AN EFFICIENT SPARSE QUADRATIC PROGRAMMING
RELAXATION BASED ALGORITHM FOR LARGE-SCALE MIMO
DETECTION∗

PING-FAN ZHAO†, QING-NA LI‡, WEI-KUN CHEN‡, AND YA-FENG LIU§

Abstract. Multiple-input multiple-output (MIMO) detection is a fundamental problem in wireless communications and it is strongly NP-hard in general. Massive MIMO has been recognized as a key technology in the fifth generation (5G) and beyond communication networks, which on one hand can significantly improve the communication performance, and on the other hand poses new challenges of solving the corresponding optimization problems due to the large problem size. While various efficient algorithms such as semidefinite relaxation (SDR) based approaches have been proposed for solving the small-scale MIMO detection problem, they are not suitable to solve the large-scale MIMO detection problem due to their high computational complexities. In this paper, we propose an efficient sparse quadratic programming (SQP) relaxation based algorithm for solving the large-scale MIMO detection problem. In particular, we first reformulate the MIMO detection problem as an SQP problem. By dropping the sparse constraint, the resulting relaxation problem shares the same global minimizer with the SQP problem. In sharp contrast to the SDRs for the MIMO detection problem, our relaxation does not contain any (positive semidefinite) matrix variable and the numbers of variables and constraints in our relaxation are significantly less than those in the SDRs, which makes it particularly suitable for the large-scale problem. Then we propose a projected Newton-based quadratic penalty method to solve the relaxation problem. By extensive numerical experiments, when applied to solve small-scale problems, the proposed algorithm is demonstrated to be competitive with the state-of-the-art approaches in terms of detection accuracy and solution efficiency; when applied to solve large-scale problems, the proposed algorithm achieves better detection performance and is more robust to the choice of the initial point than a recently proposed generalized power method.

Key words. Hypergraph Matching, MIMO Detection, Projected Newton Method, Quadratic Penalty Method, Semidefinite Relaxation, Sparse Quadratic Programming Relaxation

AMS subject classifications. 90C22, 90C20, 90C27

1. Introduction. Multiple-input multiple-output (MIMO) detection is a fundamental problem in modern communications [1, 33]. The input-output relationship of the MIMO channel is

\[ r = Hx^* + v, \]

where \( r \in \mathbb{C}^m \) denotes the vector of received signals, \( H \in \mathbb{C}^{m \times n} \) denotes an \( m \times n \) complex channel matrix (usually \( m \geq n \)), \( x^* \in \mathbb{C}^n \) denotes the vector of transmitted signals, and \( v \in \mathbb{C}^m \) denotes an additive white circularly symmetric Gaussian noise. The goal of MIMO detection is to recover the transmitted signals \( x^* \) from the received signals \( r \) based on the channel information \( H \). We refer to [8, 33] for a review of different formulations and approaches for MIMO detection and [1] for the latest progress in MIMO detection.

∗Submitted to the editors June 23, 2020.
†School of Mathematics and Statistics, Beijing Institute of Technology, Beijing (pzhaoo@bit.edu.cn).
‡Qing-Na Li is the corresponding author. School of Mathematics and Statistics/Beijing Key Laboratory on MCAAI, Beijing Institute of Technology, Beijing (qnn@bit.edu.cn, chenweikun@bit.edu.cn).
§State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing (yafliu@lsec.cc.ac.cn).
In this paper, we assume that $x^*$ in (1.1) is modulated via the $M$-Phase-Shift Keying ($M$-PSK) modulation scheme with $M \geq 2$. More exactly, each entry $x^*_j$ of $x^*$ belongs to a finite set:

$$x^*_j \in \mathcal{X} \triangleq \left\{ e^{i\theta} \left| \theta = \frac{2(k - 1)\pi}{M}, \ k = 1, \ldots, M \right. \right\}, \ j = 1, \ldots, n,$$

where $i$ is the imaginary unit and $e$ is the natural base number. The mathematical formulation for the MIMO detection problem is

$$\begin{align*}
\min_{x \in \mathbb{C}^n} & \quad F(x) \triangleq \|Hx - r\|^2_2 \\
\text{s.t.} & \quad |x_j|^2 = 1, \ j = 1, \ldots, n, \\
& \quad \text{arg}(x_j) \in \mathcal{A} \triangleq \left\{ 0, \frac{2\pi}{M}, \ldots, \frac{2(M-1)\pi}{M} \right\}, \ j = 1, \ldots, n,
\end{align*}$$

(1.3)

where $\| \cdot \|_2$ denotes the Euclidean norm and $\text{arg}(\cdot)$ denotes the argument of the complex number.

Let

$$Q = H^\dagger H \quad \text{and} \quad c = -H^\dagger r.$$

Then problem (P) is equivalent to the following problem

$$\begin{align*}
\min_{x \in \mathbb{C}^n} & \quad x^\dagger Qx + 2\text{Re}(c^\dagger x) \\
\text{s.t.} & \quad |x_j|^2 = 1, \ j = 1, \ldots, n, \\
& \quad \text{arg}(x_j) \in \mathcal{A}, \ j = 1, \ldots, n,
\end{align*}$$

(1.3)

where $(\cdot)^\dagger$ denotes the conjugate transpose and $\text{Re}(\cdot)$ denotes the real part of the complex number.

Various methods to tackle the MIMO detection problem can be summarized into several lines [33, Figure 15], including tree search [7, 23, 31], lattice reduction (LR) [11, 37], and semidefinite relaxation (SDR) [17, 27, 30]. The tree search based methods are the most popular detectors in the era of multi-antenna MIMO systems [33]. Taking the typical tree search based method, the sphere decoder (SD) algorithm [7], as an example, it is regarded as the benchmark for globally solving the MIMO detection problem. However, both the expected and worst-case complexities of the SD algorithm are exponential [9, 29]. The most popular LR algorithm is the Lenstra-Lenstra-Lovász (LLL) algorithm [11], whose worst-case computational complexity can be prohibitively high [10, 34]. Below we mainly review the SDR based approach, which is most related to this work.

The SDR based approach was first proposed for a binary PSK (BPSK) modulated code division multiple access (CDMA) system [27]. Then it was extended to the quadrature PSK (QPSK) scenario [15] and further to the high-order $M$-PSK scenario [18, 19]. In [20], a quadratic assignment problem formulation was proposed for problem (P), and a near-maximum-likelihood decoding algorithm was designed based on the resulting SDR. Other early SDR based approaches are summarized in [33, Table IX].

SDR based approaches generally perform very well for solving the MIMO detection problem. To understand the reason, various researches have been done and one
line of researches is to identify conditions under which the SDRs are tight [16, Definition 1]. For the case where $M = 2$, So [24] proposed an SDR of problem (P) and proved its tightness when the following condition

$$\lambda_{\min}(\text{Re}(H^\dagger H)) > \|\text{Re}(H^\dagger v)\|_\infty$$

is satisfied. Here $H$ and $v$ are defined in (1.1), $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue of a given matrix, and $\|\cdot\|_\infty$ denotes the $\ell_\infty$-norm. An open question proposed in [24] is that whether the (conventional) SDR is still tight under condition (1.4) for the case where $M \geq 3$. It was negatively answered in [16]. In addition, Lu et al. in [16] proposed an enhanced SDR (see (ERSDR1) further ahead) by adding some valid inequalities and showed that under condition

$$\lambda_{\min}(H^\dagger H) \sin \left(\frac{\pi}{M}\right) > \|H^\dagger v\|_\infty,$$

(ERSDR1) is tight. In [14], the relations between different SDRs were further analyzed. In particular, it was proved that (ERSDR1) and the SDR proposed in [20] are equivalent, and as a result, the SDR proposed in [20] is also tight under condition (1.5). Other representative analysis results can be found in [3, 21]. We shall talk about the relations between our relaxation and the SDRs and their tightness later in section 4.

One key advantage of the SDR based approaches, compared to SD and LLL algorithms, is that the SDR admits polynomial-time algorithms. There are well developed solvers for solving the SDR, such as SeDuMi [25], SDPT3 [28], and the latest SDPNAL+ [26, 32, 35, 36]. However, the numbers of variables and constraints in the SDRs are much larger than those of problem (CQP), and hence the SDR based approaches cannot be used to solve the large-scale MIMO detection problem. On the other hand, it was predicted that the mobile data traffic will grow exponentially in 2017-2022 [5], which calls for higher data rates, larger network capacity, higher spectral efficiency, higher energy efficiency, and better mobility [1]. Massive MIMO is a key and effective technology to meet the above requirements, where the base station (BS) is equipped with tens to hundreds of antennas, in contrast to the current BS equipped only with 4 to 8 antennas. A new challenge coming with the massive MIMO technology is the large problem size in signal processing and optimization. In particular, the MIMO detection problem of our interest in the massive MIMO setup is a large-scale strongly NP-hard problem [29]. As far as we know, there are very few works on the large-scale MIMO detection problem. One notable work is [13], which proposes a customized generalized power method (GPM) for solving the large-scale MIMO detection problem. The GPM directly solves problem (P) and at each iteration, the algorithm takes a gradient descent step with an appropriate stepsize and projects the obtained point onto the (discrete) feasible set of problem (P). However, our experiments show that the performance of the GPM heavily depends on the choice of the initial point. Consequently, models and algorithms that can be generalized to the large-scale MIMO detection problem with satisfactory detection performance are still highly in need.

Contributions. The contributions of the paper are twofold. Firstly, we propose a sparse quadratic programming (SQP) formulation for the MIMO detection problem. We prove that, somewhat surprisingly, its relaxation obtained by dropping the sparse constraint is equivalent to the original formulation. Secondly, we present a projected Newton based quadratic penalty (PN-QP) method to solve the proposed (relaxation) formulation, which is demonstrated to be quite efficient in terms of detection accuracy.
and solution efficiency. In particular, our extensive numerical results show that (i) compared to SD and SDPNAL+ (for solving (ERSDR1)), PN-QP is more effective on massive MIMO detection; (ii) compared to GPM, PN-QP achieves significantly better detection performance and is more robust to the choice of the initial point.

