J Comb Optim (2017) 33:1395–1410
DOI 10.1007/s10878-016-0045-5

A new effective branch-and-bound algorithm to the high order MIMO detection problem

Ye Tian[1] · **Ke Li**[2] · **Wei Yang**[3] · **Zhiyong Li**[4]

Published online: 28 June 2016

Abstract This paper develops a branch-and-bound method based on a new convex reformulation to solve the high order MIMO detection problem. First, we transform the original problem into a $\{-1, 1\}$ constrained quadratic programming problem with the smallest size. The size of the reformulated problem is smaller than those problems derived by some traditional transformation methods. Then, we propose a new convex reformulation which gets the maximized average objective value as the lower bound estimator in the branch-and-bound scheme. This estimator balances very well between effectiveness and computational cost. Thus, the branch-and-bound algorithm achieves a high total efficiency. Several simulations are used to compare the performances of our method and other benchmark methods. The results show that this proposed algorithm is very competitive for high accuracy and relatively good efficiency.

Keywords MIMO detection problem · Quadratic Programming · New convex reformulation · Branch-and-Bound algorithm

✉ Zhiyong Li
lzy@swufe.edu.cn

1 School of Business Administration and Collaborative Innovation Center of Financial Security, Southwestern University of Finance and Economics, Chengdu 610000, China

2 School of Statistics and Collaborative Innovation Center of Financial Security, Southwestern University of Finance and Economics, Chengdu 610000, China

3 School of Insurance and Collaborative Innovation Center of Financial Security, Southwestern University of Finance and Economics, Chengdu 610000, China

4 School of Finance and Collaborative Innovation Center of Financial Security, Southwestern University of Finance and Economics, Chengdu 610000, China

1 Introduction

Compared with the single antenna communication system, the multiple antennae communication system has many advantages, such as much larger capacity, much higher utilization rate and more substantial performance. Thus, it develops very fast in the recent decade and has a lot of applications (Ma et al. 2009; Pan et al. 2014; Sidiropoulos and Luo 2006; Wübben et al. 2011. However, accompanied with these benefits, it also brings much more complexities to the communication system. Thus, how to efficiently and effectively detect the signal vector of the transmitted symbols has become an important issue and attracted a lot of attention over the last several years.

Suppose this multi-input-multi-output (MIMO) communication system contains n transmit antennas and m receive antennas. Then the received signal vector y is given as

$$y = Hx + v, \tag{1}$$

where $x \in \mathbb{R}^n$ is a transmitted symbol vector whose elements are independently selected from a finite constellation set, H is a $m \times n$ real matrix that characterizes the input-output relation and $v \in \mathbb{R}^m$ is an additive white Gaussian noise with unit variance. Since a signal model with complex values can be easily reformulated into a real-valued model (Ma et al. 2009), we only study the real-valued model in this paper.

Note that, the objective of the MIMO problem is to detect the transmitted vector x with the minimum error probability based on the observation y and H. Therefore, the corresponding optimal solution can be found by solving the following maximum-likelihood (ML) detection problem (Verdú 1998).

$$\begin{array}{ll} \min & \|y - Hx\|^2 \\ s.t. & x_i \in \{\pm 1, ..., \pm u\}, i = 1, ..., n, \end{array} \tag{2}$$

where $\|\cdot\|$ denotes the 2-norm, u is an odd integer strictly greater than 1 and each symbol x_i is drawn from a $(u + 1)$ quadrature amplitude modulation (QAM) constellation set $S = \{\pm 1, \pm 3, .., \pm u\}$ (Damen et al. 2003). It is worth pointing out that the MIMO problem is very fundamental in the communication area and receives considerable attention. However, this problem is NP-Hard (Fincke and Pohst 1985).

One classical method to approach this problem is the lattice decoding (Gamal et al. 2004; Singh et al. 2012; Taherzadeh and Khandani 2010). It is famous for good tradeoff between detection accuracy and complexity . Thus, many researchers adopt the lattice decoding to solve the MIMO problem. Naive lattice decoding (NLD) method relaxes the symbol bound constraints and finds the closest lattice point to the received signal over the whole lattice generated by the channel (Gamal et al. 2004). For further improvement of the efficiency, some suboptimal lattice decoding methods, such as sampled decoding (Liu et al. 2011), embedded decoding (Luzzi et al. 2013) and lattice reduction-aided (LRA) methods (Wübben et al. 2011), can be combined with lattice decoding method to accelerate the lattice point search. Moreover, to fix the flaw of the NLD method which completely ignores the symbol bound (see Gamal et al.

2004; Taherzadeh and Khandani 2010 for details), some researchers developed the regularized lattice decoding (RLD) method which can prevent the lattice points going too far away from the origin point. Then, a quadratic penalization term is added to the lattice decoding metric (Jaldén and Elia 2010). Note that, though the RLD method has been empirically found to be efficient for small to moderate problems sizes, it is prohibitive for large n and higher order QAM due to its complexity (Jaldén and Ottersten 2005).

Another important approximation method is based on the semidefinite relaxation (SDR). This method first reformulates the original problem into a linear conic programming problem. Then the linear matrix inequality (LMI) representation is used to get a solvable semidefinite programming (SDP) relaxation (Sidiropoulos and Luo 2006). Therefore, the SDR detector can be polynomial-time solved. Note that, the SDR detector was first proposed for the binary phase-shift keying (BPSK) constellation (Tan and Rasmussen 2001) and then extended to QPSK (4-QAM) constellation (Ma et al. 2004). It has been verified that the SDR detector is able to provide a constant factor approximation to the optimal log-likelihood value in the low signal-to-noise ration (SNR) region almost surely (Kisialiou and Luo 2005). Moreover, (Wiesel et al. 2005) developed a polynomial-inspired SDR (PI-SDR) method for 16-QAM and proved that PI-SDR achieves an optimal Lagrangian dual lower bound of the ML. Sidiropoulos and Luo (2006) designed a bound-constrained SDR (BC-SDR) method which has a special structure. Thus, compared to PI-SDR, BC-SDR makes fast implementations more favorable. Moreover, Mao et al. (2007) proposed a virtually-antipodal SDR (VA-SDR) method for any 4^q-QAM (where $q > 1$). The researchers also compared these SDR detectors and gave the relationship among them (Ma et al. 2009). However, since the relaxed SDP problem has a large problem size and the existing SDP solvers are slow for large size cases, the actual computation time for the SDR method is very high in practice.

