IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 28, 2018, accepted September 26, 2018, date of publication October 16, 2018, date of current version October 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875022

Fast Mixed Integer Quadratic Programming for Sparse Signal Estimation

SANGJUN PARK AND HEUNG-NO LEE, (Senior Member, IEEE)
School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea

Corresponding author: Heung-No Lee (heungno@gist.ac.kr)

This work was supported by the National Research Foundation of Korea through the Korean Government (MSIP) under Grant NRF-2018R1A2A1A19018665.

ABSTRACT It has been recently shown that the l_0-norm problem can be reformulated into a mixed integer quadratic programming (MIQP) problem. CPLEX, a commercial optimization software package that can solve integer programming problems, is used to find the global solution to this MIQP problem for sparse signal estimation. However, CPLEX uses an exhaustive approach to search a feasible space to this MIQP problem. Thus, its running time grows exponentially as the problem dimension grows. This means that CPLEX quickly becomes computationally intractable for higher dimension problems. In this paper, we aim to propose a fast first-order-type method for solving this MIQP problem based on the alternating direction method. We conduct extensive simulations to demonstrate that: 1) our method is used to estimate a sparse signal by solving this problem and 2) our method is computationally tractable for problem dimensions up to the order of 1 million.

INDEX TERMS Alternating direction method, compressed sensing, mixed integer quadratic program.

I. INTRODUCTION

Compressed sensing [1] has attracted attention because it allows for the acquisition of signal samples at a rate lower than the Nyquist rate. The theory of compressed sensing is built under a sparsity assumption that an n-dimensional signal $\mathbf{x}$ can be sparsely represented using a few non-zero coefficients in a basis. This sparse signal is sampled to yield an m-dimensional measurement vector $\mathbf{b} = \mathbf{F}\mathbf{x} + \mathbf{n}$, where $\mathbf{F}$ is an $m \times n$ sensing matrix and $\mathbf{n}$ is an $m \times n$ noise vector. Since $m < n$, the problem of estimating $\mathbf{x}$ is ill-posed. However, the theory shows that $\mathbf{x}$ is reliably estimated by solving the l_0-norm problem:

$$\min_{\mathbf{x}} \tau \left\|\mathbf{x}\right\|_0 + 2^{-1} \left\|\mathbf{b} - \mathbf{F}\mathbf{x}\right\|_2^2, \tag{1}$$

where τ is a positive regularization value. In (1), the l_0-norm function is non-convex and discontinuous. Indeed, (1) is known to be NP-hard. Instead of solving (1), researches aim to solve an l_1-norm problem. This problem is formulated by relaxing the l_0-norm function in (1) and is given by

$$\min_{\mathbf{x}} \tau \left\|\mathbf{x}\right\|_1 + 2^{-1} \left\|\mathbf{b} - \mathbf{F}\mathbf{x}\right\|_2^2. \tag{2}$$

Candes and Tao [1] have proved that a solution to (2) is equivalent to a solution to (1) if $\mathbf{F}$ satisfies a *restricted isometry constant* (RIC) condition. Many l_1-norm-based methods have been proposed to solve (2). The earliest method is $l_1 l_s$ [2]. This method is based on an interior point technique and can estimate $\mathbf{x}$ from a small number of iterations. In each of iteration, $l_1 l_s$ solves a linear equation system expressed in a matrix-vector product form. The matrix in each system changes as the iteration passes. Thus, factorization methods such as the LU decomposition and the QR decomposition can be used to reduce the computations for solving this system. However, solving multiple linear equation systems can be still burdensome. This makes its computational cost too high for high-dimensional $\mathbf{x}$. Then, *gradient projection sparse recovery* [3], homotopy [4], split-Bregman [5] and *your algorithms for* l_1 (YALL1) [6] have been proposed to solve (2). These are first-order-type methods that do not require matrix-inversions in all iterations. This implies that they are computationally tractable to estimate high-dimensional $\mathbf{x}$. But, there are known problems on (2). First, the l_1-norm function yields a biased estimation for large non-zero magnitudes, while the l_0-norm function considers all non-zero magnitudes equally [7]. Second, if $\mathbf{F}$ does not satisfy the RIC condition, – either m is small or the elements of $\mathbf{F}$ are correlated – then a solution to (2) is sub-optimal [8].

In the literature, l_0-norm-based methods such as *iterative hard thresholding* (IHT) [9], variants of IHT [10]–[12], and *mean doubly augmented Lagrangian* (MDAL) [13] have been proposed to solve (1). Dong and Zhu [12] have shown that their method is superior to homotopy [4]. Dong and Zhang [13] have shown that MDAL restores images with higher quality than those recovered by split-Bregman [5]. These results in [12] and [13] suggest that more accurate sparse signal estimation is conducted using the l_0-norm function rather than the l_1-norm function.

Recently, Bourguignon *et al.* [14] have proposed an novel approach to solve (1). This approach aims to find an estimate for $\mathbf{x}$ and the positions of the non-zero elements of $\mathbf{x}$, i.e., the support set. From (1), they have made a mixed integer quadratic programming (MIQP) problem:

$$\begin{aligned} &\min_{\mathbf{u}\in\{0,1\}^n,\mathbf{x}\in\mathbb{R}^n} \tau \mathbf{1}_n^T\mathbf{u} + 2^{-1}\|\mathbf{b}-\mathbf{F}\mathbf{x}\|_2^2 \\ &\text{subject to } |\mathbf{x}| \le M\mathbf{u} \end{aligned} \tag{3}$$

where the binary vector $\mathbf{u}$ indicates the support set and M is a positive value. For (3), M can be known in practical contexts. For example, if $\mathbf{x}$ is an 8-bit greyscale image, M is set to be 255. Bertsimas *et al.* [15] have proposed methods to estimate upper bounds on M if both $\mathbf{F}$ and $\mathbf{b}$ are known. Bourguignon *et al.* [14] used CPLEX [16] to solve (3) and demonstrated that CPLEX is superior to IHT [9] for sparse signal estimation. According to explanations in [14], this result is because CPLEX exhaustively searches for a whole feasible space to find the global solution to (3) while IHT finds a local solution to (1).

CPLEX [16] is a commercial solver which can be used to solve MIQP problems. Then, CPLEX is implemented based on a branch-and-cut method [30] that is a combination of a cutting plane method [31] with a branch-and-bound method [17]. As noted in [22], the branch-and-cut method has non-polynomial computational costs in the worst-case and can be troublesome to solve MIQP problems with large variables. This implies the computational intractability of using CPLEX in solving integer programming problems with large variables. In Section V, we empirically confirm this computational intractability.

In this paper, we aim to propose a fast method based on the *alternating direction method* (ADM) for solving (3). We analyze the computational cost per iteration of the proposed method, referred to as ADM-MIQP. According to this result, we can show that ADM-MIQP is a first-order-type method. We evaluate the quality of its solution using metrics defined as follows.

First, we define *support set error* (SSE) as

$$d_1\left(\mathbf{u},\hat{\mathbf{u}}\right) := k^{-1}\sum_{i=1}^{n}\left\|\mathbf{u}\left(i\right)-\hat{\mathbf{u}}\left(i\right)\right\|_1, \tag{4}$$

where $\hat{\mathbf{u}}$ is a solution to (3) and $\mathbf{u}$ is constructed from

$$\begin{aligned} \mathbf{u}\left(i\right) &= 0 \quad \text{if } i \notin \mathcal{I} \\ \mathbf{u}\left(i\right) &= 1 \quad \text{if } i \in \mathcal{I}, \end{aligned}$$

where $\mathcal{I}$ is the support set to be detected. Second, we define *mean square error* (MSE) as

$$d_2\left(\mathbf{x},\tilde{\mathbf{x}}\right) := n^{-1}\left\|\mathbf{x}-\tilde{\mathbf{x}}\right\|_2^2, \tag{5}$$

where $\mathbf{x}$ is an original signal and $\tilde{\mathbf{x}}$ is an estimate of $\mathbf{x}$. Then, we compare ADM-MIQP with both YALL1 and MDAL in terms of SSE and MSE. We observe the following:

- ADM-MIQP significantly surpasses both MDAL and YALL1 in terms of both MSE and SSE.
- ADM-MIQP exhibits good estimation performance close to the performance of ORACLE that knows support set *a priori*.
- ADM-MIQP is computationally tractable for solving (3) with the problem dimension up to the order of one million.
- ADM-MIQP exhibits a computational cost given by $\mathrm{O}\left(n^{1.3}\right)$ in our simulations.