Two key features of our proposed approach are highlighted as follows. Firstly, in sharp contrast to the matrix based SDRs, due to the vector based formulation for the MIMO detection problem, our relaxation is particularly suitable to deal with the large-scale MIMO detection problem. Secondly, by exploring the sparse structure of the optimal solution, the computational cost of PN-QP is significantly reduced. In particular, PN-QP is designed to identify the support set of the optimal solution rather than itself, leading to a low computational cost.

The rest of this paper is organized as follows. In section 2, we introduce different formulations for the MIMO detection problem, including the SQP formulation. In section 3, we discuss the relaxation problem and present the PN-QP method. In section 4, we discuss the relations between our relaxation and the SDRs. In section 5, we perform extensive numerical experiments to compare different algorithms for solving the MIMO detection problem. Finally, we conclude the paper in section 6.

We adopt the following standard notations in this paper. Let $i$ denote the imaginary unit (satisfying $i^2 = -1$). For a given complex vector $x$, we use $x_j$ to denote its $j$-th entry, and $|x_j|$ to denote the modulus of its $j$-th entry. Let $\| \cdot \|_2$ denote the $\ell_2$-norm for vectors and Frobenius norm for matrices. We also use $\|x\|_0$ to denote its zero norm, i.e., the number of non-zero entries of the vector. Let $\text{diag}(X)$ denote the vector formed by the diagonal elements in matrix $X$, and $\text{Diag}(x)$ denote the diagonal matrix with the diagonal entries being vector $x$. For a complex matrix $C$, let $\text{Re}(C)$ and $\text{Im}(C)$ denote the real and imaginary parts of $C$, respectively, and $C^\dagger$ and $C^\top$ denote the conjugate transpose and transpose of $C$, respectively. $C \succeq 0$ means $C$ is positive semidefinite, and $\text{Tr}(C)$ denotes the trace of $C$. For two Hermitian matrices $A$ and $B$, $A \bullet B$ means $\text{Re}(\text{Tr}(A^\dagger B))$. Let $e$ be a vector of an appropriate length with all elements being one. For a sequence $\{x^k\}$, $x^k \uparrow c$ and $x^k \downarrow c$ mean that $x^k$ tends to increasingly and decreasingly to a certain value $c$, respectively. We use $\otimes$ to denote the Kronecker product. For $t \in \mathbb{R}^{nM}$, we assume that $t$ has the partition as $t = (\bar{t}_1^T, \ldots, \bar{t}_n^T)^T$, where $\bar{t}_j \in \mathbb{R}^{M}$ is the $j$-th block of $t$. Finally, the $k$-th entry in block $\bar{t}_j$ is denoted as $(\bar{t}_j)_k$.

2. Different Formulations for MIMO Detection. In this section, we introduce some formulations for the MIMO detection problem and discuss their properties.

Define

$$
\mathcal{Y} = \left\{ (\cos \theta_k, \sin \theta_k) \mid \theta_k = \frac{2(k-1)\pi}{M}, \ k = 1, \ldots, M \right\}.
$$

Then, for each $j = 1, \ldots, n$, it is easy to see that $x_j \in \mathcal{X}$ (defined in (1.2)) if and only if $(\text{Re}(x_j), \text{Im}(x_j)) \in \mathcal{Y}$. The feasible points of $\mathcal{Y}$ for $M = 4$ and $M = 8$ are illustrated in Figure 1.

Let

$$
\hat{Q} = \begin{bmatrix} \text{Re}(Q) & -\text{Im}(Q) \\ \text{Im}(Q) & \text{Re}(Q) \end{bmatrix} = (\bar{q}_{jk})_{2n \times 2n}, \quad \hat{c} = \begin{bmatrix} \text{Re}(c) \\ \text{Im}(c) \end{bmatrix}, \quad \text{and} \quad y = \begin{bmatrix} \text{Re}(x) \\ \text{Im}(x) \end{bmatrix} \in \mathbb{R}^{2n}.
$$

Problem (CQP) can be equivalently written as the following real form:

$$
\min_{y \in \mathbb{R}^{2n}} \ y^\top \hat{Q} y + 2\hat{c}^\top y
\quad \text{s.t.} \quad (y_j, y_{n+j}) \in \mathcal{Y}, \ j = 1, \ldots, n.
$$

This manuscript is for review purposes only.
Let \( t = (\bar{t}_1^\top, \ldots, \bar{t}_n^\top) \in \mathbb{R}^{nM} \) where \( \bar{t}_j \in \mathbb{R}^M \) is the assignment variable corresponding to \((y_j, y_{n+j})\), i.e.,

\[
(\bar{t}_j)_k = \begin{cases} 
1, & \text{if } (y_j, y_{n+j}) = (\cos \theta_k, \sin \theta_k); \\
0, & \text{otherwise}.
\end{cases}
\]

By definition, the constraints in problem (RQP) can be equivalently written as

\[
(2.3) \quad \begin{bmatrix} y_j \\ y_{n+j} \end{bmatrix} = \sum_{k=1}^{M} (\bar{t}_j)_k \begin{bmatrix} \cos \theta_k \\ \sin \theta_k \end{bmatrix}, \quad \bar{t}_j \in \{0, 1\}^M, \quad e^\top \bar{t}_j = 1, \quad j = 1, \ldots, n.
\]

Then problem (RQP) can be equivalently written as

\[
(2.4) \quad \min_{y \in \mathbb{R}^{2n}, t \in \{0, 1\}^{nM}} y^\top \hat{Q} y + 2\hat{c}^\top y \\
\text{s.t.} \quad y_j = \alpha^\top \bar{t}_j, \quad j = 1, \ldots, n,
\]

\[
y_{n+j} = \beta^\top \bar{t}_j, \quad j = 1, \ldots, n,
\]

\[
e^\top \bar{t}_j = 1, \quad j = 1, \ldots, n,
\]

\[
t \in \{0, 1\}^{nM},
\]

where

\[
(2.5) \quad \alpha = \left( \cos 0, \cos \left(\frac{2\pi}{M}\right), \ldots, \cos \left(\frac{2(M-1)\pi}{M}\right) \right)^\top \quad \text{and}
\]

\[
\beta = \left( \sin 0, \sin \left(\frac{2\pi}{M}\right), \ldots, \sin \left(\frac{2(M-1)\pi}{M}\right) \right)^\top.
\]

We now eliminate the variables \( y_j \) for \( j = 1, \ldots, 2n \), based on the constraints in problem (2.4). Let

\[
(2.6) \quad A = I \otimes \alpha^\top \in \mathbb{R}^{n \times nM}, \quad B = I \otimes \beta^\top \in \mathbb{R}^{n \times nM}, \quad \text{and} \quad P = \begin{bmatrix} A \\ B \end{bmatrix} \in \mathbb{R}^{2n \times nM}.
\]
We obtain the following quadratic assignment problem:

\[
\begin{align*}
\min_{t \in \mathbb{R}^{nM}} \quad & h(t) \triangleq t^\top G t + 2w^\top t \\
\text{s.t.} \quad & e^\top \bar{t}_j = 1, \quad j = 1, \ldots, n, \\
& t \in \{0, 1\}^{nM},
\end{align*}
\]

where

\[
G = P^\top \hat{Q} P \in \mathbb{R}^{nM \times nM} \quad \text{and} \quad w = P^\top \hat{c} \in \mathbb{R}^{nM}.
\]

Inspired by the sparse formulation in \[6, (2.10)\], we define the following SQP problem:

\[
\begin{align*}
\min_{t \in \mathbb{R}^{nM}} \quad & h(t) \\
\text{s.t.} \quad & e^\top \bar{t}_j = 1, \quad j = 1, \ldots, n, \\
& t \geq 0, \\
& \|t\|_0 \leq n,
\end{align*}
\]

where \(\|t\|_0 \leq n\) is the sparse constraint. We have the following result addressing the connection between problems (QAP) and (SQP1).

**Proposition 2.1.** Problems (QAP) and (SQP1) are equivalent.

**Proof.** For each feasible point \(t\) of problem (QAP), there is \(\|\bar{t}_j\|_0 = 1\), implying that \(\|t\|_0 \leq n\). Consequently, point \(t\) is feasible for problem (SQP1). On the other hand, for each feasible point \(t\) of problem (SQP1), it follows \(e^\top \bar{t}_j = 1\) for \(j = 1, \ldots, n\), and \(\|t\|_0 \leq n\), implying that \(\|\bar{t}_j\|_0 = 1\) for \(j = 1, \ldots, n\). Therefore, each entry in \(\bar{t}_j\) must be either zero or one, i.e., \(t \in \{0, 1\}^{nM}\). This shows that point \(t\) is also feasible for problem (QAP). Therefore, problems (QAP) and (SQP1) are equivalent. \(\square\)

By Proposition 2.1, problem (SQP1) is equivalent to the original problem (P). Specifically, if \(x^*\) is a global minimizer of problem (P), then \(t^*\) obtained by the following

\[
\begin{bmatrix}
\Re(x^*) \\
\Im(x^*)
\end{bmatrix} = Pt^* = \begin{bmatrix} A \\ B \end{bmatrix} t^* = \begin{bmatrix} At^* \\ Bt^* \end{bmatrix}
\]

is a global minimizer of problem (SQP1). Conversely, for a global minimizer \(t^*\) of problem (SQP1), one can get a global minimizer \(x^*\) of problem (P) by

\[
x^* = At^* + iBt^*.
\]

This reveals that there is a one-to-one correspondence between the global minimizers of problem (SQP1) and those of problem (P).

Next, we partition the matrix \(G\) (defined in (2.7)) as follows:

\[
G = \begin{bmatrix}
S_{11} & S_{12} & \cdots & S_{1n} \\
S_{21} & S_{22} & \cdots & S_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
S_{n1} & S_{n2} & \cdots & S_{nn}
\end{bmatrix},
\]
where $S_{jk} \in \mathbb{R}^{M \times M}$ for $j, k = 1, \ldots, n$. Define a new matrix $\tilde{G} \in \mathbb{R}^{nM \times nM}$ obtained by removing the diagonal blocks in $G$, i.e.,

\begin{equation}
\tilde{G} = \begin{bmatrix}
0 & S_{12} & \cdots & S_{1n} \\
S_{21} & 0 & \cdots & S_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
S_{n1} & S_{n2} & \cdots & 0
\end{bmatrix}.
\end{equation}

We have the following result.