Besides, many researchers have proposed some other methods for solving the MIMO problem. Sphere decoder method is considered as a classical method (Damen et al. 2003, Viterbo and Boutros (1999)). However, the complexity of this problem is exponential with respect to the problem size. Moreover, Goldberger and Leshem (2011) developed a new detection algorithm based on an optimal tree approximation in an unconstrained linear system. For the loop-free factor graph situation, this algorithm beats some other benchmark methods. Recently, a Lagrangian dual relaxation (LDR) for the MIMO problem was developed by Pan et al. (2014). This method finds the best diagonally regularized lattice decoder to approximate the ML detector. They proved that the corresponding LDR problem yields a duality gap no worse than that of the SDR method. Bunse-Gerstner et al. (2010) provided the first order necessary conditions for h_2-optimal model reduction for discrete MIMO systems. These conditions suggest a specific choice of interpolation data and a novel algorithm aiming for an h_2-optimal model reduction for MIMO systems. Moreover, Tian and Dang (2015) developed a canonical dual approach which finds either an optimal or approximate solution.

Note that, the branch-and-bound strategy is a typical global optimal method for solving a binary constrained quadratic programming (BQP) problem (Pardalos and Rodgers 1990). For this strategy, a tight lower bound estimator plays a key role

in cutting unnecessary branches and reduce the number of traversed nodes significantly. Thus, a well performed lower bound can improve the efficiency of the whole algorithm significantly. Besides, the computational efficiency of the lower bound estimator is also important. That is because a well performed but computationally expensive lower bound estimator will slow down the total efficiency of the algorithm. Therefore, if a lower bound estimator which is a little looser but more efficient, it can be a better choice for the BQP problem. Based on this idea, we develop a new branch-and-bound algorithm with a new efficient lower bound estimator in this paper.

The rest of this paper is arranged as follows. In Sect. 2, we reformulate the original problem into a $\{-1, 1\}$ constrained quadratic programming problem. Section 3 proposes a new lower bound estimator for the problem. Section 4 develops a branch-and-bound method for solving the binary constrained quadratic programming problem with a new convex reformulation. In Sect. 5, we show the comparison results for different methods by simulations.

2 Reformulation

Note that, problem (2) can be written in the following form:

$$\begin{array}{ll} \min & x^T Qx - f^T x + y^T y \\ s.t. & x_i \in \{\pm 1, \pm 3, ..., \pm u\}, \quad i = 1, ..., n, \end{array} \tag{3}$$

where $Q = H^T H$ is an $n \times n$ positive semidefinite matrix and $f = 2H^T y$ is a n-dimensional real vector. Since $y^T y$ is a fixed scalar, we can delete this term in the objective function in problem (3).

Let $y_i = \frac{x_i + u}{2}$ for $i = 1, ..., n$, it is easy to verify that $y_i \in \{0, 1, ..., u\}$. Let e denote the n-dimensional vector with all elements being 1. Then, problem (3) can be equivalently reformulated to the following problem:

$$\begin{array}{ll} \min & 4y^T Qy - (4ue^T Q + 2f^T)y + u^2 e^T Qe + uf^T e \\ s.t. & y_i \in \{0, 1, ..., u\}, i = 1, ..., n. \end{array} \tag{4}$$

Note that, once we get an optimal solution of problem (4), we can use $x = 2y - ue$ to get the corresponding optimal solution of problem (3). Since $u^2 e^T Qe + uf^T e$ is also a fixed scalar, we can ignore it in the objective function. Moreover, let $U = 4Q$, $d = 4uQ^T e + 2f$, we can write problem (4) as follows.

$$\begin{array}{l} \min\ y^T Uy - d^T y \\ s.t.\ y_i \in \{0, 1, ..., u\}, i = 1, ..., n. \end{array} \tag{5}$$

Let $Y = \{y \in \mathbb{N}^n \mid 0 \le y_i \le u\}$ denote the feasible domain of problem (5), where $\mathbb{N}$ denotes the set of integers. Set $t = \lfloor \log(u + 1) \rfloor$, where $\lfloor \cdot \rfloor$ denotes the biggest integer value which is smaller than or equal to that number. Then $t \ge 2$ and $u + 1 \ge 2^t$.

Now, we can define a new set $\mathcal{Z}$ as follows.

$$\mathcal{Z} = \{z \in \mathbb{N}^{(t+1)n} | \ z \in \{-1, 1\}^{(t+1)n}\}. \tag{6}$$

Note that, each element in set Z is a $(t+1)n$ dimensional vector with all elements being -1 or 1. Then, we have the following theorem to show the relationship between Y and Z.

Theorem 1 *The transformation* $y_i(z) = \sum_{j=1}^{t} 2^{j-2} z_{(j-1)n+i} + \frac{1}{2}(u+1-2^t) z_{tn+i} + \frac{u}{2}$ *is a linear and full mapping from* $\mathcal{Z}$ *to* Y.

Proof First, it is easy to see that $y_i(z)$ is a linear mapping from $\mathcal{Z}$ to Y.