The rest of this paper is organized as follows. Section II gives notations used in this paper and a summary about ADM. Section III elucidates the derivation and computational costs associated with ADM-MIQP. Also, Section III gives results of comparison between our proposed approach with that of [22] for solving our problem. Section IV gives simulation studies and shows the superiority of ADM-MIQP compared to other ADM-based methods [6], [13]. Section V gives conclusions of this paper.

II. PRELIMINARIES

A. NOTATIONS

We present some notations frequently used in this paper and their meanings in Table 1.

TABLE 1. **Summary of the notations.**

Notation	Definitions
$\mathbf{1}_n$	The $n \times 1$ vector of ones
$\mathbf{0}_n$	The $n \times 1$ vector of zeros
$\mathbf{I}_n$	The $n \times n$ identity matrix
$\mathbf{O}_n$	The $n \times n$ matrix of zeros
$\mathbf{f}(i)$	The i^{th} element of a column vector $\mathbf{f}$
$\mathbf{f}[i{:}l]$	The column vector constructed by collecting elements of a column vector $\mathbf{f}$ from the i^{th} element to l^{th} element
$\mathbf{F}_{\mathcal{I}}$	The matrix is constructed by collecting columns of $\mathbf{F}$ corresponding to indices of a set $\mathcal{I}$

B. ALTERNATION DIRECTION METHOD (ADM)

A branch and bound method [17] finds the global solution to a MIQP problem. But, since this method has non-polynomial computational costs, it is computationally intractable for solving MIQP problems with large variables. We turn instead to ADM for solving the MIQP problem (3). In this subsection, we introduce ADM and provide its recent results.

It is well-known that ADM is a powerful technique for solving a large-scale convex problem. ADM involves the following steps: *i*) ADM splits this problem into sub-problems and *ii*) solves alternatively these sub-problems until conditions are satisfied. ADM is then proven to find the global solution to this problem as the iteration continues [18], [19]. As the number of iterations approaches infinity, the solution generated by ADM converges to an optimal solution which satisfies the *Karush-Kuhn-Tucker* conditions to a convex problem.

Recently, ADM has been empirically shown to be a powerful technique to find accurate solutions to integer programming problems [20]–[22]. Yadav *et al.* [20] have used ADM to solve an image separation problem that can be modeled as a binary quadratic programming. Souto and Dinis [21] have then solved a signal decoding problem modeled as an integer quadratic programming with an equality constraint using ADM. Last, Takapoui *et al.* [22] have solved problems modeled as MIQPs with an equality constraint, and shown that ADM could be greatly faster than a commercial integer programming method. We are motivated to derive a computationally tractable and accurate method to solve (3) using ADM, inspired by these results in [20]–[22].

III. SPARSE SIGNAL ESTIMATION VIA MIQP PROBLEM

The MIQP problem (3) has an inequality constraint and this constraint can be formulated into an equality constraint. Thus, we can use the approach of [22] to solve (3) by taking a further formulation. But, no explicit discussion on how this approach can be used to solve a MIQP problem which has an inequality constraint was given in [22]. In the sub-section III.C, we derive an algorithm based on the approach of [22] for solving (3). We call it *Takapoui's Algorithm with Inequality Constraint* (TAIC). We then compare ADM-MIQP with TAIC with respect to the computational cost per iteration. We show that ADM-MIQP requires much less computation per iteration than TAIC does.

A. DERIVATION OF ADM-MIQP

It is convenient to solve a single minimization problem rather than a joint minimization problem. To this end, we define

$$\mathbf{d} = \begin{bmatrix} \mathbf{x}^T & \mathbf{u}^T \end{bmatrix}^T \in \mathbb{R}^n \times \{0, 1\}^n ,$$

which is nonconvex. Then, (3) is reformulated into

$$\min_{\mathbf{d}} 2^{-1}\mathbf{d}^T\mathbf{Q}\mathbf{d} + \mathbf{q}^T\mathbf{d} \quad \text{subject to } \mathbf{A}\mathbf{d} \leq \mathbf{0}_{2n},$$

where

$$\mathbf{Q} = \begin{bmatrix} \mathbf{F}^T\mathbf{F} & \mathbf{O}_n \\ \mathbf{O}_n & \mathbf{O}_n \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} -\mathbf{\Phi}^T\mathbf{b} \\ \tau\mathbf{1}_n \end{bmatrix},$$

and

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_n & -M\mathbf{I}_n \\ -\mathbf{I}_n & -M\mathbf{I}_n \end{bmatrix}.$$

We then define a nonnegative vector $\mathbf{z}$. Then, we obtain

$$\begin{aligned} &\min_{\mathbf{d},\mathbf{z}} 2^{-1}\mathbf{d}^T\mathbf{Q}\mathbf{d} + \mathbf{q}^T\mathbf{d} + I_{\mathcal{X}}(\mathbf{z}) \\ &\text{subject to } \mathbf{A}\mathbf{d} + \mathbf{z} = \mathbf{0}_{2n} \end{aligned} \qquad (6)$$

where $I_{\mathcal{X}}(\mathbf{z})$ is an indicator function of $\mathcal{X} := \{\mathbf{z} \,|\, \mathbf{z} \geq \mathbf{0}_{2n}\}$, i.e., $I_{\mathcal{X}}(\mathbf{z}) = 0$ for $\mathbf{z} \in \mathcal{X}$ and $I_{\mathcal{X}}(\mathbf{z}) = \infty$ for $\mathbf{z} \notin \mathcal{X}$. We apply ADM into (6) to obtain

$$\begin{aligned} \mathbf{d}_{t+1} &= \arg\min_{\mathbf{d}} 2^{-1}\mathbf{d}^T\left[\mathbf{Q} + \rho\mathbf{A}^T\mathbf{A}\right]\mathbf{d} + \mathbf{q}_1^T\mathbf{d}, \\ \mathbf{z}_{t+1} &= \arg\min_{\mathbf{z}} I_{\mathcal{X}}(\mathbf{z}) + \rho 2^{-1}\mathbf{z}^T\mathbf{z} + (\rho\mathbf{A}\mathbf{d}_{t+1} - \boldsymbol{\lambda}_t)^T\mathbf{z}, \\ \boldsymbol{\lambda}_{t+1} &= \boldsymbol{\lambda}_t - \rho\left(\mathbf{A}\mathbf{d}_{t+1} + \mathbf{z}_{t+1}\right), \end{aligned} \qquad (7)$$

where $\mathbf{q}_{1,t} = \mathbf{q} - \mathbf{A}^T(\boldsymbol{\lambda}_t - \rho\mathbf{z}_t)$, $\boldsymbol{\lambda}$ is the dual variable, and $\rho > 0$ is a penalty value. The sub-problem on $\mathbf{d}$ is an MIQP. Thus, solving this problem is difficult, but we separate it into a pair of problems in terms of $\mathbf{x}$ and $\mathbf{u}$, respectively:

$$\begin{aligned} \mathbf{x}_{t+1} &= \arg\min_{\mathbf{x}} 2^{-1}\mathbf{x}^T\left(\mathbf{F}^T\mathbf{F} + 2\rho\mathbf{I}_n\right)\mathbf{x} + \mathbf{q}_{1,t}^T[1:n]\,\mathbf{x}, \\ \mathbf{u}_{t+1} &= \arg\min_{\mathbf{u}} \rho M^2\mathbf{u}^T\mathbf{u} + \mathbf{q}_{1,t}^T[n+1:2n]\,\mathbf{u}. \end{aligned}$$

