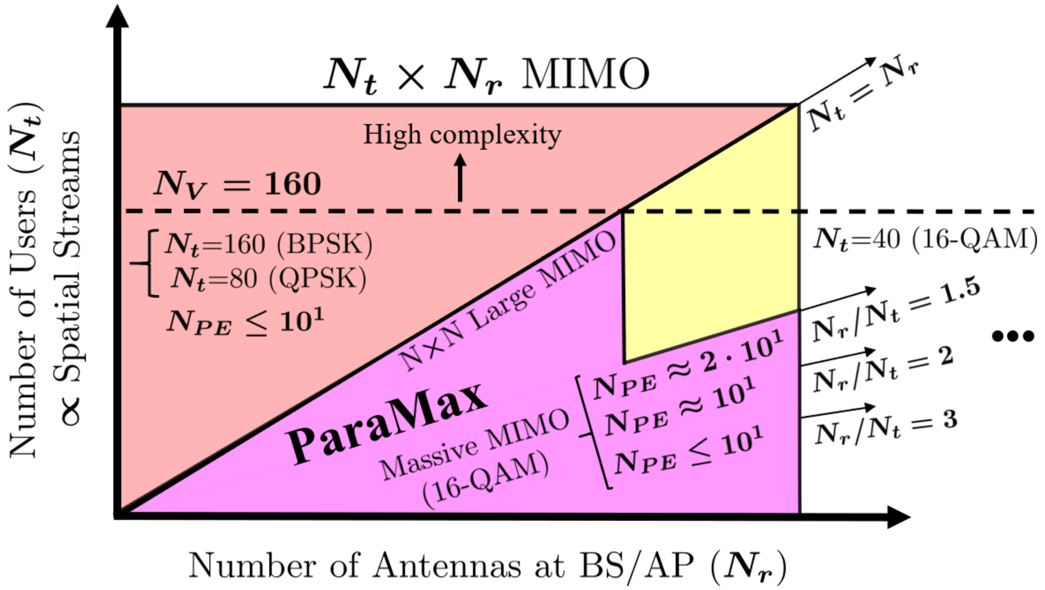


Fig. 2. Summary of ParaMax's feasible MIMO regimes (*cf.* Figure 1) and required processing element count N_{PE} per subcarrier.

is represented as a quadratic cost function of the following form:

$$\mathcal{H}(\mathbf{s}) = \sum_{ij} g_{ij} s_i s_j + \sum_i f_i s_i, \tag{1}$$

with $g_{ij} \in \mathbb{R}$ being *(anti-)ferromagnetic couplings* that indicate a preference of correlation ($s_i = s_j$ or $s_i \neq s_j$) between two spins, and $f_i \in \mathbb{R}$ *local magnetic fields* that individually act on $s_i = \pm 1$. Any optimization problem, including MIMO detection, can in theory be translated to an Ising spin model by properly choosing the $\{g_{ij}\}$ and $\{f_i\}$ [43, 74].

Simulated Annealing (SA) Heuristic. SA is inspired by the physical process of *annealing*, where a metallic material is slowly cooled from high temperature to eventually reach a molecular state or atomic configuration where the potential energy of the material is minimized. SA numerically emulates this process in order to find the global optimum (or *ground state*) of Eq. 1. To enable SA, it is necessary to simulate a *thermal bath* which imitates the cooling or annealing process interacting with the Ising spin model. More precisely, the probability that a given spin-configuration $\mathbf{s}$ is explored by the Ising spin system at a given *inverse temperature* $\beta = 1/T$ follows the *Gibbs distribution* $p(\mathbf{s}) = \exp[-\beta\mathcal{H}(\mathbf{s})]/\mathcal{Z}$, with $\mathcal{Z}$ usually called *partition* function [12, 21]. As the temperature T is lowered, the probability $p(\mathbf{s})$ of finding a state $\mathbf{s}$ with an energy larger than the minimum energy becomes exponentially lower. Therefore, sampling from the low-temperature Gibbs distribution allows rapid detection of the spin configuration with the lowest energy with high probability.

However, the calculation of the partition function $\mathcal{Z}$, and thus $p(\mathbf{s})$, is computationally challenging, particularly for low temperatures. To avoid the direct calculation of the Gibbs distribution $p(\mathbf{s})$, Metropolis *et al.* [50] proposed the use of Markov chain processes to help the system emulate the annealing and heuristic exploration of configurations at a given temperature. Specifically, they proposed a random process to "flip" a spin, with probability depending only on temperature and the Hamiltonian (but not on $\mathcal{Z}$), *i.e.*:

$$p(s_i \rightarrow -s_i) = \min\left\{1, e^{-\beta\Delta\mathcal{H}}\right\}, \tag{2}$$

with $\Delta\mathcal{H}$ the variation of energy once the spin s_i ($\forall i$) is flipped for a given initial configuration. Hence, moves that would eventually reduce the overall energy of the spin system are always accepted. Otherwise, there is a chance that such spin flip is either accepted or rejected. Metropolis *et al.* showed that the spin system will eventually thermalize to the corresponding temperature if the rejection rule in Eq. 2 (also called *Metropolis updates* or *sweeps*) is iteratively applied. Therefore, it is in principle possible to find the lowest energy spin configuration and, consequently, the solution to the original problem, by starting from a very large temperature and slowly decreasing it by iteratively applying the rule in Eq. 2.

2.2 Parallel Tempering

SA guarantees that the spin system will eventually find the lowest energy spin configurations if the temperature is lowered slowly enough. However, for hard optimization problems, it may require an exponentially long time. Indeed, a rugged energy landscape "traps" the spin system in local minima which are hard to escape: *parallel tempering* [67] helps the spin system escaping local minima and, therefore, thermalize faster at a low temperature. The basic principle of parallel tempering is simple: instead of a a single spin system, different *replicas* are simulated in parallel, each with a different temperature. After a certain amount of Metropolis updated, the temperatures of the two replicas r_1 and r_2 are exchanged following the updating rule:

$$p(r_1 \leftrightarrow r_2) = \min\left\{1, e^{\Delta\beta\Delta\mathcal{H}}\right\}, \tag{3}$$

with $\Delta\beta$ and $\Delta\mathcal{H}$ being the difference in the inverse of temperature and the difference in energies of the two replicas respectively. As one can see, two temperatures are always exchanged if a replica at higher temperature has a lower energy than a replica with a lower temperature. Otherwise, the exchange of the two temperatures is either accepted or rejected accordingly to Eq. 3. In a variety of hard optimization problems, parallel tempering drastically speeds up the thermalization of the spin system [37, 68, 78], including benchmark against quantum annealers [45–47].

2.3 MIMO Detection

The input and output relationship of a spatial multiplexing MIMO system (per subcarrier in OFDM systems [54]) with N_t input antennas at user side (or N_t single-antenna users for simplicity) and N_r output antennas ($N_t \leq N_r$) with N_R radios ($N_R \leq N_r$) at the receiver side is described as $\mathbf{y} = \mathbf{H}\bar{\mathbf{v}} + \mathbf{n}$. With $N_t \leq N_R$ (*i.e., $N_t \times N_r$* MIMO with N_t radio streams), here $\mathbf{y} \in \mathbb{C}^{N_r}$ is the received vector perturbed by *additive white Gaussian noise* (AWGN) $\mathbf{n} \in \mathbb{C}^{N_r}$, $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is the wireless channel, and $\bar{\mathbf{v}} \in O^{N_t}$ is the transmitted set of N_t symbols with *constellation O* (*e.g.*, 4-, 16-, 64-QAM), representing $N_V = N_t \log_2 |O|$ bits per channel use, with $|O|$ the size of the modulation. MIMO detection at the receiver side (AP or BS) is a technique to find a candidate solution $\hat{\mathbf{v}} \in O^{N_t}$ with an objective of detecting the transmitted symbol vector (*i.e.*, the objective is $\hat{\mathbf{v}} = \bar{\mathbf{v}}$) based on the received signal $\mathbf{y}$ and estimated $\mathbf{H}$. Pilot symbols enable the estimation of $\mathbf{H}$.

Maximum Likelihood Detection. *Maximum likelihood detection* (ML detection) is optimal in the sense that it minimizes the error probability of the detection. It is defined as

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

The search set of $\mathbf{v}$ (*i.e.*, the *search space* $\mathbb{S} \subseteq O^{N_t}$) is the set of possible solutions that the optimizer can take into account. Each element $\mathbf{v}$ in the search space is a candidate solution with which the values of the *ML objective function* $\mathcal{D}(\mathbf{v}) = \|\mathbf{y} - \mathbf{H}\mathbf{v}\|^2$ in Eq. 4 (*i.e.*, Euclidean distances) are measured and compared with each other. The best candidate, with minimum value, becomes the ML solution $\hat{\mathbf{v}}$. Further, $\mathbb{S}$ is an indicator of complexity. In principle, the ML search involves all possible candidates (*i.e.*, $\mathbb{S} = O^{N_t}$) which makes the brute-force approach intractable for large MIMO sizes with high-order modulations.

Sphere Decoder. The *Sphere Decoder* (SD) achieves optimal performance even with $\mathbb{S} \subset O^{N_t}$ by applying an adaptable search constraint in a sequential manner [3, 17, 20, 70]. The SD transforms Eq. 4 into an equivalent tree search and applies tree pruning, visiting fewer nodes and leaves without loss of optimality. However, since it is an exact algorithm (*i.e.*, achieves ML performance), the search space for SD is still exponentially large in the worst case [27]. Further, because of its sequential nature, its processes cannot be fully parallelized and the complexity (latency) varies per detection, which is not desirable for hardware implementation.

3 RELATED WORK

In this section we introduce related work on MIMO detection.

Parallel Sub-Optimal Architectures. These approaches divide the optimal SD tree search into parallel tasks in order to make use of hardware containing many processing elements (PEs) such as a GPU or FPGA [31, 55], while search algorithms become approximate. For these methods such as the Fixed-Complexity Sphere Decoder (FCSD) [5–7, 33] and K-best SD [23, 42, 53, 79], $\mathbb{S}$ is a subset of O^{N_t}, so how to select $\mathbb{S}$ for comparing $\mathcal{D}(\mathbf{v})$ is a key factor. For instance, the FCSD splits the SD tree of N_t levels into two separate search areas, one for *full search* (FS) and the other for *greedy search* (GS). During the FS, the FCSD visits all nodes at the first N_{fs} levels and then switches to GS, where only one child node with minimum partial Euclidean distance is explored for the remaining levels ($N_t - N_{fs}$). This exploration process can run in parallel.[2] The FCSD results in $\mathbb{S}$ consisting of $|O|^{N_{fs}}$ candidate solutions. Here, N_{fs} is a controllable positive integer parameter that trades off the FCSD's detection performance with its computational complexity. Note that the complexity of the FCSD (even with small N_{fs}) is still larger than linear methods and the FCSD enables only $|O|^{N_{fs}}$ parallel processes such as 16, 256, 4096 for 16-QAM with $N_{fs} = 1, 2, 3$, respectively (*i.e.,* bounded complexity but not flexible). ParaMax features flexible and scalable parallelism.

Heuristics for MIMO Detection. Heuristic approaches inspired by Biology or Combinatorial Optimization methods such as genetic algorithms, reactive tabu search, and particle swarm optimization for MIMO detection exist [2, 29, 66], but significant performance gains are not observed. Some studies have used analog quantum hardware platforms [38, 39], but these are specialized platforms that not yet generally available. Other studies on Physics-inspired SA, Gibbs distribution, and quantum search algorithm show feasibility to some extent [2, 9, 19, 22, 26], but lack comprehensive evaluations for Large and Massive MIMO systems and comparisons against other state of the art detectors.

4 DESIGN

In this section, we describe the design of ParaMax: Section 4.1 introduces the key building block of ParaMax's design, a SA-parallel tempering solver. Section 4.2 describes the complete design of ParaMax. Section 4.3 then introduces a refinement of ParaMax, *2R-ParaMax*, which uses soft information to enhance performance at the cost of some computational complexity. We evaluate both designs in Section 6.

4.1 ParaMax Ising Solver (PMIS)

The *ParaMax Ising Solver* (PMIS) is the main solver module in ParaMax. It is based on SA, featuring a parallel tempering algorithm highly-tailored to optimize the Ising model of MIMO detection. PMIS is a completely classical algorithm that

[2] For the maximum effect of the algorithm, a channel ordering scheme is used to ensure users with poor channel are detected in the FS phase of the FCSD.

