QR-Decomposition-Aided Tabu Search Detection for Large MIMO Systems

Nhan Thanh Nguyen, Kyungchun Lee, *Senior Member, IEEE*, and Huaiyu Dai, *Fellow, IEEE*

***Abstract*—In the conventional tabu search (TS) detection algorithm for multiple-input multiple-output (MIMO) systems, the cost metrics of all neighboring vectors are computed to determine the best neighbor. This can require an excessively high computational complexity, especially in large MIMO systems because the number of neighboring vectors and the dimension per vector are large. In this study, we propose an improved TS algorithm based on the QR decomposition of the channel matrix (QR-TS), which allows for finding the best neighbor with a significantly lower complexity compared with the conventional TS algorithm. Specifically, QR-TS does not compute all metrics by early rejecting unpromising neighbors, which reduces the computational load of TS without causing any performance loss. To further optimize the QR-TS algorithm, we investigate novel ordering schemes, namely the transmit-ordering (Tx-ordering) and receive-ordering (Rx-ordering), which can considerably reduce the complexity of QR-TS. Simulation results show that QR-TS reduces the complexity approximately by a factor of two compared with the conventional TS. Furthermore, when both Tx-ordering and Rx-ordering are applied, QR-TS requires approximately 60% – 90% less complexity compared with the conventional TS scheme. The proposed algorithms are suitable for both low-order and high-order modulation, and can achieve a significant complexity reduction compared to the Schnorr–Euchner and K-best sphere decoders in large MIMO systems.**

***Index Terms*—Tabu search detection, massive MIMO, ordering schemes.**

I. Introduction

In MOBILE communications, a large multiple-input multiple-output (MIMO) system, where the base station (BS) is equipped with a large number of antennas, has been recently considered as a potential technique for dramatically improving system spectral and power efficiency [1], [2]. However, the promised advantages of large MIMO require significantly increased computational complexity at the receiver compared to the conventional MIMO system [3]. In the conventional MIMO, the sphere decoder (SD) has been widely considered as the most efficient approach to achieve the near-maximum likelihood (ML) performance with reduced complexity compared to the optimal ML detector (MLD). However, the lower bound of its complexity exponentially grows with the number of transmit antennas and constellation orders [4], which leads to excessively large complexity in large MIMO systems. The variations of the original SD, such as $K-$best SD (KSD) [5], which is equivalent to the sequential $M-$algorithm [6], and fixed-complexity SD (FSD) [7], have been proposed to further reduce or limit the complexity of SD while sacrificing error performance depending on chosen design parameters. However, to guarantee near-ML performance, especially at high signal-to-noise ratios (SNRs), their complexities can be higher than that of the original SD [5], [7], [8]. Therefore, low-complexity near-optimal detection is one of the challenges in realizing large MIMO systems, and has thus attracted considerable interests recently [9]–[11].

A. Recent Works

Various algorithms for large-MIMO detection have been introduced [12]–[22]. In [12], a high-performance, low-complexity detector called likelihood ascent search (LAS) and its extension, multistage LAS [13] are proposed for large-MIMO systems with binary phase-shift keying (BPSK) modulation. The work in [16] considers the application of belief propagation to achieve near-optimal signal detection with low complexity for 64×64 MIMO with BPSK and 4-quadrature amplitude modulation (QAM). In [17], low-complexity probabilistic-data-association-based decoding algorithms are proposed for non-orthogonal space-time block-coded large MIMO systems; their performance approached the single-input single-output additive white Gaussian noise (AWGN) performance with 4-QAM but the performance gap becomes larger for higher-order modulations such as 16-QAM. In [18], an algorithm based on Monte Carlo sampling, known as the mixed Gibbs sampling (MGS) algorithm, is proposed to alleviate the stalling problem in [19]. However, MGS provides poor performance in higher-order QAM modulations; this problem is alleviated in [20]. Most previous works focus on low-order modulation (e.g., BPSK, 4-QAM) while higher-order QAM can be used to convey information at higher rates. To address this, channel-hardening-based [21] and soft-heuristic [22] algorithms are proposed in the literature for large MIMO systems such as 64×64 MIMO with QPSK and 16-QAM.

Manuscript received October 4, 2018; revised January 27, 2019; accepted March 4, 2019. Date of publication March 18, 2019; date of current version May 28, 2019. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant NRF-2014R1A1A1002653. The review of this paper was coordinated by Prof. R. Dinis. *(Corresponding author: Kyungchun Lee.)*

N. T. Nguyen is with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea (e-mail: nhan.nguyen@seoultech.ac.kr).

K. Lee is with the Department of Electrical and Information Engineering and the Research Center for Electrical and Information Technology, Seoul National University of Science and Technology, Seoul 01811, South Korea (e-mail: kclee@seoultech.ac.kr).

H. Dai is with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695 USA (e-mail: Huaiyu_Dai@ncsu.edu).

Digital Object Identifier 10.1109/TVT.2019.2905642

Additionally, the tabu search (TS) detector, which is a local neighbor search algorithm, was also introduced as a complexity-efficient scheme for symbol detection in large MIMO systems. It has been shown that the TS detection algorithm can perform very close to the ML bound with far lower complexity compared to SD or FSD in large MIMO systems [9], [23]. In [24], the authors present an approach based on reactive tabu search for near-ML decoding of non-orthogonal 64×64 STBCs with 4-QAM. However, its performance is far from optimum for higher-order QAMs such as 16- and 64-QAM [25]. The work in [23] proposes an algorithm called layered TS (LTS). This algorithm improves the conventional TS in terms of higher-order QAM performance in large MIMO systems. However, to achieve the bit-error rate (BER) of 10^{-2} in 32×32 and 64×64 MIMO systems with 16-QAM, higher complexities are required compared to conventional TS. The random-restart reactive TS (R3TS) algorithm, which runs multiple reactive TS (RTS) and chooses the best among the resulting solution vectors, is presented in [26]. It achieves improved BER performance at the expense of increased complexity. The complexity of R3TS is generally higher than that of RTS to achieve 10^{-2} BER, especially for large systems and high-order QAMs such as 64×64 MIMO with 64-QAM. Several works have been conducted to further improve TS in terms of complexity by reducing the number of examined neighbors or using a stopping criterion [27], [28]. However, it comes at the cost of performance loss; for example, to achieve BER $= 10^{-3}$ for 4×4 MIMO with 16-QAM modulation, the TS algorithm with a stopping criterion has a 3-dB SNR loss compared to the original TS [27].

B. Contributions

TS detection is considered one of the efficient symbol detection algorithms for large MIMO systems because it performs especially close to the ML bound with a significantly lower complexity compared with SD and FSD in large MIMO systems, as shown in [9] and [23]. In this work, we optimize the conventional TS algorithm in terms of computational complexity for symbol detection in large MIMO with both lower-order (e.g., QPSK) and higher-order (e.g., 16- and 64-QAM) modulations. In each searching iteration of TS detection, the algorithm needs to find the best neighboring vector of the current candidate, which is computationally expensive, especially in large MIMO. This motivates us to develop an algorithm to efficiently find the best neighbor, which can significantly reduce the overall complexity. Our main contributions include:

- We propose an improved TS algorithm based on the QR decomposition of the channel matrix (QR-TS), which allows finding the best neighbor with reduced complexity by exploiting efficient computations and early rejection while achieving exactly the same BER performance with conventional TS.
- The analytical expressions for the average computational complexities to find the best neighbor in the conventional and QR-TS algorithms are derived. This allows us to analytically compare the proposed algorithm to the conventional one in terms of computational complexity and further optimize the QR-TS algorithm.
- We exploit ordering schemes, namely, transmit-ordering (Tx-ordering) and receive-ordering (Rx-ordering), to further reduce the complexity of the QR-TS algorithm. We analytically and numerically justify that the ordering schemes are powerful in larger-sized MIMO systems. Simulation results show that QR-TS reduces the computational complexity of the conventional TS algorithm by approximately half. Moreover, if the ordering schemes are incorporated, they require only approximately one-fourth the complexity of the conventional TS. It is also shown that the proposed ordered TS schemes can achieve complexity reduction gain in both lower- and higher-order modulations.

The rest of the paper is organized as follows: Section II presents the system model and the conventional TS algorithm. Section III describes the proposed QR-TS and ordering schemes, and analyzes their complexity reduction. Section IV discusses the performance and complexity results of the proposed schemes for various MIMO configurations. Finally, conclusions are presented in Section V.

Notations: Throughout this paper, scalars, vectors, and matrices are denoted by lower-case, bold-face lower-case, and bold-face upper-case letters, respectively. The (i, j)th element of a matrix $\boldsymbol{A}$ is denoted by $a_{i,j}$, whereas $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and conjugate transpose of a vector, respectively. Furthermore, $|\cdot|$ and $\|\cdot\|$ represent the absolute value of a scalar and the norm of a vector, respectively. The probability and expectation operators are denoted by $\mathbb{P}\{\cdot\}$ and $\mathbb{E}\{\cdot\}$, respectively, while $\sim$ means $distributedas$.

II. System Model and Conventional TS

A. System Model

We consider the uplink of a multiuser MIMO system with N_r receive antennas, where the total number of transmit antennas of all users is N_t. The received signal vector $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{N_r}]^T$ is given by

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}} + \tilde{\boldsymbol{v}}, \tag{1}$$

where $\tilde{\boldsymbol{s}} = [\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{N_t}]^T$ is the vector of transmit symbols. We assume that $\mathbb{E}\{|\tilde{s}_i|^2\} = \sigma_t^2$, where σ_t^2 is the average symbol power, and $\tilde{\boldsymbol{v}} = [\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_{N_r}]^T$ is an $(N_r \times 1)$-vector of independent and identically distributed (i.i.d.) additive white Gaussian noise (AWGN) samples $\tilde{v}_i \sim \mathcal{CN}(0, \sigma_v^2)$. Furthermore, $\tilde{\boldsymbol{H}}$ denotes an $N_r \times N_t$ channel matrix consisting of entries $\tilde{h}_{i,j}$, where $\tilde{h}_{i,j}$ represents the complex channel gain between the jth transmit antenna and the ith receive antenna. For simplicity of exposition, we assume $N_t = N_r$[1]. The transmitted symbols $\tilde{s}_i, i = 1, 2, \ldots, N_t$, are independently drawn from a complex constellation $\tilde{\mathcal{A}}$ of Q points such as QAM and phase shift keying (PSK) modulations. The set of all possible transmitted vectors forms an N_t-dimensional complex constellation $\tilde{\mathcal{A}}^{N_t}$ consisting of Q^{N_t} vectors, i.e., $\tilde{\boldsymbol{s}} \in \tilde{\mathcal{A}}^{N_t}$.

The complex signal model (1) can be converted to an equivalent real signal model

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{s} + \boldsymbol{v}, \tag{2}$$

[1]Note that the proposed scheme can be applied to every system configuration.

where

$$\boldsymbol{s} = \begin{bmatrix} \Re(\tilde{\boldsymbol{s}}) \\ \Im(\tilde{\boldsymbol{s}}) \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} \Re(\tilde{\boldsymbol{y}}) \\ \Im(\tilde{\boldsymbol{y}}) \end{bmatrix}, \boldsymbol{v} = \begin{bmatrix} \Re(\tilde{\boldsymbol{v}}) \\ \Im(\tilde{\boldsymbol{v}}) \end{bmatrix},$$

and

$$\boldsymbol{H} = \begin{bmatrix} \Re\left(\tilde{\boldsymbol{H}}\right) & -\Im\left(\tilde{\boldsymbol{H}}\right) \\ \Im\left(\tilde{\boldsymbol{H}}\right) & \Re\left(\tilde{\boldsymbol{H}}\right) \end{bmatrix}$$

denote the $(N \times 1)$-equivalent real transmitted, received, AWGN noise signal vectors, and the $N \times N$ equivalent real channel matrix, respectively, with $N = 2N_t = 2N_r$. Here, $\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts of a complex number or vector, respectively. In (2), we have $\boldsymbol{s} \in \mathcal{A}^N$, where $\mathcal{A}$ is the equivalent real-valued signal constellation set of $\tilde{\mathcal{A}}$; for example, $\mathcal{A} = \{-1, 1\}$ for QPSK with $\sigma_t^2 = 2$ and $\mathcal{A} = \{-3, -1, 1, 3\}$ for 16-QAM with $\sigma_t^2 = 10$.

For the description of the TS algorithm, we use the equivalent real-valued signal model in (2). Subsequently, the maximum likelihood (ML) solution can be written as

$$\hat{\boldsymbol{s}}_{ML} = \arg \min_{\boldsymbol{s} \in \mathcal{A}^N} \phi(\boldsymbol{s}), \tag{3}$$

where $\phi(\boldsymbol{s}) = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{s}\|^2$ is the ML cost metric of $\boldsymbol{s}$. The computational complexity of ML detection in (3) is exponential with N [23], which results in extremely high complexity for massive MIMO systems, where N is very large.