Since the sub-problem on $\mathbf{x}$ has a quadratic objective function, we have an analytic closed-form solution:

$$\begin{aligned} \mathbf{x}_{t+1} &= -\left(\mathbf{F}^T\mathbf{F} + 2\rho\mathbf{I}_n\right)^{-1}\mathbf{q}_{1,t}[1:n] \\ &= \left(\mathbf{F}^T\mathbf{D}\mathbf{F}\mathbf{q}_{1,t}[1:n] - \mathbf{q}_{1,t}[1:n]\right)/(2\rho) \end{aligned} \qquad (8)$$

where the second equality is due to the Woodbury formula [23] and $\mathbf{D} := \left(\mathbf{F}\mathbf{F}^T + 2\rho\mathbf{I}_m\right)^{-1}$. The sub-problem on $\mathbf{u}$ is a binary quadratic programming. Since $\mathbf{u}^T\mathbf{u} = \mathbf{1}_n^T\mathbf{u}$, we have

$$\mathbf{u}_{t+1} = \arg\min_{\mathbf{u}} \left(\rho M^2\mathbf{1}_n + \mathbf{q}_{1,t}[n+1:2n]\right)^T\mathbf{u},$$

which has an analytic closed-form solution as follows:

$$\begin{aligned} \mathbf{u}_{t+1}(i) &= 0 \quad \text{if } \eta_t \geq 0 \\ \mathbf{u}_{t+1}(i) &= 1 \quad \text{if } \eta_t < 0 \end{aligned} \qquad (9)$$

where $\eta_t = \rho M^2 + \mathbf{q}_{1,t}(n+i)$. The sub-problem on $\mathbf{z}$ is solved to yield a solution:

$$\mathbf{z}_{t+1} = \max\left(\mathbf{0}_{2n}, \boldsymbol{\lambda}_t/\rho + \begin{bmatrix} M\,\mathbf{u}_{t+1} - \mathbf{x}_{t+1} \\ M\,\mathbf{u}_{t+1} + \mathbf{x}_{t+1} \end{bmatrix}\right), \qquad (10)$$

where "max" operation is performed element-wise.

In summary, we have formulated (6) from (3) by adding the non-negative vector and the indicator function. We then have applied ADM into (6) to produce the iterations given in (7). We next provided analytic solutions to these sub-problems. Then, we summarized ADM-MIQP in Table 2.

In [18] and [19], it has been proved that for any positive penalty value ρ, ADM can find the global solution to a convex problem. The penalty value only affects the convergence speed, not the quality of the solution. Researchers have discussed how this penalty value can be chosen to improve the speed [18], [19]. However, our problem (3) is non-convex due to the non-convex variable $\mathbf{d}$. In the literature, there are no convergence studies for non-convex problems with non-convex variables, to the best of our knowledge. It is difficult to find convergence conditions for the penalty value in the problem (3) that is being solved using ADM-MIQP. Afonso *et al.* [32] have solved (1) using their own algorithm

TABLE 2. The pseudo code of ADM-MIQP.

```
Parameters: $\mathbf{x}_0 = \mathbf{0}_n, \mathbf{u}_0 = \mathbf{0}_n, \mathbf{z}_0 = \mathbf{0}_n, \boldsymbol{\lambda}_0 = \mathbf{0}_{2n}$
            $\mathbf{F}, \mathbf{b}, \mathbf{A}, \rho, \tau, M, \varepsilon, \mathrm{MaxIter}$

Step 1:   set $\mathbf{D} := \left(\mathbf{F}\mathbf{F}^T + 2\rho\mathbf{I}_m\right)^{-1}$
Step 2:   for $t = 0, 1, 2, \ldots$ maxIter
Step 3:       update $\mathbf{x}_{t+1}$ by (8).
Step 4:       update $\mathbf{u}_{t+1}$ by (9).
Step 5:       update $\mathbf{z}_{t+1}$ by (10).
Step 6:       update $\mathbf{d}_{t+1} = \left[\mathbf{x}_{t+1}^T \quad \mathbf{u}_{t+1}^T\right]^T$.
Step 7:       update $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t - \rho\left(\mathbf{A}\mathbf{d}_{t+1} + \mathbf{z}_{t+1}\right)$.
Step 8:       If $\frac{\|\mathbf{d}_{t+1} - \mathbf{d}_t\|_2}{\|\mathbf{d}_{t+1}\|_2} \le \varepsilon$, then go to Step 10
Step 9:   end for
Step 10:  set $\tilde{\mathcal{I}} = \left\{i \middle| \mathbf{u}_t(i) = 1\right\}$.
Step 11:  set $\mathbf{x}_{sol} = \begin{cases} \mathbf{x}(i) = 0 & \text{if } i \notin \tilde{\mathcal{I}} \\ \mathbf{x}_{\tilde{\mathcal{I}}} = \mathbf{F}_{\tilde{\mathcal{I}}}^{\dagger}\mathbf{b} & o.w. \end{cases}$.
```

derived based on ADM. They have set their penalty value as $\rho = \tau/10$ in their simulations. Ghadimi *et al.* [18] have made a tool for setting the penalty value for a strictly convex problem with an inequality constraint. This tool takes a matrix given in the constraint as its input. By inspired by these works, we have relied upon extensive simulations with various penalty values given by a combination of τ and M, i.e.

$$\rho \in \left\{\tau/M, \tau/M^2, \ldots, \tau M^2\right\},$$

where M is the element of our matrix $\mathbf{A}$. Based on results of these simulations, we set the penalty value as $\rho = \tau/M$ and use this value in our simulations. In our simulations, we empirically observe that ADM-MIQP with this penalty value can be used to solve (3) for estimating a sparse signal with the accuracy of $\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|_2^2}{\|\mathbf{x}\|_2^2} \le \varepsilon$ where $\mathbf{x}$ is an original sparse signal, $\tilde{\mathbf{x}}$ is the estimate sparse signal and ε is sufficiently small.

Any warm-start techqniues can be applied into ADM-MIQP for improving its performance. We run ADM-MIQP multiple times with different initial variables randomly generated. Then, we have different sotluions, i.e.,

$$\left\{\mathbf{d}^1, \mathbf{d}^2, \cdots, \mathbf{d}^L\right\}$$

where L is the number of runs of ADM-MIQP. We then select a solution among these multiple solutions via

$$\mathbf{d}_{sol} := \underset{\mathbf{d} \in \{\mathbf{d}^1, \mathbf{d}^2, \cdots, \mathbf{d}^L\}}{\operatorname{argmin}} \quad 2^{-1}\mathbf{d}^T\mathbf{Q}\mathbf{d} + \mathbf{q}^T\mathbf{d}.$$

This selected solution is at least guaranteed to be better than the other unselected solutions in terms of the cost function.

B. COMPUTATION COSTS PER ITERATION

We aim to show that ADM-MIQP is a first-order-type method. The costs of updating $\mathbf{z}$ and $\mathbf{u}$ are both $\mathrm{O}(n)$. Then, the cost of updating $\mathbf{x}$ is $\mathrm{O}\left(mn + m^3\right)$, due to both the matrix inversion and the matrix-vector products. If $\mathbf{D}$ is stored, then this cost can be reduced to $\mathrm{O}(mn)$.

Next, in applications such as a single pixel camera [24], [25], a lensless camera [26], [27], for an image compression [28], a sensing matrix is constructed by randomly taking m rows from an orthogonal matrix. Then, $\mathbf{D}$ becomes a constant value $\frac{1}{1+2\rho}$. As a result, the update on $\mathbf{x}$ is given as

$$\mathbf{x}_{t+1} = \left((1+2\rho)^{-1}\mathbf{F}^T\mathbf{F}\mathbf{q}_{1,t}[1:n] - \mathbf{q}_{1,t}[1:n]\right)/(2\rho). \tag{11}$$

Indeed, if $\mathbf{F}$ is a partial discrete cosine transform (DCT) matrix, all matrix-vector products in (11) can be performed by the fast Fourier transform operation. That is, the update cost for$\mathbf{x}$ can be significantly reduced to $\mathrm{O}(n \log n)$.