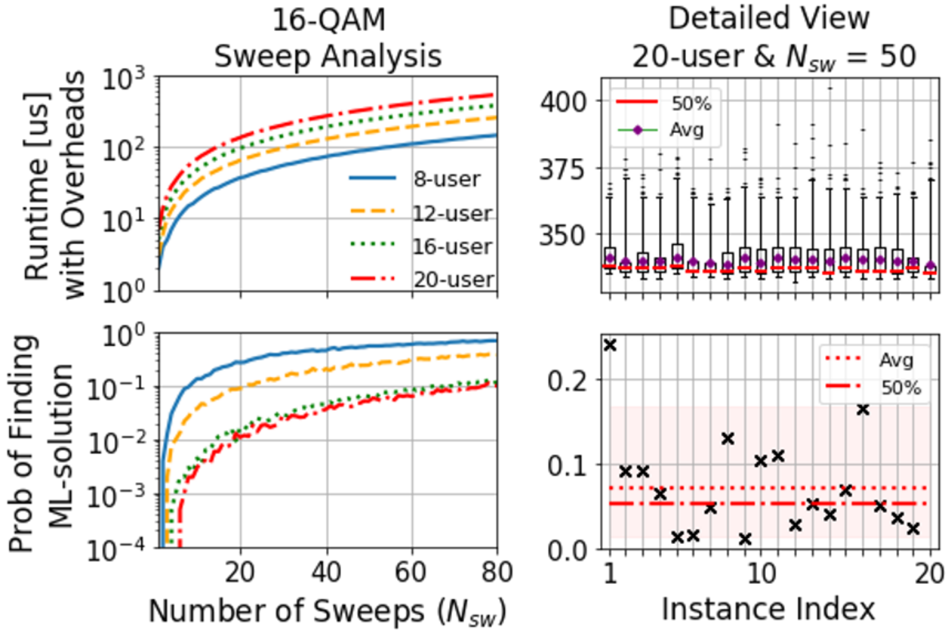


Fig. 3. Metropolis Sweep analysis of the PMIS solving MIMO detection (16-QAM at 20 dB SNR): ***overview (left)*** varying user numbers and ***detailed view (right)***.

does not require any specialized hardware for its implementation. It performs a local search by updating the spin values of a given random initial configuration according to Eq. 2. Each replica is associated with a different temperature, and temperatures may be exchanged according to the update rule in Eq. 3. Since the calculation of the energy associated with each replica can be trivially reduced to matrix-vector and vector-vector multiplications, couplings g_{ij} and local fields f_j of the Ising cost function $\mathcal{H}$ are stored as a matrix **G** and as a vector **f** respectively. Therefore, the calculation of $\mathcal{H}$, critical for the update rules in Eq. 2 and Eq. 3, is reduced to:

$$\mathcal{H}(\mathbf{s}) = \mathbf{s} \cdot [\mathbf{G} \cdot (\mathbf{s} + 2\mathbf{f})] /2, \tag{5}$$

with **s** the vector representing the spin configuration, where the factor 2 takes into account the symmetry of the matrix **G**. During our implementation, PMIS is optimized to maximize the performance for operations involving $N_V \lesssim 512$ spin variables which cover up to 512, 256, and 128 single-antenna users with BPSK, QPSK, and 16-QAM modulations, respectively. We provide further details on our PMIS implementation in Section 5.

Computational complexity. In MIMO detection, compute time complexity is a fundamental metric, along with BER and network throughput. From Eq. 5, it is clear that complexity is proportional to the square of the number of spin variables N_V ($= N_t \ \log_2 |\mathcal{O}|$). Therefore, recalling that every replica is independently updated, overall PMIS complexity scales as $N_V^2 \times N_{\text{repl}} \times N_{sw}$, with N_{repl} and N_{sw} the number of replicas and Metropolis sweeps, respectively.

Replicas and Metropolis Sweeps. To reduce the computational cost to the bare minimum, we have opted for a "bang-bang" parallel tempering approach. That is, only two replicas are used ($N_{\text{repl}} = 2$): one at very low temperature and one at higher temperature: the replica at lower temperature acts as a *greedy* searcher while the replica at a higher temperature acts as an *observer*. When the greedy searcher is stuck in a local minimum, the two replicas can exchange roles (*i.e.*, temperature) to resolve the bottleneck. While this has been successfully used in the context of quantum

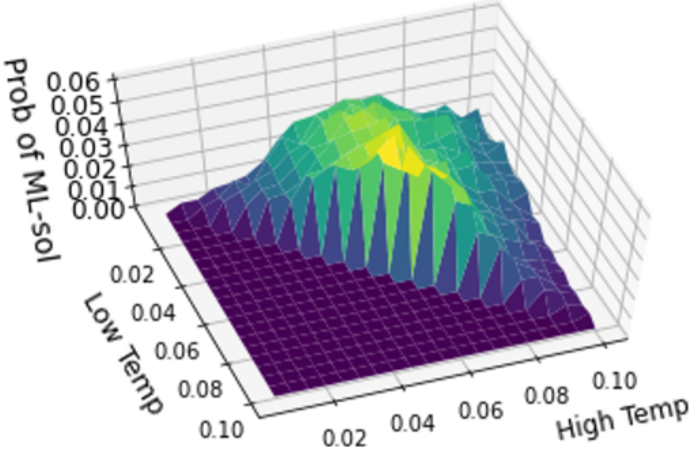


Fig. 4. Temperature range analysis of the PMIS solving MIMO detection for 20-user (16-QAM at 20 dB SNR).

annealing [77], ours is one of the first reports of a bang-bang parallel tempering schedule on a classical platform. To choose the number of Metropolis sweeps (§2.1), we empirically examine different numbers of sweeps from one to 80 with 16-QAM in Figure 3 out of 20 instances and 10,000 PMIS runs per instance. Not surprisingly, we observe a trade-off between latency (*upper*) and sampling quality (*lower*). We choose $N_{sw} = 50$ as an appropriate point that satisfies the current LTE standard's latency requirements.

Choice of temperature range. Unlike N_{repl} and N_{sw}, the temperature range does not influence ParaMax's complexity, so we choose a PMIS temperature range where it achieves the highest probability of finding the ML solution (*i.e.,* the ground state). For benchmarks we select the values $T_{\text{min}} = 0.05$ and $T_{\text{max}} = 0.06$, which perform well, as shown in Figure 4.

The foregoing description has described a single PMIS run. In ParaMax, multiple PMIS runs on multiple PEs in parallel, one PMIS run per PE. Each PMIS run is independent from the others, accepting the Ising model $\mathcal{H}$ of the MIMO detection as input, and outputting a candidate solution.

4.2 ParaMax Design

In this section we describe the complete ParaMax design (Figure 5). We describe the function of each block required for MIMO detection in §4.2.1 and the soft output generator module in §4.2.2.

4.2.1 ParaMax Detection Algorithm. We assume the base station receives a signal perturbed by AWGN and estimates the wireless channel as stated in Section 2.3.

1. ML-to-Ising Reduction. The procedure and the generalized formula of reducing the MIMO detection $\mathcal{D}(\mathbf{v})$ (= $\|\mathbf{y} - \mathbf{H}\mathbf{v}\|^2$ from Eq. 4) to the Ising form $\mathcal{H}(\mathbf{s})$ using the *spin-to-symbol mapping* were first introduced in [38]; our system assumes the same mapping. In the mapping, N_V spins represent all possible N_t symbol combinations (*i.e.*, $\log_2 |O|$ spins for a possible symbol per user), so its ground state always corresponds to the ML solution.

2. PMIS Parallel Processing. Each PMIS run samples an independent solution candidate (*i.e.,* an Ising configuration $\mathbf{s}$). Since detection is based on a heuristic, a single PMIS run's solution may not be optimal, and thus multiple runs are required to form a set of candidate solutions, gradually increasing the probability of collecting the optimal ML solution. Since each PMIS optimization run on a single PE completely independent of the others, ParaMax flexibly

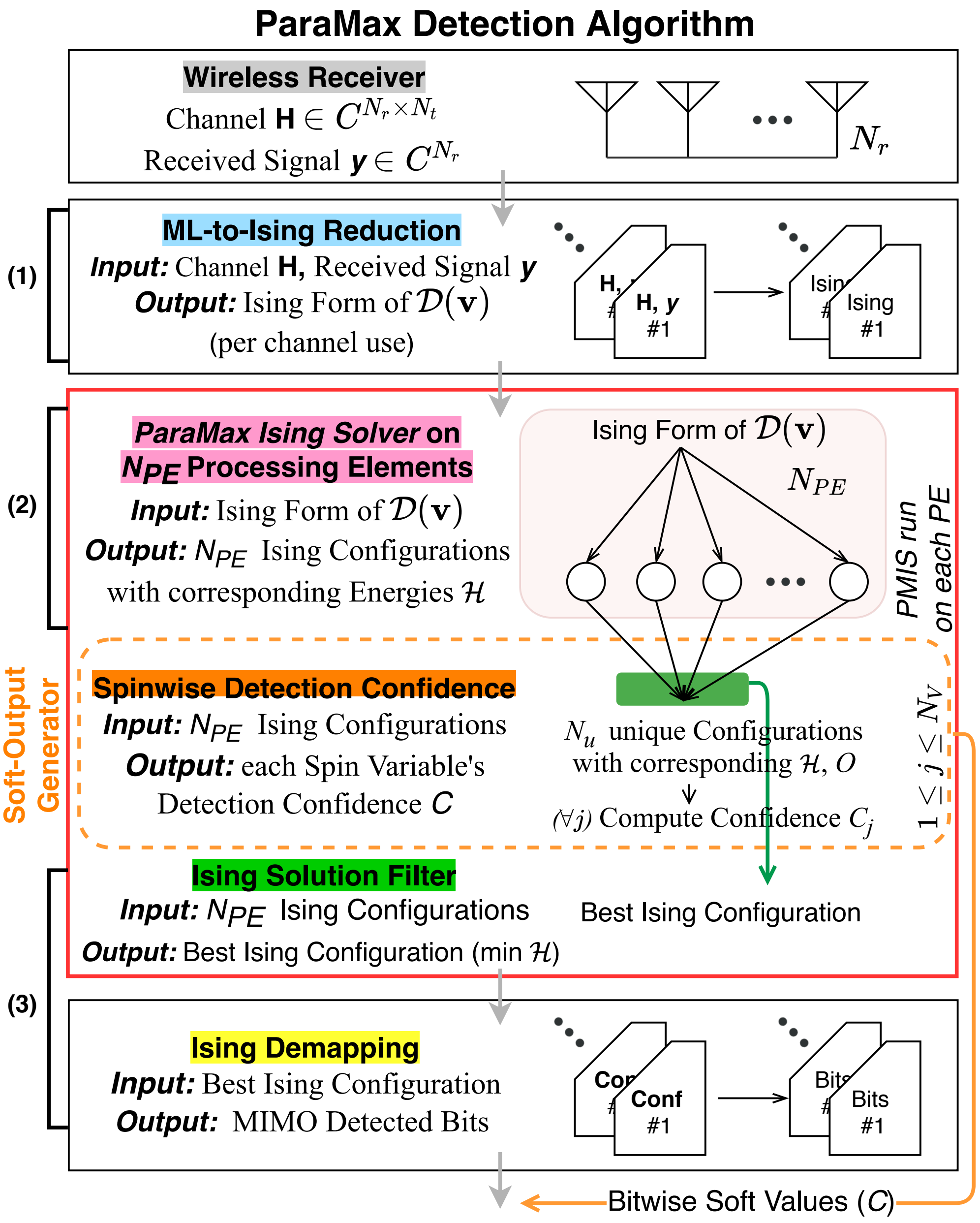

ParaMax Detection Algorithm
Wireless Receiver
Channel $\mathbf{H} \in C^{N_r \times N_t}$
Received Signal $\boldsymbol{y} \in C^{N_r}$
N_r
(1)
ML-to-Ising Reduction
Input: Channel **H,** Received Signal $\boldsymbol{y}$
Output: Ising Form of $\mathcal{D}(\mathbf{v})$
(per channel use)
H, y #1
Ising #1
(2)
ParaMax Ising Solver on
N_{PE} Processing Elements
Input: Ising Form of $\mathcal{D}(\mathbf{v})$
Output: N_{PE} Ising Configurations
with corresponding Energies $\mathcal{H}$
Ising Form of $\mathcal{D}(\mathbf{v})$
N_{PE}
PMIS run on each PE
Soft-Output Generator
Spinwise Detection Confidence
Input: N_{PE} Ising Configurations
Output: each Spin Variable's
Detection Confidence C
N_u unique Configurations
with corresponding $\mathcal{H}$, O
$(\forall j)$ Compute Confidence C_j
$1 \leq j \leq N_V$
Ising Solution Filter
Input: N_{PE} Ising Configurations
Output: Best Ising Configuration (min $\mathcal{H}$)
Best Ising Configuration
(3)
Ising Demapping
Input: Best Ising Configuration
Output: MIMO Detected Bits
Conf #1
Bits #1
Bitwise Soft Values (C)

Fig. 5. Overview of ParaMax's detection algorithm.

operates on any number of independent processing elements (N_{PE}), with highly scalable parallelism. The number of available processing elements N_{PE} is equal to the number of PMIS outputs that the ParaMax system can generate with full parallelism.

3. Ising Solution Filter and Demapping. After all N_{PE} PMIS parallel runs, the corresponding N_{PE} outputs are

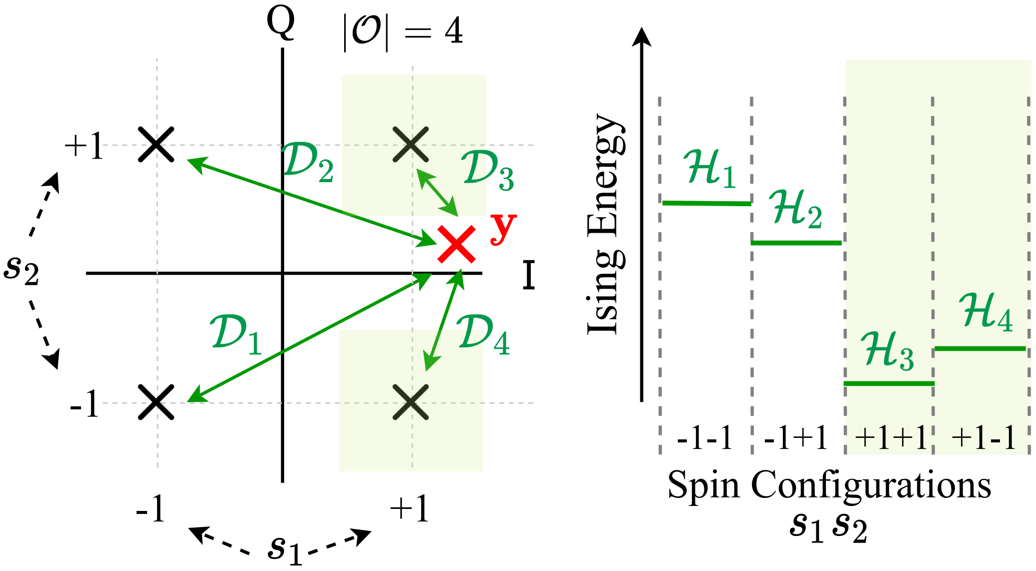


Fig. 6. Equivalent representations of 1×1 QPSK detection: Euclidean distances $\mathcal{D}(\mathbf{v})$ in the I-Q plane *(left)*, and Ising energies $\mathcal{H}(\mathbf{s})$ *(right)*. Shadings highlight likely solutions.

collected in a table of Ising spin configurations. Before further processing, the list is sorted in order of solution quality, based on the Ising energy $\mathcal{H}(\mathbf{s})$ of each output. The *Ising solution filter* returns only the configuration $\hat{\mathbf{s}}$ with the best (minimum) $\mathcal{H}(\hat{\mathbf{s}})$, which is equivalent to the wireless symbol $\hat{\mathbf{v}}$, *i.e.,* $\mathbf{v}$ with the minimum $\mathcal{D}(\mathbf{v})$ (among candidate solutions), after proper demapping (spins ⟶ symbols). Finally, $\hat{\mathbf{v}}$ is converted into N_V MIMO detected bits.