**Theorem 2.2.** Problem (SQP1) is equivalent to the following problem:

\begin{equation}
\min_{t \in \mathbb{R}^{nM}} f(t) \triangleq t^\top \tilde{G} t + 2w^\top t \\
\text{s.t.} \quad e^\top \tilde{t}_j = 1, \; j = 1, \ldots, n, \\
\quad t \succeq 0, \\
\quad \|t\|_0 \leq n.
\end{equation}

**Proof.** The proof is relegated to Appendix A.

Below, we give a property of the objective function $f(t)$ in problem (SQP2) stating that $f(t)$ is a linear function with respect to $\tilde{t}_j$, which follows from the fact that the diagonal block in $\tilde{G}$ is zero. Such a property is similar to that in [6, Proposition 3] for hypergraph matching.

**Proposition 2.3.** For each block $\tilde{t}_j$, $j = 1, \ldots, n$, $f(t)$ in problem (SQP2) is a linear function of $\tilde{t}_j$, i.e., $\nabla_{\tilde{t}_j} f(t)$ is independent of $\tilde{t}_j$.

3. Algorithms for MIMO Detection. In this section, we first show the equivalence between problem (SQP2) and its relaxation problem obtained by dropping the sparse constraint. Then we present the PN-QP method to solve the relaxation problem.

3.1. Relaxation of Problem (SQP2). By dropping the sparse constraint in problem (SQP2), i.e., $\|t\|_0 \leq n$, we get the following relaxation problem:

\begin{equation}
\min_{t \in \mathbb{R}^{nM}} f(t) \\
\text{s.t.} \quad e^\top \tilde{t}_j = 1, \; j = 1, \ldots, n, \\
\quad t \succeq 0.
\end{equation}

Problem (RSQP) is actually equivalent to problem (SQP2). To prove it, we need the following result.

**Lemma 3.1** (Corollary 2 in [6]). Consider

\begin{equation}
\min_{t \in \mathbb{R}^{nM}} \hat{f}(t) \\
\text{s.t.} \quad e^\top \tilde{t}_j = 1, \; j = 1, \ldots, n, \\
\quad t \succeq 0,
\end{equation}

where $\hat{f}(t)$ is a linear function of $\tilde{t}_j$, $j = 1, \ldots, n$. Then there exists a global minimizer $t^*$ of problem (3.1) such that $\|t^*\|_0 = n$. 

This manuscript is for review purposes only.
Since, by Proposition 2.3, \( f(t) \) satisfies the condition in Lemma 3.1, we immediately have the following result.

**Theorem 3.2.** There exists a global minimizer \( t^* \) of problem (RSQP) such that \( \|t^*\|_0 = n \). Furthermore, such \( t^* \) is a global minimizer of problem (SQP2).

**Remark 3.3.** Suppose that one gets a global minimizer of problem (RSQP), denoted as \( t^* \in \mathbb{R}^{nM} \). As pointed out in [6, Remark 3], we can get a global minimizer \( t^\bigcirc \) of problem (SQP2) in the following way. For each block \( \bar{t}^\bigcirc_j \), pick up any nonzero entry in \( \bar{t}^\bigcirc_j \), say \( p_j \), and set

\[
(\bar{t}^\bigcirc_j)_{p_j} = 1, \quad (\bar{t}^\bigcirc_j)_l = 0, \quad l \in \{1, \ldots, M\} \setminus p_j.
\]

Repeatedly applying the above procedure, we will obtain a global minimizer of problem (SQP2).

From the strong NP-hardness of problem (P) and the equivalence between problems (SQP2) and (RSQP) (cf. Theorem 3.2), problem (RSQP) is also strongly NP-hard. However, it allows much flexibility to develop efficient numerical algorithms for solving the MIMO detection problem, since it is a continuous optimization problem with multiple simple simplex constraints.

An interesting question is that under what condition, each global minimizer of problem (RSQP) is also a global minimizer of problem (SQP2), i.e., problem (RSQP) does not have a global minimizer that is not optimal for problem (SQP2). We have the following result.

**Theorem 3.4.** Suppose that \( t^* \) is the unique global optimal solution of problem (SQP2). Then, \( t^* \) is the unique global minimizer of problem (RSQP).

**Proof.** We use the contradiction argument. Assume that \( t^* \) is not the unique global minimizer of problem (RSQP), then there must exist another global minimizer \( t^0 \) of problem (RSQP) such that \( t^0 \neq t^* \). This, together with the assumption that \( t^* \) is the unique global minimizer of problem (SQP2) and \( e^T \bar{t}^0_j = 1 \), implies that \( \|t^0\|_0 \geq n + 1 \) must hold, and there must exist a block \( \bar{t}^0_j \) such that \( \|\bar{t}^0_j\|_0 \geq 2 \). Without loss of generality, let \( \|\bar{t}^0_1\|_0 \geq 2 \), \( (\bar{t}^0_1)_1 > 0 \), and \( (\bar{t}^0_1)_2 > 0 \). Applying the rounding procedure in (3.2) by setting

\[
\begin{align*}
(\bar{t}^0_1)_1 &= 1; \\
(\bar{t}^0_1)_2 &= 0
\end{align*}
\quad \text{and} \quad
\begin{align*}
(\bar{t}^0_2)_1 &= 0; \\
(\bar{t}^0_2)_2 &= 1,
\end{align*}
\]

respectively, we will obtain two different global minimizers of problem (RSQP). Repeatedly applying the rounding procedure in (3.2) to other blocks of these two points, we can obtain two different global minimizers of problem (SQP2), which contradicts with the assumption that \( t^* \) is the unique global minimizer of problem (SQP2). Consequently, \( t^* \) is the unique global minimizer of problem (RSQP).

**Theorem 3.4** implies that if the vector of transmitted signals \( x^* \) is the unique global minimizer of problem (P), then the corresponding \( t^* \) obtained via (2.8) is a unique global minimizer of problem (RSQP).

We illustrate several formulations for the MIMO detection problem in Figure 2, which demonstrates the equivalence between problems (P), (CQP), (RQP), (SQP1), (SQP2), as well as (RSQP).

It should be emphasized that problem (RSQP) is a vector based formulation and its size is much smaller (than that of SDRs for problem (P)), and thus it is
more suitable to be used for designing algorithms for the large-scale problems. More detailed comparisons between problem (RSQP) and various SDRs will be shown in the next section.

3.2. Numerical Algorithm for Problem (RSQP). In this subsection, we present the numerical algorithm for solving problem (RSQP). Recall that problem (RSQP) is a nonlinear programming problem with \( n \) simplex constraints. Hence, one can use a solver for constrained optimization problems like \texttt{fmincon} in MATLAB to solve it. However, due to the special property as stated in Remark 3.3, once the support set of the global minimizer of problem (RSQP) is correctly identified, we can apply the rounding procedure in (3.2) to obtain a global minimizer of problem (SQP2). Based on such observations, instead of directly solving problem (RSQP) by treating it as a general constrained optimization problem, we prefer to design an algorithm to (quickly) identify the support set of the global minimizer of problem (RSQP). Due to this, such an algorithm does not necessarily need to strictly satisfy the equality constraints during the algorithmic procedure, i.e., it is reasonable to allow the violations of the equality constraints to some extent. Therefore, we choose the quadratic penalty method to solve problem (RSQP). More precisely, at each iteration \( k \), the quadratic penalty method solves the following subproblem:

\[
\min_{t \in \mathbb{R}^{nM}} f_{\omega_k}(t) = f(t) + \frac{\omega_k}{2} \sum_{j=1}^{n} (e^\top \bar{t}_j - 1)^2 \\
\text{s.t.} \quad t \geq 0,
\]

where \( \omega_k > 0 \) is the penalty parameter. Next, we provide more details on the stopping criteria of the quadratic penalty algorithm and the algorithm for solving the subproblem.

Let \( t^k \) be an (approximate) solution of subproblem (3.3). As for the stopping criteria, we check whether the support set of \( t^k \) is the same as that of the previous step and whether the size of the support set of \( t^k \) is equal to \( n \), i.e.,

\[
\mathcal{K}(t^k) = \mathcal{K}(t^{k-1}) \quad \text{and} \quad \|t^k\|_0 = n,
\]

where \( \mathcal{K}(t) \) is the support set of \( t \in \mathbb{R}^{nM} \) defined as

\[
\mathcal{K}(t) = \{\ell \mid t_\ell > 0, \forall \ell = 1, \ldots, nM\}.
\]

If (3.4) is satisfied, which implies that we reach a feasible point of problem (SQP2) via the rounding procedure in (3.2), we terminate the iteration.

As for subproblem (3.3), it is equivalent to

\[
\min_{t \in \mathbb{R}^{nM}} f_{\omega_k}(t) + \delta_{\mathbb{R}^{nM}_+}(t),
\]

where \( \delta_{\mathbb{R}^{nM}_+}(t) \) is the indicator function defined by 0 if \( t \in \mathbb{R}^{nM}_+ \) and \(+\infty\) otherwise. The first-order optimality condition of subproblem (3.6) is

\[
\text{dist} \left( 0, \nabla f_{\omega_k}(t^k) + \partial \delta_{\mathbb{R}^{nM}_+}(t^k) \right) = 0,
\]
where $\partial \delta_{R_{nM}}(\cdot)$ is the sub-differential of $\delta_{R_{nM}}(\cdot)$. Here we solve subproblem (3.3) inexactly to get a solution $t^k$ such that the first-order optimality condition holds approximately. That is, we solve subproblem (3.6) until $t^k$ satisfies

$$\left\| \text{dist} \left( 0, \nabla f_{\omega_k}(t^k) + \partial \delta_{R_{nM}}(t^k) \right) \right\|_2 \leq \tau_k,$$

where $\tau_k \downarrow 0$.