C. COMPUTATION COSTS PER ITERATION

We now derive the algorithm called TAIC (*Takapoui's Algorithm with Inequality Constraint*) by following the approach of [22] for solving the MIQP problem (3) which only has the inequality constraint. As shown in the subsection III.A, it is noted that (3) is equal to (7). Then, we define the symbols as follows:

$$\begin{aligned}
\tilde{\mathbf{d}} &:= \begin{bmatrix} \mathbf{d} \\ \mathbf{z} \end{bmatrix} \in \mathbb{R}^n \times \{0, 1\}^n \times \mathbb{R}_+^{2n}, \\
\tilde{\mathbf{A}} &:= \begin{bmatrix} \mathbf{A} & \mathbf{I}_{2n} \end{bmatrix} \in \mathbb{R}^{2n \times 4n}, \\
\tilde{\mathbf{q}} &= \begin{bmatrix} \mathbf{q} \\ \mathbf{0}_{2n} \end{bmatrix} \in \mathbb{R}^{4n \times 1} \text{ and } \tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & \mathbf{O}_{2n} \\ \mathbf{O}_{2n} & \mathbf{O}_{2n} \end{bmatrix} \in \mathbb{R}^{4n \times 4n},
\end{aligned}$$

where $\mathbf{z}$ is a slack variable. With these symbols, we can reformulate (7) into an MIQP problem with an equality

$$\min_{\tilde{\mathbf{d}}} 2^{-1}\tilde{\mathbf{d}}^T\tilde{\mathbf{Q}}\tilde{\mathbf{d}} + \tilde{\mathbf{q}}^T\tilde{\mathbf{d}} \quad \text{subject to } \tilde{\mathbf{A}}\tilde{\mathbf{d}} = \mathbf{0}_{2n}, \tilde{\mathbf{d}} \in \tilde{\mathcal{X}} \tag{12}$$

where $\tilde{\mathcal{X}} := \mathbb{R}^n \times \{0, 1\}^n \times \mathbb{R}_+^{2n}$ is a non-convex set. Similar to (6), we also reformulate (12) into a standard form of ADM as follows:

$$\begin{aligned}
&\min_{\tilde{\mathbf{d}}, \tilde{\mathbf{z}}} 2^{-1}\tilde{\mathbf{d}}^T\tilde{\mathbf{Q}}\tilde{\mathbf{d}} + \tilde{\mathbf{q}}^T\tilde{\mathbf{d}} + I_{\tilde{\mathcal{X}}}\left(\tilde{\mathbf{z}}\right) \\
&\text{subject to } \begin{bmatrix} \tilde{\mathbf{A}} \\ \mathbf{I}_{4n} \end{bmatrix}\tilde{\mathbf{d}} - \begin{bmatrix} \mathbf{O}_{2n \times 4n} \\ \mathbf{I}_{4n} \end{bmatrix}\tilde{\mathbf{z}} = \mathbf{0}_{6n}
\end{aligned} \tag{13}$$

where $I_{\tilde{\mathcal{X}}}\left(\tilde{\mathbf{z}}\right)$ is an indicator function of $\tilde{\mathcal{X}}$ and $\mathbf{O}_{2n \times 4n}$ is the $2n \times 4n$ matrix of zeros. TAIC is then implemented via

$$\begin{aligned}
\tilde{\mathbf{d}}_{t+1} &= \arg\min_{\tilde{\mathbf{d}}} \ 2^{-1}\tilde{\mathbf{d}}^T\tilde{\mathbf{Q}}\tilde{\mathbf{d}}^T + \tilde{\mathbf{q}}^T\tilde{\mathbf{d}}^T \\
&\quad + 2^{-1}\rho\left\|g\left(\tilde{\mathbf{d}}, \tilde{\mathbf{z}}_t, \tilde{\boldsymbol{\lambda}}_t\right)\right\|_2^2, \\
\tilde{\mathbf{z}}_{t+1} &= \arg\min_{\tilde{\mathbf{z}}} \ I_{\tilde{\mathcal{X}}}\left(\tilde{\mathbf{z}}\right) + 2^{-1}\rho\left\|g\left(\tilde{\mathbf{d}}_{t+1}, \tilde{\mathbf{z}}, \tilde{\boldsymbol{\lambda}}_t\right)\right\|_2^2, \\
\tilde{\boldsymbol{\lambda}}_{t+1} &= \tilde{\boldsymbol{\lambda}}_t - \rho g\left(\tilde{\mathbf{d}}_{t+1}, \tilde{\mathbf{z}}_{t+1}, \mathbf{0}_{6n}\right),
\end{aligned} \tag{14}$$

where $\tilde{\boldsymbol{\lambda}}$ is the dual variable, $\rho > 0$ is a penalty value, and

$$g\left(\tilde{\mathbf{d}}, \tilde{\mathbf{z}}, \tilde{\boldsymbol{\lambda}}\right) := \begin{bmatrix} \tilde{\mathbf{A}} \\ \mathbf{I}_{4n} \end{bmatrix} \tilde{\mathbf{d}} - \begin{bmatrix} \mathbf{O}_{2n\times 4n} \\ \mathbf{I}_{4n} \end{bmatrix} \tilde{\mathbf{z}} + \frac{\tilde{\boldsymbol{\lambda}}}{\rho}.$$

It is noted that (7) is formed by adding *one slack variable* to (3). But, (13) is formed by adding *two slack variables* into (3). Thus, there is an intuition that TAIC requires more computational costs per iteration than ADM-MIQP does.

To investigate the validation of our intuition, we restrict our attentions to the sub-problem on $\tilde{\mathbf{d}}$ in (14) that can be simplified to

$$\tilde{\mathbf{d}}_{t+1} = \arg\min_{\tilde{\mathbf{d}}} 2^{-1} \tilde{\mathbf{d}}^T \tilde{\mathbf{D}} \tilde{\mathbf{d}}^T + \mathbf{h}_t^T \tilde{\mathbf{d}} \tag{15}$$

where $\tilde{\mathbf{D}} := \left[\tilde{\mathbf{Q}} + \rho\left(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}} + \mathbf{I}_{4n}\right)\right] \in \mathbb{R}^{4n\times 4n}$ and

$$\mathbf{h}_t := \tilde{\mathbf{q}} - \rho \begin{bmatrix} \tilde{\mathbf{A}} \\ \mathbf{I}_{4n} \end{bmatrix}^T \left(\begin{bmatrix} \mathbf{O}_{2n\times 4n} \\ \mathbf{I}_{4n} \end{bmatrix} \tilde{\mathbf{z}}_t - \tilde{\boldsymbol{\lambda}}_t \right).$$

The sub-problem on $\mathbf{d}$ in (7) has been decomposed into a pair of problems on $\mathbf{x}$ and $\mathbf{u}$, respectively. But, $\tilde{\mathbf{D}}$ is a non-diagonal matrix that implies that the sub-problem in (15) cannot be decomposed. We then consider an analytic closed form solution to (15) as follows:

$$\tilde{\mathbf{d}}_{t+1} = -\tilde{\mathbf{D}}^{-1}\mathbf{h}_t. \tag{16}$$

For saving computational costs, the inverse matrix in (16) can be stored. Even with this stored matrix, TAIC takes $\mathrm{O}\left(16n^2\right)$ computational cost per iteration for conducting (16) due to the matrix-vector product. This cost can be negligible for a small value of n. For a large value of n, it cannot be ignored. On the other hands, ADM-MIQP takes $\mathrm{O}\left(mn\right)$ computational cost per iteration. It can be seen that TAIC takes more computational costs per iteration for updating the other variables than ADM-MIQP does. Thus, it can be concluded that the cost of ADM-MIQP is greatly less than that of TAIC.

IV. SIMULATIONS STUIDES

We conduct simulations to show that ADM-MIQP gives a solution to (3). We compare ADM-MIQP with MDAL and YALL1. The reasons for selecting both MDAL and YALL1 as comparative approaches are *a*) these methods are also based on ADM and *b*) are known to be computationally tractable. We define a Gaussian sparse vector ensemble and a Gaussian noise vector ensemble as follows.

Definition 1: The Gaussian sparse vector ensemble is an ensemble of n-dimensional k-sparse vectors, where each vector $\mathbf{x}$ is generated as follows: *a*) the positions of the non-zero values of $\mathbf{x}$ are randomly selected, *b*) the non-zero values are taken from the standard normal distribution and *c*) $\mathbf{x}$ is normalized to produce the l_2-norm for $\mathbf{x}$ unit.