B. Conventional TS Algorithm

In this subsection, the main concepts of the conventional TS detection algorithm are briefly explained. We focus on the analysis of the computational complexity of this algorithm, which is useful for comparison with the proposed algorithm in Section III-C. For a detailed description of the TS algorithm, please refer to [23], [24].

The conventional TS algorithm searches candidates and their neighbors over M iterations. It uses a tabu list $\mathcal{L}$ of length $P (P \leq M)$ to record the visited points to avoid cycling. The output of the TS algorithm is the best solution vector $\hat{\boldsymbol{s}}_{TS}$ found until M searching iterations.

1) Initialization Phase: The algorithm starts with an initial candidate vector, which is assumed to be the ZF solution $\boldsymbol{x}_{ZF} = \lceil \boldsymbol{H}^{\dagger}\boldsymbol{y} \rfloor$, where $\boldsymbol{H}^{\dagger}$ is the pseudoinverse of $\boldsymbol{H}$ and $\lceil \cdot \rfloor$ is a quantized vector. With the assumption of $N_t = N_r$ and full rank of $\boldsymbol{H}$, we have $\boldsymbol{H}^{\dagger} = \boldsymbol{H}^{-1}$ and $\boldsymbol{x}_{ZF} = \lceil \boldsymbol{H}^{-1}\boldsymbol{y} \rfloor$. Assuming that the well-known Gaussian elimination method is used, solving for $\boldsymbol{x}_{ZF}$ requires $\frac{1}{3}N^3 + \frac{1}{2}N^2 - \frac{5}{6}N$ multiplications and $\frac{1}{3}N^3 + \frac{1}{2}N^2 - \frac{5}{6}N$ additions [29]. Furthermore, the metric of the initial candidate is computed as $\phi(\boldsymbol{c}_{\{1\}}) = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{c}_{\{1\}}\|^2$, which requires $N^2 + N$ multiplications and $N^2 + N - 1$ additions. Consequently, the total complexity required for the initialization phase becomes $\frac{1}{3}N^3 + \frac{3}{2}N^2 + \frac{1}{6}N$ multiplications and $\frac{1}{3}N^3 + \frac{3}{2}N^2 + \frac{1}{6}N - 1$ additions.

2) Searching Phase: In each searching iteration, the TS algorithm examines the neighboring vectors of the current candidate and subsequently moves to the best neighboring vector. In this work, neighbor $\boldsymbol{x}$ of a given candidate $\boldsymbol{c}$ is defined as a non-tabu vector inside alphabet $\mathcal{A}^N$ with the smallest distance to $\boldsymbol{c}$. Therefore, the neighbor set of $\boldsymbol{c}$ is given by

$$\mathcal{N}(\boldsymbol{c}) = \left\{ \boldsymbol{x} \in \mathcal{A}^N \backslash \mathcal{L}, |\boldsymbol{x} - \boldsymbol{c}| = \theta_{min} \right\}, \tag{4}$$

where $\mathcal{A}^N \backslash \mathcal{L}$ denotes the alphabet $\mathcal{A}^N$ excluding the tabu vectors kept in $\mathcal{L}$, and θ_{min} is the minimum distance between two constellation points in a plane. The maximum numbers of neighbors in $\mathcal{N}(\boldsymbol{c})$ for different modulation schemes are presented in Table I. It is noteworthy that for BPSK and QPSK modulations, each candidate vector $\boldsymbol{c}$ has a maximum of N neighboring vectors. However, in higher-order modulation schemes, such as 16- and 64-QAM, the number of neighbors of $\boldsymbol{c}$ depends on its elements. The elements with the highest absolute amplitude level have only a single neighboring symbol each, whereas each of the other elements has two neighboring symbols. Therefore, in higher-order QAM schemes, the maximum number of neighbors of $\boldsymbol{c}$ is given by $2N - \alpha$, where $\alpha \in [0, N]$ is the number of elements of $\boldsymbol{c}$ having the highest absolute amplitude level. The subtractions to one in Table I imply the exclusion of the neighbor that is maintained in the tabu list to avoid cycling.

TABLE I
MAXIMUM NUMBER OF NEIGHBORS OF A CANDIDATE $\boldsymbol{c}$, WITH α IS THE NUMBER OF ELEMENTS OF $\boldsymbol{c}$ HAVING THE HIGHEST ABSOLUTE AMPLITUDE LEVEL, $0 \leq \alpha \leq N$

Modulation scheme	BPSK	QPSK	16-QAM	64-QAM
Maximum number of neighbors	$N-1$	$N-1$	$2N-\alpha-1$ $\alpha=3$	$2N-\alpha-1$ $\alpha=7$

In $\mathcal{N}(\boldsymbol{c})$, the best neighbor $\boldsymbol{x}^{\star}$ is the one with the smallest metric, i.e.,

$$\boldsymbol{x}^{\star} = \arg \min_{\boldsymbol{x} \in \mathcal{N}(\boldsymbol{c})} \phi(\boldsymbol{x}).$$

In every iteration, the cost metric of $\boldsymbol{x}^{\star}$ is compared to that of $\hat{\boldsymbol{s}}_{TS}$. If a new smaller metric is found, i.e., $\phi(\boldsymbol{x}^{\star}) < \phi(\hat{\boldsymbol{s}}_{TS})$, the algorithm updates the best solution $\hat{\boldsymbol{s}}_{TS}$ to $\boldsymbol{x}^{\star}$. It is worth noting that in the TS algorithm, to escape from local minima, the move from $\boldsymbol{c}$ to $\boldsymbol{x}^{\star}$ is made in every iteration even when $\boldsymbol{x}^{\star}$ is worse than $\boldsymbol{c}$ in terms of the ML cost, i.e., $\phi(\boldsymbol{x}^{\star}) > \phi(\boldsymbol{c})$. After each move, the current candidate and tabu list are updated as $\boldsymbol{c}_{\{m\}} = \boldsymbol{x}^{\star}_{\{m-1\}}$ and $\mathcal{L}_{\{m\}} = \{\mathcal{L}_{\{m-1\}}, \boldsymbol{x}^{\star}_{\{m-1\}}\}$, where the subscript $\{i\}$ indicates the ith iteration. If the tabu list is full, its first element will be released to make space for another vector. This process is terminated after M iterations or if a termination condition is satisfied, and the final solution is determined as the best solution vector found.

The metric of a neighbor $\boldsymbol{x}$ can be expressed as

$$\phi(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 = \|\boldsymbol{u} + \boldsymbol{H}\Delta\boldsymbol{x}\|^2 = \|\boldsymbol{u} + \boldsymbol{h}_d\delta_d\|^2, \tag{5}$$

where $\boldsymbol{u} = \boldsymbol{y} - \boldsymbol{H}\boldsymbol{c}$, $\Delta\boldsymbol{x} = \boldsymbol{c} - \boldsymbol{x} = [0, \ldots, 0, \delta_d, 0, \ldots, 0]^T$ has only one nonzero element $\delta_d = c_d - x_d$, and $\boldsymbol{h}_d$ is the dth column of $\boldsymbol{H}$. In this study, we refer to d as the "difference position" of a neighbor. For example, if the current candidate is $\boldsymbol{c} = [1, -3, 1, 3]^T$, then $d = 3$ is the difference position for $\boldsymbol{x} = [1, -3, -1, 3]^T$ because $\boldsymbol{c}$ and $\boldsymbol{x}$ are only different with respect to the third element. Let $\mathcal{D} = \{d_1, d_2, \ldots, d_L\}$ be the set

of difference positions of L neighbors in $\mathcal{N}(\boldsymbol{c})$. Since $\Delta\boldsymbol{x}$ is a single-nonzero vector, we obtain the last equation in (5). Let $\bar{L}$ be the average number of neighbors in an iteration. From (5), the average complexity to find the best neighbor in an iteration includes $2\bar{L}N$ multiplications and $\bar{L}(2N-1)$ additions. Furthermore, $\bar{L}$ comparisons on average are required to determine and update the best neighbor in each searching iteration of the TS algorithm. It is observed that the computational load of TS comes primarily from the computations of the neighbors' metrics to find the best neighbor in each iteration. This motivates us to develop a novel TS algorithm that allows determining the best neighbor with low complexity to reduce the overall computational load of the TS algorithm.

III. Proposed QR-TS Algorithm

In this section, we introduce the proposed QR-TS algorithm that aims to optimize the conventional TS in terms of computational complexity. The complexity analysis is also given in this section.

A. Basic Ideas

The metric in (5) can be rewritten as

$$\phi(\boldsymbol{x}) = \sum_{n=1}^{N} \left|u_n + h_{n,d}\delta_d\right|^2. \tag{6}$$

It implies that to compute $\phi(\boldsymbol{x})$, the algorithm needs to consider N terms, i.e., $|u_n + h_{n,d}\delta_d|^2, n = 1, \ldots, N$. However, in large MIMO systems, N is large, and hence, the computation of $\phi(\boldsymbol{x})$ becomes computationally expensive. The QR-TS algorithm allows the computation of $\phi(\boldsymbol{x})$ with considerably lower complexity as a benefit of the QR decomposition of the channel matrix $\boldsymbol{H}$. Furthermore, it also computes the neighbors' metrics more efficiently by accumulating only a subset of N terms.

By using the QR decomposition, we have $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{Q}$ is a unitary matrix, and $\boldsymbol{R}$ is an upper triangular matrix. Subsequently, the cost metric of $\boldsymbol{x}$ can be rewritten as

$$\begin{aligned}\phi(\boldsymbol{x}) &= \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|^2 = \left\|\boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{R}\boldsymbol{x}\right\|^2 \\ &= \left\|\boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{x}\right\|^2 = \left\|\boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\left(\boldsymbol{c} - \Delta\boldsymbol{x}\right)\right\|^2. \end{aligned} \tag{7}$$

By defining $\boldsymbol{z} = \boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{c}$, (7) can be rewritten as

$$\phi(\boldsymbol{x}) = \|\boldsymbol{z} + \boldsymbol{R}\Delta\boldsymbol{x}\|^2 = \|\boldsymbol{z} + \boldsymbol{r}_d\delta_d\|^2 \tag{8}$$

$$= \underbrace{\sum\nolimits_{n=1}^{d} \left|z_n + r_{n,d}\delta_d\right|^2}_{\triangleq \phi_A} + \underbrace{\sum\nolimits_{n=d+1}^{N} \left|z_n\right|^2}_{\triangleq \phi_B}, \tag{9}$$

where $\boldsymbol{r}_d$ is the dth column of $\boldsymbol{R}$, and z_n is the nth element of $\boldsymbol{z}$. Equation (9) motivates the development of two main ideas to reduce the computational complexity of QR-TS, which are efficient metric computation and early rejection.

1) Efficient Metric Computation: In $\boldsymbol{r}_d$, only the first d elements are non-zeros, which makes the sum ϕ_B in (9) independent of $\boldsymbol{x}$. Therefore, ϕ_B in (9) can be computed for all neighbors with minimal complexity with the following techniques:

TABLE II
Neighbors' Metric

d	Metric of the neighbor corresponding to d
1	$\underbrace{\left\|z_N\right\|^2 + \ldots\ldots\ldots + \left\|z_3\right\|^2 + \left\|z_2\right\|^2}_{\phi_B = e_1} + \left\|z_1 + r_{1,1}\delta_1\right\|^2$
2	$\underbrace{\left\|z_N\right\|^2 + \ldots\ldots\ldots + \left\|z_3\right\|^2}_{\phi_B = e_2} + \sum_{n=1}^{2}\left\|z_n + r_{n,2}\delta_2\right\|^2$
$\ldots$	$\ldots$
d	$\underbrace{\left\|z_N\right\|^2 + \ldots + \left\|z_{d+1}\right\|^2}_{\phi_B = e_d} + \sum_{n=1}^{d}\left\|z_n + r_{n,d}\delta_d\right\|^2$
$\ldots$	$\ldots$
$N-1$	$\underbrace{\left\|z_N\right\|^2}_{\phi_B = e_{N-1}} + \sum_{n=1}^{N-1}\left\|z_n + r_{n,N-1}\delta_{N-1}\right\|^2$
N	$\underbrace{0}_{e_N} + \sum_{n=1}^{N}\left\|z_n + r_{n,N}\delta_N\right\|^2$

- *Within one searching iteration,* $\boldsymbol{z}$ is constant for every neighbor since $\boldsymbol{z} = \boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{c}$, whereas $\boldsymbol{Q}, \boldsymbol{R}$, and $\boldsymbol{y}$ are fixed for M iterations, and $\boldsymbol{c}$ is invariant with neighbors. Therefore, the algorithm needs to compute $\sum_{n=i+1}^{N}|z_n|^2, i = 1, 2, \ldots, N-1$ only once and store them in an array $\boldsymbol{e}$ to reuse it for every neighbor. Here, we define

$$\boldsymbol{e} = \left[\sum_{n=2}^{N}\left|z_n\right|^2, \sum_{n=3}^{N}\left|z_n\right|^2, \ldots, \left|z_N\right|^2, 0\right]. \tag{10}$$