Note that subproblem (3.3) is a non-convex quadratic programming problem with simple lower bound constraints. As mentioned above, we prefer to identify the support set of the global minimizer of subproblem (3.3) rather than find the global minimizer itself (in order to reduce the computational cost). The strategy of identifying the active set is therefore crucial in solving subproblem (3.3). From this point of view, the active set methods are particularly suitable to solve subproblem (3.3). Therefore, we choose the typical active set method, projected Newton method proposed in [2], which is demonstrated to be highly efficient in solving large-scale problems such as calibrating least squares covariance matrices [12].

Overall, we give the details of the PN-QP method in Algorithm 3.1.

**Algorithm 3.1** PN-QP Method.

1. **Initialization:** $t^0 \in \mathbb{R}^{nM}, k := 1, \rho > 1, \text{and } \omega_k > 0$;
2. **while** $k \leq \text{maxiter}$ **do**
3. Solve subproblem (3.3) by the projected Newton method to get $t^k$ such that $t^k$ satisfies (3.8);
4. **if** condition (3.4) is satisfied **then**
5. Break;
6. **end if**
7. Update $\omega_{k+1}$ by $\omega_{k+1} = \rho \omega_k$, $k := k + 1$;
8. **end while**
9. Repeatedly using the rounding procedure in (3.2) on $t^k$ to obtain a point $t^* \in \{0, 1\}^{nM}$;
10. **return** $t^*$.

We have the following classic convergence result of the quadratic penalty method.

**Theorem 3.5.** Suppose that in Algorithm 3.1, $t^k$ satisfies (3.8), $\tau_k \downarrow 0$, and $\omega_k \uparrow +\infty$. Then any accumulation point of the sequence generated by Algorithm 3.1 is a stationary point of problem (RSQP).

**Proof.** The proof is relegated to Appendix B.

**4. Relations to the SDRs.** Recall that the enhanced SDR studied in [16] is tight under condition (1.5). Now a question is whether we can show the tightness result of the SDR of our proposed formulation (SQP2) under the same condition. The following shows that the answer is yes. Next we give the definition of tightness [16, Definition 1]:

**Definition 4.1.** An SDR of problem (P) is called tight if the following two conditions hold:
- the gap between the SDR and problem (P) is zero; and
- the SDR recovers the true vector of transmitted signals.
Firstly, let us briefly describe the enhanced SDR in [16]:

\[
\begin{align*}
\min_{y \in \mathbb{R}^{2n}, t \in \mathbb{R}^{nM}, \ Y \in \mathbb{R}^{2n \times 2n}} & \quad \bar{Q} \bullet Y + 2\bar{c}^T y \\
\text{s.t.} & \quad Y(j) = \sum_{k=1}^{M} (\bar{t}_j)_k U_k, \quad j = 1, \ldots, n, \\
& \quad \sum_{k=1}^{M} (\bar{t}_j)_k = 1, \quad j = 1, \ldots, n, \\
& \quad \begin{bmatrix} 1 & y^T \\ y & Y \end{bmatrix} \succeq 0, \\
& \quad t \geq 0,
\end{align*}
\]

(ERSDR1)

where

\[
Y(j) = \begin{bmatrix} 1 & y_j & y_{n+j} \\ y_j & Y_{jj} & Y_{j(n+j)} \\ y_{n+j} & Y_{(n+j)j} & Y_{(n+j)(n+j)} \end{bmatrix}, \quad j = 1, \ldots, n,
\]

and

\[
U_k = \begin{bmatrix} 1 & \cos \theta_k \\ \sin \theta_k \end{bmatrix} \begin{bmatrix} 1 & \cos \theta_k \\ \sin \theta_k \end{bmatrix}, \quad k = 1, \ldots, M.
\]

To see the relation between problems (ERSDR1) and (SQP2), we need the following SDR of problem (QAP), which was proposed in [20]:

\[
\begin{align*}
\min_{T \in \mathbb{R}^{nM \times nM}, \ t \in \mathbb{R}^{nM}} & \quad \bar{f}_1(T, t) \triangleq G \bullet T + 2w^T t \\
\text{s.t.} & \quad e^T \bar{t}_j = 1, \quad j = 1, \ldots, n, \\
& \quad T_{jj} = \text{Diag}(\bar{t}_j), \quad j = 1, \ldots, n, \\
& \quad T \succeq tt^T, \quad t \geq 0,
\end{align*}
\]

(ERSDR2)

where \( G \) and \( w \) are defined in (2.7), and \( T_{jj} \in \mathbb{R}^{M \times M} \) is the \( j \)-th diagonal block of \( T \).

Recall that problem (QAP) is equivalent to problem (SQP1). Problem (ERSDR2) is the relaxation of problem (SQP1). In [14, Theorem 1], the authors have established the equivalence of problems (ERSDR1) and (ERSDR2). As a result, under condition (1.5), problem (ERSDR2) is also tight [14, Theorem 2]. Now for problem (SQP2), a similar SDR can be obtained as follows:

\[
\begin{align*}
\min_{T \in \mathbb{R}^{nM \times nM}, \ t \in \mathbb{R}^{nM}} & \quad \tilde{f}_2(T, t) \triangleq \bar{G} \bullet T + 2w^T t \\
\text{s.t.} & \quad e^T \bar{t}_j = 1, \quad j = 1, \ldots, n, \\
& \quad T_{jj} = \text{Diag}(\bar{t}_j), \quad j = 1, \ldots, n, \\
& \quad T \succeq tt^T, \quad t \geq 0,
\end{align*}
\]

(ERSDR3)

where \( \bar{G} \) is defined in (2.11).

**Theorem 4.2.** Problem (ERSDR3) is equivalent to problem (ERSDR2).
Proof. Note that as the two problems share the same constraints, we only need to prove that the difference of the two objective functions $\bar{f}_1(T, t)$ and $\bar{f}_2(T, t)$ is a constant. Indeed, with constraints $e^\top \bar{t}_j = 1$, $t \succeq 0$, $T_{jj} = \text{Diag}(\bar{t}_j)$, $j = 1, \ldots, n$, and (iv) in Proposition A.1, we have

\begin{equation}
S_{jj} \cdot T_{jj} = S_{jj} \cdot \text{Diag}(\bar{t}_j) = \sum_{k=1}^M q_{jj}(\bar{t}_j)_k = q_{jj} e^\top \bar{t}_j = q_{jj},
\end{equation}

where $S_{jj} \in \mathbb{R}^{M \times M}$ is the $j$-th diagonal block of $G$. Consequently,

\begin{equation}
\sum_{j=1}^n S_{jj} \cdot T_{jj} = \sum_{j=1}^n q_{jj} = \sum_{j=1}^n \sum_{k=1}^m |h_{kj}|^2 = \|H\|_2^2.
\end{equation}

This, together with the definitions of $G$ and $\tilde{G}$, shows that

\begin{equation}
\bar{f}_1(T, t) - \bar{f}_2(T, t) = \|H\|_2^2.
\end{equation}

This proves that problems (ERSDR3) and (ERSDR2) are indeed equivalent. 

With [14, Theorem 2] and Theorem 4.2, we immediately have the following result.

THEOREM 4.3. Suppose that $M \geq 2$. If the inputs $H$ and $v$ in (1.1) satisfy (1.5), then problem (ERSDR3) is tight for problem (P).

Now, the relations between the series of “ERSDRs” and other formulations discussed above can be summarized in Figure 3. Problems (ERSDR1), (ERSDR2), and (ERSDR3) are SDRs of problems (RQP), (SQP1), and (SQP2), respectively. These “ERSDRs” are equivalent.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig3.png}
\caption{Relations between the series of “ERSDRs” and other formulations.}
\end{figure}

To conclude this section, we summarize the scale of the above problems in terms of the number of variables and the number of different types of constraints in Table 1. In Table 1, ‘=’ means equality constraints, ‘$\succeq$’ means positive semidefinite constraints, and ‘$\geq$’ means lower bound constraints. It can be seen from Table 1 that problem (RSQP) only involves one vector variable $t \in \mathbb{R}^{nM}_+$ and $n$ linear equality constraints, both of which are significantly smaller than those of other relaxation problems. In addition, all constraints in problem (RSQP) are linear. In sharp contrast, all SDR problems contain a positive semidefinite constraint. Our proposed relaxation problem (RSQP) enables us to develop fast algorithms for solving the large-scale MIMO detection problem. Our proposed PN-QP method for solving the MIMO detection problem is customized based on problem (RSQP), and as it will be shown in section 5, it is much more efficient compared to state-of-the-art ERSDR based approaches.

This manuscript is for review purposes only.
5. Numerical Results. In this section, we conduct extensive numerical tests to verify the efficiency of the proposed PN-QP algorithm. The algorithm is implemented in MATLAB (R2017a) and all the experiments are performed on a Lenovo ThinkPad laptop with Intel dual core i5-6200 CPU and 8 GB of memory running in Windows 10. We generate the instances of problem (P) following the way in [13, 16], which is detailed as follows:

Step 1: Generate each entry of the channel matrix $H \in \mathbb{C}^{m \times n}$ according to the complex standard Gaussian distribution (with zero mean and unit variance);
Step 2: Generate each entry of the noise vector $v \in \mathbb{C}^{m}$ according to the complex Gaussian distribution with zero mean and variance $\sigma^2$;
Step 3: Choose $k_j$ uniformly and randomly from $\{0, 1, \ldots, M-1\}$, and set $x^*_j = e^{\frac{2\pi k_j}{M}}$ for each $j \in \{1, \ldots, n\}$, where $x^*$ is the vector of transmitted signals;
Step 4: Compute the vector of received signals $r \in \mathbb{C}^{m}$ as in (1.1).

Generally, in practical digital communications, $M$ is taken as an exponential power of 2. Therefore, in our following tests, we always choose $M = 2^l$, where $l$ is a positive integer.

In our setting, we define the signal-to-noise ratio (SNR) as

$$\text{SNR} = 10 \log_{10} \left( \frac{E[\|Hx^*\|^2]}{E[\|v\|^2]} \right) = 10 \log_{10} \left( \frac{m \sigma_x^2}{\sigma_v^2} \right),$$

where $\sigma_x^2 = E[\|x^*\|^2]$, $\sigma_v^2 = E[\|v\|^2]$, and $E[\cdot]$ is the expectation operator. Then according to our ways of generating instances (i.e., $\sigma_x^2 = n$, and $\sigma_v^2 = m \sigma^2$), we have $\text{SNR} = 10 \log_{10} \left( \frac{n}{\sigma_v^2} \right)$ in our tests. Generally, the MIMO detection problem is more difficult when the SNR is low and when the numbers of inputs and outputs are equal (i.e., $m = n$).