Definition 2: The Gaussian noise vector ensemble is an ensemble of m-dimensional noise vectors whose elements are independent and identically distributed Gaussian with zero mean and variance σ^2.

We define the *signal-to-noise ratio* (SNR) as

$$\text{SNR [dB]} := 10\log_{10}\left(\|\mathbf{Fx}\|_2^2/\left(m\sigma^2\right)\right).$$

We set the parameters of ADM-MIQP, MDAL, and YALL1 as follows. The regularization value is set as $\tau = \sigma\sqrt{2\log n}$ if SNR [dB] is finite and $\tau = 10^{-4}$ if SNR [dB] is infinite. The M value is set as $M = \max_i |\mathbf{x}(i)|$. As we have stated in the sub-section III.A, our penalty value is set as $\rho = \tau/M$. The penalty value of YALL1 is set as $\rho = \|\mathbf{b}\|_1/m$, used in [6]. But, MDAL with the penalty value used in [13] failed to yield an accurate solution in our simulation. We conducted extensive simulations to find the penalty value for MDAL. Thus, in our simulations, it was set as $\rho = 10\tau$. We terminated these methods either when the number of iterations exceeded 2000 or when $\frac{\|\mathbf{x}_{t+1}-\mathbf{x}_t\|_2}{\|\mathbf{x}_{t+1}\|_2} \leq 10^{-4}$, as was done in [6], for YALL1, and when $\frac{\|\mathbf{x}_{t+1}-\mathbf{x}_t\|_2}{\|\mathbf{b}\|_2} \leq 10^{-4}$, as in [13] for MDAL and when $\frac{\|\mathbf{d}_{t+1}-\mathbf{d}_t\|_2}{\|\mathbf{d}_{t+1}\|_2} \leq 10^{-4}$ for ADM–MIQP.

We kept in mind that a solution for $\mathbf{x}$ in (3) must satisfy a convex constraint

$$\mathbf{x} \in \{\mathbf{x}\,|-M \leq \mathbf{x}(i) \leq M\},$$

where $i = 1, 2, \ldots, n$. However, both YALL1 and MDAL are not designed to use this constraint. Therefore, we extended these methods to use the constraint for a fair comparison. Since the constraint is convex, this extension was easily carried out by adding the following:

$$\mathbf{x}_t(i) = \min\left(\max\left(\mathbf{x}_t(i), M\right), -M\right),$$

where $\mathbf{x}_t(i)$ is the i^{th} element of an intermediate solution at the t^{th} iteration. All simulations are conducted on a computer with Intel (R) Core (TM) i7-3820 processor clocked at 3.6 GHz. The MATLAB codes are in [29].

A. CONVERGENCE BEHAVIORS OF ADM-MQIP

We remind that both SSE defined in (4) and MSE defined in (5) can be used to evaluate the quality of a solution given by ADM-MIQP. We use both of the metrics to study how this solution behaves. Since the elements of $\mathbf{u}$ are either 0 or 1, we have

$$\begin{aligned} kd_1\left(\mathbf{u}, \mathbf{u}_t\right) + nd_2\left(\mathbf{x}, \mathbf{x}_t\right) &= \|\mathbf{u} - \mathbf{u}_t\|_2^2 + \|\mathbf{x} - \mathbf{x}_t\|_2^2 \\ &= \left\| \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} - \begin{bmatrix} \mathbf{x}_t \\ \mathbf{u}_t \end{bmatrix} \right\|_2^2 \\ &= \|\mathbf{d} - \mathbf{d}_t\|_2^2 \end{aligned} \tag{17}$$

where $\mathbf{d}_t$ is the t^{th} solution of the ADM-MIQP and $\mathbf{d}$ is a feasible solution to (3). Thus, if both the metrics are small, the l_2-norm between the t^{th} solution and the $\mathbf{d}$ is also small. Based on this relation, we define the convergence of ADM-MIQP.

Definition 3: A solution $\mathbf{d}_t = \left[\mathbf{x}_t^T\ \mathbf{u}_t^T\right]^T$ of by ADM-MIQP is convergent to a point $\mathbf{d} = \left[\mathbf{x}^T\ \mathbf{u}^T\right]^T$ to (3) if there exists a positive integer T such that for every positive

ε_1 and ε_2, we then have $d_1(\mathbf{u}, \mathbf{u}_t) < \varepsilon_1$ and $d_2(\mathbf{x}, \mathbf{x}_t) < \varepsilon_2$ for all $T \le t \le maxIter$, where maxIter is the maximum number of iteration.

To show that ADM-MIQP can find a converged solution to the MIQP problem (3), the problem dimension n, the number of measurements m and the sparsity level k were set as 1024, 307 and 30, respectively. Two values for SNR [dB] were considered: 35 and 45, respectively. We generated 1000 independent realizations of the set $(\mathbf{F}, \mathbf{x}, \mathbf{n})$ where $\mathbf{F}$ was made by randomly taking 307 rows of the 1024 × 1024 DCT matrix, $\mathbf{x}$ was taken from the Gaussian sparse vector ensemble, and $\mathbf{n}$ was taken from the Gaussian noise vector ensemble. We determined average values for both MSE defined in (5) and SSE defined in (4). We then plotted the results in Figs. 1 and 2, respectively.

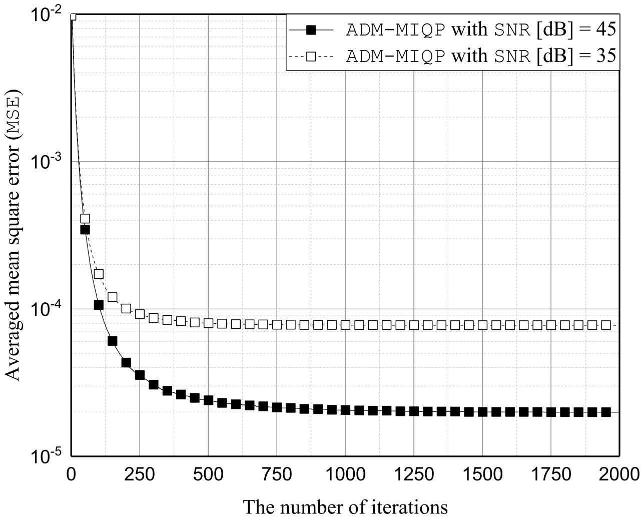


FIGURE 1. It plots the average MSE of ADM-MIQP depending on the number of iterations. The problem dimension n, the number of measurements m and the sparsity level k are set to be 1024, 307 and 30, respectively.

For all the SNRs investigated, both MSE and SSE gradually decreased and were eventually saturated. For SNR [dB] = 45, at the 250th and 500th iterations, MSEs were $3.5 \times 1\mathrm{E}^{-5}$ and $2.4 \times 1\mathrm{E}^{-5}$, respectively. Finally, MSE converged to $2 \times 1\mathrm{E}^{-5}$ after $\mathrm{O}\left(10^3\right)$ iterations. This means that an estimate of $\mathbf{x}$ can converge to an original sparse signal. Next, we considered SSE at SNR [dB] = 45. At the 250th and 500th iterations, SSEs were 0.041 and 0.031, respectively. Eventually, SSE converged to 0.029 after $\mathrm{O}\left(10^3\right)$ iterations. This suggests that the detected support set converges to an original support set. Due to (17), after $\mathrm{O}\left(10^3\right)$ iterations, we observed

$$\|\mathbf{d} - \mathbf{d}_t\|_2^2 < \mathrm{O}\left(10^{-c}\right)$$

where $c \approx 1$. This observation shows the convergence of ADM-MIQP under the definition 3.

B. COMPARISON STUIDES AND DISCUSSION

Let $\alpha := m/n$ be an *under-sampling* ratio and $\beta := k/m$ be an *over-sampling* ratio. The phase transition for a given method shows how accurately this method can estimate sparse signals in the (α, β) plane with n. We conducted simulations to study the phase transitions in computations obtained by ADM-MIQP, MDAL and YALL1. Then, the aims of this phase transition study include being aware of the overall performance of ADM-MIQP and understanding which of these ADM-based methods, each of which solves different problems to estimate sparse signals, achieves the best performance for this sparse signal estimation.