Then we show that for any $z \in Z$, the corresponding y belongs to Y. Note that, $y_i(z)$ can also be written as $y_i(z) = \sum_{j=2}^{t} 2^{j-2} z_{(j-1)n+i} - 2^{t-1} z_{tn+i} + \frac{1}{2} z_i + \frac{u+1}{2} z_{tn+i} + \frac{u}{2}$. The first part $\sum_{j=2}^{t} 2^{j-2} z_{(j-1)n+i} - 2^{t-1} z_{tn+i}$ is always an integer, the possible fraction can only occur in the second part $\frac{1}{2} z_i + \frac{u+1}{2} z_{tn+i} + \frac{u}{2}$. However, it is easy to verify that this part is always an integer as $z_i, z_{tn+i} \in \{-1, 1\}$ for $u \geq 3, t \geq 2$. Moreover, for $i = 1, ..., n$, $\max\{y_i\} = \sum_{j=1}^{t} 2^{j-2} + \frac{u+1-2^t}{2} + \frac{u}{2} = u$ and $\min\{y_i\} = -\sum_{j=1}^{t} 2^{j-2} - \frac{u+1-2^t}{2} + \frac{u}{2} = 0$. Therefore, y_i is an integer between 0 and u, y belongs to Y.

Now we show that for an arbitrary integer $0 \leq y_i \leq u$, there exist $z_{(j-1)n+i} \in -1, 1$ for $j = 1, ..., t+1$ such that the transformation holds. If $0 \leq y_i \leq 2^t - 1$, then y_i can be written as $y_i = 2^{t-1} a_{t-1} + 2^{t-2} a_{t-2} + ... + 2a_1 + 1a_0$ for particular $a_i \in 0, 1$, $j = 0, ..., t-1$. Then, let $z_{jn+i} = 2a_j - 1$ for $j = 0, ..., t-1$ and $y_{tn+i} = -1$. Thus, $z_{(j-1)n+i} \in \{-1, 1\}$ for $j = 1, ..., t+1$. It is easy to verify that $\sum_{j=1}^{t} 2^{j-2} z_{(j-1)n+i} + \frac{u+1-2^t}{2} z_{tn+i} + \frac{u}{2} = \sum_{j=0}^{t-1} 2^j a_j = y_i$. If $2^t - 1 < y_i \leq u$, since $2^t \leq u + 1 \leq 2^{t+1}$, we have $-2^t \leq 2^t - (u+1) \leq 0$. Therefore, $0 \leq y_i + 2^t - (u+1) \leq 2^t - 1$. Then follow the similar way, we can write $y_i + 2^t - (u+1)$ as $y_i + 2^t - (u+1) = 2^{t-1} a_{t-1} + 2^{t-2} a_{t-2} + ... + 2a_1 + 1a_0$ for particular $a_i \in 0, 1$, $j = 0, ..., t-1$. This time, let $z_{jn+i} = 2a_j - 1$ for $j = 0, ..., t-1$ and $z_{tn+i} = 1$. Thus, $z_{(j-1)n+i} \in \{-1, 1\}$ for $j = 1, ..., t+1$. Consequently, we have $\sum_{j=1}^{t} 2^{j-2} z_{(j-1)n+i} + \frac{u+1-2^t}{2} z_{tn+i} + \frac{u}{2} = \sum_{j=0}^{t-1} 2^j a_j + (u+1) - 2^t = x_i$. Therefore, the transformation is also a full mapping from $\mathcal{Z}$ to Y. □

Observation The size of the reformulated problem is smaller than those problems derived by some traditional transformation methods (Verdú 1998) due to taking advantage of the special structure of the original problem.

Then, by using the transformation in problem (5), we can get the following problem:

$$\begin{array}{ll} \min & z^T M z - b^T z + c \\ s.t. & z_i \in \{-1, 1\}, i = 1, ..., (t+1)n, \end{array} \tag{7}$$

where

$$M = \begin{pmatrix} \frac{1}{4}U & \frac{1}{2}U & \cdots & 2^{t-3}U & \frac{1}{4}(u+1-2^t)U \\ \frac{1}{2}U & U & \cdots & 2^{t-2}U & \frac{1}{2}(u+1-2^t)U \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 2^{t-3}U & 2^{t-2}U & \cdots & 2^{2t-4}U & 2^{t-3}(u+1-2^t)U \\ \frac{1}{4}(u+1-2^t)U & \frac{1}{2}(u+1-2^t)U & \cdots & 2^{t-3}(u+1-2^t)U & \frac{1}{4}(u+1-2^t)^2U \end{pmatrix},$$

$$b = \begin{pmatrix} \frac{1}{4}[2d - uUe] \\ \frac{1}{2}[2d - uUe] \\ \vdots \\ 2^{t-3}[2d - uUe] \\ \frac{1}{4}(u+1-2^t)[2d - uUe] \end{pmatrix}, \quad c = \frac{1}{8}u^2e^TUe - \frac{u}{2}d^Te.$$

Moreover, we have the next result.

Theorem 2 *Solving problem (7) is equivalent to solving problem (5).*

Proof Assume y^* is an optimal solution of problem (5), then Theorem 1 indicates that there is feasible solution z^* of problem (7) such that both problems have the same objective value now. Thus, the optimal value of problem (7) is no more than that of problem (5). On the other hand, assume z^* is an optimal solution of problem (7), then by using the transformation in Theorem 1, we cant get a feasible solution y^* of problem (5) such that both problems have the same objective value now. Therefore, the optimal value of problem (5) is no more than that of problem (7). Hence, both problems have the same optimal value. And an optimal solution z^* of problem (7) can lead to an optimal solution y^* of problem (5). □

Let $r = (t+1)n$ and ignore the fixed scalar c, the corresponding problem can be finally written as follows.

$$\begin{array}{ll} \min & F(z) = z^TMz - b^Tz \\ s.t. & z_i \in \{-1, 1\}, i = 1, ..., r. \end{array} \tag{8}$$

3 New lower bound estimator

In this section, we will propose a new quadratic convex reformulation which can be used as a new lower bound estimator. Note that, if $z \in \{-1, 1\}^r$, we can represent these binary variables by the constraints $z_i^2 - 1 = 0, i = 1, ..., r$. Then the Lagrangian function for problem (P6) is

$$L(z, \lambda) = F(z) + \sum_{i=1}^{r} \lambda_i(x_i^2 - 1) = z^T[M + \mathrm{Diag}(\lambda)]z - b^Tz - e^T\lambda, \tag{9}$$

where $\lambda \in \mathbb{R}^r, e \in \mathbb{R}^r$ denotes the vector with all elements being 1 and $\mathrm{Diag}(\lambda)$ denotes a $r \times r$ diagonal matrix with λ_i being the i_{th} diagonal element. Let $M_\lambda = M + \mathrm{Diag}(\lambda)$, we can choose a proper λ to make $L(z, \lambda)$ convex for z. Then problem (P6) can be relaxed as the following problem:

$$\begin{aligned} P(\lambda) = \min\ & L(z, \lambda) \\ s.t.\ & z \in [-1, 1]^r, \end{aligned} \qquad \text{(LP)}$$