5.1. Performance of PN-QP. When we apply the rounding procedure in (3.2) in Algorithm 3.1, we choose the index with the largest entry in each block. First, we demonstrate the efficiency of PN-QP by an example with $(m, n, M) = (4, 4, 8)$, and SNR = 30 dB. The initial point of PN-QP is chosen as $t^0 = e$. We selectively plot the iterates $\{t^k\}$ in Figure 4 with $k = 0, 9, 23$. In Figure 4, the ‘*’ denotes the vector $t^*$ corresponding to the vector of transmitted signals $x^*$ in (1.1) and the ‘◦’ denotes the iterate $t^k$ generated by PN-QP. It can be seen from Figure 4 that as the iteration goes on, $t^k$ becomes more and more sparse, and eventually, the support set of $t^k$ at $k = 23$ coincides with that of the true minimizer of problem (RSQP).

5.2. Comparison with Other Algorithms. In this subsection, we will compare the numerical performance of PN-QP for solving problem (RSQP) with different models and the corresponding algorithms, which are detailed below.

- Problem (P) solved by GPM [13]: GPM is essentially a gradient projection method whose projection step is taken directly over the discrete set $\mathcal{X}$. Due to

| Problem     | Number of variables | Number of constraints |
|-------------|---------------------|-----------------------|
| (ERSDR1)    | $2n + nM$           | $1(2n \times 2n)$     |
| (ERSDR2)    | $nM$                | $1(nM \times nM)$     |
| (ERSDR3)    | $nM$                | $1(nM \times nM)$     |
| (RSQP)      | $nM$                | $1(nM \times nM)$     |

Table 1
Comparison of different relaxations.
its low computational complexity, it is able to solve the large-scale problem. Moreover, in our implementation, we modify its output by choosing the best point among all generated iterates (to improve its performance), instead of simply using the last iterate $x^k$ as the output.

- **Problem (P) solved by SD$^1$ [7]:** SD is a typical tree search based method which searches for constellation points limited to a sphere with a predetermined radius centered on the vector of received signals $r$ to find the global solution. However, the complexity of SD is generally exponential, and hence it is impractical to solve the large-scale problem.

- **Problem (ERSDR1) solved by SDPNAL+ v1.0$^2$ [26, 36]:** As shown in Table 1, the size of problems (ERSDR2) and (ERSDR3) is significantly larger than that of (ERSDR1), and the three problems are (mathematically) equivalent due to [14, Theorem 1] and Theorem 4.2. Therefore, in our following tests, we do not compare the performance of solving problems (ERSDR2) and (ERSDR3). For $(t^k, y^k, y^k)$ of problem (ERSDR1) returned by SDPNAL+, we perform the following rounding procedure to obtain a feasible

---

$^1$The code is downloaded from https://ww2.mathworks.cn/matlabcentral/fileexchange/22890-sphere-decoderfor-mimo-systems and modified by adopting the techniques proposed in [4] to further improve its efficiency.

$^2$https://blog.nus.edu.sg/matohkc/softwares/sdpnalplus/
AN EFFICIENT ALGORITHM FOR LARGE-SCALE MIMO DETECTION

1.5

2.1

\((y_k^j, y_{n+j}^j)\) to \(Y\) in (2.1), and get \((y_k^\star, y_{n+j}^\star)\) by

\[
(y_k^\star, y_{n+j}^\star) \in \arg \min_{(u_1, u_2) \in Y} (u_1 - y_k^j)^2 + (u_2 - y_{n+j}^j)^2, \quad j = 1, \ldots, n.
\]

Then return \(x_j = y_k^\star + iy_{n+j}^\star, \quad j = 1, \ldots, n\).

In our experiments, we set \(M \in \{4, 8, 16\}\), \(n \in \{16, 32, 64, 128, 256, 512\}\), and \(m = n\) or \(m = 2n\). We use the following two metrics to evaluate the performance of different algorithms: the symbol error rate (denoted by SER) \([13, 30]\) as well as the running time in seconds (denoted by Time). More specifically, the SER is used to evaluate the detection error rate of different algorithms, which is calculated by

\[
\text{The number of incorrectly recovered entries compared to } x^*.
\]

The length of transmitted signals \(n\).

Time is used to evaluate the speed of different algorithms, which is particularly important for solving large-scale problems. We limit the maximum running time of each algorithm to be 3600 seconds. That is, we will terminate the algorithm if its running time is over 3600 seconds and we use “—” to denote such case. The reported results below are obtained by averaging over 100 randomly generated instances.

**Initial Points.** Since both PN-QP and GPM require the initial points, we first compare the effects of different initial points on the detection results of the two algorithms. We test three initial points: the zero vector \(0\), the all-one vector \(e\), and the approximate solution \(x_{ml}\) obtained by the minimum mean square error (MMSE) detector \([13]\). We report the results in Table 2.

It can be observed from Table 2 that the SER by PN-QP hardly changes for the three choices of the initial point. In comparison, the SER by GPM varies quite a lot among the three choices of the initial point, and \(x_{ml}\) is definitely the best initial point for GPM which will lead to a much smaller SER. In terms of the running time, it seems that \(x_{ml}\) for PN-QP is preferable for small-scale problems whereas \(e\) leads to the smallest running time for PN-QP among the three choices when the size of problem \(n\) is large (i.e., \(n = 512\)). For GPM, \(x_{ml}\) is also the winner from the perspective of time. Based on the above observations, in our following test, we choose \(e\) as the initial point for PN-QP and the MMSE estimator \(x_{ml}\) as the initial point for GPM. Here we would like to highlight that the numerical results with \(n \geq 128\) which we show in this paper have not appeared in literature.

**Results on Small- to Medium-Scale Problems.** Next, we compare the performance of the four algorithms in the case of small and medium scale (i.e., \(n \leq 128\)). We also report the no interference lower bound (LB) results. This approach solves the MIMO detection problem with respect to each component \(x_j\) assuming all the others being fixed to be the true transmitted signals. Again, the SER is obtained by dividing the total number of incorrectly estimated elements over the length of the transmitted signals. The solution returned by LB can be viewed as the best possible result that the MIMO detection problem can be solved theoretically. Therefore, the above no interference LB can be used as the theoretical (and generally unachievable especially in the low SNR scenarios) lower bound of the SER of all the other approaches.

It can be observed from Table 3 that as the SNR decreases, the SER achieved by each algorithm increases, implying that the problem becomes more difficult. The running time for SD becomes longer and even prohibitively high as the SNR decreases/\(n\) increases, despite that SD provides the best SER among the four algorithms. SDP-NAL+ for solving (ERSDR1) returns reasonably good SER. However, as \(n\) increases,
| \((m, n, M)\) | SNR (dB) | \(0\) | \(e\) | \(x_{ml}\) | \(0\) | \(e\) | \(x_{ml}\) | \(0\) | \(e\) | \(x_{ml}\) |
|---|---|---|---|---|---|---|---|---|---|---|
| \((64, 32, 4)\) | 45 | 0.033 | 0.026 | 0.017 | 0.001 | 0.001 | 0.001 | 0.00 | 0.00 | 0.00 |
| | 30 | 0.028 | 0.023 | 0.014 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| | 15 | 0.034 | 0.029 | 0.018 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| \((32, 32, 4)\) | 45 | 0.063 | 0.049 | 0.028 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| | 30 | 0.058 | 0.044 | 0.025 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| | 15 | 0.059 | 0.046 | 0.032 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| \((64, 32, 8)\) | 45 | 0.094 | 0.097 | 0.078 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| | 30 | 0.093 | 0.094 | 0.072 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| | 15 | 0.101 | 0.107 | 0.086 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| \((32, 32, 8)\) | 45 | 0.174 | 0.167 | 0.141 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| | 30 | 0.172 | 0.164 | 0.137 | 0.001 | 0.001 | 0.000 | 0.00 | 0.00 | 0.00 |
| | 15 | 0.200 | 0.202 | 0.176 | 0.001 | 0.001 | 0.001 | 4.44 | 4.44 | **1.38** |
| \((1024, 512, 4)\) | 45 | 3.112 | 1.829 | 2.266 | 0.208 | 0.309 | **0.144** | 0.00 | 0.00 | 0.00 |
| | 30 | 3.101 | 1.823 | 2.251 | 0.206 | 0.302 | **0.145** | 0.00 | 0.00 | 0.00 |
| | 15 | 3.111 | 1.841 | 2.256 | 0.203 | 0.268 | **0.145** | 0.00 | 0.00 | 0.00 |
| \((512, 512, 4)\) | 45 | 3.765 | 2.245 | 2.705 | 1.056 | 1.091 | **0.093** | 0.00 | 0.00 | 0.00 |
| | 30 | 3.778 | 2.254 | 2.707 | 1.054 | 1.127 | **0.092** | 0.00 | 0.00 | 0.00 |
| | 15 | 3.864 | 2.354 | 2.835 | 1.125 | 1.146 | **0.123** | 0.00 | 0.00 | 0.00 |
| \((1024, 512, 8)\) | 45 | 3.765 | 6.071 | 7.942 | 2.003 | 2.716 | **0.145** | 0.00 | 0.00 | 0.00 |
| | 30 | 9.199 | 6.031 | 7.870 | 1.932 | 2.667 | **0.144** | 0.00 | 0.00 | 0.00 |
| | 15 | 9.375 | 6.245 | 8.084 | 0.439 | 2.515 | **0.166** | 0.00 | 0.00 | 0.00 |
| \((512, 512, 8)\) | 45 | 13.479 | 9.159 | 11.345 | 1.675 | 1.637 | **0.092** | 0.00 | 0.00 | 0.00 |
| | 30 | 13.611 | 9.221 | 11.419 | 1.629 | 1.591 | **0.104** | 0.00 | 0.00 | 0.00 |
| | 15 | 15.172 | 10.804 | 12.990 | 1.551 | 1.656 | **1.189** | 0.43 | **0.34** | 0.38 |

**Table 2**

Comparison of initial points for PN-QP and GPM.
its running time increases rapidly. This can be explained by the numbers of the variables (one $2n \times 2n$ matrix variable and one $2n + nM$ vector variable) and constraints (one $4n \times 4n$ positive semidefinite constraint, $6n$ equality constraints, and $nM$ inequality constraints) in Table 1. Consequently, we will not compare with SD and SDPNAL+ in our subsequent tests on large-scale problems.