4.2.2 Spinwise Soft Information Output. For most heuristics-based solvers, only the lowest-energy Ising configuration is returned (regardless of how many times it occurs among N_{PE} PMIS outputs) and any outputs other than it are discarded. In ParaMax, however, we utilize all N_{PE} PMIS outputs to generate soft information (*i.e.,* detection confidences, for each spin in a given configuration). In general, soft-output MIMO detectors' soft values are utilized for iterative MIMO detection or channel coding [6, 40, 41, 58]. In this work, we design the former (2R-ParaMax) in Section 4.3.

ParaMax collects candidate solutions from N_{PE} independent PMIS runs. Among these, multiple occurrences of a certain spin configuration (with agreeing spin variables) are very likely to be observed, which could be used to identify spins easy (or hard) to detect (*i.e.* variables that are very likely to be assigned a certain value in the unknown optimal ML solution). Figure 6 shows an illustrative example of detecting a received 1×1 QPSK signal $\mathbf{y}$ in two equivalent representations, one in the I-Q plane with Euclidean distance $\mathcal{D}$ *(left)*, and the other with Ising energies $\mathcal{H}$ *(right)*. In this example, the first spin variable s_1 (corresponding to the symbol's real part) is likely to be detected as +1 for most PMIS runs, since the difference in Ising energy from all configurations that have $s_1 = +1$ (resulting in $\mathcal{H}_3$ or $\mathcal{H}_4$ in Figure 6, *right*) and $s_1 = -1$ (resulting in $\mathcal{H}_1$ or $\mathcal{H}_2$ in Figure 6, *right*) is significant. The spin s_1 is easy to detect compared to spin s_2 (corresponding to the symbol's imaginary part). Multiple occurrences of PMIS runs agreeing on $s_1 = +1$ indicate this, while PMIS runs will disagree on the value of s_2, because the two most frequent spin configurations' energies ($\mathcal{H}_3$ and $\mathcal{H}_4$) themselves disagree on the value of s_2.

This phenomenon becomes even clearer for high-order modulations, since in 16-QAM or higher modulations, the

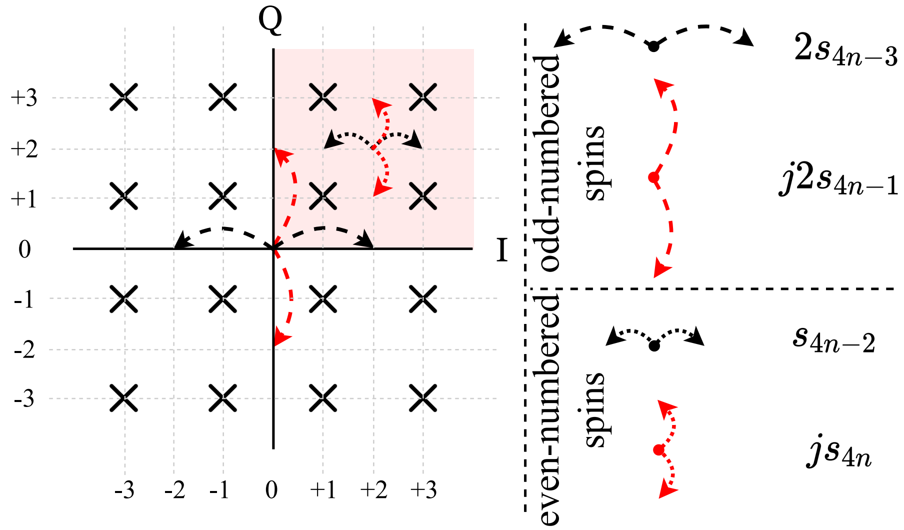


Fig. 7. Mapping between four Ising spins (variables) to n-th user's 16-QAM symbols (× symbols) on the constellation. Shading denotes an inner quadrant. In this mapping, odd numbered spins are easier to detect than even numbered spins.

value of the spin coefficients in the Ising spin-to-symbol mapping varies across spins. For example, Figure 7 shows the spin-to-symbol mapping of 16-QAM for the n^{th} user, where the user's possible symbol maps one-to-one with spins $s_{4n-3}, \ldots, s_{4n}$, that is $\mathbf{v_n} = (2s_{4n-3} + s_{4n-2}) + j(2q_{4n-1} + s_{4n})$. Here, spins s_{4n-3} and s_{4n-1}, the odd-numbered spins, determine the symbol's quadrant, while spins s_{4n-2} and s_{4n}, the even-numbered spins, determine the symbol in the given quadrant. Here, the odd-numbered spins for 16-QAM are in general easier to detect than the even-numbered spins because of higher robustness to AWGN (detection reliability). Table 1 presents empirical spinwise error rates of ParaMax for 8 × 8 16-QAM detection. These differences in robustness indicate that using ParaMax's soft information would be particularly helpful for further processing, similarly for unequal error protection (UEP) [10, 30, 48, 72, 73]).

Table 1. ParaMax's spinwise error rate (conditioned on 103,850 incorrect outputs) for eight-user, 16-QAM MIMO detection.

Mean Spinwise Error Rate			
Oven-numbered spins:		**Even-numbered spins:**	
$4n-3^{\text{rd}}$	$4n-1^{\text{st}}$	$4n-2^{\text{nd}}$	$4n^{\text{th}}$
0.167	0.152	0.329	0.352
≈ Either/both: 0.32		≈ Either/both: 0.68	

Soft information computation. Based on the equivalence of of the I-Q and spin configuration representations, the occurrence count of a given spin value for a certain spin s_j across all PMIS outputs samples the distance of the symbol corresponding to that spin's value, and hence estimates that spin's likelihood of correctness. More specifically, after collecting N_{PE} PMIS outputs sorted by Ising energies $\mathcal{H}_i$ $(1 \leq i \leq N_u)$, the system has N_u $(N_u \leq N_{PE})$ unique outputs with corresponding *occurrence counts* O_i $(1 \leq i \leq N_u)$. The detection confidence C_j of spin s_j $(1 \leq j \leq N_V)$ is defined as:

$$C_j = \left(\sum_{i=1}^{N_u} O_i^{s_j^i = s_j^1} \cdot \left| \frac{\mathcal{H}_i}{\mathcal{H}_1} \right| \right) \Bigg/ \left(\sum_{i=1}^{N_u} O_i \cdot \left| \frac{\mathcal{H}_i}{\mathcal{H}_1} \right| \right), \tag{6}$$

where $O_i^{s_j^i = s_j^1}$ is a count of occurrences of the i^{th} ranked configuration (defined in §2.1 on p. 3), only when the i^{th} configuration's j^{th} spin is equal to the first-ranked configuration's j^{th} spin (*i.e.*, $O_i^{s_j^i = s_j^1}$ is either O_i or zero). The spinwise detection confidences C_j ($0 < C \leq 1.0$) are the soft values ParaMax outputs in this step. Note that the reliability of each C_j increases as the best observed Ising energy among the collected N_{PE} outputs ($\mathcal{H}_1$) becomes closer to the unknown ground state (of energy $\mathcal{H}$), which implies as N_{PE} increases the quality of soft values improves.[3] Similar algorithm is introduced using quantum annealing [35], where only partial outputs are used.

4.3 2R-ParaMax: Iterative Soft Detection

We now introduce a method of using the soft information described in the prior section to enhance the operation of ParaMax. We call this protocol 2R-ParaMax. The main idea is to iterate the PMIS block twice, once for generating soft confidence information, and again to obtain a final detection result based on the confidences from the first iteration. Intermediate processing between the first and second iterations functions pre-decision of spins with high detection confidence. An error correction post-processing is applied at the end of the second round, both of which have linear complexity. The end result is a more accurate MIMO detection result, at the expense of a modestly increased latency, and so this might be employed for challenging wireless channels and/or large numbers of users. With reference to Figure 5, the structure of 2R-ParaMax PMIS block is shown in Figure 8. This block is replacing the third block marked in red of Figure 5. The other blocks are exactly the same as described in ParaMax. We also note that the soft information generated by the second round can also be used for the channel decoding or further iterations of the algorithm.

Intermediate Pre-decision. The intermediate pre-decision module identifies those spins with a high detection confidence (over a threshold C_{th}) from the first round of PMIS outputs in order to reduce the number of spin variables involved in second-round of PMIS runs, simplifying second-round detection. That is, if the j^{th} spin's detection confidence $C_j \geq C_{\text{th}}$ ($1 \leq j \leq N_V$), then we pre-decide the j^{th} spin variable to be the value of the corresponding spin of the best solution in the first round (*i.e.*, s_j^1).

After thus obtaining a pre-decided set of spin indices, the next step is to update the Ising form accordingly. For each spin index k in $\mathcal{F}$ and for each Ising problem index i, we set $f_i' = f_i + g_{ik} \cdot s_k^1$, if $i < k$ and $f_i' = f_i + g_{ki} \cdot s_k^1$, if $i > k$, and then remove f_k, g_{ik} and g_{ki}. The result is a reduced Ising problem that contains $N_V' = N_V - |\mathcal{F}|$ spins only. Table 2 summarizes the average ratio of decided spins ($|\mathcal{F}|/N_V$) and the success ratio ($|\mathcal{F}_{\text{ML}}|/|\mathcal{F}|$) of the pre-decision process. Here, $\mathcal{F}_{\text{ML}}$ denotes an index group of spins where spins decided by the pre-decision process are exactly the same as the corresponding spins in the ML solution. In 2R-ParaMax, we apply $C_{th} \geq 0.97$, which ensures $|\mathcal{F}_{\text{ML}}|/|\mathcal{F}| = 1.0$ (for lower N_{PE}, higher C_{th} applied). With the updated Ising form $\mathcal{H}'$ (with f' and g' for N_V' spins), we execute a second round of PMIS to generate N_{PE} outputs and then filter the best output consisting of N_V' spins with minimum Ising energy in terms of $\mathcal{H}'$. When the filtered configuration is combined with the pre-decided spins appropriately, the full configuration consisting of N_V spins can be restored and demapped into symbols. This full configuration is further compared against the best PMIS outputs of the first round based on the original Ising form $\mathcal{H}$. The final best configuration is returned as 2R-ParaMax's detection solution, which guarantees that 2R-ParaMax's bare minimum performance is ParaMax's performance.

5 IMPLEMENTATION

We now describe our ParaMax implementation.

Computing Environments. CPU-based experiments are executed on an Intel i9-9820X at 3.30GHz with 20 cores,

[3] While ParaMax is an inherent soft-output MIMO detector, requiring simple computations, conventional soft-output MIMO detectors require additional computations of exponential complexity, of log-likelihood ratio (LLR) for all coded bits to generate soft values at channel decoder [58, 65, 71].