Note that, the optimal value of problem (LP) is a lower bound of problem (P6). Moreover, if M_λ is positive semidefinite, then problem (LP) is a convex programming problem which can be solved very efficiently. Therefore, this convex reformulation can be used as a lower bound estimator for each step in the branch-and-bound scheme. However, different λ lead to different qualities of the lower bounds. Hence, the next important issue is to determine a proper vector λ for the Lagrangian function $L(z, \lambda)$.

Billionnet and Elloumi (2007) solved the following problem to get the convex reformulation with the tightest lower bound.

$$\max_{M_\lambda \succeq 0} \min_{z \in [-1,1]^r} L(z, \lambda), \tag{10}$$

where $M_\lambda \succeq 0$ denotes that matrix M_λ is positive semidefinite.

Problem (10) can be reformulated as a semidefinite programming problem. And the numerical tests demonstrate its effectiveness in improving the lower bound in the branch-and-bound algorithm for the first several steps. However, as the algorithm continues, some variables of z have been fixed to -1 or 1, then this reformulation may not be the best one. Lu and Guo (2015) pointed out that the reformulation with the tightest continuous relaxation can not guarantee the further performance for a branch-and-bound method . Let $\mathcal{K} \subseteq \{1, ..., r\}$ denote the index subset. Suppose z_i is fixed to -1 or 1 for $i \in \mathcal{K}$ in the current branch node. Let $\mathcal{K}^c = \{1, ..., r\} - \mathcal{K}$.

$$\begin{aligned} P_{\mathcal{K}}(\lambda) = \min\ & L(z, \lambda) \\ s.t.\ & z_i = -1 \text{ or } 1, i \in \mathcal{K}, \\ & z_i \in [-1, 1], i \in \mathcal{K}^c. \end{aligned} \qquad \text{(RP)}$$

Then in order to get the tightest continuous relaxation bound for the current branch node, we need to solve the following problem:

$$\begin{aligned} \max\ & P_{\mathcal{K}}(z, \lambda_{\mathcal{K}}) \\ s.t.\ & M_{\mathcal{K}^c\mathcal{K}^c} + \mathrm{Diag}(\lambda_{\mathcal{K}^c}) \succeq 0, \\ & \lambda_i = 0, i \in \mathcal{K}, \end{aligned} \qquad \text{(SRP)}$$

where $M_{\mathcal{K}^c\mathcal{K}^c}$ denotes the sub-matrix of M with sub-rows and sub-columns in $\mathcal{K}^c$, $\lambda_{\mathcal{K}}$ and $\lambda_{\mathcal{K}^c}$ denote the sub-vectors of λ with elements in $\mathcal{K}$ and $\mathcal{K}^c$, respectively. Though problem (SRP) can provide the tightest lower bound for the continuous relaxation at the current branch, it is not wise to do it for each branch. That is because the computation cost on solving an SDR problem is about $\mathcal{O}(n^{3.5})$ (Vandenberghe and Boyd 1996). This is relatively high for a lower bound estimator. Therefore, if problem (SRP) is

recomputed at each iteration, the total computation cost is not affordable. Hence, we aim to get a lower bound estimator which has a continuous relaxation bound with high efficiency and preserving relatively good quality. In this way, we can achieve a higher total efficiency for solving the problem.

Instead of finding the best reformulation of $L(z, \lambda)$, we consider the average value of $L(z, \lambda)$ over $[-1, 1]^r$.

$$\begin{aligned} A(\lambda) &= \int_{z\in[-1,1]^r} L(z,\lambda)dz \\ &= \int_{z\in[-1,1]^r} z^T \mathrm{Diag}(\lambda) z - e^T\lambda + \int_{z\in[-1,1]^r} F(z)dz \\ &= -\frac{1}{3}e^T\lambda + \int_{z\in[-1,1]^r} F(z)dz. \end{aligned}$$

Then, we aim to find the vector λ^* which has the largest average value of $L(z, \lambda)$ over $z \in [-1, 1]^r$. Since $\int_{z\in[-1,1]^r} F(z)dz$ is a fixed constant, we can solve the following convex problem to get the corresponding optimal λ^*.

$$\begin{aligned} &\max\ -\tfrac{1}{3}e^T\lambda \\ &s.t.\ M + \mathrm{Diag}(\lambda) \succeq 0, \qquad \text{(AP)} \end{aligned}$$

Since problem (AP) focuses on the overall tightness of the convex reformulation, the corresponding lower bound estimator performs better in the deep branch nodes. Thus it should be more effective in a branch-and-bound method with continuous relaxation bounds and improve the total efficiency of the algorithm.

Moreover, the channel matrix in the MIMO problem is unchangeable, thus problem (P6) has a fixed quadratic term $z^T Mz$. Note that, the solution of (AP) only depends on the matrix M. Hence we may precompute the optimal solution of problem (AP) only once and store it in advance for further applications. It is worth pointing out that this property can accelerate the computing process of this branch-and-bound algorithm.

4 Algorithm

In this section, we design a branch-and-bound algorithm with our new lower bound estimator to solve the MIMO problem.

Branch-and-Bound Algorithm
Preparing Step: Use the linear transformation to reformulate the MIMO problem into a $\{-1, 1\}$ constrained quadratic programming problem.
Step 1 (Initialization Step): Solve problem (AP) to get the optimal λ^*. Construct a branch tree with an initial node and set $k = 1$. Solve the continuous relaxation $\min_{z\in[-1,1]^r} L(z, \lambda^*)$ to get the optimal solution z^* and lower bound l_0. Round the solution z^* to get the nearest feasible solution $\bar{z}$ in $\{-1, 1\}^r$ and the corresponding objective value $\bar{v}_0$. Set $l = l_0$, $u = \bar{v}_0$ and $z_{best} = \bar{z}$ as the current best lower bound, objective value and feasible solution, respectively.