Compared with GPM, PN-QP achieves better detection performance as the SNR decreases. Moreover, its running time increases slowly as $n$ increases. This is mainly due to the vector formulation of problem (RSQP). For example, for $(m, n, M) = (128, 128, 8)$ and SNR = 14 dB, it takes about one second for PN-QP to return a solution with SER = 45%, whereas SDPNAL+ takes about two minutes to return a solution with a larger SER 10.41%, and GPM returns a solution with the SER being 16.63% instantly (i.e., 0.018 seconds). For such example, SD fails to return a solution within one hour. Among all four algorithms, GPM is the fastest one but its SER performance is not as good as others.

To better understand the detection performance of the four algorithms, we plot Figure 5, showing the SER with respect to the SNR for each algorithm. It can be seen from Figure 5 that for $(m, n, M) = (16, 16, 8)$, SD performs the best since its SER curve coincides with LB when the SNR is large. SDPNAL+ also performs very well since the curve of SDPNAL+ becomes parallel to LB, i.e., a constant SER gap. However, this is not the case for PN-QP and GPM. For $(m, n, M) = (32, 16, 8)$, PN-QP, SDPNAL+, and SD are competitive, whereas SD is the best one.

Table 3

| $(m, n, M)$ | SNR (dB) | Time(s) | SER(%) |
|-------------|----------|---------|--------|
| (32, 32, 8) | 22       | 0.329   | 11.145 |
|             | 20       | 0.334   | 11.171 |
|             | 18       | 0.315   | 10.969 |
|             | 16       | 0.267   | 8.335  |
|             | 14       | 0.276   | 7.537  |
|             | 12       | 0.296   | 7.884  |
| (64, 64, 8) | 22       | 0.381   | 34.573 |
|             | 20       | 0.421   | 40.113 |
|             | 18       | 0.421   | 36.913 |
|             | 16       | 0.398   | 32.544 |
|             | 14       | 0.448   | 31.760 |
|             | 12       | 0.456   | 29.165 |
| (128, 128, 8) | 22   | 0.770   | 128.872|
|             | 20      | 0.785   | 127.363|
|             | 18      | 0.796   | 127.690|
|             | 16      | 0.866   | 124.961|
|             | 14      | 0.907   | 128.751|
|             | 12      | 0.979   | 127.204|

Results on Large-Scale Problems. We further compare the performance of PN-QP and GPM on large-scale problems, i.e., $n \geq 128$. According to Table 4, as the SNR decreases, PN-QP provides a lower SER than GPM, implying that PN-QP achieves better detection performance. For instance, for $(m, n, M) = (512, 512, 16)$ with SNR = 20 dB, PN-QP returns a solution with SER = 3.69%, whereas the SER of GPM is 34.19%. We also present more comparisons of the two algorithms in Figure 6. It can be observed from Figure 6 that as the SNR increases, the SER curve of PN-QP...
tends to coincide with the LB. In contrast, there is a large gap between the SER curve of GPM with the LB, especially in the case where \( m = n \).

Based on the above comparisons, our proposed algorithm for solving the MIMO detection problem is more efficient and robust than existing algorithms for solving large-scale problems. Specifically, compared to SDPNAL+ and SD, PN-QP is more efficient; compared to GPM, PN-QP achieves better detection performance and is more robust to the choice of the initial point. To conclude, PN-QP is demonstrated to be a competitive candidate for solving the large-scale MIMO detection problem.

6. Conclusions. In this paper, we proposed an efficient algorithm called PN-QP for solving the large-scale MIMO detection problem, motivated by the massive MIMO technology. The proposed algorithm is essentially a quadratic penalty method applied to solve an SQP relaxation, i.e., problem \((\text{RSQP})\), of the original problem. Two key features of the proposed algorithm, which make it particularly suitable to solve the large-scale problems, are: (i) it is based on the relaxation problem \((\text{RSQP})\), whose numbers of variables and constraints are significantly less than those of the SDRs; and (ii) our proposed algorithm is custom-designed to identify the support set of the optimal solution by judiciously exploiting the special structure of the problem, instead of finding the solution itself, which thus substantially reduces the computational complexity of the proposed algorithm. Our extensive simulation results show that our proposed algorithm compares favorably with the state-of-the-art algorithms (including SD and SDR based approaches) for solving the MIMO detection problem. In particularly, when applied to solve large-scale problems, our proposed algorithm not only achieves significantly better detection performance but also more robust to the choice of the initial point than GPM.
Table 4

Time and SER comparison of PN-QP and GPM for solving large-scale MIMO detection problems.

| (m, n, M) | SNR (dB) | Time(s) | SER(%) |
|-----------|----------|---------|--------|
|           |          | PN-QP | GPM | PN-QP | GPM | LB |
| (256, 128, 8) | 30 | 0.475 | 0.005 | 0.00 | 0.00 | 0.00 |
|           | 25 | 0.450 | 0.003 | 0.00 | 0.00 | 0.00 |
|           | 20 | 0.456 | 0.003 | 0.00 | 0.00 | 0.00 |
|           | 15 | 0.478 | 0.004 | 0.00 | 0.01 | 0.00 |
| (128, 128, 8) | 30 | 0.676 | 0.002 | 0.00 | 0.00 | 0.00 |
|           | 25 | 0.689 | 0.003 | 0.00 | 0.00 | 0.00 |
|           | 20 | 0.704 | 0.004 | 0.00 | 0.00 | 0.00 |
|           | 15 | 0.793 | 0.011 | 0.00 | 0.12 | 13.49 |
| (512, 256, 8) | 30 | 1.465 | 0.002 | 0.00 | 0.00 | 0.00 |
|           | 25 | 1.478 | 0.002 | 0.00 | 0.00 | 0.00 |
|           | 20 | 1.479 | 0.002 | 0.00 | 0.00 | 0.00 |
|           | 15 | 1.522 | 0.021 | 0.00 | 0.00 | 0.00 |
| (256, 256, 8) | 30 | 2.659 | 0.013 | 0.00 | 0.00 | 0.00 |
|           | 25 | 2.652 | 0.016 | 0.00 | 0.00 | 0.00 |
|           | 20 | 2.687 | 0.019 | 0.00 | 0.00 | 0.00 |
|           | 15 | 2.951 | 0.114 | 0.00 13.87 | 0.21 |
| (1024, 512, 8) | 30 | 5.930 | 0.143 | 0.00 | 0.00 | 0.00 |
|           | 25 | 5.953 | 0.140 | 0.00 | 0.00 | 0.00 |
|           | 20 | 5.997 | 0.142 | 0.00 | 0.00 | 0.00 |
|           | 15 | 6.174 | 0.165 | 0.00 | 0.00 | 0.00 |
| (512, 512, 8) | 30 | 9.028 | 0.104 | 0.00 | 0.00 | 0.00 |
|           | 25 | 9.064 | 0.114 | 0.00 | 0.00 | 0.00 |
|           | 20 | 9.376 | 0.146 | 0.00 | 0.00 | 0.00 |
|           | 15 | 10.621 | 1.114 | 0.00 10.73 | 0.23 |
| (256, 128, 16) | 30 | 3.469 | 0.006 | 0.00 | 1.98 | 0.00 |
|           | 25 | 3.315 | 0.016 | 0.00 | 15.51 | 0.00 |
|           | 20 | 3.585 | 0.023 | 0.00 | 33.73 | 0.63 |
|           | 15 | 3.645 | 0.046 | 33.86 | 47.11 | 13.08 |
| (128, 128, 16) | 30 | 6.219 | 0.019 | 0.00 | 0.00 | 0.00 |
|           | 25 | 6.333 | 0.020 | 0.00 | 0.00 | 0.00 |
|           | 20 | 6.511 | 0.022 | 0.00 | 0.00 | 0.00 |
|           | 15 | 7.277 | 0.044 | 4.27 | 4.59 | 2.98 |
| (512, 256, 16) | 30 | 11.215 | 0.033 | 0.00 | 0.00 | 0.00 |
|           | 25 | 11.382 | 0.170 | 0.00 | 16.39 | 0.00 |
|           | 20 | 12.179 | 0.200 | 5.04 | 34.14 | 0.64 |
|           | 15 | 12.505 | 0.211 | 33.35 | 48.71 | 12.38 |
| (520, 512, 16) | 30 | 23.078 | 0.148 | 0.00 | 0.00 | 0.00 |
|           | 25 | 23.418 | 0.150 | 0.00 | 0.00 | 0.00 |
|           | 20 | 24.074 | 0.182 | 0.01 | 0.01 | 0.01 |
|           | 15 | 26.468 | 0.397 | 4.39 | 4.70 | 2.81 |
| (512, 512, 16) | 30 | 41.136 | 0.292 | 0.00 | 0.72 | 0.00 |
|           | 25 | 41.936 | 2.378 | 0.00 | 18.53 | 0.00 |
|           | 20 | 46.996 | 2.831 | 3.69 | 34.19 | 0.65 |
|           | 15 | 45.173 | 2.425 | 34.11 | 48.16 | 12.09 |

Appendix A. Proof of Theorem 2.2. We need the following results to prove Theorem 2.2.