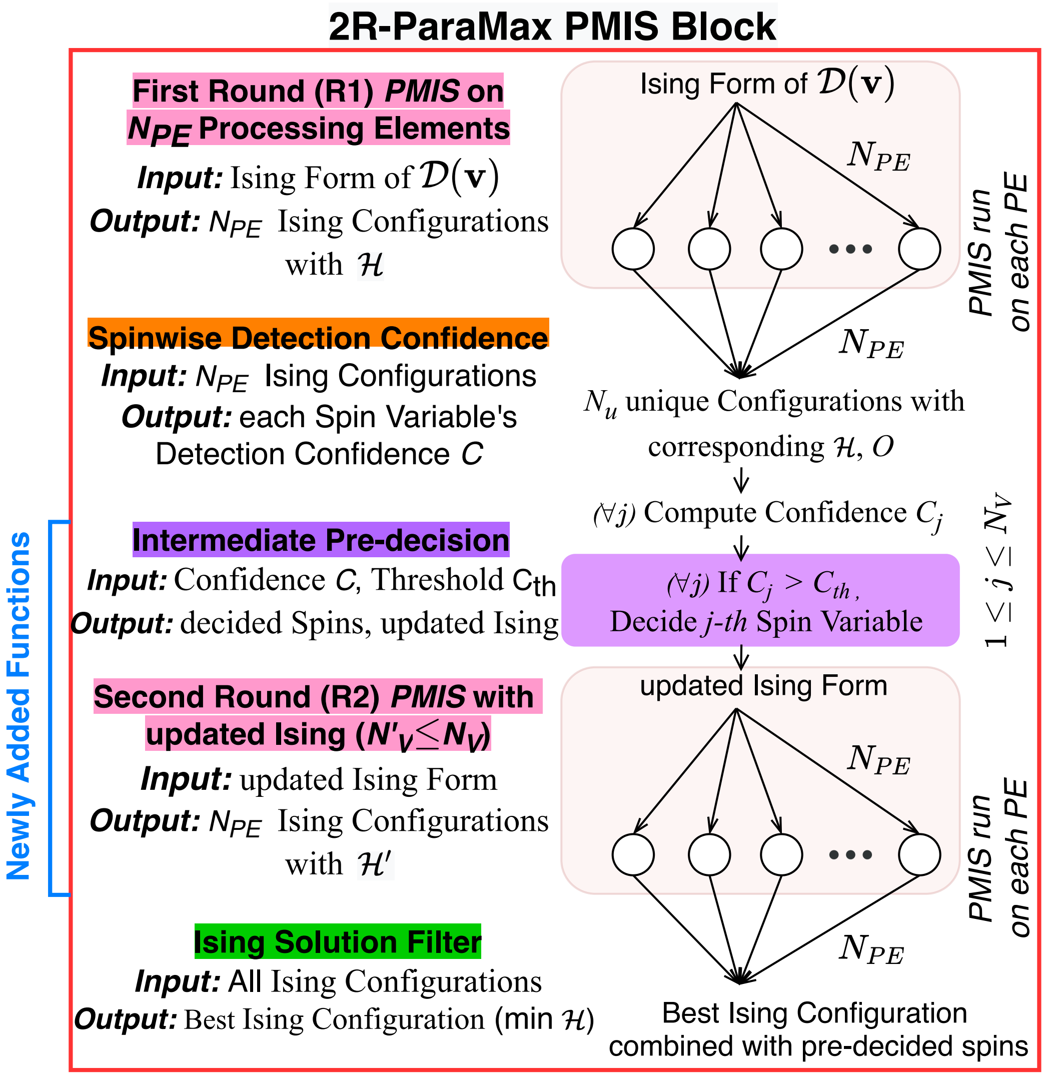

2R-ParaMax PMIS Block
First Round (R1) PMIS on N_{PE} Processing Elements
Input: Ising Form of $\mathcal{D}(\mathbf{v})$
Output: N_{PE} Ising Configurations with $\mathcal{H}$
Spinwise Detection Confidence
Input: N_{PE} Ising Configurations
Output: each Spin Variable's Detection Confidence C
Intermediate Pre-decision
Input: Confidence C, Threshold C_{th}
Output: decided Spins, updated Ising
Second Round (R2) PMIS with updated Ising ($N'_V \leq N_V$)
Input: updated Ising Form
Output: N_{PE} Ising Configurations with $\mathcal{H}'$
Ising Solution Filter
Input: All Ising Configurations
Output: Best Ising Configuration (min $\mathcal{H}$)
Newly Added Functions
Ising Form of $\mathcal{D}(\mathbf{v})$
N_{PE}
N_{PE}
PMIS run on each PE
N_u unique Configurations with corresponding $\mathcal{H}$, O
$(\forall j)$ Compute Confidence C_j
$(\forall j)$ If $C_j > C_{th}$, Decide j-th Spin Variable
$1 \leq j \leq N_V$
updated Ising Form
N_{PE}
N_{PE}
PMIS run on each PE
Best Ising Configuration combined with pre-decided spins

Fig. 8. Structure of 2R-ParaMax PMIS Block (*cf.* Figure 5's third block in red). The other blocks in 2R-ParaMax are exactly the same as ones in ParaMax.

Table 2. 2R-ParaMax's intermediate pre-decision process tested for 5,000 different instances of 20 × 20 16-QAM detection at SNR 20 dB on $N_{PE} = 200$.

C_{th}	**0.91**	**0.93**	**0.95**	**0.97**	**0.99**
$\lvert\mathcal{F}\rvert/N_V$	0.35	0.32	0.28	0.21	0.01
$\lvert\mathcal{F}_{ML}\rvert/\lvert\mathcal{F}\rvert$	0.97	0.99	1.0	1.0	1.0

2,189 threads, and 126 GB RAM. GPU-based experiments are tested based on the CUDA (Compute Unified Device Architecture [60]) 10.2 with GeForce RTX 2080 Ti of 4,352 CUDA cores and 68 streaming multiprocessors.

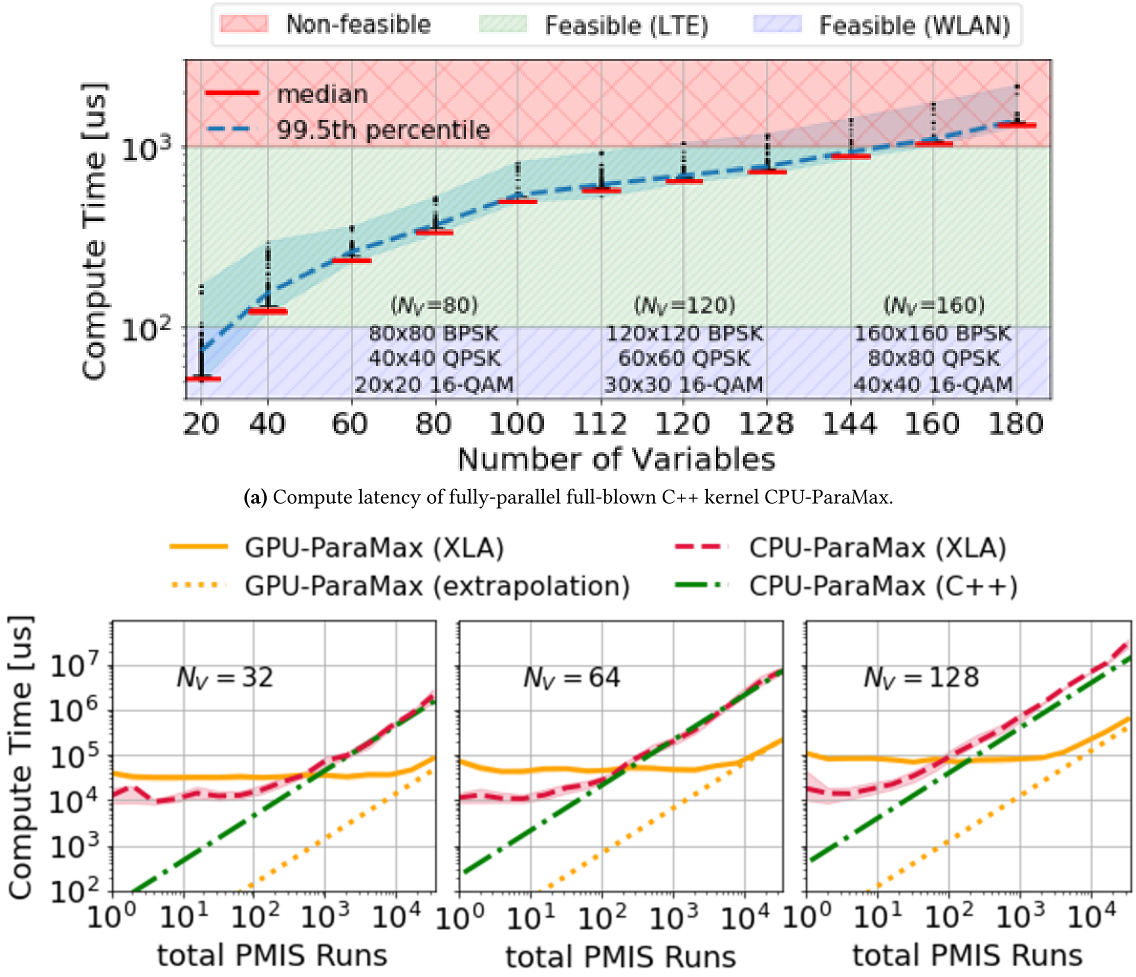
Wireless MIMO Channels. Both simulation-based and trace-driven real world wireless channels are used for our experiments. In the case of the simulation-based channel, independent and identically distributed (i.i.d) Gaussian channels with AWGN are synthesized for various SNR settings. For trace-driven channels, we use non-line of sight wideband MIMO channel traces at 2.4 GHz, between 96 base station antennas (N_r) and eight static users (N_t), the largest MU-MIMO dataset provided in Argos [63]. Among $N_r = 96$, we single out 8 to 32 (in steps of four) antennas to test the most challenging MIMO regimes (*e.g.* $N_t \geq N_r/4$). Since trace based channels include measured noise and limited user numbers, we use synthesized channels, unless otherwise stated, in order to precisely control SNRs and evaluate various MIMO regimes such as $N_t > 8$. Based on both channel settings, we generate large-scale Ising models $\mathcal{H}$ of MIMO detection (100,000-1,000,000 random instances per scenario) in order to measure detection BER up to $(N_V \cdot \text{total Insts})^{-1}$, approximately 10^{-7}.

PMIS CPU-Implementation. While the front-end of our PMIS implementation is in Python, the core is completely written in $C^{++}11$ standard [34]. We assign only a single core and a single thread (a single PE) to the calculation of each PMIS run by manually modifying the OpenMP [11, 14] and C++ parallelization settings. Furthemore, to maximize the performance of PMIS (to satisfy limited processing time in wireless standards), the following *innovations* have been implemented: **(1) Use of static memory:** Static allocation, unlike dynamic allocation, happens at global scope and it is pre-populated when the library is loaded. Moreover, since the size of arrays is known in advance, compilers can further optimize math operations on static arrays. **(2) Parameter pack expansion:** Loops in the matrix-matrix and vector-matrix multiplications are the most expensive part in PMIS implementation. To further reduce the computational cost, most of the critical loops are statically unrolled using features like the parameter pack expansion, introduced in the $C^{++}11$ standard. **(3) Intel SIMD instructions:** Most of the modern CPU architecture have intrinsic operations to allow multiple operations on contiguous arrays of floats. In PMIS, we have used Intel SIMD instructions to vectorialize operations like matrix-vector and vector-vector multiplications [56].

PMIS GPU-Implementation. The core design of CPU-ParaMax is based on a highly optimized C++ implementation of SA. Therefore, the natural extension of ParaMax to GPU consists into the implementation of the core SA engine to GPU. However, unlike CPUs, GPUs achieve the best performance for large arrays where multiple synchronous operations are applied at the same time. Indeed, while GPUs have more cores than CPUs, each single GPU core is typically much slower. Therefore, to maximize the performance of SA implemented on GPUs, we have designed a GPU kernel based on the JAX/XLA language that updates multiple PMIS runs at the same time. More precisely, the spin configuration $\mathbf{s}$ for a single PMIS run (in Eq. 5) is now extended to a matrix $\mathbf{S} (= \mathbf{s}^k)$, with k corresponding to the PMIS index. Since PMIS runs are completely independent from each other, $\mathbf{s}^k$ $(\forall k)$ can be updated independently and synchronously.

6 EVALUATION

In this section, we evaluate ParaMax in various aspects. Section 6.1 evaluates ParaMax's detection latency against other CPU- and GPU-based detectors. Section 6.2 illustrates sampling performance of ParaMax comparing against simulated annealing and required the number of processing elements for ParaMax to achieve near-ML performance in both Large and Massive MIMO. Section 6.3 and 6.4 show the ParaMax's bit error rate and system throughput performance respectively, compared against other state-of-the-art detectors in both Large and Massive MIMO.



(a) Compute latency of fully-parallel full-blown C++ kernel CPU-ParaMax.

(b) Single Device: 9.3 TFLOP GPU-ParaMax vs 0.028 TFLOP CPU-ParaMax.

Fig. 9. ParaMax Detection Latency.

6.1 Detection Latency

Fully-Parallel Full-Blown ParaMax. Figure 9(a) shows the detection time of ParaMax as a function of N_V ($= N_t \log_2 |\mathcal{O}|$) per channel use, where the background color coding indicates approximate feasibility for the wireless standards (WLAN and LTE). As N_V increases (*i.e.*, N_t and/or modulation increases), computing time tends to scale as N_V^2 (cf. N_r does not affect N_V and thus compute time). The available largest MIMO sizes that reach the borderline-limit of acceptable detecting time in the LTE standards are $160 \times N_r$, $80 \times N_r$, $40 \times N_r$ for BPSK, QPSK, 16-QAM modulation, respectively. While slight variations of runtime are observed that can cause overall latency increase and hardware synchronization issues, this can be resolved by an integrated system hardware since the origin of the variations is caused related to our system by the kernel allocation of jobs and concurrent (unrelated) system processes. In general, 2R-ParaMax requires approximately $1.4 - 1.6\times$ ParaMax latency.

CPU- vs GPU-ParaMax Comparison. Figure 9(b) shows the comparison of compute time between the XLA kernel (compiled for both CPU and GPU) and the original C++ kernel. The runtime for the C++ kernel has been obtained by computing the average runtime for a single PMIS run and then projected for multiple PMIS runs. XLA (Accelerated Linear Algebra) is a domain-specific compiler for linear algebra that can be compiled and optimized separately for either CPU or GPU. As one can see, for a sufficiently large number of parallel PMIS runs, while the kernel runtime compiled for CPUs have a consistent runtime with the full-blown C++ implementation of ParaMax, the kernel compiled for GPUs shows a speed-up, where total PMIS runs can be defined as N_{PE} multiplied by the number of subcarriers (N_{SC}). While GPU-ParaMax can achieve speed-up for over hundreds of PMIS runs, it cannot satisfy time requirements for standards. Recall that current GPUs are not designed to make full use of resources for small-size systems (*i.e.,* few PMIS runs). Thus we also extrapolate its performance to estimate what we could achieve in GPU without these limitations. Note that unlike on CPUs where a single core can be used to carry out any calculation, GPU cores are designed to work in concert to manipulate large block of data in parallel, and users cannot assign specific resources to a certain computation [31]. Therefore, we define a single PE for GPU-ParaMax as the extrapolation of a single PMIS run from large number of PMIS runs. In 5G New Radio, this extrapolation becomes more reasonable (while still being approximate), since 5G systems will support over three thousands of subcarriers, and slightly more time[4] (4 ms) than LTE (3 ms) will be allowed for enhanced mobile broadband (eMBB) [1, 18, 76] (cf. 1 ms for 5G Ultra-Reliable Low-Latency Communication (URLLC)).