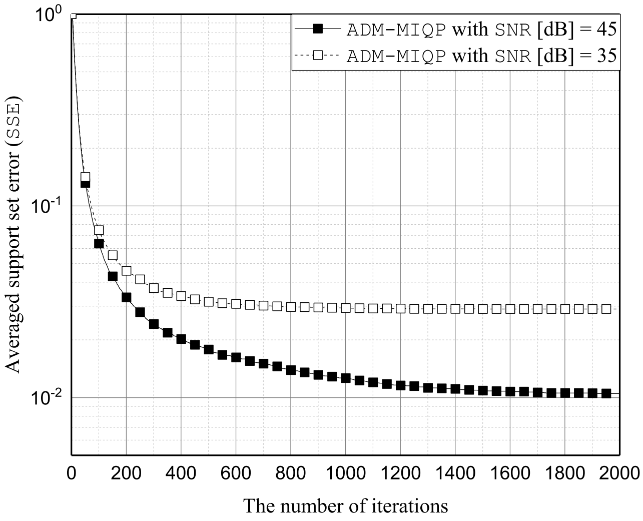


FIGURE 2. It plots the average SSE of ADM-MIQP depending on the number of iterations. The problem dimension n, the number of measurements m and the sparsity level k are set to be 1024, 307 and 30, respectively.

The problem dimension n was set as 1024. Then, a 15 × 15 uniformly spaced grid on the (α, β) plane was constructed for $\alpha, \beta \in \{0.15, 0.175, \cdots, 0.5\}$. We generated 1000 independent realizations of the set $(\mathbf{F}, \mathbf{x})$, where $\mathbf{F}$ was derived by randomly taking m rows of the 1024 × 1024 DCT matrix and $\mathbf{x}$ was taken from the Gaussian sparse vector ensemble. The estimate $\tilde{\mathbf{x}}$ was considered to be successful if $\left\|\mathbf{x} - \tilde{\mathbf{x}}\right\|_2^2 / \|\mathbf{x}\|_2^2 \le 10^{-4}$. In Fig. 3, we illustrated the phase transitions for all these methods. The solid line represents a 99% probability of success. That is, for points lying in the graphical area below this line, there was at least 99% probability of success in problem solving. The area beneath the dashed-line then represents a 50% probability of success.

First, we fixed the over-sampling ratio. We then considered the under-sampling ratio to attain a 99% probability of success. The under-sampling ratio for ADM-MIQP was found to be the smallest. As an example, for a fixed $\beta = 0.25$, we observed that the under-sampling ratios of ADM-MIQP, MDAL, and YALL1 were 0.25, 0.275, and 0.325, respectively. The under-sampling ratio was proportional to m because n was fixed. This implies that ADM-MIQP requires the smallest value of m for sparse signal estimation, when compared with the other methods.

Second, we fixed the under-sampling ratio and considered the over-sampling ratio to achieve a 99% probability of success. We observed that for ADM-MIQP, the over-sampling

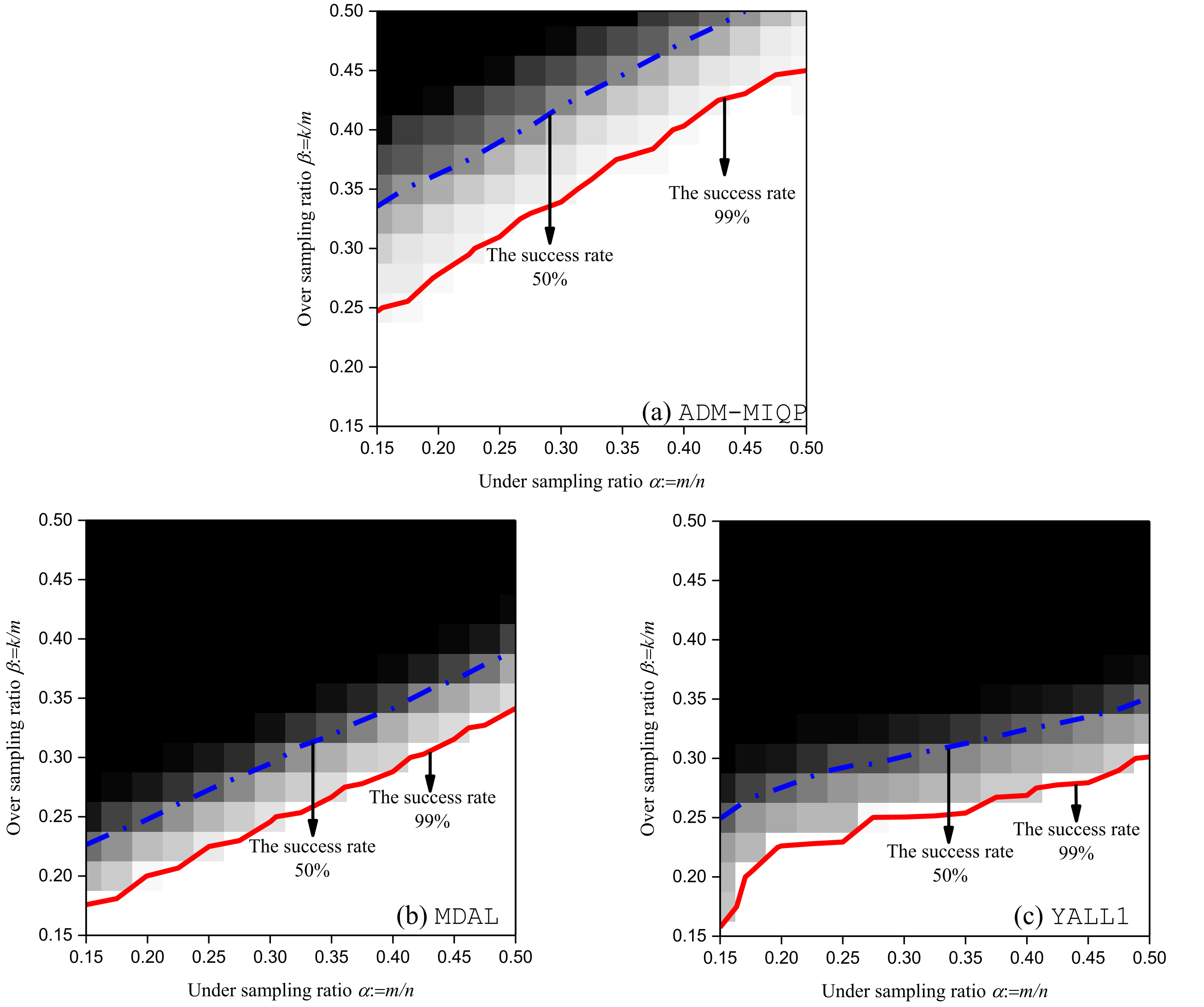


FIGURE 3. It plots the empirical phase transitions of the ADM-based methods such as ADM-MIQP, MDAL and YALL1, respectively.

ratio was the largest. For a fixed $\alpha = 0.3$, the over-sampling ratios of ADM-MIQP, MDAL and YALL1 were 0.325, 0.25, and 0.225, respectively. The over-sampling ratio was proportional to k for a fixed under-sampling ratio. This shows that ADM-MIQP can estimate $\mathbf{x}$ with the higher value of k in which the other methods cannot.

Next, we conducted simulations to study the performance of all these methods by varying k for a fixed n and m under noisy cases. To this end, SNR [dB], n and m were set as 35, 1024, and 307, respectively and k was varied between 30 and 100. Then, we generated 1000 independent realizations of the set $(\mathbf{F}, \mathbf{x}, \mathbf{n})$ where $\mathbf{F}$, $\mathbf{x}$, and $\mathbf{n}$ were obtained through the manner discussed in the sub-section III.A. Then, we obtained the average MSE for each method and plotted these values in Fig. 4.

For any k, ADM-MIQP can achieve the lowest MSE when compared with MDAL and YALL1. This means that ADM-MIQP can more accurately estimate $\mathbf{x}$ than the other methods can. The MSE gap between ADM-MIQP and ORACLE is small. At $k = 40$, as an example, MSEs of ADM-MIQP and ORACLE are $7 \times 1\mathrm{E}^{-6}$ and $4 \times 1\mathrm{E}^{-6}$, respectively. This suggests that ADM-MIQP can achieve a performance close to that achieved by ORACLE.