Step 2: If $u - l < \epsilon$ or $k > k_{max}$, stop. Return z_{best} and u as the optimal solution and objective value, go to Step 5. Otherwise, find the leaf node with the smallest continuous relaxation bound among all leaf nodes. Choose the index $i^* = argmin_{i=1,...,r}((z_i^k)^2 - 1)$ and branch the current node into two child nodes with constraints $z_i = -1$ and $z_i = 1$. Set $k = k + 1$ and go to the next step.
Step 3: Solve the two continuous relaxation problems to get the optimal solutions $(z^k)^*$ and objective values l_k. If one of the $(z^k)^*$ is feasible for problem (P6), stop. Return the corresponding $(z^k)^*$ and l^k as the optimal solution and objective value of problem (P6). Otherwise, let l be the smallest lower bound at current iteration. Round the corresponding solution $(z^k)^*$ to get the nearest feasible solution $\bar{z}^k$ in $\{-1, 1\}^r$ and the corresponding objective value $\bar{v}_k$. If $\bar{v}_k < u$, let $u = \bar{v}_k$ and $z_{best} = \bar{z}^k$. Then go back to Step 2.
Step 4: Use the linear mapping $y_i(z) = \sum_{j=1}^{t} 2^{j-2} z_{(j-1)n+i} + \frac{1}{2}(u+1-2^t) z_{tn+i} + \frac{u}{2}$ to get the corresponding y values.

Note that, this branch-and-bound algorithm picks the leaf node with the smallest continuous relaxation bound in each iteration for further branching. And for unsatisfied solution, we choose the index $i^* \in \{1, ..., r\}$ in which z_{i^*} is the furthest one from the feasible domain $\{-1, 1\}$.

In real numerical tests, we can solve problem (AP) for some child nodes in the first several iterations. Then we can store and use the corresponding optimal $\lambda_{\mathcal{K}}^*$ in each branch for further branching. In this way, we can get some good lower bounds at the beginning and accelerate the whole computation process.

In the next section, we will compare this branch-and-bound algorithm with some other benchmark algorithms for the MIMO problem.

5 Comparisons by simulation

Some benchmark approximating methods are compared in the numerical tests, such as inexact ML sphere decoding (ML-SD) (Damen et al. 2003), semidefinite relaxation (SDR) (Sidiropoulos and Luo 2006), MMSE lattice decoding (MMSE-LD) (Wübben et al. 2011) and canonical dual approach (CDP) (Tian and Dang 2015). Moreover, based on the same reformulated problem (problem 8), we compared two branch-and-bound algorithms with the tightest lower bound estimator (10) and our proposed average value estimator (3), which are denoted as (BAB_T) and (BAB_A), respectively.

5.1 An intuitive example

Before the numerical tests, we first provide an intuitive example to show the advantage of (BAB_A) in comparison with (BAB_T). The example is a 10-dimensional problem as follows:

$$\begin{aligned} &\min\ z^T Q z - b^T z \\ &s.t.\ \ z_i \in \{-1, 1\}, i = 1, ..., 10, \end{aligned} \tag{11}$$

where

$$Q = \begin{pmatrix} 8 & 17 & 0 & 0 & -7 & -18 & -24 & 7 & 0 & 0 \\ 17 & -58 & -21 & 0 & -36 & 18 & 0 & 0 & -2 & 0 \\ 0 & -21 & -56 & 0 & 3 & 15 & -21 & 29 & 1 & 0 \\ 0 & 0 & 0 & -35 & 21 & -45 & 28 & 0 & -1 & 25 \\ -7 & -36 & 3 & 21 & -81 & 0 & 3 & -6 & 2 & 23 \\ -18 & 18 & 15 & -45 & 0 & 50 & -33 & -4 & 0 & 9 \\ -24 & 0 & -21 & 28 & 3 & -33 & 50 & 0 & 15 & 0 \\ 7 & 0 & 29 & 0 & -6 & -4 & 0 & 9 & 41 & 37 \\ 0 & -2 & 1 & -1 & 2 & 0 & 15 & 41 & -32 & 0 \\ 0 & 0 & 0 & 25 & 23 & 9 & 0 & 37 & 0 & 66 \end{pmatrix},$$

$$b = (-138, 100, 152, -294, 240, 272, -144, -326, 88, -564)^T.$$

For algorithm ($\mathrm{BAB_T}$), by using the corresponding reformulated objective function $L(z, \lambda_{\mathrm{T}}^*)$, we obtained a lower bound -2362.8 at the beginning. Then we branched x_6 at the first step and obtained a lower bound -2362.4 in the right child ($x_6 = 1$). Then, we branched x_7 at the second step and obtained a lower bound -2354.1 in the left child ($x_7 = -1$). Following the similar procedure until the 8th iteration, then we found the optimal solution $z^* = (-1, 1, 1, -1, 1, 1, -1, -1, 1, -1)^T$ of problem (11) with the optimal value -2337. Above all, we used eight iterations to complete algorithm ($\mathrm{BAB_T}$).

In comparison, for algorithm ($\mathrm{BAB_A}$), by using the corresponding reformulated objective function $L(z, \lambda_{\mathrm{A}}^*)$, we obtained a lower bound -2429.2 at the beginning. Then we branched x_6 at the first step and obtained a lower bound -2340.6 in the right child ($x_6 = 1$). Then we branched x_7 at the second step and obtained a lower bound -2337 in the left child ($x_7 = -1$). Note that, the approximated feasible integer solution of this node has an objective value which equals the current best lower bound, hence it is already the optimal solution of problem (11). Above all, we only used three iterations to complete algorithm ($\mathrm{BAB_A}$).