	20-user MIMO (16-QAM)	**ZF-SIC**	**ParaMax**	**FCSD**		
				N_{fs}=2	N_{fs}=3	N_{fs}=4
	Parallelism #	×	*Flexible*	16^2	16^3	16^4
	Required N_{PE}	1		256	4,096	65,536
Latency [us]	CPU	25	357	405	5,821	93,714
	GPU	83,861	extr. 31	319	378	1,841
		CPU	CPU	GPU	GPU	GPU
	Min time	**25**	**357**	**319**	**378**	1,841

Table 3. Available number of parallel processes, required N_{PE} for fully parallel processing, and average detection runtime of various MIMO detectors both on CPU and GPU. ParaMax's compute time is for a single PMIS run on a single PE (*i.e.,* fully-parallel ParaMax) and GPU-ParaMax reports extrapolated compute time.

Comparison against Conventional Detectors. We compare ParaMax latency against various detectors implemented on the MIMOPACK library [57], which is one of the fastest open-source MIMO detector implementations based on the (CUDA) C programming. The results for 20-user 16-QAM are summarized in Table 3. In the case of the zero-forcing successive interference cancellation (ZF-SIC or V-BLAST with ordering scheme) [75], while its complexity is slightly higher than linear detectors such as ZF and MMSE, compute time is still few tens of microseconds. However, their computations (both ZF and ZF-SIC) are not appropriate for parallel processing, causing extra overheads such as job scheduling and data transition among computing resources. In the case of the FCSD, we consider three different N_{fs} that trade-offs the FCSD's detection performance with its complexity. As long as the available number of PEs is large enough to allow full parallelism ($|O|^{N_{fs}} \leq$ total PEs), the compute time remains in a few hundreds of microseconds, satisfying LTE requirements.

6.2 Heuristic Detection Sampling

For ParaMax, we can report the expected number of sampling repetitions to reach ML-performance, which can be computed using the *probability of obtaining the ML-solution in one sample* of a given MIMO detection scenario (*i.e.,* MIMO size, modulation, and SNR), averaged across the problem distribution (P_{ML}) [47]. Since P_{ML} cannot be determined *a priori* by theoretical means, we obtain it through empirical evaluation of statistically significant 1,000,000 PMIS runs

[4] Available compute time is for all BS-processing including channel decoding.

across 100 detection instances per scenario. In order to compute this average probability, we use the ML-solutions found by expensive runs of the Sphere Decoder. Since each run is independent, the probability for ParaMax's to find the optimal ML-solution:

$$\mathcal{P}(\text{ParaMax}_{\text{ML}}) = 1 - (1 - P_{\text{ML}})^{N_{PE}}. \tag{7}$$

Inverting Eq. 7, we can obtain the required number of PMIS repetitions (samples) to achieve the ML-detection with a target probability $\mathcal{P}_{\mathbf{T}}(\text{ParaMax}_{\text{ML}})$ as:

$$\text{required } N_{PE} = \frac{\log(1 - \mathcal{P}_{\mathbf{T}}(\text{ParaMax}_{\text{ML}}))}{\log(1 - P_{\text{ML}})}. \tag{8}$$

In Figure 10(c),[5] we plot P_{ML} and corresponding required N_{PE} for different $\mathcal{P}_{\mathbf{T}}(\text{ParaMax}_{\text{ML}})$.

Very Large MIMO with Low-Order Modulations. Figure 10(a) plots P_{ML} as a function of $N \times N$ Large MIMO detection with different heuristic-based detectors (SA, ParaMax, and 2R-ParaMax) for various N_V and modulations. Surprisingly, for the BPSK and QPSK modulations, all tested heuristic detectors achieve $P_{\text{ML}} \approx 1.0$, which implies nearly all PMIS runs successfully reach the ML-solution. For ParaMax and 2R-ParaMax, this tendency is observed up to $N_V = 512$ while we plot here only up to $N_V = 128$ to save space. Only a few processing elements are enough to perform ML-detection up to 512×512 MIMO with BPSK and 256×256 MIMO with QPSK. While ParaMax becomes currently unpractical at around $N_V = 160$ (Figure 9), this MIMO size and its requirement for optimal detection is promising for city-scale *Internet of Things* (IoT) applications envisioned in 5G networks or beyond. Those scenarios will handle hundreds or thousands of devices per BS with low-order modulations [49, 51, 52], and may accept longer processing time than ordinary data communications.

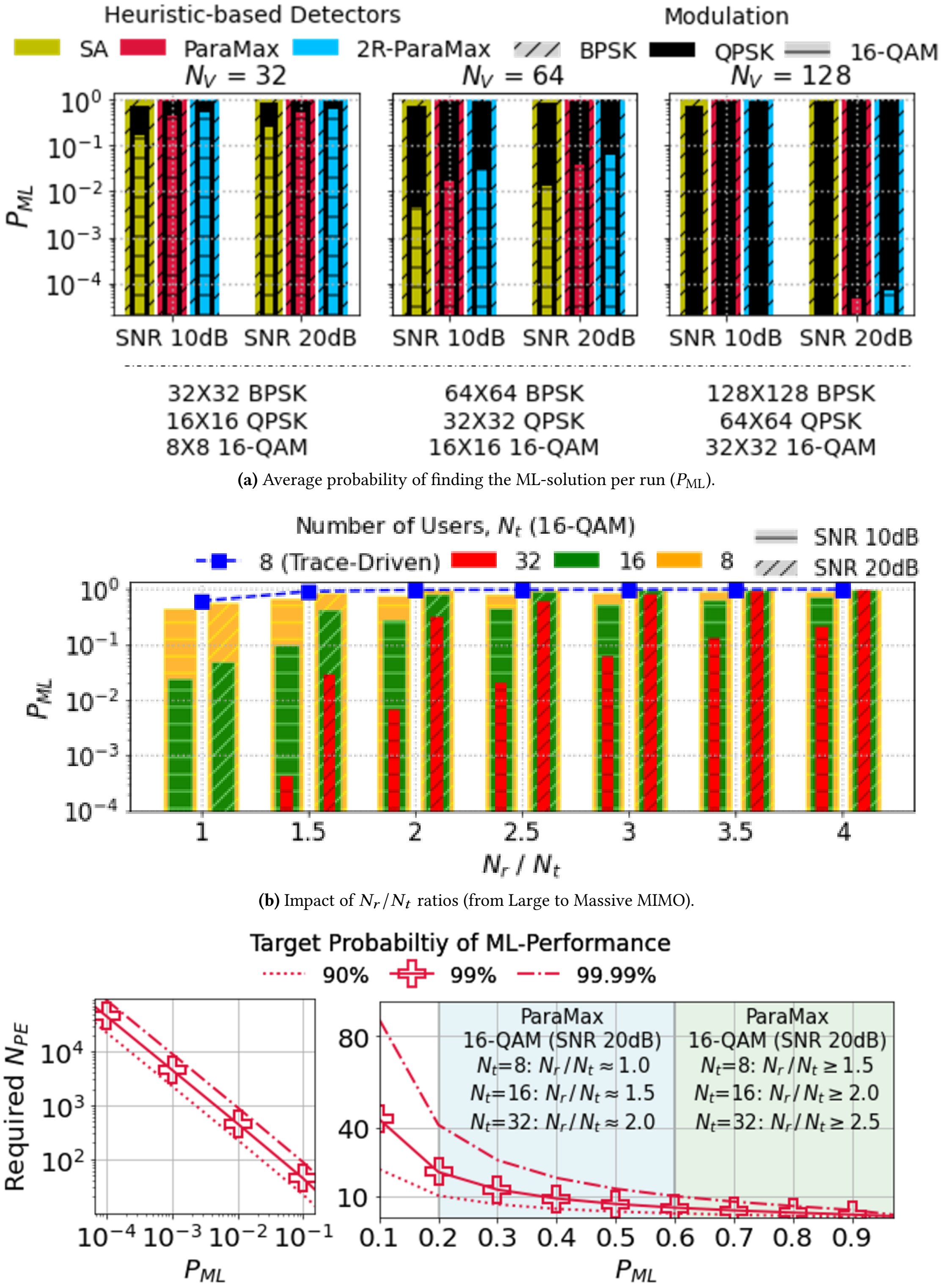
From Large to Massive MIMO with 16-QAM. In the case of 16-QAM in Figure 10(a), P_{ML} notably drops as N_V increases for all heuristic-based detectors and we observe higher P_{ML} for 2R-ParaMax, ParaMax, and SA. Given that Large MIMO detection with high-order modulations is a challenging problem in general, we add more receiver antennas (N_r) to see the impact of N_r/N_t ratio on P_{ML}. Figure 10(b) shows this relationship for various user numbers for different SNRs. As N_r/N_t increases (*i.e,* from Large MIMO to Massive MIMO), P_{ML} rapidly increases and then is converged to 1.0. While P_{ML} for larger number of users (N_t) at lower SNRs tends to increase slower, $P_{\text{ML}} \approx 10^{-2}$ can still be achieved around $N_r/N_t = 2$, where the required N_{PE} for ML-detection is around 1,000 (see Figure 10(c), where we summarize the applicability of ParaMax). We observed that the trace-driven channel with noise shows better performance (faster convergence than 20 dB SNR).

6.3 Bit Error Rate (BER) Performance

This section presents ParaMax's detection BER. Recall that N_{PE} is the number of processing elements (PEs) assigned to ParaMax per subcarrier, where each PE performs a PMIS run. Since we assume fully-parallel ParaMax for minimum detection latency, N_{PE} is also equal to the number of PMIS runs. Note that regardless of computing platforms (CPU, GPU, or FPGA), the detection performance (BER and throughput) as a function of N_{PE} is the same, as long as they can satisfy limited time requirements supporting all subcarriers (unless there exists a serious precision issue), while the definition of a single PE and total available PEs per device can vary depending on platforms and/or implementation details. In the next subsection (Sec 6.4), we evaluate ParaMax on multi-subcarrier systems, considering its detection latency, available parallelism, available compute time in wireless standards, and impact of forward error control (FEC).

Overview: BER from Massive to Large MIMO. Figure 11 shows BER performance in various MIMO regimes with

[5] Note that the formulas hold also for any non-parallel iterative method with independent sampling, where N_{PE} is simply the number of required repetitions.

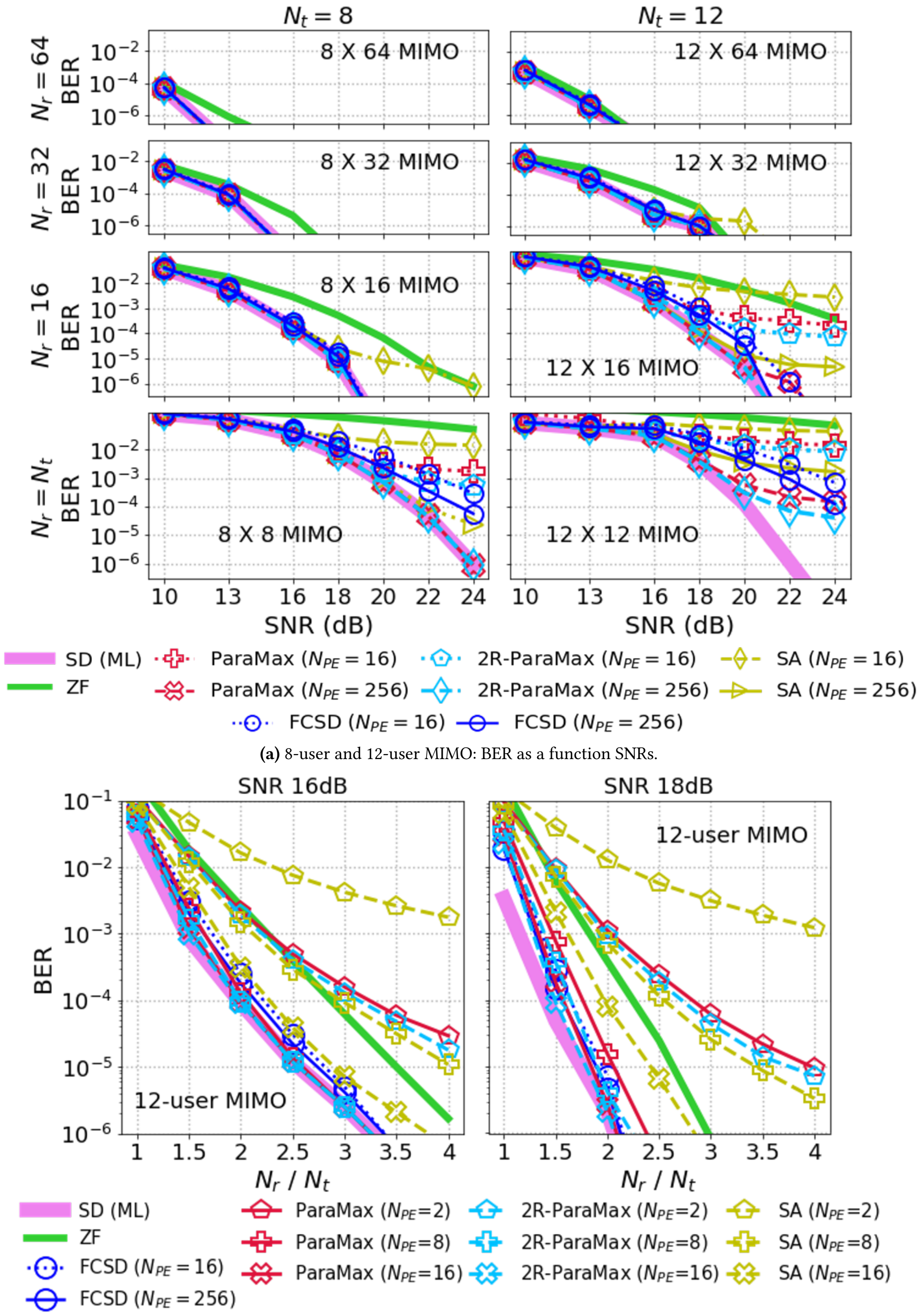


(a) Average probability of finding the ML-solution per run (P_{ML}).