Since both MDAL and YALL1 are originally designed to find an estimate of $\mathbf{x}$, not the support set, we needed to construct the support set based on the estimate $\tilde{\mathbf{x}}$ in order to measure SSEs for these methods. For this purpose, we set a threshold value

$$\zeta = 0.8 \min_i |\mathbf{x}(i)|$$

and constructed the support set $\hat{\mathbf{u}}$ by

$$\begin{aligned} \hat{\mathbf{u}}(i) &= 0 \quad \text{if } \left|\tilde{\mathbf{x}}(i)\right| < \zeta, \\ \hat{\mathbf{u}}(i) &= 1 \quad \text{if } \left|\tilde{\mathbf{x}}(i)\right| \geq \zeta. \end{aligned}$$

where $i = 1, 2, \ldots, n$. Under the same conditions used in the experiment depicted in Fig. 4, we independently made 1000 realizations of the set $(\mathbf{F}, \mathbf{x}, \mathbf{n})$. We then determined the

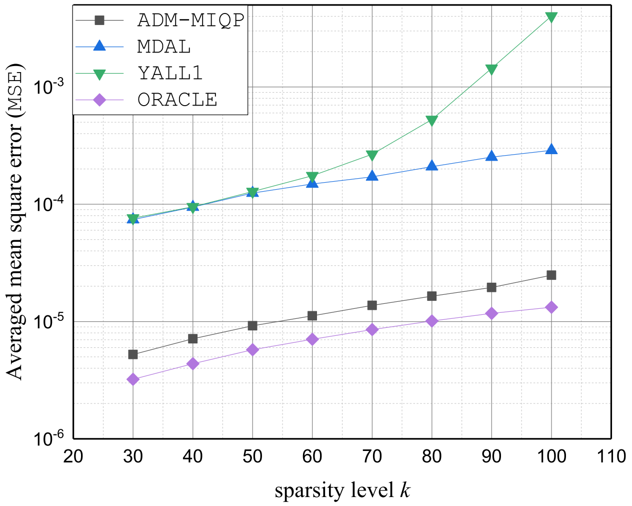


FIGURE 4. It plots the average MSEs of ADM-MIQP, MDAL, YALL1 and ORACLE depending on the sparsity level k. The problem dimension n, the number of measurements m and SNR [dB] are set to be 1024, 307 and 35, respectively.

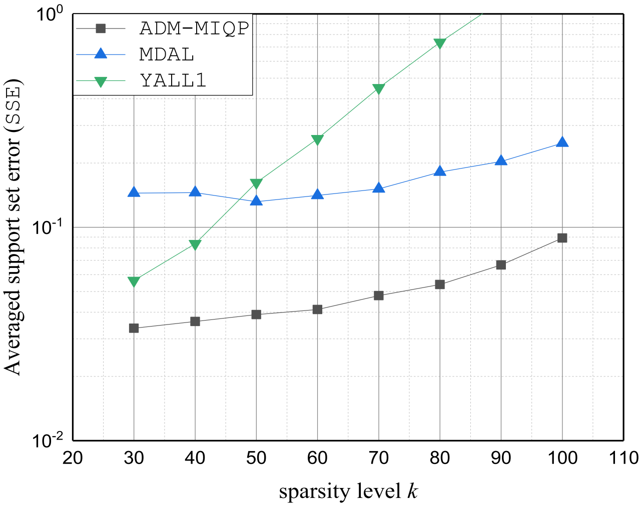


FIGURE 5. It plots the average SSEs of ADM-MIQP, MDAL and YALL1 depending on the sparsity level k. The problem dimension n, the number of measurements m and SNR [dB] are set to be 1024, 307 and 35, respectively.

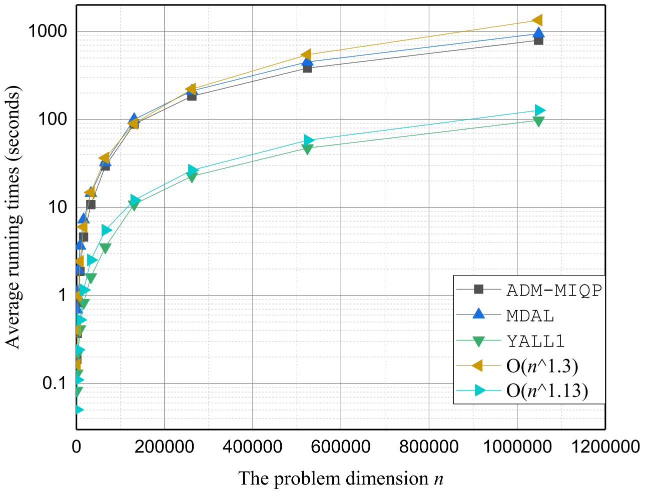


FIGURE 6. It plots the average running times of ADM-MIQP, MDAL, YALL1 and CPLEX depending on the problem dimension n with $m = \lfloor 0.3n \rfloor$, $k = \lfloor 0.3m \rfloor$ and SNR [dB] $= 45$. ADM-MIQP, MDAL and YALL1 have the polynomial computational order.

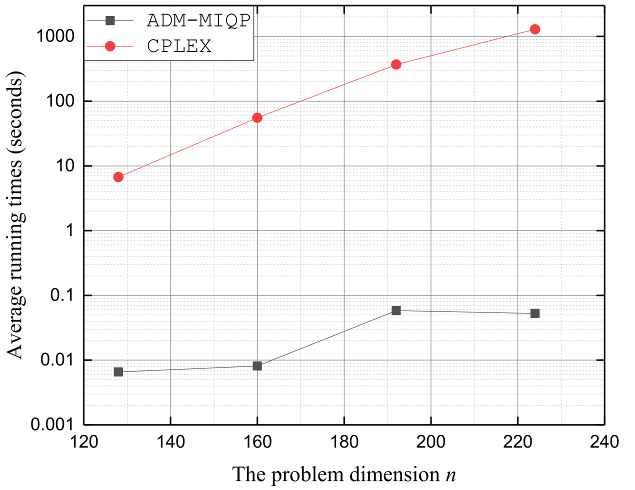


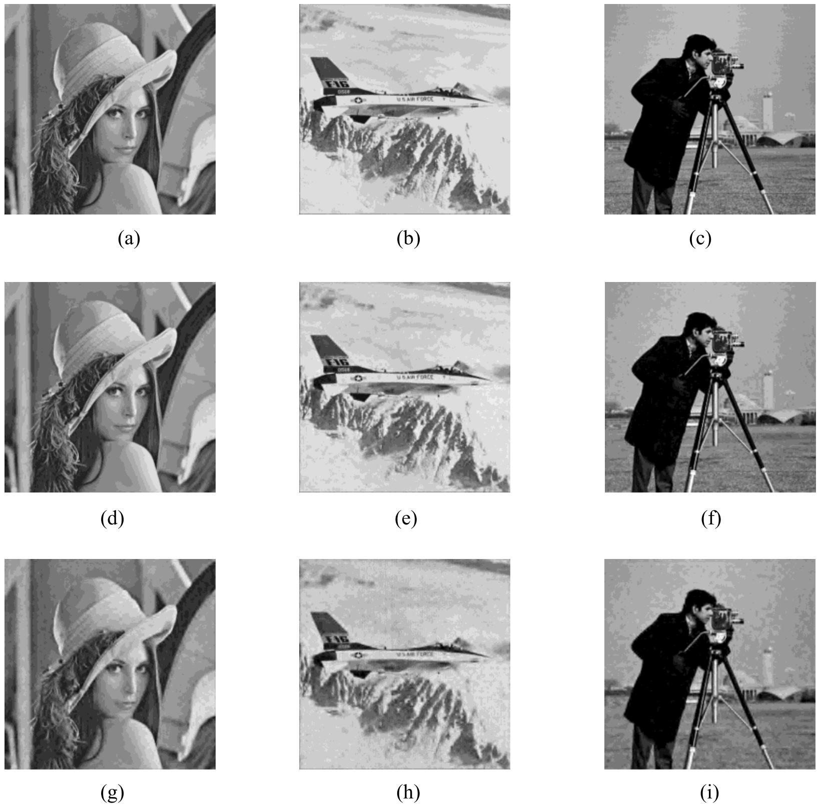
FIGURE 7. It plots the average running times of ADM-MIQP and CPLEX depending on the problem dimension n with $m = \lfloor 0.3n \rfloor$, $k = \lfloor 0.2m \rfloor$ and SNR [dB] $= 45$. This figure shows that ADM-MIQP is significantly faster than CPLEX.