From this intuitive example, we can easily see that although the lower bound of $\mathrm{BAB_A}$ is looser than that of $\mathrm{BAB_T}$ at the beginning, it quickly catches up and completes in less iterations. Since the new lower bound estimator considers the overall tightness of the relaxation, it may obtain better lower bounds than the traditional branch-and-bound algorithm when some of the variables have been branched in the branch-and-bound scheme.

5.2 Numerical tests

Following the numerical tests in other papers, we also use several simulations to compare the performances of different methods. The testing data sets are generated as follows. The channel matrix H comprises independent and identically distributed (i.i.d.) elements drawn from a zero-mean normal distribution of unit variance. The symbol vector x is elementwise independent and identically uniformly distributed. Besides, v is a white Gaussian noise with zero mean and variance σ_v^2. Note that, the

signal-to-noise ratio (SNR) is defined as $\frac{E(\|Hx\|^2)}{E(\|v\|^2)} = n\frac{\sigma_x^2}{\sigma_v^2}$, where σ_x^2 is the variance of the elements of x. To measure the performances of different methods, three metrics are used in this paper: symbol error rate (SER), average computational time and worst-case computational time.

For the branch-and-bound algorithms, the accuracy criterion ϵ and maximum iteration number k_{max} are set to be 10^{-5} and 50% of the size of the problem, respectively. Moreover, we compute λ_k for each brand in the first p_{th} iterations. Here, p is 5% of the size of the problem. It's easy to see that, for the same problem, the bigger the value of k_{max} and p are, the more computational time the algorithms should cost while the more accurate result the algorithms may get. Therefore, we need to choose a good balance point with which the algorithms can get a relatively accurate result while keep a relatively high efficiency. After many numerical tests, we choose 50% and 5% as the parameters pin this paper.

Like other papers, we use two QAM order sizes as 16-QAM and 64-QAM in the test. Moreover, four different problem sizes are tested as $(m_1, n_1) = (8, 8)$, $(m_2, n_2) = (16, 16)$,$(m_3, n_3) = (64, 64)$ and $(m_4, n_4) = (128, 128)$. To get reasonable statistical results, 100 samples are tested for each SNR value in every case. All the simulations are implemented using MATLAB 7.9.0 on a computer with Intel Core i5 CPU 3.3 Ghz and 4G memory. Moreover, the solver "cvx" (Grant and Boyd 2010) is incorporated in solving the SDP problems. Note that, since the existing semi-definite programming (SDP) solvers are unable to solve high-dimensional problems, we don't compute the result for the SDR method in the following three cases: 64 64-QAM, 128 16-QAM, 128 64-QAM. Moreover, for each case, the lower bound of our computational accuracy is 10^{-4}. Therefore, we don't test the case where its optimal solution achieves this accuracy.

Figures 1, 2, 3, and 4 plot the average SER performances versus SNR with problem sizes $(m, n) = (8, 8), (16, 16), (32, 32)$ and $(128, 128)$, respectively. Tables 1 and 2 provide the average computational times in seconds (here each result is the average value of all samples and SER values). Note that, the notation "8 16-QAM"

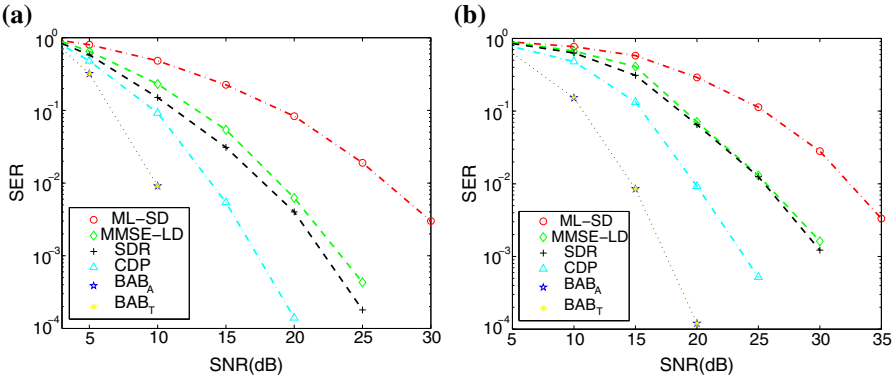

Fig. 1 Comparison of symbol error rate (SER) versus signal-to-noise ratio (SNR) for the new algorithm BAB_A and other methods, $(m, n) = (8, 8)$

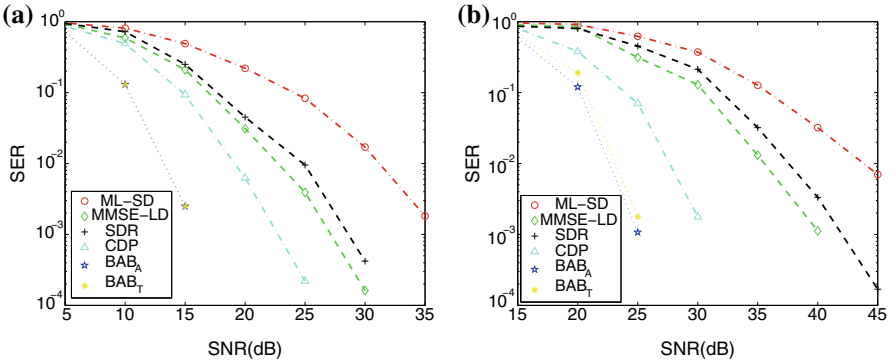

Fig. 2 Comparison of symbol error rate (SER) versus signal-to-noise ratio (SNR) for the new algorithm $\mathrm{BAB_A}$ and other methods, $(m, n) = (16, 16)$