  Obviously, the ith element of $\boldsymbol{e}$, e_i, can be recursively computed with low complexity by $e_i = e_{i+1} + |z_{i+1}|^2$, $i = 1, 2, \ldots, N-1$, and $e_N = 0$. After computing $\boldsymbol{e}$, we use it to obtain the neighbors' metrics:

$$\phi(\boldsymbol{x}) = e_d + \sum_{n=1}^{d}\left|z_n + r_{n,d}\delta_d\right|^2. \tag{11}$$

  This idea is further illustrated in Table II. It is shown that the sums ϕ_B of different neighbors have many common terms, which allows QR-TS to save a substantial number of operations compared to the conventional algorithm.
- *Over two successive searching iterations,* only a few first elements of $\boldsymbol{z}$ are updated. Note that between two successive iterations, m and $m+1$, $\boldsymbol{c}_{\{m\}}$ and $\boldsymbol{c}_{\{m+1\}}$ are neighbors of each other. In particular, $\boldsymbol{c}_{\{m+1\}}$ is the best neighbor of $\boldsymbol{c}_{\{m\}}$, i.e., $\boldsymbol{x}^{\star}_{\{m\}} = \boldsymbol{c}_{\{m+1\}}$. Let $d^{\star}$ be the difference position of $\boldsymbol{x}^{\star}_{\{m\}}$. Then, $\boldsymbol{c}_{\{m\}}$ and $\boldsymbol{c}_{\{m+1\}}$ are different only in the $d^{\star}$th element. Therefore, we have

$$\begin{aligned}&\boldsymbol{z}_{\{m+1\}} - \boldsymbol{z}_{\{m\}} \\ &= \left(\boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{c}_{\{m+1\}}\right) - \left(\boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{c}_{\{m\}}\right) \\ &= \boldsymbol{R}\left(\boldsymbol{c}_{\{m\}} - \boldsymbol{c}_{\{m+1\}}\right) = \boldsymbol{R}\Delta\boldsymbol{x}_{\{m\}} = \boldsymbol{r}_{d^{\star}}\delta_{d^{\star}} \\ &= \left[r_{1,d^{\star}}\delta_{d^{\star}}, \ldots, r_{d^{\star},d^{\star}}\delta_{d^{\star}}, 0, \ldots, 0\right]^T,\end{aligned}$$

  which implies that over two successive iterations, only the first $d^{\star}$ elements of $\boldsymbol{z}$ should be updated. Hence, in Table II, the algorithm only needs to update the first $d^{\star} - 1$ elements of $\boldsymbol{e}$, i.e., $e_1, e_2, \ldots, e_{d^{\star}-1}$, and the complexity for the others can be saved. Because it is relatively cheap

to compute ϕ_B for the neighbors, it is first computed in the process of calculating the cumulative metrics of neighbors. These advantages of QR-TS can significantly reduce the complexity of the TS algorithm, which will be further discussed in Section III-C.

In prior works regarding MIMO detection, the QR decomposition is used to simplify the metric function [30], [31], which results in overall complexity reduction. However, we note that in the proposed QR-TS algorithm, the QR decomposition is incorporated into not only the simplified cost metric but also the efficient metric computation, as described in this subsection, which can potentially result in larger complexity reduction.

2) Early Rejection: It is shown in (11) that to calculate the metric of the neighbor $\boldsymbol{x}$, d terms, i.e., $|z_1 + r_{1,d}\delta_d|^2$, ..., $|z_d + r_{d,d}\delta_d|^2$, need to be computed. It requires lower complexity than the conventional TS algorithm, where N terms are calculated as in (6). However, it is still inefficient and requires high complexity because of the large number of neighbors in large MIMO systems. In fact, the neighbors' metrics are distributed in a wide range, and a subset of neighbors have much larger metrics compared to the others. If these neighbors can be rejected early, the complexity of the TS algorithm can be reduced without causing any performance loss. Therefore, instead of considering the metric in (11), the QR-TS scheme computes the neighbors' metrics recursively as follows:

$$\begin{aligned}\phi_\eta(\boldsymbol{x}) &= \phi_B + \sum_{n=1}^{\eta} |z_n + r_{n,d}\delta_d|^2 \\ &= \phi_{\eta-1}(\boldsymbol{x}) + |z_\eta + r_{\eta,d}\delta_d|^2, \eta \le d, \end{aligned} \tag{12}$$

where $\phi_\eta(\boldsymbol{x})$ is the cumulative metric of $\boldsymbol{x}$ at the layer η, and $\phi_0(\boldsymbol{x}) = \phi_B = e_d$. Whether to reject or to keep a neighbor, after each layer, the cumulative metric $\phi_\eta(\boldsymbol{x})$ is compared to the threshold γ. As soon as $\phi_\eta(\boldsymbol{x})$ exceeds γ, the algorithm stops examining this neighbor. For $\eta < d$, we have $\phi_\eta(\boldsymbol{x}) < \phi(\boldsymbol{x})$, and $\phi_\eta(\boldsymbol{x})$ is referred to as an "incomplete metric." In contrast, the term "complete metric" indicates $\phi_\eta(\boldsymbol{x})$ for $\eta = d$. It is obvious that an incomplete metric requires lower complexity than a complete metric. To guarantee that the exact best neighbor is found, the threshold γ is set to infinity initially. Subsequently, if a smaller-than-γ complete metric is found, both γ and $\boldsymbol{x}^\star$ are updated. Through this process, after all neighbors are examined, γ finally becomes the smallest complete metric among the neighbors' complete metrics, and the best neighbor is the one with the metric γ, i.e., $\phi(\boldsymbol{x}^\star) = \gamma$. Consequently, the early rejection techniques allows QR-TS to find the best neighbor without computing all L complete metrics, which results in an overall complexity reduction.

The application of efficient metric computations and early rejection is further illustrated in Fig. 1. In this tree diagram, $\varphi_n^{[l]}$ given by

$$\varphi_n^{[l]} = \begin{cases} |z_n|^2, & n > d \\ |z_n + r_{n,d}\delta_d|^2, & n \le d \end{cases}$$

represents the metric of layer n of the lth neighbor with $n = 1, 2, \ldots, N$ and $l = 1, 2, \ldots, L$. Therefore, the incomplete metric of the lth neighbor, i.e., $\phi_{\eta_l}^{[l]}(\boldsymbol{x})$, is given by

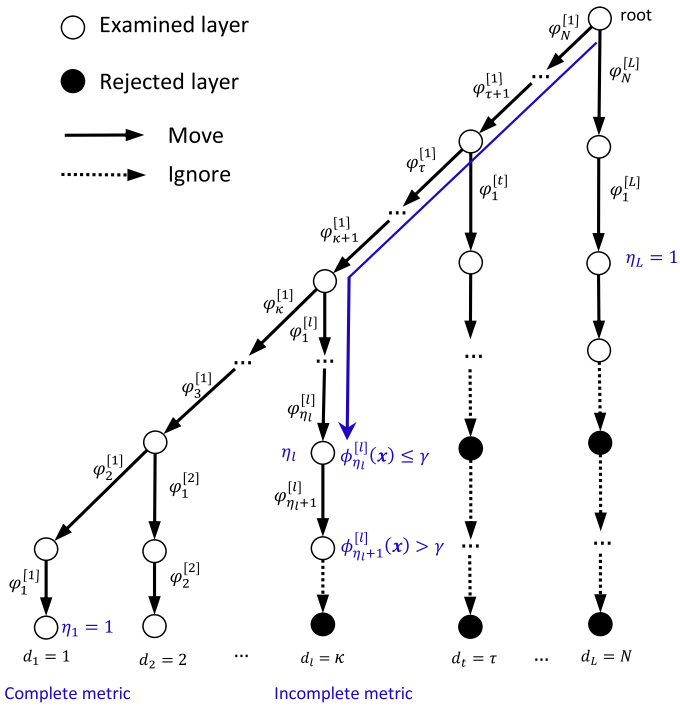


Fig. 1. Low-complexity computation and early rejection of QR-TS.

$$\phi_{\eta_l}^{[l]}(\boldsymbol{x}) = \sum_{n=d+1}^{N} \varphi_n^{[l]} + \sum_{n=1}^{\eta_l} \varphi_n^{[l]} = e_l + \sum_{n=1}^{\eta_l} \varphi_n^{[l]}. \tag{13}$$

In Fig. 1, the diagonal paths from the root represent the steps to find the vector $\boldsymbol{e}$ in (10) with low complexity. By contrast, the vertical paths show the process to compute the second sum in (13) to find the complete/incomplete metrics of the neighbors. For example, the early rejection process of the lth neighbor with difference position κ is illustrated by the κth vertical path in the figure. In this example, when $\phi_{\eta_l+1}^{[l]}(\boldsymbol{x}) > \gamma$, the QR-TS algorithm rejects this neighbor and moves on to the next neighbor.

The operation of the proposed QR-TS to find the best neighbor is summarized in Algorithm 1. Steps 1–4 are to find the sum ϕ_B for every neighbor, which is then stored in array $\boldsymbol{e}$. It is noted that all elements of $\boldsymbol{e}$ need to be computed in the first iteration. However, from the second iteration, only the first $d^\star - 1$ elements of $\boldsymbol{e}$ are updated, where $d^\star$ is the difference position between the current candidate $\boldsymbol{c}_{\{m\}}$ and the previous candidate $\boldsymbol{c}_{\{m-1\}}$. In steps 5 and 7, $\boldsymbol{x}_{[i]}$ is the ith element in the neighbor set. In steps 8–15, $\boldsymbol{e}$ is used to find the incomplete and complete metrics of neighbors. Steps 16–19 update the threshold γ and the best neighbor as soon as a smaller-than-γ complete metric is found.

B. QR-TS With Ordering Schemes

1) Receive Ordering (Rx-Ordering): In the computation of ϕ_A in (9), if the elements of received signals are arranged in the decreasing order of layer metrics $|z_n + r_{n,d}\delta_d|^2, n = 1, 2, \ldots, d$, the cumulative metric $\phi(\boldsymbol{x})$ can exceed the threshold γ earlier. As a result, the computational complexity in (9) can be reduced. Motivated by this fact, in the Rx-ordering scheme, we

Algorithm 1: Find the Best Neighbor in the QR-TS Algorithm.

Input: $\boldsymbol{c}, \boldsymbol{z}, \boldsymbol{R}, \mathcal{N}(\boldsymbol{c}), \mathcal{D}$, and $d^\star$.
Output: The best neighbor $\boldsymbol{x}^\star$.

1: $e_N = 0$
2: **for** $i = d^\star - 1 \to 1$ **do**
3: $\quad e_i = e_{i+1} + |z_{i+1}|^2$
4: **end for**
5: $\gamma = \infty, \boldsymbol{x}^\star = \boldsymbol{x}_{[1]}$
6: **for** $i = 1 \to L$ **do**
7: $\quad \boldsymbol{x} = \boldsymbol{x}_{[i]}, d = d_i$
8: $\quad \eta = 1, \phi_\eta(\boldsymbol{x}) = e_d$
9: $\quad$ **for** $\eta = 1 \to d$ **do**
10: $\quad\quad$ **if** $\phi_\eta(\boldsymbol{x}) \le \gamma$ **then**
11: $\quad\quad\quad \phi_\eta(\boldsymbol{x}) = \phi_\eta(\boldsymbol{x}) + |z_\eta + r_{\eta,d}\delta_d|^2$
12: $\quad\quad$ **else**
13: $\quad\quad\quad$ break
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: $\quad$ **if** $\phi_\eta(\boldsymbol{x}) < \gamma$ **then**
17: $\quad\quad \gamma = \phi_\eta(\boldsymbol{x})$
18: $\quad\quad \boldsymbol{x}^\star = \boldsymbol{x}$
19: $\quad$ **end if**
20: **end for**

sort the elements of $\boldsymbol{z} + \boldsymbol{R}\Delta\boldsymbol{x}$ by the expectations of layer metrics, i.e., $\mathbb{E}\{|z_n + r_{n,d}\delta_d|^2\}$, which is given in the following lemma.

Lemma 1: The average layer metric can be expressed as

$$\mathbb{E}\left\{|z_n + r_{n,d}\delta_d|^2\right\} \approx \sigma_v^2 + |r_{n,d}|^2\,\delta^2, \tag{14}$$

where $\delta^2 = |\delta_d|^2$.

Proof: See Appendix A. ■

As a result, within an iteration, the expected metric of the nth layer of a neighbor only depends on $|r_{n,d}|^2$. In Rx-ordering, we aim to accelerate the increase of the cumulative metric. Therefore, the layer with the largest metric should be considered first. In other words, for the computation of ϕ_A in Rx-ordered QR-TS, the terms corresponding to the larger $|r_{n,d}|^2$ are computed and accumulated earlier. It is worth noting that Rx-ordering only changes the computation order in the summation expressed in (12) and does not cause any changes in R and Q.