**Proposition A.1.** Let Q, Q, and G be defined in (1.3), (2.2), and (2.7), respectively. We have

(i) $q_{jj} = \sum_{k=1}^{m} |h_{kj}|^2$ for $j = 1, \ldots, n$;

(ii) $q_{kj} = q_{jk} = q_{(n+j)(n+k)} = q_{(n+k)(n+j)}$, $q_{(n+j)k} = q_{k(n+j)} = -q_{(n+k)j} = -q_{j(n+k)}$, $q_{jj} = q_{(n+j)(n+j)} = q_{jj}$, and $q_{(n+j)j} = q_{j(n+j)} = 0$ for $j, k = 1, \ldots, n$;

This manuscript is for review purposes only.
Fig. 6. The SER performance of PN-QP and GPM under different SNRs.

(iii) $S_{jk} = \hat{q}_{jk} \alpha^T + \hat{q}_{(n+j)k} \beta^T + \hat{q}_{(n+j)(n+k)} \beta^T + \hat{q}_{(n+j)(n+k)} \beta^T$ and $S_{jk} = S^T_{kj}$ for $j, k = 1, \ldots, n$; and

(iv) $\text{diag}(S_{jj}) = q_{jj} e$ for $j = 1, \ldots, n$.

Proof. (i) By the definition of $Q$ in (1.3), there is

$$q_{jj} = \sum_{k=1}^{m} \langle h_{kj} \rangle^2 = \sum_{k=1}^{m} |h_{kj}|^2,$$

which gives (i).

(ii) By the definitions of $\hat{Q}$ in (2.2) and $Q$ in (1.3), the first three results in (ii) hold naturally. Note that due to (i), there is $\text{diag}(\text{Im}(Q)) = 0$, implying the fourth result in (ii).

(iii) Let $P$ have the partition as $P = [P_1 \cdots P_n]$, where $P_j \in \mathbb{R}^{2n \times M}$ takes
the following form:
\[
P_j^T = [0 \cdots 0 \alpha \cdots 0 \beta \cdots 0]^T.
\]
(A.1)

By the definition and partition of \( G \) in (2.7) and (2.10), we have
\[
G = P^T \hat{Q} P = [P_1 \cdots P_n]^T \hat{Q} [P_1 \cdots P_n]
\]
\[
= \begin{bmatrix}
P_1^T \hat{Q} P_1 & \cdots & P_1^T \hat{Q} P_n \\
\vdots & \ddots & \vdots \\
P_n^T \hat{Q} P_1 & \cdots & P_n^T \hat{Q} P_n
\end{bmatrix}
= \begin{bmatrix}
S_{11} & \cdots & S_{1n} \\
\vdots & \ddots & \vdots \\
S_{n1} & \cdots & S_{nn}
\end{bmatrix},
\]
which, together with (2.2) and (A.1), further implies
\[
S_{jk} = P_j^T \hat{Q} P_k
= [\hat{q}_j \alpha + \hat{q}_{(n+j)} \beta \cdots \hat{q}_{(2n)} \alpha + \hat{q}_{(n+j)(2n)} \beta] P_k
= \hat{q}_j \alpha \alpha^T + \hat{q}_{(n+j)} \beta \alpha^T + \hat{q}_{(n+j)(n+k)} \beta \beta^T.
\]
The proof of the first result in (iii) is finished. With the first result in (iii), as well as the first two results in (ii), there is
\[
S_{jk} = \hat{q}_j \alpha \alpha^T + \hat{q}_{(n+j)} \beta \alpha^T + \hat{q}_{(n+j)(n+k)} \beta \beta^T
= \hat{q}_k \alpha \alpha^T + \hat{q}_{(n+j)} \beta \alpha^T + \hat{q}_{(n+k)(n+j)} \beta \beta^T
= (\hat{q}_j \alpha \alpha^T + \hat{q}_{(n+j)} \beta \alpha^T + \hat{q}_{(n+k)(n+j)} \beta \beta^T)^T
= \tilde{S}_{kj}.
\]
We get the second result in (iii).

(iv) Due to (iii), there is
\[
S_{jj} = q_j \alpha \alpha^T + \hat{q}_{(n+j)} \beta \alpha^T + \hat{q}_{(n+j)(n+k)} \beta \beta^T
= q_j \alpha \alpha^T + \beta \beta^T.
\]
Recall the definitions of \( \alpha, \beta \) in (2.5) and \( \theta_k \) in (2.1). The \( k \)-th diagonal entry of \( S_{jj} \) takes the following form:
\[
q_j (\cos^2 (\theta_k) + \sin^2 (\theta_k)) = q_{jj}, \quad k = 1, \ldots, M.
\]
Therefore, \( \text{diag}(S_{jj}) = q_{jj} e \). The proof is finished.

**Proposition A.2.** Under the constraints in problem (SQP2), there is
\[
(A.2) \quad \sum_{j=1}^n \tilde{e}_j^T S_{jj} \tilde{e}_j = \| H \|_2^2.
\]

**Proof.** Under the constraints in problem (SQP2), for each \( \tilde{e}_j \), there exists \( l_j \in \{1, \ldots, M\} \), such that \( \tilde{l}_j = e_{l_j} \), where \( e_{l_j} \) denotes the \( l_j \)-th column in the identity matrix \( I \in \mathbb{R}^{M \times M} \). As a result, with (i) and (iv) in Proposition A.1, there is
\[
\sum_{j=1}^n \tilde{e}_j^T S_{jj} \tilde{e}_j = \sum_{j=1}^n \tilde{e}_j^T S_{jj} e_{l_j} = \sum_{j=1}^n q_{jj} = \sum_{j=1}^n \sum_{k=1}^m |h_{kj}|^2 = \| H \|_2^2,
\]
which gives (A.2). The proof is finished.
Now we are ready to prove Theorem 2.2.

Proof. Using Proposition A.2, we have
\[ h(t) - f(t) = \sum_{j=1}^{n} \bar{t}_j^T S_{jj} \bar{t}_j = \|H\|_2^2. \]

This, together with the fact that the constraints of problems (SQP1) and (SQP2) are the same, implies that problems (SQP1) and (SQP2) are equivalent. □

Appendix B. Proof of Theorem 3.5. The proof is similar to that in [22, Theorem 17.2]. However, there is a nonsmooth term \( \delta_{\mathbb{R}^+} \) in subproblem (3.6). Therefore, for the sake of completeness, we include our proof as follows.

Proof. For \( \gamma \in \mathbb{R}^M \), let \( \mathcal{I}(\gamma) \) be the set of the indices of zero elements, i.e.,
\[ \mathcal{I}(\gamma) = \{ \ell \mid \gamma_\ell = 0, \ \forall \ \ell = 1, \ldots, M \}. \] (B.1)

Hence, the support set
\[ \mathcal{K}(\gamma) = \{1, \ldots, M\} \setminus \mathcal{I}(\gamma). \] (B.2)

Let \( e(t) \) be defined as
\[ e(t) = \begin{bmatrix} (e^T \bar{t}_1 - 1)e \\ \vdots \\ (e^T \bar{t}_n - 1)e \end{bmatrix}. \]

The first-order optimality condition in (3.7) can be reformulated as
\[ \text{dist} \left( 0, \nabla f(t) + \omega_k e(t) + \partial \delta_{\mathbb{R}^+} (t) \right) = 0. \] (B.3)

Consequently, by (3.8) we have
\[ \|e(t^k)\|_2 = \frac{\text{dist}(0, \omega_k e(t^k))}{\omega_k} \leq \frac{1}{\omega_k} \left( \text{dist} \left( 0, \nabla f(t^k) + \omega_k e(t^k) + \partial \delta_{\mathbb{R}^+} (t^k) \right) \right. \]
\[ + \text{dist} \left( \omega_k e(t^k), \nabla f(t^k) + \omega_k e(t^k) + \partial \delta_{\mathbb{R}^+} (t^k) \right) \] \[ \leq \frac{1}{\omega_k} \left( \tau_k + \text{dist} \left( 0, \nabla f(t^k) + \partial \delta_{\mathbb{R}^+} (t^k) \right) \right). \] (B.4)

Let \( t^* \) be an accumulation point of \( \{t^k\} \). That is, there exists an infinite subsequence \( k \in \mathbb{K} \) such that \( \lim_{k \to +\infty} t^k = t^* \). According to line 3 in Algorithm 3.1, for \( k \in \mathbb{K} \), by the continuity of \( \nabla f(\cdot) \) as well as the upper semicontinuity of \( \partial \delta_{\mathbb{R}^+} (\cdot) \), there is \( \nabla f(t^k) \to \nabla f(t^*) \) and \( \partial \delta_{\mathbb{R}^+} (t^k) \to \partial \delta_{\mathbb{R}^+} (t^*) \), as \( k \to +\infty \). Together with \( \omega_k \uparrow +\infty \), (B.4) implies that \( \|e(t^k)\|_2 \to 0 \), i.e.,
\[ e^T \bar{t}^* - 1 = 0, \ j = 1, \ldots, n. \] (B.5)

Let \( \lambda_j^k = -\omega_k (e^T \bar{t}_j^k - 1), \ j = 1, \ldots, n \). We will show that
\[ \lim_{k \to +\infty, k \in \mathbb{K}} \lambda^k = \lambda^*, \] (B.6)
and there must exist $\mu^* \in \mathbb{R}^{nM}$, such that $(t^*, \lambda^*, \mu^*)$ satisfies the first-order optimality condition of problem (RSQP). Note that by $\tau_k \downarrow 0$, $t^*$ satisfies (B.3) when $k \to +\infty$. It is equivalent to the following

$$\lim_{k \to +\infty} (t^k - \lfloor t^k - \nabla f_{\omega_k}(t^k) \rfloor^+) = 0,$$

where $\lfloor \cdot \rfloor^+$ denotes $\max \{0, \cdot \}$.

Recall $\mathcal{I}(\cdot)$ and $\mathcal{K}(\cdot)$ defined in (B.1) and (B.2). There is

$$\lim_{k \to +\infty} (\bar{t}^k)_s - (\bar{t}^k)_s - (\nabla f_{\omega_j}(t^k))_s - \lambda^k_j = 0, \ s \in \mathcal{K}(\bar{t}^k), \ j = 1, \ldots, n,$$

(B.7)

$$\lim_{k \to +\infty} (\bar{t}^k)_l - (\bar{t}^k)_l - (\nabla f_{\omega_j}(t^k))_l - \lambda^k_j = 0, \ l \in \mathcal{I}(\bar{t}^k), \ j = 1, \ldots, n.$$  

(B.8)

Note that $\lim_{k \to +\infty} (\bar{t}^k)_s = (\bar{t}^*)_s > 0$ and $\lim_{k \to +\infty} (\bar{t}^k)_l = (\bar{t}^*)_l = 0, \ s \in \mathcal{K}(\bar{t}^*_s), \ l \in \mathcal{I}(\bar{t}^*_s)$.