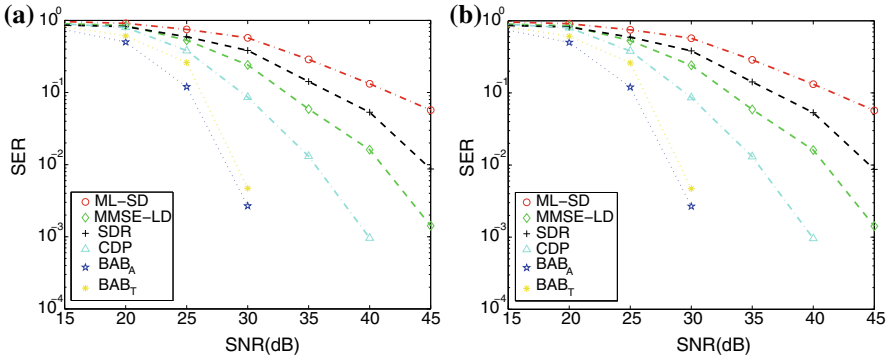

Fig. 3 Comparison of symbol error rate (SER) versus signal-to-noise ratio (SNR) for the new algorithm $\mathrm{BAB_A}$ and other methods, $(m, n) = (64, 64)$

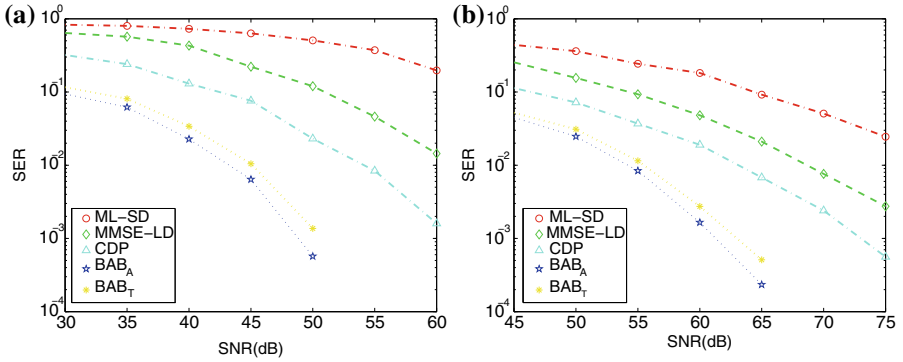

Fig. 4 Comparison of symbol error rate (SER) versus signal-to-noise ratio (SNR) for the new algorithm $\mathrm{BAB_A}$ and other methods, $(m, n) = (128, 128)$

Table 1 Average computational times for the new algorithm BAB_A and other methods with different problem sizes in 16-QAM

Methods	Cases			
	8 16-QAM	16 16-QAM	64 16-QAM	128 16-QAM
ML-SD	4.13	14.88	51.05	136.46
SDR	3.37	14.04	175.29	–
MMSE-LD	5.47	19.56	57.37	150.21
CDP	3.92	14.38	46.78	127.48
BAB_T	7.49	20.03	82.95	185.33
BAB_A	6.34	16.79	55.72	145.23

Table 2 Average computational times for the new algorithm BAB_A and other methods with different problem sizes in 64-QAM

Methods	Cases			
	8 64-QAM	16 64-QAM	64 64-QAM	128 64-QAM
ML-SD	7.30	18.57	78.32	146.57
SDR	5.91	19.36	–	–
MMSE-LD	10.29	27.01	81.48	153.72
CDP	7.56	20.43	69.58	130.40
BAB_T	14.88	31.02	100.75	237.24
BAB_A	11.45	24.72	79.37	183.86

Table 3 Average computational times for the new algorithm BAB_A and other methods with various signal-to noise ratio (SNR) on 8-64 QAM

Methods	SNR(dB)						
	5	10	15	20	25	30	35
ML-SD	8.32	8.02	7.44	7.21	6.85	6.71	6.58
SDR	5.92	5.98	5.86	5.92	5.91	5.88	–
MMSE-LD	11.43	11.05	10.68	10.01	9.43	9.11	–
CDP	7.59	7.89	7.91	7.16	7.23	–	–
BAB_T	15.84	15.04	14.17	14.46	–	–	–
BAB_A	12.32	11.85	11.37	10.26	–	–	–

denotes the 16-QAM problem whose size is $(m, n) = (8, 8)$. Moreover, for two scenarios (8-16 QAM and 16-64 QAM), we provide the information about the average computational times for each SER value (here each result is the average value of all samples but with a fixed SER value) in Tables 3 and 4, respectively. Besides, we also provide the computational times for the worst-case we met in the test for each case in Tables 5 and 6. Note that, in all tables, "−" denotes there is no result for this case.

Table 4 Average computational times for the new algorithm BAB_A and other methods with various signal-to noise ratio (SNR) on 64-16 QAM

Methods	SNR(dB)						
	15	20	25	30	35	40	45
ML-SD	54.01	52.65	51.90	51.14	50.75	49.37	47.51
SDR	180.63	174.28	171.37	177.45	177.68	175.33	170.27
MMSE-LD	61.47	62.05	57.35	54.81	56.39	56.26	53.29
CDP	50.01	45.26	48.30	46.11	45.73	45.27	–
BAB_T	85.04	82.35	83.81	80.60	–	–	–
BAB_A	57.32	57.20	54.01	54.36	–	–	–

Table 5 Computational times for the new algorithm BAB_A and other methods in the worst-cases on 16-QAM

Methods	Cases			
	8 16-QAM	16 16-QAM	64 16-QAM	128 16-QAM
ML-SD	6.28	22.75	78.11	183.79
SDR	4.01	16.85	197.48	–
MMSE-LD	6.33	27.61	75.67	194.65
CDP	8.89	31.83	82.46	217.32
BAB_T	10.42	27.46	97.38	213.28
BAB_A	8.85	26.49	89.53	197.46

Table 6 Computational times for the new algorithm BAB_A and other methods in the worst-cases on 64-QAM

Methods	Cases			
	8 64-QAM	16 64-QAM	64 64-QAM	128 64-QAM
ML-SD	11.35	26.27	100.74	225.20
SDR	7.64	27.31	–	–
MMSE-LD	16.08	38.18	119.37	242.36
CDP	15.21	39.84	128.39	240.55
BAB_T	20.44	45.08	165.23	302.41
BAB_A	17.26	38.61	147.46	287.34

From these figures and tables, we can see that the branch-and-bound algorithms have much better performances than other state-of-the-art methods in all situations. Though the computational times of the branch-and-bound algorithms are relatively a bit longer than other methods, considering the good quality of the performances, the branch-and-bound algorithms are very competitive for the MIMO problems, especially for the situations with high accuracy requirement. Moreover, under the same branch-and-bound scheme, (BAB_A) beats (BAB_T) in terms of accuracy and efficiency.