2) Transmit Ordering (Tx-Ordering): If we order the channel columns such that the neighbors with smaller cost metrics are examined earlier, the threshold γ will decrease more quickly, which certainly reduces the complexity of QR-TS by increasing the chances of early rejection. This motivates the development of an ordering scheme at the transmit side that is "Tx-ordering." Intuitively, the cost metrics of all neighbors depend on the channel matrix. However, in the QR-TS algorithm, within one searching iteration, the cost metrics of different neighbors depend on different channel columns $\boldsymbol{r}_d$, as shown in (8). Therefore, an ordering scheme based on channel columns can result in complexity reduction. For the development of Tx-ordering, we investigate the average metric of a neighbor, which is given in the following lemma.

Lemma 2: The average metric of a neighbor is given as

$$\mathbb{E}\left\{\phi\left(\boldsymbol{x}\right)\right\} = N\sigma_v^2 + \delta^2\left\|\boldsymbol{h}_d\right\|^2. \tag{15}$$

Proof: See Appendix B. ■

In (15), N, σ_v^2, and δ^2 are constant. Therefore, the average metric of a neighbor depends only on the norm of the corresponding channel column, i.e., $\|\boldsymbol{h}_d\|$, which implies that the neighbors can be ordered optimally based on the norms of the channel columns.

In the QR-TS algorithm, in contrast to the conventional TS, the complexity in metrics computation is different among the neighbors. Specifically, it is shown from (9) that the computation of a neighbor metric with smaller d requires lower complexity. Therefore, the neighbors with the smaller value d should be considered earlier. If we denote $\underline{\mathcal{D}} = \{d_{[1]}, d_{[2]}, \ldots, d_{[L]}\}$ as the increasing order of elements of $\mathcal{D}$, the neighbors should be examined in the order $\underline{\mathcal{N}}(\boldsymbol{c}) = \{\boldsymbol{x}_{[1]}, \boldsymbol{x}_{[2]}, \ldots, \boldsymbol{x}_{[L]}\}$, where $\boldsymbol{x}_{[i]}$ corresponds to $d_{[i]}$. Hence, (15) can be rewritten as

$$\mathbb{E}\left\{\phi\left(\boldsymbol{x}_{[i]}\right)\right\} = N\sigma_v^2 + \delta^2\left\|\boldsymbol{h}_{d_{[i]}}\right\|^2. \tag{16}$$

However, in Tx-ordering, to accelerate the decreasing of the threshold γ, the neighbor with a smaller average metric should be examined earlier. Therefore, the neighbors in $\underline{\mathcal{N}}(\boldsymbol{c})$ should be sorted such that $\mathbb{E}\{\phi(\boldsymbol{x}_{[1]})\} \le \mathbb{E}\{\phi(\boldsymbol{x}_{[2]})\} \le \ldots \le \mathbb{E}\{\phi(\boldsymbol{x}_{[L]})\}$, which is guaranteed by the condition $\|\boldsymbol{h}_1\|^2 \le \|\boldsymbol{h}_2\|^2 \le \ldots \le \|\boldsymbol{h}_N\|^2$, as indicated by (16). In conclusion, in Tx-ordering, the channel columns are arranged by the increasing order of their norms to improve the efficiency of early rejection. Subsequently, the neighbors with smaller d are examined earlier to further optimize the overall complexity.

Algorithm 2 summarizes the proposed QR-TS detection algorithm with the ordering schemes. In Algorithm 2, steps 1–5 and 7–16 correspond to Tx-ordering and Rx-ordering, respectively. When Tx-ordering and Rx-ordering are not applied, $\underline{\boldsymbol{H}}$ and $\underline{\boldsymbol{r}_d}$ are set to the original unordered ones, as indicated in steps 4 and 15, respectively. Steps 17–19 are for the initialization phase, whereas steps 20–38 present the iterative searching phase. In each iteration, the neighbors are ordered for Tx-ordering in step 22. Algorithm 1 is then used to find the best neighbor in step 27. It is worth noting that for each neighbor corresponding to d, the elements of $[z_1, z_2, \ldots, z_d]$ are also sorted in the order $\mathcal{S}_d$ to generate $\underline{\boldsymbol{z}}$. The metric is now expressed as

$$\phi(\boldsymbol{x}) = \left\|\underline{\boldsymbol{z}} + \underline{\boldsymbol{r}_d}\delta_d\right\|^2. \tag{17}$$

Finally, after M searching iterations, the output in step 39 is the best solution found. Henceforth, we refer to the case when both Tx- and Rx-ordering are applied as "double-ordering."

It is also noteworthy that the proposed Tx-ordering scheme has a different objective, procedure, and benefits compared with the ordering scheme used in the LTS algorithm [23]. Specifically, in LTS, post-detection SNR-based ordering is performed to order symbols such that the interference among the layers is minimized. This scheme can improve detection performance, but requires high complexity [23]. In contrast, the proposed Tx-ordering reduces the complexity of the QR-TS algorithm with no loss in performance.

Algorithm 2: QR-TS Detection with Tx- and Rx-ordering.

Input: $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N]$, $\boldsymbol{y}$
1: **if** Tx-ordering is used **then**
2: Find $\underline{\boldsymbol{H}}$ by sorting the columns of $\boldsymbol{H}$ in the ascending order of $\|\boldsymbol{h}_n\|^2$.
3: **else**
4: $\underline{\boldsymbol{H}} = \boldsymbol{H}$
5: **end if**
6: Obtain $\boldsymbol{Q}$ and $\boldsymbol{R}$ by QR-decomposition of $\underline{\boldsymbol{H}}$.
7: $\boldsymbol{z} = \boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{c}$
8: **if** Rx-ordering is used **then**
9: **for** $d = 1$ to N **do**
10: $\mathcal{S}_d = \{1, \ldots, d\}$
11: Find $\underline{\mathcal{S}_d}$ by sorting the elements of $\mathcal{S}_d$ in the descending order of $|r_{n,d}|^2, n = 1, 2, \ldots, d$.
12: Find $\underline{\boldsymbol{r}_d}$ by sorting the first d elements of $\boldsymbol{r}_d$ in the order $\underline{\mathcal{S}_d}$.
13: **end for**
14: **else**
15: $\underline{\boldsymbol{r}_d} = \boldsymbol{r}_d$
16: **end if**
17: $\boldsymbol{x}_{ZF} = \ulcorner \boldsymbol{H}^{\dagger}\boldsymbol{y} \lrcorner$
18: Initialize $\boldsymbol{c}_{\{1\}}$ based on $\boldsymbol{x}_{ZF}$, push $\boldsymbol{c}_{\{1\}}$ to the tabu list $\mathcal{L}$, $p = 1$ (current length of $\mathcal{L}$).
19: $\hat{\boldsymbol{s}}_{TS} = \boldsymbol{c}_{\{1\}}, \phi(\hat{\boldsymbol{s}}_{TS}) = \phi(\boldsymbol{c}_{\{1\}}), \boldsymbol{c} = \boldsymbol{c}_{\{1\}}, d^{\star} = N$
20: **for** $m = 1$ to M **do**
21: Find the neighbor set $\mathcal{N}(\boldsymbol{c})$ and the corresponding set of difference positions $\mathcal{D}$.
22: **if** Tx-ordering **then**
23: Sort the neighbors in the ascending order of the elements of $\mathcal{D}$: $\underline{\mathcal{N}}(\boldsymbol{c}) = \{\boldsymbol{x}_{[1]}, \boldsymbol{x}_{[2]}, \ldots, \boldsymbol{x}_{[L]}\}$.
24: **else**
25: $\underline{\mathcal{N}}(\boldsymbol{c}) = \mathcal{N}(\boldsymbol{c})$
26: **end if**
27: Sort the first d elements of $\boldsymbol{z}$ in the order $\underline{\mathcal{S}_d}$.
28: Use Algorithm 1 to find the best neighbor $\boldsymbol{x}^{\star}$ in the neighbor set $\underline{\mathcal{N}}(\boldsymbol{c})$
29: Set $d^{\star}$ to the difference position of $\boldsymbol{x}^{\star}$.
30: Move to the best neighbor, $\boldsymbol{c} = \boldsymbol{x}^{\star}$, and update $\boldsymbol{z}$.
31: **if** $\phi(\boldsymbol{c}) < \phi(\hat{\boldsymbol{s}}_{TS})$ **then**
32: Update the best solution: $\hat{\boldsymbol{s}}_{TS} = \boldsymbol{c}, \phi(\hat{\boldsymbol{s}}_{TS}) = \phi(\boldsymbol{c})$.
33: **end if**
34: **if** Tabu list $\mathcal{L}$ is full **then**
35: Release the first element in $\mathcal{L}$, $p = p - 1$.
36: **end if**
37: Push $\boldsymbol{c}$ to $\mathcal{L}$.
38: Update the length of the tabu list, $p = p + 1$.
39: **end for**
40: The final solution is the best solution $\hat{\boldsymbol{s}}_{TS}$ found so far.

C. Complexity Analysis

In this section, we examine the amount of computation reduction achieved by the proposed QR-TS scheme by comparing it to the conventional algorithm in terms of complexity.

1) Initialization Phase: First, we analyze the initialization phase, where the ZF-based candidate $\boldsymbol{c}_{\{1\}}$ is determined. In this work, we assume that the QR Householder method is used to perform QR decomposition, which requires $\frac{2}{3}N^3$ multiplications and $\frac{2}{3}N^3$ additions [32]. In QR-TS, unlike the conventional TS algorithm, the initial solution $\boldsymbol{c}_{\{1\}}$ can be determined as $\boldsymbol{c}_{\{1\}} = \boldsymbol{x}_{ZF} = \ulcorner \boldsymbol{H}^{\dagger}\boldsymbol{y} \lrcorner = \ulcorner \boldsymbol{R}^{-1}\boldsymbol{Q}^T\boldsymbol{y} \lrcorner$. Here, the computation of $\boldsymbol{Q}^T\boldsymbol{y}$ requires N^2 multiplications and $N^2 - N$ additions. Because $\boldsymbol{R}$ is an upper triangular matrix, solving for $\boldsymbol{x}_{ZF}$ by performing the backward substitution costs $\frac{1}{2}N^2 + \frac{1}{2}N$ multiplications and $\frac{1}{2}N^2 - \frac{1}{2}N$ additions [32], [33].

The cost metric of $\boldsymbol{c}_{\{1\}}$ is computed as

$$\phi\left(\boldsymbol{c}_{\{1\}}\right) = \left\|\boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{c}_{\{1\}}\right\|^2, \tag{18}$$

which requires $\frac{1}{2}N^2 + \frac{3}{2}N$ multiplications and $\frac{1}{2}N^2 + \frac{1}{2}N$ additions, with the note that most operations are to compute $\boldsymbol{R}\boldsymbol{c}_{\{1\}}$ because $\boldsymbol{Q}^T\boldsymbol{y}$ is already computed while finding $\boldsymbol{c}_{\{1\}}$. Therefore, the total complexity of the initialization phase of QR-TS is $\frac{2}{3}N^3 + 2N^2 + 2N$ multiplications and $\frac{2}{3}N^3 + 2N^2 - N$ additions.

2) Searching Phase: Similar to the computation of $\boldsymbol{u}$ in the conventional TS, the computation of $\boldsymbol{z}$ is only required in the first searching iteration. In the subsequent iterations, $\boldsymbol{z}$ is automatically obtained during the neighbor-searching process. Specifically, $\boldsymbol{z}$ in the $(m+1)$th iteration is expressed as

$$\boldsymbol{z}_{\{m+1\}} = \boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{c}_{\{m+1\}} = (\boldsymbol{z} + \boldsymbol{R}\Delta\boldsymbol{x})_{\{m\}}, \tag{19}$$

where $\boldsymbol{z}_{\{m\}} = \boldsymbol{Q}^T\boldsymbol{y} - \boldsymbol{R}\boldsymbol{c}_{\{m\}}$ according to the definition of $\boldsymbol{z}$ in (8). In the mth iteration, as an essential procedure to find the best neighbor, $(\boldsymbol{z} + \boldsymbol{R}\Delta\boldsymbol{x})_{\{m\}}$ are computed for $L_{\{m\}}$ neighbors, and one of them is $\boldsymbol{z}_{\{m+1\}}$, as shown in (19). Therefore, the computational loads for $\boldsymbol{z}$ in the searching phase is ignored in the following analysis.

a) Complexity in computing ϕ_B: The most intuitive factor for the computational saving of QR-TS is the QR decomposition advantage. Because $\boldsymbol{R}$ is an upper triangular matrix and $\Delta\boldsymbol{x}$ is a single-nonzero vector, the computational complexity for $\boldsymbol{R}\Delta\boldsymbol{x}$ is much lower than that for $\boldsymbol{H}\Delta\boldsymbol{x}$. Furthermore, the zeros in $\boldsymbol{R}\Delta\boldsymbol{x} = \boldsymbol{r}_d\delta_d = [r_{1,d}\delta_d, \ldots, r_{d,d}\delta_d, 0, \ldots, 0]^T$ lead to the low-complexity computation as described in Section III-A1, where the computations of $\phi_B = \sum_{n=d+1}^{N} |z_n|^2$ are performed only once for all L neighbors. More specifically, the complexity to find ϕ_B for all neighbors are equal to that to find $\boldsymbol{e}$ in (10). According to Fig. 1 and steps 2–4 in Algorithm 1, it requires only $N - 1$ multiplications and $N - 2$ additions.

b) Average complexity to find the best metric: Another important factor for the low complexity of QR-TS is the early rejection scheme described in Section III-A2. We recall that in QR-TS, it takes only $N - 1$ multiplications and $N - 2$ additions to compute ϕ_B. Furthermore, $\sum_{i=1}^{L} 2\eta_i$ multiplications and $\sum_{i=1}^{L} 2\eta_i - 1$ additions are required to compute the summation in (12) for all neighbors corresponding to the vertical paths in Fig. 1. Let $\bar{\eta}$ be the average value of $\eta_i, i = 1, 2, \ldots, L$. The average computational cost of QR-TS to find the best neighbor in an iteration are $N - 1 + 2\bar{L}\bar{\eta}$ multiplications and $N - 2 + \bar{L}(2\bar{\eta} - 1)$ additions. Furthermore, it can be observed in Algorithm 1 that the QR-TS algorithm requires an average of $\bar{L}(\bar{\eta} + 1)$ comparisons for steps 10 and 16 in each iteration.