(b) Impact of N_r/N_t ratios (from Large to Massive MIMO).

(c) Required N_{PE} for ParaMax to perform near-ML detection.

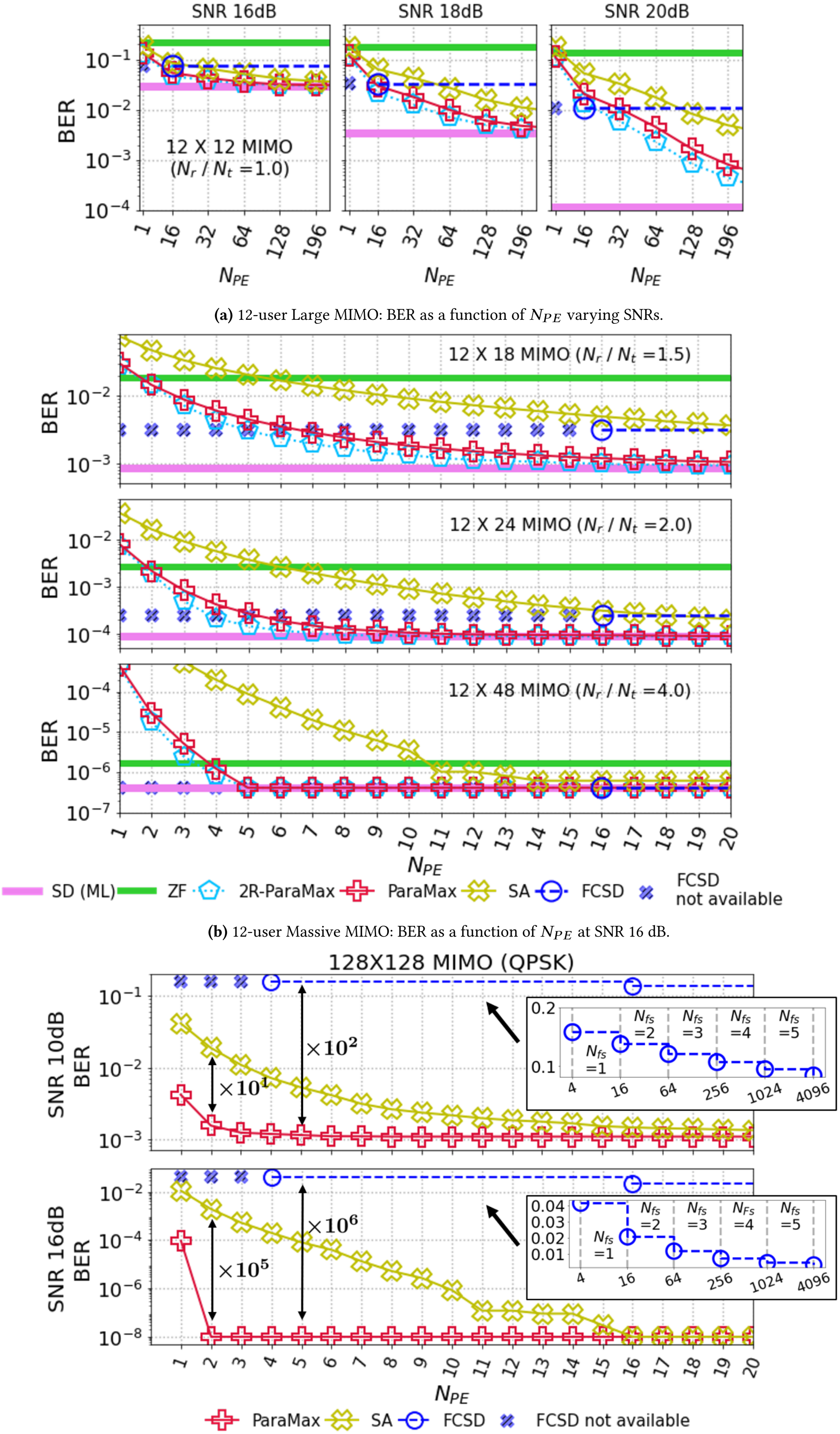
Fig. 10. Detection Sampling Evaluation. Figure 10(a) plots P_{ML} for three heuristic-based detectors (colors) varying modulations (hatch patterns), SNRs, and N_V. Figure 10(b) shows impact of N_r/N_t (from 1 to 4, from Large to Massive MIMO) on ParaMax's P_{ML} for different N_t user numbers (colors) and SNRs (hatch patterns). Figure 10(c) plots relationship between P_{ML} and N_{PE} for ML performance with various $\mathcal{P}_{\mathrm{T}}(\mathrm{ParaMax}_{\mathrm{ML}})$. Approximate MIMO feasibility of ParaMax is provided for two blocks of high P_{ML}.



(a) 8-user and 12-user MIMO: BER as a function SNRs.

(b) 12-user MIMO: BER as a function of N_r / N_t ratio.

Fig. 11. (Overview) BER from Massive to Large MIMO. Comparisons of detection BER in Large and Massive MIMO for various detectors across MIMO regimes and/or SNRs with 16-QAM.



(a) 12-user Large MIMO: BER as a function of N_{PE} varying SNRs.

(b) 12-user Massive MIMO: BER as a function of N_{PE} at SNR 16 dB.

(c) 128 × 128 Very Large MIMO with QPSK modulation.

16-QAM. We consider ParaMax and 2R-ParaMax with 16 and 256 PEs, comparing them against other detectors such as ZF (linear), SA (heuristic), FCSD (tree search-based), and optimal SD (ML), where SA and FCSD (with channel ordering [16]) are comparison schemes of parallel architecture-based detectors. As expected, linear-based ZF, which requires high N_r/N_t ratio for proper detection, performs poorly as the regime goes from Massive MIMO to Large MIMO (*upper* to *lower* in Figure 11(a)) and with more users (*left* to *right* in Figure 11(a)), showing several orders of magnitude worse BER performance against the other detectors, particularly at high SNRs. In the case of the parallel architecture-based detectors, it is observed that for the same PEs, ParaMax and 2R-ParaMax outperform SA and FCSD detectors in all MIMO regimes and SNRs tested except that in 12 × 12 MIMO at high SNRs, FCSD outperforms ParaMax and 2R-ParaMax when with 16 PEs. However, 2R-ParaMax reaches lower BER than FCSD when with 256 PEs. Figure 11(b) plots BER with 16-QAM as a function N_r/N_t ratio with smaller N_{PE} such as 2, 8, and 16. For low N_r/N_t ratios, parallel architecture-based detectors even with 2 PEs can obtain lower BER than ZF. As the ratio increases, all detectors achieve better BER for the same N_{PE}, but more PEs are required to beat ZF.

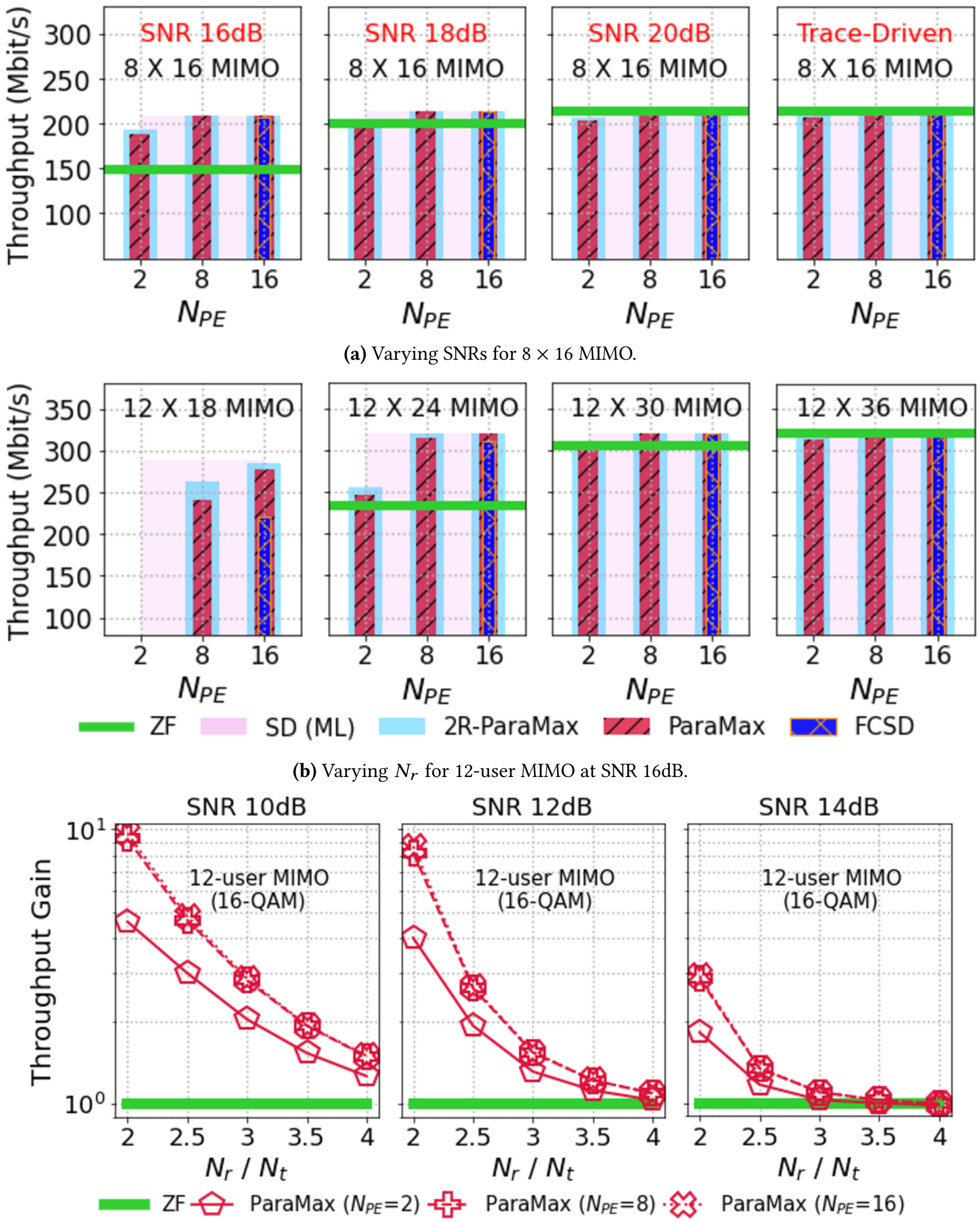
Detailed View: BER as a function of N_{PE}. We evaluate BER as a function of N_{PE} for 12-user Large and Massive MIMO in Figure 12 to show the detailed performance comparison. Note that ZF is not suitable for parallelization, so it achieves the same performance, regardless of the number of PEs. Figure 12(a) presents BER for 12 × 12 Large MIMO ($N_r = N_t$) with 16-QAM at various SNRs. ParaMax can support any number of PEs and approach the optimal performance as N_{PE} increases (*i.e.,* fine parallelism granularity), while the FCSD requires at least 16 PEs to operate the fully-parallel algorithm for the minimum N_{fs}, and the FCSD with N_{PE}=16^1 performs equivalently until N_{PE} reaches 16^2 (*i.e.,* no gain between 16 PEs and 256 PEs). Figure 12(b) focuses on Massive MIMO ($N_r \geq N_t$), showing the impact of N_r/N_t ratio on both BER and N_{PE}. Higher ratios (*upper* to *lower* in Figure 12(b)) lead to lower BER for the same PEs and smaller N_{PE} for the near-ML BER, especially compared against 12 × 12 Large MIMO (Figure 12(a)). Precisely, to reach the near-ML BER at SNR 16 dB for 12 users, 12-BS antenna MIMO requires around 60 PEs, 18-BS antenna MIMO requires 18 PEs, and 48-BS antenna MIMO requires only 5 PEs. Compared to SA and FCSD, ParaMax's BER drops more rapidly as N_{PE} increases for any N_r/N_t ratios. For example, to reach BER $\approx 2 \cdot 10^{-4}$ at 12 × 24 MIMO, where the FCSD and SA requires (over) 16 PEs, ParaMax requires 6 PEs.

We also test 128-user Very Large MIMO with the QPSK modulation in Figure 12(c). As analyzed in Figure 10(a), we observe that very small N_{PE} can result in BER convergence for the QPSK, which is very likely the optimal BER, although we cannot evaluate SD because of extremely high complexity. At SNR 16 dB, ParaMax achieves over five orders of magnitude better BER than SA at 2 PEs and over six orders of magnitude better than FCSD at 4 PEs. Furthermore, the FCSD with thousands of PEs (*i.e.,* with high N_{fs}) cannot even reach the ParaMax's performance with a single PE.

6.4 System Throughput Performance

This section evaluates throughput on multi-subcarrier systems. While detection BER is a fundamental metric for MIMO detection, the detector of the lowest BER does not necessarily imply the best throughput scheme, since real-world wireless systems include FEC techniques for error correction at the channel decoder under MIMO detector. Further, since the systems support many subcarriers with limited compute time, the required total computing resources to support them is another important metric for evaluation.