Therefore, (BAB_A) can be a very effective and efficient tool in solving the MIMO problems.

6 Conclusion

In this paper, we designed a very effective algorithm while preserving a relatively high efficiency to solve the high-order MIMO problem. First, we reformulated the MIMO problem into a $\{-1, 1\}$ constrained quadratic programming problem. Then, we managed to get a good lower bound estimator by the quadratic convex reformulation. Considering the balance between tightness and computation cost, we chose the convex reformulation with the maximized average objective value over the relaxed convex domain. Based on that, we designed a branch-and-bound algorithm. To see the effectiveness and efficiency of the algorithm, we compare it with some benchmark methods. The simulation results demonstrate that our branch-and-bound algorithm achieves a very good accuracy while preserving a relatively high efficiency. Therefore, this algorithm has a big potential to be applied in some real applications.

Acknowledgements Tian's research has been supported by the Chinese National Science Foundation #11401485 and #71331004.

References

Billionnet A, Elloumi S (2007) Using a mixed integer quadratic programming solver for the unconstrained quadratic 0–1 problem. Math Program 109:55–68

Bunse-Gerstner A, Kubaliń D, Vossen G, Wilczek D (2010) h_2-norm optimal model reduction for large scale discrete dynamical MIMO systems. J Comput Appl Math 233:1202–1216

Damen M, El Gamal H, Caire G (2003) On maximum-likelihood detection and the search for the closest lattice point. IEEE Trans Inf Theory 49:2389–2402

Fincke U, Pohst M (1985) Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. Math Computat 44:463–471

El Gamal H, Caire G, Damen M (2004) Lattice coding and decoding achieve the optimal diversity-multiplexing tradeoff of MIMO channels. IEEE Trans Inf Theory 50:968–985

Goldberger J, Leshem A (2011) MIMO detection for high-order QAM based on a Gaussian tree approximation. IEEE Trans Inf Theory 57:4973–4982

Grant M, Boyd S (2010) CVX: matlab Software for Disciplined Programming. Version 1.2. http://cvxr.com/cvx

Jaldén J, Elia P (2010) DMT optimality of LR-aided linear decoders for a general class of channels, lattice designs, and system models. IEEE Trans Inf Theory 56:4765–4780

Jaldén J, Ottersten B (2005) On the complexity of sphere decoding in digital communications. IEEE Trans Signal Process 53:1474–1484

Kisialiou M, Luo Z (2005) Performance Analysis of Quasi-Maximum-Likelihood Detector Based on Semi-definite Programming. In: Proceedings of the IEEE International Conference on Acoustics Speech and Signal Process, vol III, pp 433–436

Liu S, Ling C, Stehlé D (2011) Decoding by sampling: A randomized lattice algorithm for bounded-distance decoding. IEEE Trans Inf Theory 57:5933–5945

Lu C, Guo X (2015) Convex reformulation for binary quadratic programming problems via average objective value maximization. Optim Lett 9:523–535

Luzzi L, Stehlé D, Ling C (2013) Decoding by embedding: Correct decoding radius and DMT optimality. IEEE Trans Inf Theory 59:960–2973

Ma W, Davidson T, Wong K, Ching P (2004) A block alternating likelihood maximization approach to multiuser detection. IEEE Trans Signal Process 52:2600–2611

Ma W, Su C, Jaldém J, Chang T, Chi C (2009) The equivalence of semidefinite relaxation MIMO detectors for higher-order QAM. IEEE J Select Top Signal Process 3:1038–1052
Mao Z, Wang X, Wang X (2007) Semidefinite programming relaxation approach for multiuser detection of QAM signals. IEEE Trans Wire Commun 6:4275–4279
Pan J, Ma W, Jaldém J (2014) MIMO detection by Lagrangian dual maximum-likilihood relaxation: Reinterpreting regularized lattice decoding. IEEE Trans Signal Process 62:511–524
Pardalos P, Rodgers G (1990) Computational aspects of a branch and bound algorithm for quadratic zero-one programming. Computing 45:131–144
Sidiropoulos N, Luo Z (2006) A semidefinite relaxation approach to MIMO detection for higher-order constellations. IEEE Signal Process lett 13:525–528
Singh A, Elia P, Jaldén J (2012) Achieving a vanishing SNR gap to exact lattice decoding at a subexponential complexity. IEEE Trans Inf Theory 58:3692–3707
Taherzadeh M, Khandani A (2010) On the limitations of the naive lattice decoding. IEEE Trans Inf Theory 56:4820–4826
Tan P, Rasmussen L (2001) The application of semidefinite programming for detection in CDMA. IEEE J Select Areas Commun 19:1442–1449
Tian Y, Dang JF (2015) MIMO detection for high order QAM by canonical dual approach. J Appl Math. doi:10.1155/2015/201369
Tse D, Viswanath P (2005) Fundamentals of wireless communication. Cambridge University Press, Cambridge
Vandenberghe L, Boyd S (1996) Semidefinite programming. SIAM Rev 38:49–95
Verdú S (1998) Multiuser detectection. Cambridge University Press, Cambridge
Viterbo E, Boutros J (1999) A universal lattice code decoder for fading channels. IEEE Trans Inform Theory 45:1639–1642
Wang Z, Fang SC, Gao D, Xing W (2008) Global extremal conditions for multi-integer quadratic programming. J Ind Manage Optim 4:213–225
Wiesel A, Eldar Y, Shamai S (2005) Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels. IEEE Signal Process Lett 13:525–528
Wübben D, Seethaler D, Jaldén J, Matz G (2011) Lattice reduction. IEEE Signal Process Mag 28:70–91