In Table III, we summarize and compare the computational complexities of the conventional TS and the proposed QR-TS

TABLE III
COMPARISON OF THE NUMBER OF MULTIPLICATIONS, ADDITIONS, AND COMPARISONS OF THE CONVENTIONAL TS AND QR-TS ALGORITHMS

Process		Conventional TS			QR-TS		
		Mul.	Add.	Comp.	Mul.	Add.	Comp.
Initialization phase		$N^3/3+3N^2/2$ $+N/6$	$N^3/3+3N^2/2$ $+N/6-1$	0	$2N^3/3$ $+2N^2+2N$	$2N^3/3$ $+2N^2-N$	0
Find the best neighbor in one iteration	Average	$2\bar{L}N$	$\bar{L}(2N-1)$	$\bar{L}$	$N-1+2\bar{L}\bar{\eta}$	$N-2$ $+\bar{L}(2\bar{\eta}-1)$	$\bar{L}(\bar{\eta}+1)$
	Upper bound (QPSK, BPSK)	$2N^2-2N$	$2N^2-3N+1$	$N-1$	N^2-1	N^2-N-1	$N^2/2+N/2-1$
	Upper bound (16-QAM, 64-QAM)	$4N^2$ $-2(\alpha+1)N$	$4N^2$ $-2(\alpha+2)N$ $+(\alpha+1)$	$2N-\alpha-1$	$2N^2-\alpha N-1$	$2N^2$ $-(\alpha+2)N$ $+(\alpha-1)$	$N^2-(\alpha-3)N/2$ $-(\alpha+1)$

algorithms. Additionally, we present the upper bounds on their complexities to find the best neighbor. It should be noted that $\bar{L} \leq N-1$ for BPSK/QPSK and $\bar{L} \leq 2N-\alpha-1$ for higher-order modulation schemes, such as 16- and 64-QAM, and that $\bar{\eta} \leq \bar{d} \approx \frac{N}{2}$. These analytical results will be verified numerically in Section IV. From Table III, it is clear that, in large-sized MIMO systems, the QR-TS algorithm requires a complexity approximately two times lower to find the best neighbor compared with the conventional TS algorithm. Furthermore, the average overall complexities to find the best neighbor in the conventional TS and QR-TS algorithms can be expressed as

$$\mathcal{C}_{TS} = 4\bar{L}N, \tag{20}$$

$$\mathcal{C}_{QR-TS} = 2N + 5\bar{\eta}\bar{L} - 3, \tag{21}$$

respectively. The upper bounds of $\mathcal{C}_{TS}$ for BPSK/QPSK and higher-order modulation schemes are given by

$$\mathcal{C}_{TS}^{ub,low} = 4N^2 - 4N, \tag{22}$$

$$\mathcal{C}_{TS}^{ub,high} = 8N^2 - 4(\alpha+1)N, \tag{23}$$

respectively. On the other hand, the corresponding upper bounds of $\mathcal{C}_{QR-TS}$ are

$$\mathcal{C}_{QR-TS}^{ub,low} = \frac{5}{2}N^2 - \frac{1}{2}N - 3, \tag{24}$$

$$\mathcal{C}_{QR-TS}^{ub,high} = 5N^2 - \frac{5\alpha+1}{2}N - 3. \tag{25}$$

It is worth noting that the complexity of QR-TS significantly depends on $\bar{\eta}$ as shown in (21). In the QR-TS algorithm, N and $\bar{L}$ can be considered as constants, and $\bar{\eta}$ is the unique factor that decides the overall complexity, which motivates us to develop techniques to reduce $\bar{\eta}$, or equivalently, accelerate the increase of cumulative metrics. In the next section, the ordering techniques to achieve this goal will be presented.

3) Complexity of Ordering Schemes: In Algorithm 2, the computations for the Tx- and Rx-ordering schemes are performed in steps 2 and 11, respectively. To perform step 2 for Tx-ordering, the norms of the N columns of $\boldsymbol{H}$, i.e., $\|\boldsymbol{h}_n\|^2, n = 1, \ldots, N$ are required. However, for the real channel matrix expressed in (2), we have $\|\boldsymbol{h}_i\|^2 = \|\boldsymbol{h}_{i+\frac{N}{2}}\|^2, i = 1, \ldots, \frac{N}{2}$. Therefore, only $\frac{N}{2}$ norm values, i.e., $\|\boldsymbol{h}_i\|^2, i = 1, \ldots, \frac{N}{2}$, need to be computed, which requires $\frac{N^2}{2}$ multiplications and $\frac{N^2-N}{2}$ additions. Furthermore, assuming that the quick-sort algorithm [34] is used to sort the channel columns, $\frac{N}{4}(\log_2 N - 1)$ comparisons are required on average. Therefore, the total complexity of Tx-ordering is $N^2 - \frac{3N}{4} + \frac{N}{4}\log_2 N$ operations.

In Rx-ordering, for each d, the computations and sorting of $|r_{n,d}|^2, n = 1, 2, \ldots, d$ in step 11 require d multiplications and $\frac{1}{2}$ $d \log_2 d$ comparisons on average if the quick-sort algorithm [34] is used. For $d = 1, 2, \ldots, N$, the total complexity of Rx-ordering becomes $\sum_{d=1}^{N}(d + \frac{1}{2}d\log_2 d) = \frac{1}{2}(N^2 + N + \log_2 \mathcal{H}(N))$ operations, where $\mathcal{H}(\cdot)$ is the hyperfactorial function [35].

In Algorithm 2, the computations of both Tx- and Rx-ordering are performed outside the iterative searching process of M iterations. Therefore, although $N^2 - \frac{3N}{4} + \frac{N}{4}\log_2 N$ and $\frac{1}{2}(N^2 + N + \log_2 \mathcal{H}(N))$ operations are required for Tx- and Rx-ordering, respectively, these amounts are much smaller than the total complexities required to find the best neighbors in M searching iterations. As a result, despite the additional required complexity for ordering, the complexity advantages of the proposed ordering schemes are still significant. This will be numerically verified in the next section.

IV. SIMULATION RESULTS

In this section, we numerically evaluate the BER performance and computational complexities of the proposed schemes, which are also compared to those of the conventional TS, LTS [23], Schnorr–Euchner SD (SE-SD) [36], and KSD receivers [5], [7]. For a fair comparison, in the simulations, we used the real-valued signal model for QR-TS, LTS, and conventional TS algorithm. In our simulations, each channel coefficient is assumed to be an i.i.d. zero-mean complex Gaussian random variable with a variance of $1/2$ per dimension. The SNR is defined as the ratio of the average symbol power σ_t^2 to the noise power σ_v^2.

A. BER Performance of the Proposed Schemes

In Fig. 2, we compare the BER performance of the proposed QR-TS algorithms to those of the conventional TS and SE-SD receivers in various MIMO environments, namely, 16×16 and 32×32 MIMO with QPSK, 8×8 and 16×16 MIMO with 16-QAM, and 8×8 MIMO with 64-QAM.[2] For each environment, it is shown that with the chosen M and P values, which

[2]We note that the complexity reduction through effective metric computation, early rejection, and ordering schemes can be achieved for every antenna configuration. However, in this study, we focus on a typical MIMO assumption of $N_r = N_t$, which is generally considered a practical antenna configuration to simultaneously achieve high data rates and relatively low complexities.

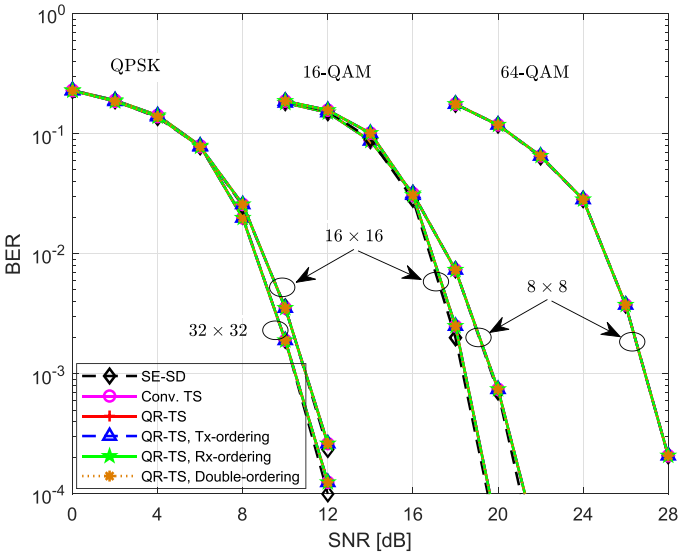


Fig. 2. BER performance comparison for 16×16 and 32×32 MIMO with QPSK, 8×8 and 16×16 MIMO with 16-QAM, and 8×8 MIMO with 64-QAM.

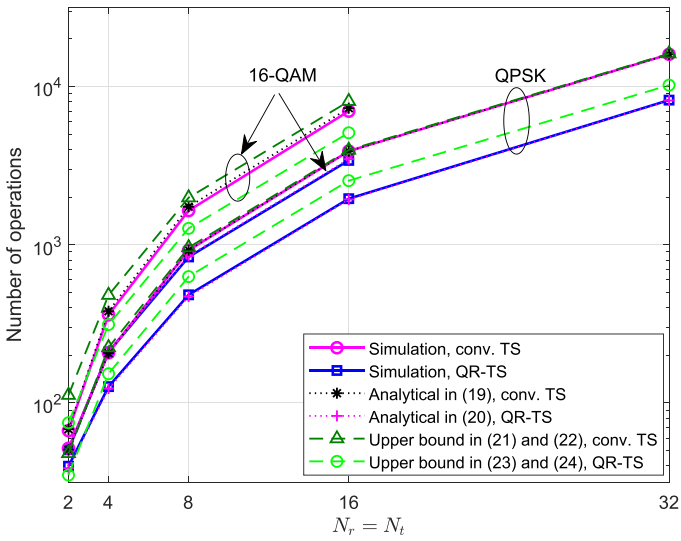


Fig. 3. Average computational complexity to find the best neighbor for the cases of $M = \{6, 40, 150, 400, 800, 2000\}$ corresponding to $N_t = \{2, 4, 8, 16, 32, 64\}$, SNR = 12 dB, QPSK, and $M = \{100, 400, 2500, 8000\}$ corresponding to $N_t = \{2, 4, 8, 16\}$, SNR = 22 dB, and 16-QAM. $N_r = N_t$ and $P = M/2$ are assumed.

TABLE IV
M, P FOR QR-TS AND THE CONVENTIONAL TS TO ACHIEVE SE-SD PERFORMANCES IN VARIOUS SYSTEMS, AND $\bar{L}$, $\bar{\eta}$ OBTAINED BY SIMULATIONS

Modulation scheme	$N_t \times N_r$	$[M, P]$	$\bar{L}$	$\bar{\eta}$
QPSK	2×2	$[6, 3]$	2.5	2.13
	4×4	$[40, 20]$	6.45	3.42
	8×8	$[150, 75]$	14.44	6.12
	16×16	$[400, 200]$	30.50	12.28
	32×32	$[800, 400]$	62.43	25.75
	64×64	$[2000, 1000]$	126.20	56.25
16-QAM	2×2	$[100, 50]$	4.16	2.13
	4×4	$[400, 200]$	11.87	3.38
	8×8	$[2500, 1250]$	27.15	6.04
	16×16	$[8000, 4000]$	57.02	11.98

are presented in Table IV, the BER performances provided by the conventional TS and QR-TS algorithms are approximately the same as that of the SE-SD decoder. It is also shown that the proposed schemes including the QR-TS and its variations with the ordering schemes totally preserve the BER performance of the conventional TS algorithm.