Moreover, $\lim_{k \to +\infty} \nabla f(t^k) = \nabla f(t^*)$. By (B.7), for $j = 1, \ldots, n$, there is

$$\lim_{k \to +\infty} (\bar{t}^k)_s - (\nabla f_{\omega_j}(t^k))_s - \lambda^k_j = 0, \ s \in \mathcal{K}(\bar{t}^*), \ j = 1, \ldots, n.$$  

(B.9)

Due to the continuity of $\lfloor \cdot \rfloor^+$, for sufficiently large $k \in \mathbb{K}$ and $s \in \mathcal{K}(\bar{t}^*_s)$, there is

$$\lim_{k \to +\infty} (\bar{t}^k)_s - (\nabla f_{\omega_j}(t^k))_s - \lambda^k_j = \lim_{k \to +\infty} (\bar{t}^k)_s - (\nabla f_{\omega_j}(t^k))_s - \lambda^k_j = (\bar{t}^*)_s.$$  

(B.10)

Consequently, (B.9) and (B.10) imply that $\lambda^k_j$ converges as $k \to +\infty$, $k \in \mathbb{K}$ and (B.6) holds. Then (B.10) gives

$$\lim_{k \to +\infty} (\nabla f_{\omega_j}(t^*))_s - \lambda^*_j = 0, \ s \in \mathcal{K}(\bar{t}^*_s), \ j = 1, \ldots, n.$$  

(B.11)

Combining (B.8) and (B.11) implies

$$\lim_{k \to +\infty} (\nabla f_{\omega_j}(t^*))_l - \lambda^*_j = 0, \ l \in \mathcal{I}(\bar{t}^*_s), \ j = 1, \ldots, n.$$  

(B.12)

Define $\mu^* \in \mathbb{R}^{nM}$ as

$$\lim_{k \to +\infty} (\bar{t}^k)_s - (\nabla f_{\omega_j}(t^k))_s - \lambda^k_j = \lim_{k \to +\infty} (\bar{t}^k)_s - (\nabla f_{\omega_j}(t^k))_s - \lambda^k_j = (\bar{t}^*)_s.$$  

(B.13)

$\mu^* \in \mathbb{R}^{nM}$, and $(\bar{t}^k)_l = (\nabla f_{\omega_j}(t^k))_l - \lambda^k_j$, $l \in \mathcal{I}(\bar{t}^*_s), \ j = 1, \ldots, n$.

(B.14)

With (B.11)–(B.13), one can easily show that $(t^*, \lambda^*, \mu^*)$ satisfies the first-order optimality condition of problem (RSQP), where $\lambda^*$ and $\mu^*$ correspond to the Lagrange multipliers of equality and inequality constraints, respectively. The proof is finished.

Acknowledgments. We would like to thank Dr. Huikang Liu for kindly sharing the GPM code with us.
REFERENCES

[1] M. A. Albreem, M. Juntti, and S. Shahabuddin, *Massive MIMO detection techniques: A survey*, IEEE Communications Surveys and Tutorials, 21 (2019), pp. 3109–3132.

[2] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM Journal on Control and Optimization, 20 (1982), pp. 221–246.

[3] S. A. Busari, K. M. S. Huq, S. Mumtaz, L. Dai, and J. Rodriguez, *Millimeter-wave massive MIMO communication for future wireless systems: A survey*, IEEE Communications Surveys and Tutorials, 20 (2018), pp. 836–869.

[4] A. M. Chan and I. Lee, *A new reduced-complexity sphere decoder for multiple antenna systems*, in Proceedings of IEEE International Conference on Communications, New York, 2002, pp. 460–464.

[5] Cisco, *Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022*, 2019, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html (accessed 2019-12-10).

[6] C. Cui, Q.-N. Li, L. Qi, and H. Yan, *A quadratic penalty method for hypergraph matching*, Mathematical Programming, 70 (2018), pp. 237–259.

[7] O. Damen, A. Chkeif, and J.-C. Belfiore, *Lattice code decoder for space-time codes*, IEEE Communications Letters, 4 (2000), pp. 161–163.

[8] J. Jaldén, *Detection for multiple input multiple output channels*, PhD thesis, KTH Royal Institute of Technology, 2006.

[9] J. Jaldén and B. Ottersten, *On the complexity of sphere decoding in digital communications*, IEEE Transactions on Signal Processing, 53 (2005), pp. 1474–1484.

[10] J. Jaldén, D. Seethaler, and G. Matz, *Worst- and average-case complexity of LLL lattice reduction in MIMO wireless systems*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, 2008, pp. 2685–2688.

[11] A. K. Lenstra, H. W. Lenstra, and L. Lovász, *Factoring polynomials with rational coefficients*, Mathematische Annalen, 261 (1982), pp. 515–534.

[12] Q.-N. Li and D.-H. Li, *A projected semismooth Newton method for problems of calibrating least squares covariance matrix*, Operations Research Letters, 39 (2011), pp. 103–108.

[13] H. Liu, M.-C. Yue, A. M.-C. So, and W.-K. Ma, *A discrete first-order method for large-scale MIMO detection with provable guarantees*, in Proceedings of IEEE Workshop on Signal Processing Advances in Wireless Communications, Sapporo, 2017, pp. 669–673.

[14] Y.-F. Liu, Z. Xu, and C. Lu, *On the equivalence of semidefinite relaxations for MIMO detection with general constellations*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Brighton, 2019, pp. 4549–4553.

[15] M. P. Lotter and P. V. Rooyen, *Space division multiple access for cellular CDMA*, in Proceedings of IEEE International Symposium on Spread Spectrum Techniques and Applications, Sun City, 1998, pp. 959–964.

[16] C. Lu, Y.-F. Liu, W.-Q. Zhang, and S. Zhang, *Tightness of a new and enhanced semidefinite relaxation for MIMO detection*, SIAM Journal on Optimization, 29 (2019), pp. 719–742.

[17] C. Lu, Y.-F. Liu, and J. Zhou, *An efficient global algorithm for nonconvex complex quadratic problems with applications in wireless communications*, in Proceedings of IEEE/CIC International Conference on Communications in China, Qingdao, 2017, pp. 1–5.

[18] Z.-Q. Luo, X. Luo, and M. Kusalić, *An efficient quasi-maximum likelihood decoder for PSK signals*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Hong Kong, 2003, pp. 561–564.

[19] W.-K. Ma, P.-C. Ching, and Z. Ding, *Semidefinite relaxation based multiuser detection for M-ary PSK multiuser systems*, IEEE Transactions on Signal Processing, 52 (2004), pp. 2862–2872.

[20] A. Mobasher, M. Taherzadeh, R. Sotirov, and A. K. Khandani, *A near-maximum-likelihood decoding algorithm for MIMO systems based on semi-definite programming*, IEEE Transactions on Information Theory, 53 (2007), pp. 3869–3886.

[21] A. F. Molisch, V. V. Ratnam, S. Han, Z. Li, S. L. H. Nguyen, L. Li, and K. Haneda, *Hybrid beamforming for massive MIMO: A survey*, IEEE Communications Magazine, 55 (2017), pp. 134–141.

[22] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, 2006.

[23] M. Pohst, *On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications*, ACM Sigsam Bulletin, 15 (1981), pp. 37–44.

[24] A. M.-C. So, *Probabilistic analysis of the semidefinite relaxation detector in digital communications*, in Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, Austin, 2010, pp. 698–711.
[25] J. F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optimization Methods and Software, 11 (1999), pp. 625–653.

[26] D. Sun, K.-C. Toh, Y. Yuan, and X.-Y. Zhao, SDPNAL+: A MATLAB software for semidefinite programming with bound constraints (version 1.0), Optimization Methods and Software, 35 (2020), pp. 87–115.

[27] P. H. Tan and L. K. Rasmussen, The application of semidefinite programming for detection in CDMA, IEEE Journal on Selected Areas in Communications, 19 (2001), pp. 1442–1449.

[28] R. H. Tutuncu, K.-C. Toh, and M. J. Todd, Solving semidefinite-quadratic-linear programs using SDPT3, Mathematical Programming, 95 (2003), pp. 189–217.

[29] S. Verdú, Computational complexity of optimum multiuser detection, Algorithmica, 4 (1989), pp. 303–312.

[30] H.-T. Wai, W.-K. Ma, and A. M.-C. So, Cheap semidefinite relaxation MIMO detection using row-by-row block coordinate descent, in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Prague, 2011, pp. 3256–3259.

[31] Z. Xie, C. K. Rushforth, R. T. Short, and T. K. Moon, Joint signal detection and parameter estimation in multiuser communications, IEEE Transactions on Communications, 41 (1993), pp. 1208–1216.

[32] L. Yang, D. Sun, and K.-C. Toh, SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints, Mathematical Programming Computation, 7 (2015), pp. 331–366.

[33] S. Yang and L. Hanzo, Fifty years of MIMO detection: The road to large-scale MIMOs, IEEE Communications Surveys and Tutorials, 17 (2015), pp. 1941–1988.

[34] H. Yao, Efficient signal, code, and receiver designs for MIMO communication systems, PhD thesis, Massachusetts Institute of Technology, 2003.

[35] X.-Y. Zhao, A semismooth Newton-CG augmented Lagrangian method for large scale linear and convex quadratic SDPs, PhD thesis, National University of Singapore, 2009.

[36] X.-Y. Zhao, D. Sun, and K.-C. Toh, A Newton-CG augmented Lagrangian method for semidefinite programming, SIAM Journal on Optimization, 20 (2010), pp. 1737–1765.

[37] Q. Zhou and X. Ma, Element-based lattice reduction algorithms for large MIMO detection, IEEE Journal on Selected Areas in Communications, 31 (2013), pp. 274–286.