We first consider a WLAN wireless system with 64 OFDM subcarriers with 1/2 rate convolutional coding, where optimal achievable (ML-based) throughput on the system has been measured via over-the-air experiments in [31]. We translate the measured detection BER into the corresponding convolutional code-applied BER (*i.e.,* coded BER). Among the provided data we select achievable optimal throughput for 8 × 8 MIMO and 12 × 12 MIMO at SNR 21.6 dB

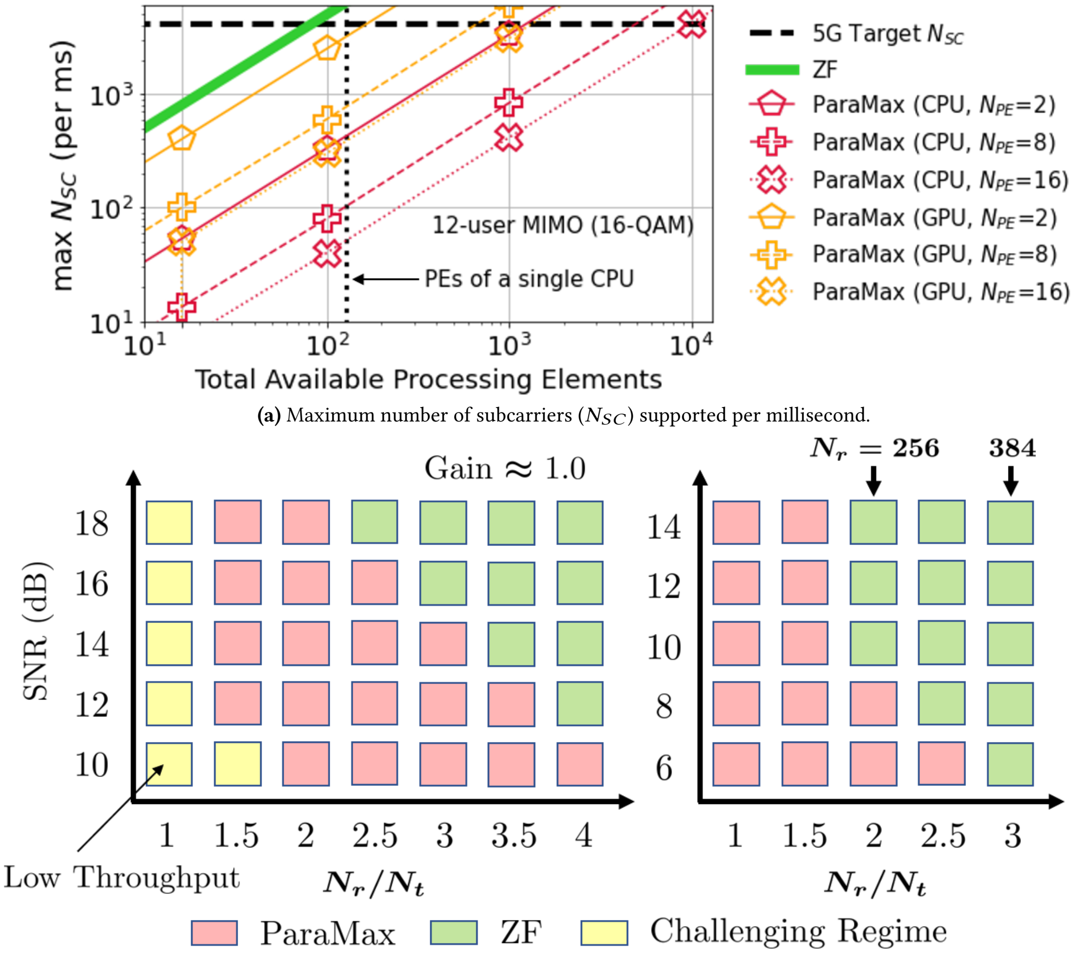


(a) Varying SNRs for 8 × 16 MIMO.

(b) Varying N_r for 12-user MIMO at SNR 16dB.

(c) Achievable gain (vs ZF) for the same number of subcarriers (N_{SC}).

Fig. 13. System Throughput of Massive MIMO in WLAN. Achievable system throughput comparisons of various detectors as a function of N_{PE} with 16-QAM in different scenarios (varying MIMO sizes or SNRs) for minimum detection latency.



(a) Maximum number of subcarriers (N_{SC}) supported per millisecond.

SNR (dB) \ N_r/N_t	1	1.5	2	2.5	3	3.5	4
18	🟨	🟥	🟥	🟩	🟩	🟩	🟩
16	🟨	🟥	🟥	🟥	🟩	🟩	🟩
14	🟨	🟥	🟥	🟥	🟥	🟩	🟩
12	🟨	🟥	🟥	🟥	🟥	🟥	🟩
10	🟨	🟨	🟥	🟥	🟥	🟥	🟥

N_r/N_t	1	1.5	2	2.5	3
14	🟥	🟥	🟩	🟩	🟩
12	🟥	🟥	🟩	🟩	🟩
10	🟥	🟥	🟩	🟩	🟩
8	🟥	🟥	🟥	🟩	🟩
6	🟥	🟥	🟥	🟥	🟩

🟥 ParaMax 🟩 ZF 🟨 Challenging Regime

(b) Best throughput scheme for various MIMO/SNR regimes: 12-user MIMO with 16-QAM (left) and 128-user MIMO QPSK (right).

Fig. 14. Projection to 5G Systems. Fig 14(a) shows maximum number of subcarriers that ParaMax can support as a function of total available PEs on computing devices. Fig 14(b) presents the estimated best-throughput detectors for various regimes, supporting 5G target N_{SC}, on a CPU platform with ten state-of-the-art 128-core CPUs (total PEs ≈ 10^3). Here, ParaMax's $N_{PE} \leq 4$.

as a baseline throughput, assuming the coded BER of optimal Sphere Decoder (SD) we test at the same scenario (*i.e.,* same MIMO size and SNR) is close to their optimal coded BER. We compute the achievable optimal throughput for various scenarios, considering SNR and optimal *Frame Error Rate* (FER) difference (ratio) against the baseline, for the frame size of 1500-byte and FER obtained from our coded BER. Then we compute throughput of various detectors considering FER difference between SD and corresponding detectors at each scenario. In WLAN scenarios, we maintain the minimum detection latency. For this, the system is expected to have total PEs of $N_{PE} \times (N_{SC} = 64)$ on a computing platform, where N_{SC} is the number of subcarriers. For example, for ParaMax to support 2 PMIS runs (*i.e.,* 2 PEs) per subcarrier, a single state-of-the-art CPU (with 128 cores) is enough, while multiple CPUs are required to support more PEs per subcarrier. Since MIMO detection is completely independent across subcarriers, ParaMax's job scheduling and allocation for multiple devices are quite straightforward.

Figure 13 shows throughput comparison against ZF and FCSD for various scenarios in Massive MIMO. Figure 13(a) demonstrates the impact of SNRs for the given MIMO size. While for ParaMax and 2R-ParaMax, 8 PEs (per subcarrier) are enough to reach the optimal performance for all tested SNRs, ZF could not reach the optimal performance except over SNR 20 dB including trace-driven channel and noise. Figure 13(b) shows the impact of N_r/N_t ratio varying receiver antenna numbers N_r at fixed SNR 16 dB. As seen in the previous section, less PEs are required at high N_r/N_t ratios to achieve the near-ML performance. We observe that for the same scenarios, even less PEs are required to achieve the near-optimal "throughput" performance (Figure 13(b)) than to achieve the near-optimal "BER" performance (Figure 12(b)) because of the impact of FEC. Figure 13(c) plots ParaMax's throughput gains versus ZF for various N_r/N_t ratios and SNRs. The high gains are achieved at low SNRs and/or low N_r/N_t ratios. These throughput gains can be generalized to any N_{SC} (even at different standards with slight modifications), as long as both schemes can support all subcarriers satisfying the corresponding limited compute time requirement.

In general, cellular-networked systems, such as 4G or 5G systems, support many more subcarriers, allowing more compute time than WLAN. To project the throughput performance onto 5G scenarios, we plot the maximum N_{SC} that can be supported per millisecond with 5G target, as a function of total available PEs on a computing platform, considering detector's latency and parallelism based on assigned PEs per subcarrier in Figure 14(a) (we report GPU-ParaMax based on extrapolated data, as discussed in section 6.1). The figure implies how many computing resources are required to support 5G systems. A ZF-based system can support 5G target N_{SC} even with a single state-of-the-art 128-core CPU due to its short latency and minimum PE usage. In the case of ParaMax, tens of CPUs are required to support 2 PEs per subcarrier and hundreds for 16 PEs.

Furthermore, we predict the best throughput scheme (ZF vs. ParaMax) for various MIMO regimes and SNRs, assuming a computing platform with ten CPUs in Figure 14(b) (*left*) for 12-user 16-QAM and Figure 14(b) (*right*) for 128-user QPSK based on achievable ParaMax throughput gains (vs ZF), although 128-user QPSK MIMO is currently unpractical (Figure 9). For gain ≈ 1.0, we report the ZF as the best scheme since it takes less compute time, while we report challenging regimes where ZF does not perform well and ParaMax requires at least several tens of PEs per subcarrier for the near-ML performance. We observe that ParaMax enables many challenging regimes of ZF (*i.e.*, low Nr/Nt ratio and/or low SNRs) by assigning reasonably more PEs.[6] In the case of the QPSK, at $Nr/Nt = 2$ (relatively small ratio), ZF can outperform ParaMax at some SNRs, but for this, 256-BS antennas are required to support 128 users, which is the double size N_r of the-state-of-the arts. Of course, ParaMax requires more computing resources (10-100×) than ZF, but the trend at emerging system-on-chip architectures with more and more PEs, as well as C-RAN architectures promisingly envisioned in 5G, support the direction of massively parallel architectures-based designs requiring low interaction among PEs.

7 DISCUSSION

In this section, we investigate several challenges and opportunities of ParaMax that are likely to further advance the system.

Fully-optimized and adaptive ParaMax. Considering that parallel tempering-related parameters are selected within a challenging scenario (16-QAM) in Section 4.1, ParaMax could be fully-optimized for many different scenarios based on given user numbers, N_r/N_t ratios, SNRs, modulation sizes, available total PEs, and/or wireless standards. Moreover

[6] Advanced FEC schemes such as LDPC and Polar codes that are applied in 5G systems can enable the near-ML performance with ZF for more MIMO regimes and SNRs. However, even more users and lower SNRs will keep bringing out the same scenarios, where ParaMax outperforms ZF, due to the fundamental detection BER gap.

there is a well known trade-off between number of sweeps (latency) and required N_{PE} for near-ML performance (compute resources) that could be explored (*e.g.,* for 5G URLLC).

Higher-order modulations. As modulation size increases, ParaMax's detection is degraded rapidly and becomes not operable for Large MIMO with high-order modulations such as 64-QAM or higher, requiring over 10^3 PEs even for 4×4 Large MIMO to achieve near ML-performance. For Massive MIMO, it is expected that even higher N_r/N_t ratios than 16-QAM are required. Perhaps, more replicas or Metropolis sweeps ease the problem along with further optimization on ParaMax's free-parameters related to parallel tempering such as temperature range for PMIS tuning. However, these gains will be obtained at the expense of longer latency. An implementation of ParaMax on dedicated hardware might improve the performance and reduce the computational cost order further.

Compatibility with specialized hardware. ParaMax does not require any specific hardware. However, another important aspect of ParaMax is that it is immediately compatible with future implementations that aim to deploy programmable specialized hardware (for Physics-based algorithms) designed to optimize problems in the Ising form including quantum devices such as quantum annealers [38] and gate-model quantum computers running the QAOA algorithm [24], as well as novel paradigm of classical calculation such as Optical Coherent Ising Machines [25], CMOS-based annealers [4] and Oscillator-based platforms [13].

8 CONCLUSION

In this work, we present ParaMax, a soft MU-MIMO detector system for Large and Massive MIMO networks that first makes use of parallel tempering for MIMO detection. Our performance evaluation shows that ParaMax enables currently-challenging MIMO regimes for commonly-used linear detectors, achieving the near-ML performance by assigning reasonably more compute resources. ParaMax also outperforms conventional parallel architecture-based detectors such as FCSD and SA-based detectors, requiring less processing elements to achieve the near-ML performance.

ACKNOWLEDGEMENTS

We thank the anonymous shepherd and reviewers of this paper, the NASA Quantum AI Laboratory (QuAIL), and the Princeton Advanced Wireless Systems (PAWS) Group for their extensive technical feedback. This research is supported by National Science Foundation (NSF) Award CNS-1824357 and CNS-1824470, and an award from the Princeton University School of Engineering and Applied Science Innovation Fund. Salvatore Mandrá is supported by NASA Ames Research Center (ARC), particularly the autonomous ATM part of the Transformative Tools and Technologies Project under the NASA Transformative Aeronautic Concepts Program, and DARPA under IAA 8839, Annex 125. Minsung Kim was also supported by the USRA Feynman Quantum Academy funded by the NAMS R&D Student Program at NASA ARC (FA8750-19-3-6101).

REFERENCES

[1] 3GPP Technical Specification 36.211 version 11.5.0 Release 11: Evolved Universal Terrestrial Radio Access (E-UTRA) Physical channels and Modulation.

[2] T. Abrão, L. D. de Oliveira, F. Ciriaco, B. A. Angélico, P. J. E. Jeszensky, F. J. C. Palacio. S/mimo mc-cdma heuristic multiuser detectors based on single-objective optimization. *Wireless personal communications*, **53**(4), 529–553, 2010.