For the subsequent simulations carried to compare the computational complexity, we chose the values of M and P such that QR-TS achieves a performance approximately equal to that of SE-SD, at BERs between 10^{-3} and 10^{-4} in each environment. Therefore, the determined values of M and P guarantee that the considered algorithms require nearly the same SNRs to achieve BERs in the 10^{-3}–10^{-4} range. This ensures a fair comparison between the algorithms in terms of complexity. We note that in the TS algorithm, the length of the tabu list, i.e., P, affects the performance. Specifically, a small P can result in a high chance of cycling in the search, whereas large P can result in a high chance of forbidding necessary moves to obtain the ML solution. Based on the simulations, we observed that that the optimal performance is achieved approximately for $P = M/2$. Therefore, we assume $P = M/2$ for our simulations, which is also used in [27]. Table IV lists the values of M and P for each environment, which are used in the simulations to compare the complexity of the algorithms.

B. Complexity to Find the Best Neighbor

In Section III-B, we showed that the proposed QR-TS algorithm can significantly reduce the complexity required to find the best neighbor via efficient computation and early rejection. In this section, we numerically verify it. First, via simulations, we obtained the average number of neighbors, $\bar{L}$, and the average number of accumulated layers in (13), $\bar{\eta}$, for each environment. The values of $\bar{L}$ and $\bar{\eta}$ are presented in Table IV and used to compute the average complexity to find the best neighbor.

In Fig. 3, the average complexity required to find the best neighbor in the conventional TS and the proposed QR-TS algorithms are plotted. Specifically, the simulation, analytical results, and upper bounds are compared for the cases of $N_t = \{2, 4, 8, 16, 32, 64\}$ with QPSK and $N_t = \{2, 4, 8, 16\}$ with 16-QAM. In Fig. 3, the analytical results and upper bounds for the conventional TS and QR-TS algorithms obtained using (20)–(25) are shown. Furthermore, to plot the upper bounds of the complexities in (23) and (25) for 16-QAM, we assume $\alpha = 0$, which results in the highest complexities. The values for M, P, $N_t, \bar{L}$, and $\bar{\eta}$ are chosen from Table IV. It is clear that the analytical expressions correctly reflect the computational costs of the considered algorithms. It is also shown that the upper bounds provide close approximations to the computational complexities, especially for QPSK modulation. For 16-QAM modulation, the upper bounds are not as close to the simulation and analytical results because the highest complexity is plotted for 16-QAM, i.e., $\alpha = 0$. In this figure, it is shown that the QR-TS

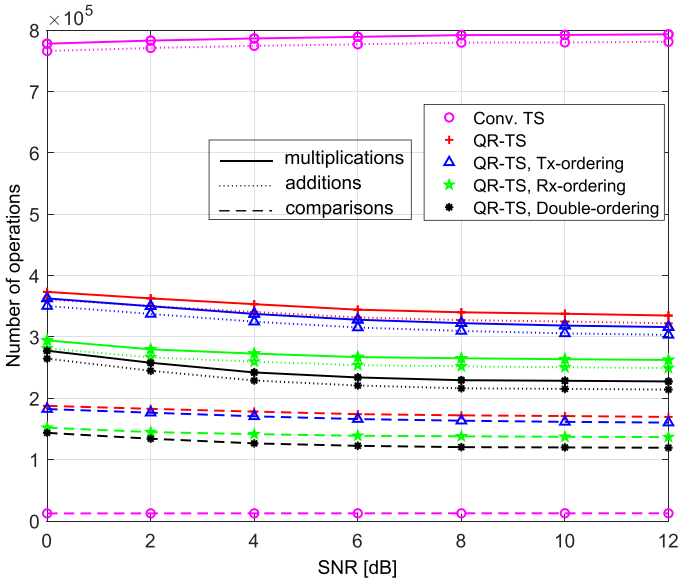


Fig. 4. Average numbers of multiplications, additions, and comparisons of the proposed schemes in comparison with those of the conventional TS for 16×16 MIMO, $M = 400$, $P = 200$, and QPSK.

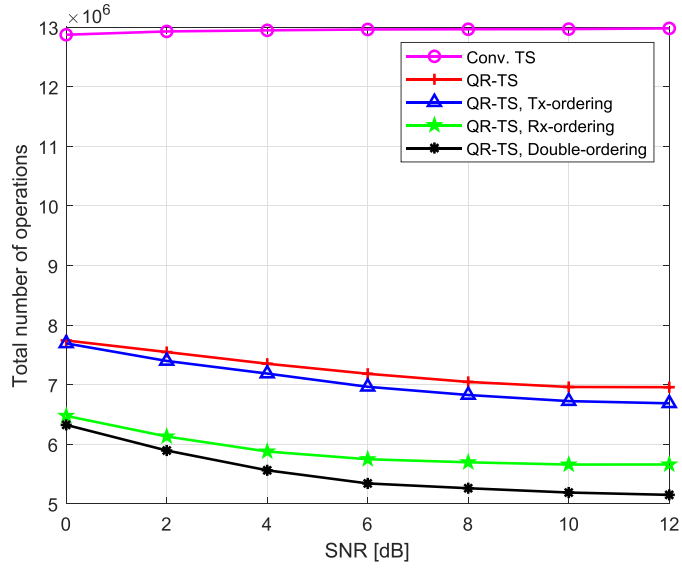
algorithm requires a much lower average complexity to find the best neighbor compared to the conventional TS.

C. Complexity Comparison With Conventional TS

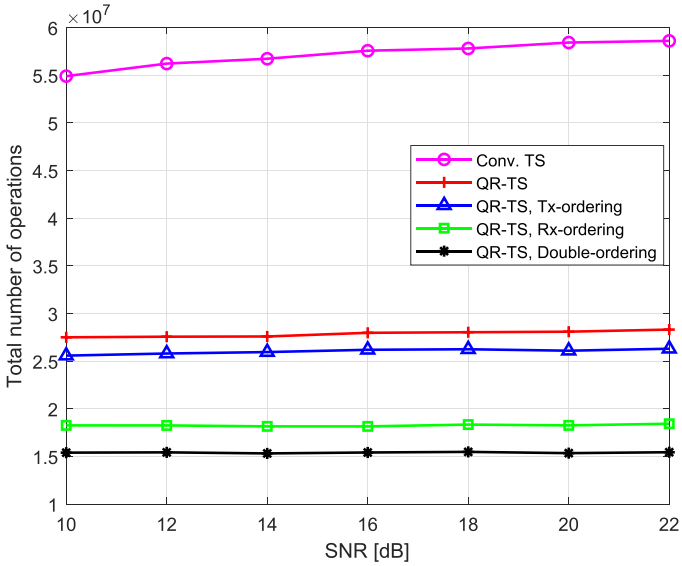
In this subsection, the complexities of the proposed QR-TS algorithms are compared with that of the conventional TS algorithm. Fig. 4 shows the numbers of additions, multiplications, and comparisons required by the conventional TS algorithm and the proposed schemes. In Fig. 4, it can be observed that QR-TS and its variants using the ordering schemes require significantly lower complexities compared with the conventional TS algorithm in terms of the numbers of additions and multiplications involved. It should be noted that although Fig. 4 shows that the proposed schemes require a larger number of comparisons than the conventional TS algorithm, its proportion to the overall complexity is small.

For simplicity, in the remaining comparisons of complexity, only the overall complexity, which is the total number of additions, multiplications, and comparisons, is presented. In Fig. 5, we compare the overall complexities of the proposed schemes to that of the conventional TS algorithm in various environments, namely, 32×32 MIMO with QPSK and $[M, P] = [800, 400]$ in Fig. 5(a), and 16×16 MIMO with 16-QAM and $[M, P] = [8000, 4000]$ in Fig. 5(b). From Fig. 5, the following observations are noted:

- It is clear that the QR-TS is capable of significantly reducing the complexity of the TS algorithm. More specifically, the QR-TS algorithm requires approximately only half the complexity needed for the conventional TS.
- When ordering techniques are applied, the QR-TS can further reduce the complexity. It is shown that Rx-ordering achieves a greater complexity reduction than Tx-ordering. Furthermore, when double-ordering is employed, the complexity reduction ratio of the QR-TS compared with the conventional TS algorithm is 60.8% and 74.1% for 32×32 MIMO with QPSK modulation and 16×16 MIMO with 16-QAM modulation, respectively.
- In the proposed schemes, the complexity reduction becomes more significant as the SNR increases. The reason is that a higher SNR results in a higher chance of early rejection.



(a) 32×32 MIMO, $M = 800$, $P = 400$, and QPSK



(b) 16×16 MIMO, $M = 8000$, $P = 4000$, and 16-QAM

Fig. 5. Average computational complexities of the proposed schemes in comparison with that of the conventional TS algorithm.

The complexity reduction of the proposed schemes for the case $N_r > N_t$ with QPSK modulation is shown in Fig. 6 for $N_r = 32$ and $N_t = \{2, 4, 8, 16\}$. The SNR for each antenna configuration is selected as indicated in Fig. 6 such that all the examined algorithms achieve BERs between 10^{-3} and 10^{-2}. Furthermore, M and P are selected from Table IV based on N_t. It is observed that for $N_r > N_t$, the proposed schemes require

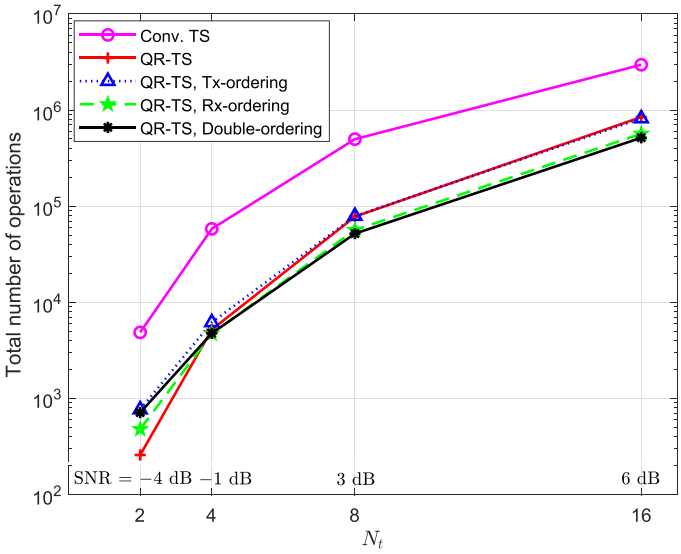


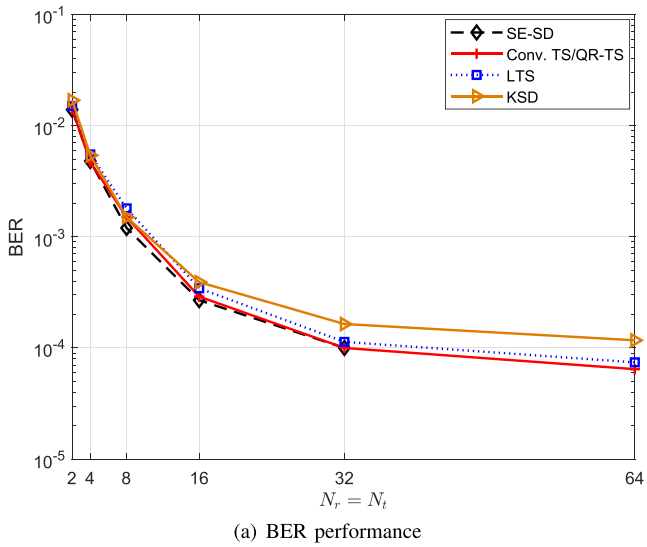
Fig. 6. Comparison of complexities of conventional TS and QR-TS for $N_r = 32$, $N_t = \{2, 4, 8, 16\}$, corresponding to $M = \{6, 40, 150, 400\}$, $P = M/2$, SNR $= \{-4, -1, 3, 6\}$ dB, and QPSK.

significantly lower complexities compared with the conventional TS algorithm. For example, in the case of $(N_t, N_r) = (8, 32)$ with SNR $= 3$ dB, the QR-TS with double ordering achieves approximately 90% complexity reduction with respect to the conventional TS scheme. In the case of $N_t \ll N_r$ such as $(N_t, N_r) = (2, 32)$, the number of neighbors in each searching iteration becomes small. Therefore, the complexity reduction from the ordering cannot compensate for the complexities required for ordering. Consequently, in this case, the QR-TS algorithm without ordering becomes more computationally efficient than that with ordering schemes.

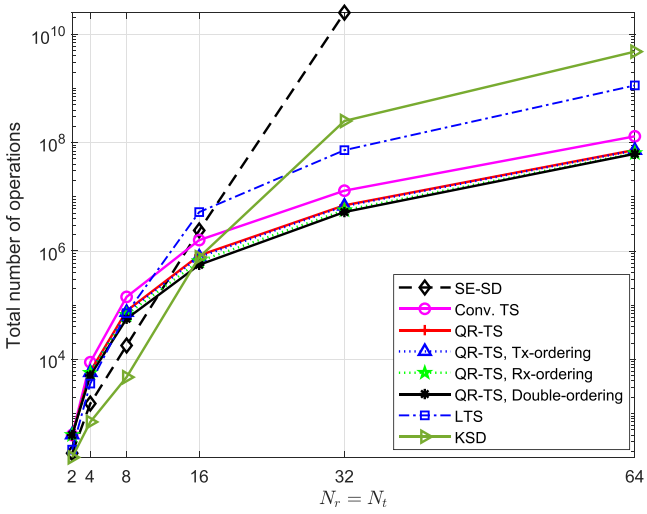
D. Complexity Comparison With the LTS, SE-SD, and KSD Schemes

Finally, we compare the computational complexities of the proposed schemes with those of the LTS [23], SE-SD [36], and KSD schemes [5], [7] as well as with conventional TS. We note that SE-SD and KSD are two of the major variants of the original SD algorithm, which are capable of significantly reducing the complexity of SD while achieving the near-optimal performance [5], [36]. Among the variants of TS-based detection, LTS, which also uses QR-decomposition, is chosen for comparison to show the different effects of QR-decomposition to the complexity reduction in LTS and the proposed QR-TS algorithm. The comparisons are made for various environments corresponding to $N_t = N_r \in \{2, 4, 8, 16, 32, 64\}$, with QPSK modulation. For these simulations, we set SNR $= 12$ dB, for which all the examined algorithms achieve BERs between 10^{-3} and 10^{-4} in large MIMO systems. To ensure a fair comparison in terms of complexity, we first performed simulations to find the appropriate parameters for the LTS and KSD algorithms so that they provide approximately the same BER performances as the TS, QR-TS, and SE-SD algorithms.

It should be noted that QR-TS and its ordering schemes totally preserve the performance of the conventional TS scheme, as shown in Figs. 2 and 3. Therefore, in Fig. 7(a), only a



(a) BER performance



(b) Computational complexity

Fig. 7. Comparison of the performances and complexities of the conventional TS, QR-TS, LTS, KSD, and SE-SD schemes with QPSK and SNR $= 12$ dB. The parameters are assumed to be $N_t = N_r \in \{2, 4, 8, 16, 32, 64\}$, corresponding to $M = \{6, 40, 150, 400, 800, 2000\}$, $P = M/2$ for TS and QR-TS, $M_{LTS} = \{4, 10, 30, 280, 500, 1000\}$, $P_{LTS} = M_{LTS}/2$, $\delta = 0.25$ for LTS, and $K = \{3, 4, 10, 10000, 100000, 500000\}$ for KSD.

single curve is plotted to present their performances. For LTS, the skipping criterion δ is assumed to be 0.25, as in [23]. For $N_t = N_r \in \{2, 4, 8, 16, 32, 64\}$, the number of searching iterations is set to $M_{LTS} = \{4, 10, 30, 280, 500, 1000\}$, respectively, while the value of K in KSD [5], [7] is chosen to be $K = \{3, 4, 10, 10000, 100000, 500000\}$, respectively. Because of the extremely high complexity of the SE-SD scheme for $N_t = 64$, which makes it infeasible to numerically test it in a reasonable time, SE-SD is not examined for $N_t = 64$. In Fig. 7(a), it can be seen that under these assumptions for the parameters the performance of the KSD and LTS algorithms is close to that of the SE-SD and QR-TS algorithms.

In Fig. 7(b), the computational complexities of the algorithms is compared under the constraint that the compared algorithms must achieve approximately the same BER performance, as shown in Fig. 7(a). From Fig. 7(b), the following observations can be made:

- The SE-SD and KSD receivers have relatively low complexities for small values of N_t, which implies that they are more suitable for small MIMO systems. However, for $N_t \geq 16$, QR-TS and its ordered versions require significantly lower complexities compared with those required by the SE-SD and KSD schemes.
- The complexity of LTS is comparable to that of QR-TS for small values of N_t. However, because joint TS detection can be required in each layer detection in LTS, its computational burden becomes substantially higher in large MIMO systems than those of the conventional TS and QR-TS algorithms. This is in good agreement with the results presented in [23].
- Among the compared schemes, the proposed QR-TS algorithms require lower complexities than the other schemes for large values of N_t. In particular, the QR-TS algorithm with double-ordering exhibits the lowest complexity.

V. Conclusion

In this paper, we introduced a low-complexity symbol detection algorithm called QR-TS for massive MIMO systems. The QR-TS algorithm is capable of significantly reducing the complexity of the TS algorithm using low-complexity metric computation and early rejection schemes, which are based on the QR decomposition of the channel matrix. The simulation and analytical results show that QR-TS reduces the complexity of the TS algorithm approximately by half without any performance loss. To further optimize the QR-TS algorithm, we exploit the ordering schemes, namely, Tx-ordering and Rx-ordering. The Tx-ordering scheme rearranges the channel columns and neighbors based on channel column norms, which helps to reduce the threshold γ for early rejection more quickly. In the computation of a neighbor metric, the Rx-ordering scheme considers layers with larger average metrics sooner, allowing the cumulative metric to reach the threshold γ more rapidly. The numerical results show that the QR-TS in combination with the two ordering schemes require only approximately one-fourth the complexity of the conventional TS. It is also shown that the proposed algorithms achieve substantial complexity reduction for both lower-and higher-order modulation schemes, and the complexity reduction with respect to the SD decoders, such as SE-SD and KSD, becomes more significant in larger MIMO systems. We note that the proposed QR-TS scheme and existing TS-based algorithms such as LTS [23], R3TS [26], and TS with early termination [27], [28] are not mutually exclusive. In other words, QR-TS can be employed for these algorithms to perform the local search process in a more computationally efficient manner.

Appendix A
Proof of Lemma 1

The expectations of layer metrics, i.e., $\mathbb{E}\{|z_n + r_{n,d}\delta_d|^2\}$, can be expressed as

$$\mathbb{E}\left\{|z_n + r_{n,d}\delta_d|^2\right\} = \mathbb{E}\left\{|z_n|^2 + |\delta_d|^2 |r_{n,d}|^2 + 2z_n r_{n,d}\delta_d\right\}. \tag{26}$$

For each modulation scheme, $|\delta_d|^2$ is a constant. For example, in QPSK/16-QAM, $\delta_d = \pm 2$, and hence $|\delta_d|^2 = 4$. Therefore, we drop the index d in $|\delta_d|^2$ for simplicity, which yields

$$\mathbb{E}\left\{|z_n + r_{n,d}\delta_d|^2\right\} = \mathbb{E}\left\{|z_n|^2\right\} + |r_{n,d}|^2 \delta^2 + 2r_{n,d}\mathbb{E}\left\{z_n\delta_d\right\}. \tag{27}$$

We first evaluate $\mathbb{E}\{|z_n|^2\}$. By denoting the nth rows of $\boldsymbol{Q}^T$ and $\boldsymbol{R}$ as $\boldsymbol{q}_n^T$ and $\boldsymbol{p}_n^T$, respectively, we obtain

$$\begin{aligned} z_n &= \boldsymbol{q}_n^T\boldsymbol{y} - \boldsymbol{p}_n^T\boldsymbol{c} = \boldsymbol{q}_n^T(\boldsymbol{H}\boldsymbol{s} + \boldsymbol{v}) - \boldsymbol{p}_n^T\boldsymbol{c} \\ &= \boldsymbol{p}_n^T(\boldsymbol{s} - \boldsymbol{c}) + \boldsymbol{q}_n^T\boldsymbol{v}, \end{aligned} \tag{28}$$

which leads to

$$\begin{aligned} |z_n|^2 &= \left|\boldsymbol{p}_n^T(\boldsymbol{s} - \boldsymbol{c}) + \boldsymbol{q}_n^T\boldsymbol{v}\right|^2 \\ &= \left|\boldsymbol{p}_n^T(\boldsymbol{s} - \boldsymbol{c})\right|^2 + \left|\boldsymbol{q}_n^T\boldsymbol{v}\right|^2 + 2\boldsymbol{p}_n^T(\boldsymbol{s} - \boldsymbol{c})\,\boldsymbol{q}_n^T\boldsymbol{v}. \end{aligned} \tag{29}$$

Because $\boldsymbol{v}$ is a zero-mean random vector and is independent of $\boldsymbol{s}$ and $\boldsymbol{c}$, the last term in (29) has zero mean, i.e., $\mathbb{E}\{\boldsymbol{p}_n^T(\boldsymbol{s} - \boldsymbol{c})\boldsymbol{q}_n^T\boldsymbol{v}\} = 0$. Hence, we have

$$\begin{aligned} \mathbb{E}\left\{|z_n|^2\right\} &= \mathbb{E}\left\{\left|\boldsymbol{p}_n^T(\boldsymbol{s} - \boldsymbol{c})\right|^2\right\} + \mathbb{E}\left\{\left|\boldsymbol{q}_n^T\boldsymbol{v}\right|^2\right\} \\ &= \mathbb{E}\left\{\left|\sum_{i=1}^{N} r_{n,i}(s_i - c_i)\right|^2\right\} + \mathbb{E}\left\{\left|\sum_{i=1}^{N} q_{n,i}v_i\right|^2\right\} \\ &= \mathbb{E}\left\{\sum_{i=1}^{N} |r_{n,i}(s_i - c_i)|^2\right\} \\ &\quad + 2\underbrace{\mathbb{E}\left\{\sum_{i=1}^{N}\sum_{j>i}^{N} r_{n,i}(s_i - c_i)r_{n,j}(s_j - c_j)\right\}}_{=0} \end{aligned} \tag{30}$$

$$\begin{aligned} &+ \mathbb{E}\left\{\sum_{i=1}^{N} |q_{n,i}|^2 |v_i|^2\right\} \\ &+ 2\underbrace{\mathbb{E}\left\{\sum_{i=1}^{N}\sum_{j>i}^{N} q_{n,i}v_i q_{n,j}v_j\right\}}_{=0} \end{aligned} \tag{31}$$

$$\begin{aligned} &= \sum_{i=1}^{N} |r_{n,i}|^2 \mathbb{E}\left\{|s_i|^2\right\} + \sum_{i=1}^{N} |r_{n,i}|^2 \mathbb{E}\left\{|c_i|^2\right\} \\ &\quad - 2\sum_{i=1}^{N} |r_{n,i}|^2 \mathbb{E}\left\{s_i c_i\right\} + \sum_{i=1}^{N} |q_{n,i}|^2 \mathbb{E}\left\{|v_i|^2\right\}. \end{aligned} \tag{32}$$

In (30), we use $\mathbb{E}\{\sum_{i=1}^{N}\sum_{j>i}^{N} r_{n,i}(s_i - c_i)r_{n,j}(s_j - c_j)\} = 0$ because $s_i - c_i$ and $s_j - c_j$ are independent zero-mean random variables when $i \neq j$. Similarly, in (31), we have $\mathbb{E}\{\sum_{i=1}^{N}\sum_{j>i}^{N} q_{n,i}v_i q_{n,j}v_j\} = 0$ because the elements of $\boldsymbol{v}$ are independent and have $\mathcal{CN}(0, \sigma_v^2)$ distribution. Because s_i and c_i are drawn from the alphabet $\mathcal{A}$, we have $\mathbb{E}\{|c_i|^2\} = \mathbb{E}\{|s_i|^2\} = \sigma_t^2$, where σ_t^2 is the average symbol power.

By letting $\varepsilon = \mathbb{E}\{s_i c_i\}, i = 1, 2, \ldots, N$, (32) can be rewritten as

$$\mathbb{E}\left\{|z_n|^2\right\} = 2\left(\sigma_t^2 - \varepsilon\right)\sum_{i=1}^{N}|r_{n,i}|^2 + \sigma_v^2\sum_{i=1}^{N}|q_{n,i}|^2. \quad (33)$$

Because ε is independent of the index i, we drop the index i in both s_i and c_i for simplicity. Then, we have s and c, whose outcomes belong to the alphabet $\mathcal{A}$. Because $\mathcal{A}$ consists of Q points, there are a total of Q^2 combinations of $\{s, c\}$, which are $\{s_{(n)}, c_{(m)}\}, n, m = 1, 2, \ldots, Q$, where $s_{(n)}$ and $c_{(m)}$ are the nth and mth elements of $\mathcal{A}$, respectively. If $n = m$, we have $s_{(n)} = c_{(m)}$; otherwise, $s_{(n)} \neq c_{(m)}$. There are Q combinations of $\{s_{(n)}, c_{(n)}\}$, which have equal probabilities of $\frac{P_0}{Q}$, where $P_0 = \mathbb{P}\{s = c\}$. Therefore, ε can be expressed as

$$\begin{aligned}\varepsilon = \mathbb{E}\{sc\} &= \sum_{n=1}^{Q}\sum_{m=1}^{Q} s_{(n)}c_{(m)}\mathbb{P}\left\{s_{(n)}, c_{(m)}\right\} \\ &= \frac{P_0}{Q}\sum_{n=1}^{Q} s_{(n)}^2 + \sum_{n=1}^{Q}\sum_{m\neq n}^{Q} s_{(n)}c_{(m)}\mathbb{P}\left\{s_{(n)}, c_{(m)}\right\} \\ &= \sigma_t^2 P_0 + \sum_{n=1}^{Q}\sum_{m\neq n}^{Q} s_{(n)}c_{(m)}\mathbb{P}\left\{s_{(n)}, c_{(m)}\right\}. \quad (34)\end{aligned}$$

In the TS algorithm, the searching process starts from an initial solution that is usually a linear solution such as ZF or MMSE, and it is close to the true solution at high SNRs. Furthermore, as the iteration proceeds, it is likely that the TS solution gets closer to $\boldsymbol{s}$. Therefore, at high SNRs, we have $P_0 \gg \mathbb{P}\{s_{(n)}, c_{(m)}\}$, with $n \neq m$, i.e., $P_0 \approx 1, \mathbb{P}\{s_{(n)}, c_{(m)}\} \approx 0$. We note that large MIMO typically works in the high-SNR region. Therefore, we take the approximation of $\mathbb{P}\{s_{(n)}, c_{(m)}\} \approx 0, n \neq m$, and from (34), we have $\varepsilon \approx \sigma_t^2$.