[3] E. Agrell, T. Eriksson, A. Vardy, K. Zeger. Closest point search in lattices. *IEEE transactions on information theory*, **48**(8), 2201–2214, 2002.

[4] 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.

[5] L. Barbero, J. Thompson. Fixing the complexity of the sphere decoder for MIMO detection. *IEEE Transactions on Wireless Communications*, **7**(6), 2131–2142, 2008.

[6] L. G. Barbero, T. Ratnarajah, C. Cowan. A low-complexity soft-mimo detector based on the fixed-complexity sphere decoder. *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2669–2672. IEEE, 2008.

[7] L. G. Barbero, J. S. Thompson. Extending a fixed-complexity sphere decoder to obtain likelihood information for turbo-mimo systems. *IEEE Transactions on Vehicular Technology*, **57**(5), 2804–2814, 2008.

[8] E. Björnson, E. G. Larsson, M. Debbah. Massive mimo for maximal spectral efficiency: How many users and pilots should be allocated? *IEEE Transactions on Wireless Communications*, **15**(2), 1293–1308, 2015.

[9] P. Botsinis, S. X. Ng, L. Hanzo. Quantum search algorithms, quantum wireless, and a low-complexity maximum likelihood iterative quantum multi-user detector design. *IEEE access*, **1**, 94–122, 2013.

[10] I. Boyarinov, G. Katsman. Linear unequal error protection codes. *IEEE Transactions on Information Theory*, **27**(2), 168–175, 1981.

[11] R. Chandra, L. Dagum, D. Kohr, R. Menon, D. Maydan, J. McDonald. *Parallel programming in OpenMP.* Morgan kaufmann, 2001.

[12] G. E. Crooks. Measuring thermodynamic length. *Physical Review Letters*, **99**(10), 100,602, 2007.

[13] G. Csaba, W. Porod. Coupled oscillators for computing: A review and perspective. *Applied Physics Reviews*, **7**(1), 011,302, 2020.

[14] L. Dagum, R. Menon. Openmp: an industry standard api for shared-memory programming. *IEEE computational science and engineering*, **5**(1), 46–55, 1998.

[15] E. Dahlman, S. Parkvall, J. Skold. *4G: LTE/LTE-advanced for mobile broadband.* Academic press, 2013.

[16] X. Dai, S. Cheung, T. Yuk. Simplified ordering for fixed-complexity sphere decoder. *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, 804–808, 2010.

[17] M. O. Damen, H. El Gamal, G. Caire. On maximum-likelihood detection and the search for the closest lattice point. *IEEE Transactions on information theory*, **49**(10), 2389–2402, 2003.

[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] B. Farhang-Boroujeny, H. Zhu, Z. Shi. Markov chain monte carlo algorithms for cdma and mimo communication systems. *IEEE transactions on Signal Processing*, **54**(5), 1896–1909, 2006.

[20] U. Fincke, M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of computation*, **44**(170), 463–471, 1985.

[21] H.-O. Georgii. *Gibbs measures and phase transitions*, vol. 9. Walter de Gruyter, 2011.

[22] Y. GuangDa, H. FengYe, H. JinFeng. The multi-user detection for the mimo-ofdm system based on the genetic simulated annealing algorithm. *Proceedings. The 2009 International Workshop on Information Security and Application (IWISA 2009)*, 334. Citeseer, 2009.

[23] 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.

[24] S. Hadfield, Z. Wang, E. G. Rieffel, B. O'Gorman, D. Venturelli, R. Biswas. Quantum approximate optimization with hard and soft constraints. *Workshop on Post Moores Era Supercomputing (PMES)*, 2017. doi:10.1145/3149526.3149530.

[25] R. Hamerly, T. Inagaki, P. L. McMahon, D. Venturelli, A. Marandi, T. Onodera, E. Ng, C. Langrock, K. Inaba, T. Honjo, K. Enbutsu, T. Umeki, R. Kasahara, S. Utsunomiya, S. Kako, K.-i. Kawarabayashi, R. L. Byer, M. M. Fejer, H. Mabuchi, D. Englund, E. Rieffel, H. Takesue, Y. Yamamoto. Experimental investigation of performance differences between coherent ising machines and a quantum annealer. *arXiv preprint arXiv:1805.05217*, 2018.

[26] M. Hansen, B. Hassibi, A. G. Dimakis, W. Xu. Near-optimal detection in mimo systems using gibbs sampling. *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, 1–6. IEEE, 2009.

[27] B. Hassibi, H. Vikalo. On the sphere-decoding algorithm i. expected complexity. *IEEE transactions on signal processing*, **53**(8), 2806–2818, 2005.

[28] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 1970.

[29] J. C. Hedstrom, C. H. Yuen, R.-R. Chen, B. Farhang-Boroujeny. Achieving near map performance with an excited markov chain monte carlo mimo detector. *IEEE Transactions on Wireless Communications*, **16**(12), 7718–7732, 2017.

[30] U. Horn, K. Stuhlmüller, M. Link, B. Girod. Robust internet video transmission based on scalable coding and unequal error protection. *Signal Processing: Image Communication*, **15**(1-2), 77–94, 1999.

[31] C. Husmann, G. Georgis, K. Nikitopoulos, K. Jamieson. Flexcore: Massively parallel and flexible processing for large MIMO access points. *14th*

USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), 197–211, 2017.

[32] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, **31**(1), 253–258, 1925.

[33] J. Jaldén, L. G. Barbero, B. Ottersten, J. S. Thompson. The error probability of the fixed-complexity sphere decoder. *IEEE Transactions on Signal Processing*, **57**(7), 2711–2720, 2009.

[34] N. M. Josuttis. *The C++ standard library: a tutorial and reference.* Addison-Wesley, 2012.

[35] H. Karimi, G. Rosenberg. Boosting quantum annealer performance via sample persistence. *Quantum Information Processing*, **16**(7), 166, 2017.

[36] S. Kasi, K. Jamieson. Towards quantum belief propagation for ldpc decoding in wireless networks. *arXiv preprint arXiv:2007.11069*, 2020.

[37] H. G. Katzgraber, S. Trebst, D. A. Huse, M. Troyer. Feedback-optimized parallel tempering monte carlo. *Journal of Statistical Mechanics: Theory and Experiment*, **2006**(03), P03,018, 2006.

[38] 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*, 241–255. ACM, 2019.

[39] ——. Towards hybrid classical-quantum computation structures in wirelessly-networked systems. *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, 110–116, 2020.

[40] E. G. Larsson, J. Jalden. Fixed-complexity soft mimo detection via partial marginalization. *IEEE transactions on Signal Processing*, **56**(8), 3397–3407, 2008.

[41] E. G. Larsson, P. Stoica, J. Li. On maximum-likelihood detection and decoding for space-time coding systems. *IEEE Transactions on Signal Processing*, **50**(4), 937–944, 2002.

[42] Q. Li, Z. Wang. Reduced complexity k-best sphere decoder design for mimo systems. *Circuits, Systems & Signal Processing*, **27**(4), 491–505, 2008.

[43] A. Lucas. Ising formulations of many np problems. *Frontiers in Physics*, **2**, 5, 2014.

[44] S. Malkowsky, J. Vieira, L. Liu, P. Harris, K. Nieman, N. Kundargi, I. C. Wong, F. Tufvesson, V. Öwall, O. Edfors. The worldâĂŹs first real-time testbed for massive mimo: Design, implementation, and validation. *IEEE Access*, **5**, 9073–9088, 2017.

[45] S. Mandra, H. G. Katzgraber. A deceptive step towards quantum speedup detection. *Quantum Science and Technology*, **3**(4), 04LT01, 2018.

[46] S. Mandra, Z. Zhu, H. G. Katzgraber. Exponentially biased ground-state sampling of quantum annealing machines with transverse-field driving hamiltonians. *Physical review letters*, **118**(7), 070,502, 2017.

[47] S. Mandra, Z. Zhu, W. Wang, A. Perdomo-Ortiz, H. G. Katzgraber. Strengths and weaknesses of weak-strong cluster problems: A detailed overview of state-of-the-art classical heuristics versus quantum approaches. *Physical Review A*, **94**(2), 022,337, 2016.

[48] B. Masnick, J. Wolf. On linear unequal error protection codes. *IEEE Transactions on Information Theory*, **13**(4), 600–607, 1967.

[49] M. H. Mazaheri, S. Ameli, A. Abedi, O. Abari. A millimeter wave network for billions of things. *Proceedings of the ACM Special Interest Group on Data Communication*, 174–186, 2019.

[50] N. Metropolis, S. Ulam. The monte carlo method. *Journal of the American statistical association*, **44**(247), 335–341, 1949.

[51] M. Miller. *The internet of things: How smart TVs, smart cars, smart homes, and smart cities are changing the world.* Pearson Education, 2015.

[52] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac. Internet of things: Vision, applications and research challenges. *Ad hoc networks*, **10**(7), 1497–1516, 2012.

[53] S. Mondal, A. Eltawil, C.-A. Shen, K. N. Salama. Design and implementation of a sort-free k-best sphere decoder. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **18**(10), 1497–1501, 2009.

[54] R. v. Nee, R. Prasad. *OFDM for wireless multimedia communications.* Artech House, Inc., 2000.

[55] 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.

[56] J. S. Plank, K. M. Greenan, E. L. Miller. Screaming fast galois field arithmetic using intel simd instructions. *FAST*, 299–306, 2013.

[57] C. Ramiro, A. M. Vidal, A. Gonzalez. Mimopack: a high-performance computing library for mimo communication systems. *The Journal of Supercomputing*, **71**(2), 751–760, 2015.

[58] S. Roger, C. Ramiro, A. Gonzalez, V. Almenar, A. M. Vidal. An efficient gpu implementation of fixed-complexity sphere decoders for mimo wireless systems. *Integrated Computer-Aided Engineering*, **19**(4), 341–350, 2012.

[59] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, F. Tufvesson. Scaling up mimo: Opportunities and challenges with very large arrays. *IEEE signal processing magazine*, **30**(1), 40–60, 2012.

[60] J. Sanders, E. Kandrot. *CUDA by example: an introduction to general-purpose GPU programming.* Addison-Wesley Professional, 2010.

[61] D. Schoolar. Massive mimo comes of age. *Samsung Official Whitepaper*, 2017.

[62] C. Shepard, J. Ding, R. Guerra, L. Zhong. Understanding real many-antenna MU-MIMO channels. *Proc. of the IEEE Asilomar Conf.*, 2017.

[63] C. Shepard, J. Ding, R. E. Guerra, L. Zhong. Understanding real many-antenna mu-mimo channels. *Signals, Systems and Computers, 2016 50th Asilomar Conference on*, 461–467. IEEE, 2016.

[64] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, L. Zhong. Argos: Practical many-antenna base stations. *Proceedings of the 18th annual international conference on Mobile computing and networking*, 53–64. ACM, 2012.

[65] C. Studer, M. Wenk, A. Burg, H. Bolcskei. Soft-output sphere decoding: Performance and implementation aspects. *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, 2071–2076. IEEE, 2006.

[66] P. Švač, F. Meyer, E. Riegler, F. Hlawatsch. Soft-heuristic detectors for large mimo systems. *IEEE Transactions on Signal Processing*, **61**(18), 4573–4586, 2013.

[67] R. H. Swendsen, J.-S. Wang. Replica monte carlo simulation of spin-glasses. *Physical review letters*, **57**(21), 2607, 1986.

[68] S. Trebst, M. Troyer, U. H. Hansmann. Optimized parallel tempering simulations of proteins. *The Journal of chemical physics*, **124**(17), 174,903, 2006.

[69] J. Vieira, S. Malkowsky, K. Nieman, Z. Miers, N. Kundargi, L. Liu, I. Wong, V. Öwall, O. Edfors, F. Tufvesson. A flexible 100-antenna testbed for massive mimo. *2014 IEEE Globecom Workshops (GC Wkshps)*, 287–293. IEEE, 2014.

[70] E. Viterbo, J. Boutros. A universal lattice code decoder for fading channels. *IEEE Trans. Inf. Theory*, **45**(5), 1639–1642, 1999.

[71] R. Wang, G. B. Giannakis. Approaching mimo channel capacity with reduced-complexity soft sphere decoding. *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No. 04TH8733)*, vol. 3, 1620–1625. IEEE, 2004.

[72] L. Wei. Coded modulation with unequal error protection. *IEEE Transactions on Communications*, **41**(10), 1439–1449, 1993.

[73] L.-F. Wei. Coded modulation with unequal error protection. *IEEE Transactions on Communications*, **41**(10), 1439–1449, 1993.

[74] D. J. Welsh. The computational complexity of some classical problems from statistical physics, 1990.

[75] P. W. Wolniansky, G. J. Foschini, G. D. Golden, R. A. Valenzuela. V-blast: An architecture for realizing very high data rates over the rich-scattering wireless channel. *1998 URSI international symposium on signals, systems, and electronics. Conference proceedings (Cat. No. 98EX167)*, 295–300. IEEE, 1998.

[76] 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.

[77] Z.-C. Yang, A. Rahmani, A. Shabani, H. Neven, C. Chamon. Optimizing variational quantum algorithms using Pontryagin's minimum principle. *Physical Review X*, **7**(2), 021,027, 2017.

[78] A. Young, H. G. Katzgraber. Absence of an almeida-thouless line in three-dimensional spin glasses. *Physical review letters*, **93**(20), 207,203, 2004.

[79] B. Zheng, M. Wen, F. Chen, N. Huang, F. Ji, H. Yu. The k-best sphere decoding for soft detection of generalized spatial modulation. *IEEE Transactions on Communications*, **65**(11), 4803–4816, 2017.