Therefore, from (33), $\mathbb{E}\{|z_n|^2\}$ can be approximated as

$$\mathbb{E}\left\{|z_n|^2\right\} \approx \sigma_v^2\sum_{i=1}^{N}|q_{n,i}|^2.$$

Because $\boldsymbol{Q}^T$ is a unitary matrix, its rows have the unit squared-norm, i.e., $\|\boldsymbol{q}_n\|^2 = \sum_{i=1}^{N}|q_{n,i}|^2 = 1$. Consequently, we obtain

$$\mathbb{E}\left\{|z_n|^2\right\} = \sigma_v^2. \quad (35)$$

Now, we consider the second expectation in (27). From (28), we get

$$\begin{aligned}\mathbb{E}\left\{z_n\delta_d\right\} &= \mathbb{E}\left\{\left[\boldsymbol{p}_n^T(\boldsymbol{s}-\boldsymbol{c}) + \boldsymbol{q}_n^T\boldsymbol{v}\right]\delta_d\right\} \\ &= \mathbb{E}\left\{\boldsymbol{p}_n^T\boldsymbol{s}\delta_d\right\} - \mathbb{E}\left\{\boldsymbol{p}_n^T\boldsymbol{c}\delta_d\right\} + \mathbb{E}\left\{\boldsymbol{q}_n^T\boldsymbol{v}\delta_d\right\} \\ &= \sum_{i=1}^{N} r_{n,i}\left(\mathbb{E}\left\{s_i\delta_d\right\} - \mathbb{E}\left\{c_i\delta_d\right\}\right) + q_{n,i}\mathbb{E}\left\{v_i\delta_d\right\} \\ &= \sum_{i=1}^{N} r_{n,i}\left(\mathbb{E}\left\{s_i(c_d - x_d)\right\} - \mathbb{E}\left\{c_i(c_d - x_d)\right\}\right) \\ &\quad + q_{n,i}\mathbb{E}\left\{v_i\delta_d\right\} \\ &= \sum_{i=1}^{N} r_{n,i}\left(\mathbb{E}\left\{(s_i - c_i)(c_d - x_d)\right\}\right) + q_{n,i}\mathbb{E}\left\{v_i\delta_d\right\}. \quad (36)\end{aligned}$$

In (36), $\mathbb{E}\{v_i\delta_d\} = 0$ because $\mathbb{E}\{v_i\} = 0$ and v_i is independent of δ_d. We note that s_i, c_i, and x_d are discrete random variables with zero means, i.e., $\mathbb{E}\{s_i\} = \mathbb{E}\{c_i\} = \mathbb{E}\{x_d\} = 0$. For $i \neq d$, s_i and c_i are independent of x_d and c_d, which leads to $\mathbb{E}\{s_i c_d\} = \mathbb{E}\{s_i x_d\} = \mathbb{E}\{c_i c_d\} = \mathbb{E}\{c_i x_d\} = 0$. Consequently, (36) is reduced to

$$\mathbb{E}\left\{z_n\delta_d\right\} = r_{n,d}\mathbb{E}\left\{(s_d - c_d)(c_d - x_d)\right\}. \quad (37)$$

Because $\mathbb{P}\{s_d = c_d\} = P_0 \approx 1$ at high SNRs, from (37), we can write

$$\mathbb{E}\left\{z_n\delta_d\right\} \approx 0. \quad (38)$$

From (27), (35), and (38), we obtain (14) in Lemma 1.

Appendix B
Proof of Lemma 2

The average metric of a neighbor can be expressed as

$$\mathbb{E}\left\{\phi(\boldsymbol{x})\right\} = \sum_{n=1}^{d}\mathbb{E}\left\{\left|z_n + r_{n,d}\delta_d\right|^2\right\} + \sum_{n=d+1}^{N}\mathbb{E}\left\{|z_n|^2\right\}. \quad (39)$$

By inserting (35) and (14) into (39), we obtain

$$\begin{aligned}\mathbb{E}\left\{\phi(\boldsymbol{x})\right\} &= \sum_{n=1}^{d}\left(\sigma_v^2 + |r_{n,d}|^2\delta^2\right) + \sum_{n=d+1}^{N}\sigma_v^2 \\ &= N\sigma_v^2 + \delta^2\sum_{n=1}^{d}|r_{n,d}|^2 \\ &= N\sigma_v^2 + \delta^2\|\boldsymbol{r}_d\|^2.\end{aligned}$$

By exploiting the property $\|\boldsymbol{h}_d\| = \|\boldsymbol{r}_d\|$, we obtain (15) in Lemma 2.

References

[1] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, Apr. 2013.

[2] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[3] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.

[4] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[5] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.

[6] J. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Trans. Commun.*, vol. COM: 32, no. 2, pp. 169–176, Feb. 1984.

[7] L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2131–2142, Jun. 2008.
[8] S. Han and C. Tellambura, "A complexity-efficient sphere decoder for MIMO systems," in *Proc. IEEE Int. Conf. Commun.*, 2011, pp. 1–5.
[9] F. Rusek *et al.*, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.
[10] A. Chockalingam and B. S. Rajan, *Large MIMO Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[11] M. Mandloi and V. Bhatia, "Low-complexity near-optimal iterative sequential detection for uplink massive MIMO systems," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 568–571, Mar. 2017.
[12] K. V. Vardhan, S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "A low-complexity detector for large MIMO systems and multicarrier CDMA systems," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 3, pp. 473–485, Apr. 2008.
[13] S. K. Mohammed, A. Zaki, A. Chockalingam, and B. S. Rajan, "High-rate space–time coded large-MIMO systems: Low-complexity detection and channel estimation," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 6, pp. 958–974, Dec. 2009.
[14] X. Qin, Z. Yan, and G. He, "A near-optimal detection scheme based on joint steepest descent and Jacobi method for uplink massive MIMO systems," *IEEE Commun. Lett.*, vol. 20, no. 2, pp. 276–279, Feb. 2016.
[15] M. Mandloi and V. Bhatia, "Error recovery based low-complexity detection for uplink massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 302–305, Jun. 2017.
[16] P. Som, T. Datta, A. Chockalingam, and B. S. Rajan, "Improved large-MIMO detection based on damped belief propagation," in *Proc. IEEE Workshop Inf. Theory*, 2010, pp. 1–5.
[17] S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "Low-complexity near-map decoding of large non-orthogonal STBCS using PDA," in *Proc. IEEE Int. Symp. Inf. Theory*, 2009, pp. 1998–2002.
[18] T. Datta, N. A. Kumar, A. Chockalingam, and B. S. Rajan, "A novel Monte-Carlo-sampling-based receiver for large-scale uplink multiuser MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 62, no. 7, pp. 3019–3038, Sep. 2013.
[19] M. Hansen, B. Hassibi, A. G. Dimakis, and W. Xu, "Near-optimal detection in MIMO systems using Gibbs sampling," in *Proc. IEEE Global Telecommun. Conf.*, 2009, pp. 1–6.
[20] M. Mandloi and V. Bhatia, "Layered Gibbs sampling algorithm for near-optimal detection in large-MIMO systems," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2017, pp. 1–6.
[21] T. L. Narasimhan and A. Chockalingam, "Channel hardening-exploiting message passing (CHEMP) receiver in large-scale MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 847–860, Oct. 2014.
[22] P. Švač, F. Meyer, E. Riegler, and F. Hlawatsch, "Soft-heuristic detectors for large MIMO systems," *IEEE Trans. Signal Process.*, vol. 61, no. 18, pp. 4573–4586, Sep. 2013.
[23] N. Srinidhi, T. Datta, A. Chockalingam, and B. S. Rajan, "Layered tabu search algorithm for large-MIMO detection and a lower bound on ML performance," *IEEE Trans. Commun.*, vol. 59, no. 11, pp. 2955–2963, Nov. 2011.
[24] N. Srinidhi, S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "Low-complexity near-ML decoding of large non-orthogonal STBCs using reactive tabu search," in *Proc. IEEE Int. Symp. Inf. Theory.*, 2009, pp. 1993–1997.
[25] N. Srinidhi, S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "Near-ML signal detection in large-dimension linear vector channels using reactive tabu search," 2009, arXiv:0911.4640.
[26] T. Datta, N. Srinidhi, A. Chockalingam, and B. S. Rajan, "Random-restart reactive tabu search algorithm for detection in large-MIMO systems," *IEEE Commun. Lett.*, vol. 14, no. 12, pp. 1107–1109, Dec. 2010.
[27] H. Zhao, H. Long, and W. Wang, "Tabu search detection for MIMO systems," in *Proc. IEEE 18th Int. Symp. Pers., Indoor Mobile Radio Commun.*, 2007, pp. 1–5.
[28] K. S. Gyamfi, J. Baek, and K. Lee, "Lattice reduction aided tabu search with channel-dependent stopping criterion for MIMO systems," *Electron. Lett.*, vol. 51, no. 24, pp. 2062–2064, 2015.
[29] R. W. Farebrother, *Linear Least Squares Computations*. New York, NY, USA: Marcel Dekker, 1988.
[30] K. J. Kim, J. Yue, R. A. Iltis, and J. D. Gibson, "A QRD-M/Kalman filter-based detection and channel estimation algorithm for MIMO-OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 710–721, Mar. 2005.
[31] X. Dai, R. Zou, J. An, X. Li, S. Sun, and Y. Wang, "Reducing the complexity of quasi-maximum-likelihood detectors through companding for coded MIMO systems," *IEEE Trans. Veh. Tech.*, vol. 61, no. 3, pp. 1109–1123, Mar. 2012.
[32] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2012, vol. 3.
[33] R. Hunger, *Floating Point Operations in Matrix-Vector Calculus*. Munich, Germany: Inst. Circuit Theory Signal Process. Munich, Munich Univ. Technol., 2005.
[34] R. Sedgewick, "Implementing quicksort programs," *Commun. ACM*, vol. 21, no. 10, pp. 847–857, 1978.
[35] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2nd ed. Noida, India: Pearson Edu. India, 1994.
[36] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

Nhan Thanh Nguyen received the B.S. degree in electronics and communications engineering from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2014, and the M.S. degree in electrical and information engineering from the Seoul National University of Science and Technology (SeoulTech), Seoul, South Korea, in 2017. Since September 2017, he has been working toward the Ph.D. degree in electrical and information engineering with SeoulTech. His research interests include signal processing and optimization for wireless communication with a focus on massive MIMO systems.

Kyungchun Lee received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2000, 2002, and 2007, respectively. From April 2007 to June 2008, he was a Postdoctoral Researcher with the University of Southampton, U.K. From July 2008 to August 2010, he was with Samsung Electronics, Suwon, South Korea. Since September 2010, he has been with the Seoul National University of Science and Technology, Seoul, South Korea. In 2017, he was a Visiting Assistant Professor with NC State University, Raleigh. His research interests include wireless communications, signal processing, and applied machine learning. He was a recipient of the Best Paper Award at the IEEE International Conference on Communications in 2009.

Huaiyu Dai (F'17) received the B.E. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 2002. He was with Bell Labs, Lucent Technologies, Holmdel, NJ, USA, in summer 2000, and with AT&T Labs-Research, Middletown, NJ, USA, in summer 2001. He is currently a Professor in electrical and computer engineering with NC State University, Raleigh, holding the title of University Faculty Scholar. His research interests are in the general areas of communication systems and networks, advanced signal processing for digital communications, and communication theory and information theory. His current research focuses on networked information processing and cross-layer design in wireless networks, cognitive radio networks, network security, and associated information-theoretic and computation-theoretic analysis. He has served as an Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He is currently an Area Editor in charge of wireless communications for IEEE TRANSACTIONS ON COMMUNICATIONS. He cochaired the Signal Processing for Communications Symposium of IEEE Globecom 2013, the Communications Theory Symposium of IEEE ICC 2014, and the Wireless Communications Symposium of IEEE Globecom 2014. He was a co-recipient of best paper awards at 2010 IEEE International Conference on Mobile Ad-hoc and Sensor Systems, 2016 IEEE INFOCOM BIGSECURITY Workshop, and 2017 IEEE International Conference on Communications.