average SSE for each of the methods and plotted the results in Fig. 5. As with MSE, for any k, ADM-MIQP was found to achieve the lowest SSE. As an example, at $k = 80$, SSEs of ADM-MIQP, MDAL, and YALL1 were 0.04, 0.14, and 0.26, respectively. This means that ADM-MIQP can more accurately detect the support set than the other methods can. Next, at $k = 60$, we counted the number of events for which $\sum_{i=1}^{n} \left\| \mathbf{u}(i) - \hat{\mathbf{u}}(i) \right\|_1 \leq 6$, i.e., for which the support set error could occur within 10%. The results for ADM-MIQP, MDAL, and YALL1 were 962, 227, and 349 events respectively. This suggests that ADM-MIQP surpasses the other methods.

Thus far, we have shown that ADM-MIQP is superior to other ADM-based methods in terms of MSE and SSE. There are multiple reasons for why this is the case.

First, ADM-MIQP is designed to solve (3). The binary vector $\mathbf{u}$ in (3) indicates the support set and $\mathbf{1}_n^T\mathbf{u}$ counts the number of ones in $\mathbf{u}$. This means that ADM-MIQP aims to find a solution that both the cardinality of the support set and the data-fidelity are jointly minimized. Minimizing the cardinality of the support set is a characteristic of l_0-norm based methods. This is the reason for the superiority of our method over YALL1.

Second, Dong and Zhang [13] have empirically reported that MDAL finds a local solution to the l_0-norm problem. By contrast, methods based on ADM tend to find the global solution to a MIQP problem, as reported in [20]–[22]. Then, as reported in [14], CPLEX is capable of finding the global solution to (3). To understand whether ADM-MIQP finds the global solution or not, we compared the solution of



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE 8. The original grayscale images of size 512 × 512 are shown in the first row. The images recovered by ADM-MIQP are shown in the second row. The images recovered by MDAL are shown in the third row. The PSNR value of each recovered image is averaged 10 trials at $m = \lfloor 0.15n \rfloor$ and $k = \lfloor 0.05n \rfloor$. (a) Lena. (b) Airplane. (c) Cameraman. (d) 33.83 dB. (e) 31.19 dB. (f) 33.71 dB. (g) 26.91 dB. (h) 22.27 dB. (i) 27.31 dB.

ADM-MIQP and that of CPLEX. We independently generated 100 realizations of the set ($\mathbf{F}$, $\mathbf{x}$) by assuming that n, m, and k were 200, 80, and 10 respectively, where $\mathbf{F}$ was a partial orthogonal sensing matrix and $\mathbf{x}$ was taken from the Gaussian sparse vector ensemble. We determined the average of the objective function

$$\tau \mathbf{1}_n^T \mathbf{u} + 2^{-1} \left\| \mathbf{b} - \mathbf{F}\mathbf{x} \right\|_2^2$$

for each method, as well as the average for *normalized* MSE,

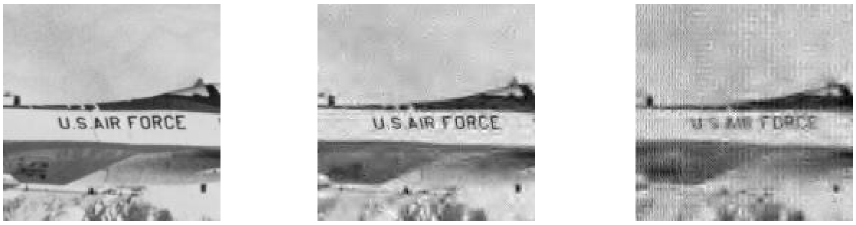
$$\|\mathbf{x}_{\mathrm{C}} - \mathbf{x}_{\mathrm{A}}\|_2^2 / \|\mathbf{x}_{\mathrm{C}}\|_2^2$$

where $\mathbf{x}_{\mathrm{A}}$ is an estimate of $\mathbf{x}$ obtained by ADM-MIQP and $\mathbf{x}_{\mathrm{C}}$ is an estimate of $\mathbf{x}$ obtained by CPLEX. The value of the objective function of CPLEX, and that of ADM-MIQP, were 0.0099 and 0.0094, respectively, and the *normalized* MSE was 0.0033. The gap between these values and the *normalized* MSE were both small. This indicates that ADM-MIQP indeed finds the global solution to (3). This makes ADM-MIQP a superior approach to MDAL.

We observed that ADM-MIQP is computationally tractable for solving (3) up to the problem dimension n of the order of one million. To this end, SNR [dB] was set as 45 and n was varied from 1024 to 1048576. For a fixed n, we altered m and k to $m = \lfloor 0.3n \rfloor$ and $k = \lfloor 0.3m \rfloor$. The number of iterations was set as 1000. At each point (n, m, k, SNR [dB]), we generated 500 independent realizations of the set ($\mathbf{F}$, $\mathbf{x}$, $\mathbf{n}$), where $\mathbf{F}$, $\mathbf{x}$, and $\mathbf{n}$ are obtained by the approach given in the sub-section IV.A. We determined the average running time for each method and plotted the results in Fig. 6.

In Fig. 6, the average running times for each method grow linearly with n. We calculated the order of the average running times for ADM-MIQP, MDAL and YALL1 with respect to n. The orders are roughly $\mathrm{O}\left(n^{1.3}\right)$, $\mathrm{O}\left(n^{1.3}\right)$, and $\mathrm{O}\left(n^{1.13}\right)$ respectively. These orders show that ADM-MIQP has polynomial computation costs, leading to that ADM-MIQP is still computationally tractable for solving (3) with the large problem dimension. Finally, YALL1 was found to be a faster method than ADM-MIQP. This is because the l_1-norm problem (2), solved by YALL1, is easier to solve than (3). Despite this, if the running time for ADM-MIQP is acceptable, ADM-MIQP gains significant improvements on sparse signal estimation.

We conducted simulations to compare ADM-MIQP with CPLEX in terms of the running time. MaxIter was set as 1000. SNR [dB] was set as 45 and n was varied from 128 to 224. Then, both m and k were altered to $m = \lfloor 0.3n \rfloor$ and $k = \lfloor 0.2m \rfloor$. At each point (n, m, k, SNR [dB]), we made 50 independent realizations of the set ($\mathbf{F}$, $\mathbf{x}$, $\mathbf{n}$), where $\mathbf{x}$ and $\mathbf{n}$ are obtained by the approach given in the sub-section IV.A and $\mathbf{F}$ is a partial orthogonal matrix.



(a) (b) (c)

FIGURE 9. The images are corresponding to the part of the original and each recovered airplane images. (a) The original image. (b) The image recovered by ADM-MIQP. (c) The image recovered by MDAL.

In Fig. 7, the average running time of CPLEX rapidly grows with n. Even n was roughly doubled, the time rapidly increased. At $n = 128$ and $n = 224$, the times are 6.7 secs and 2113 secs, respectively. This observation can be in accordance with the statement in Section I that CPLEX has the computational intractability in solving (3) with large variables. On the other hands, the average running time of ADM-MIQP does not rapidly grow with n. This observation shows that ADM-MIQP is faster than CPLEX.

C. AN IMAGE RECOVERY EXAMPLE

We conducted an image recovery experiment to demonstrate the successful application of ADM-MIQP. For this study, the discrete wavelet transform was applied onto each image. The k largest magnitude values of the transformed image were retained. For each image, k non-zero values were stacked to form a sparse vector, to be compressed to get an m-dimensional measurement vector using a partial DCT matrix. Both MDAL and ADM-MIQP were used to recover the image. To evaluate the qualities of the recovered images, we used the following *peak-signal-to-noise ratio* (PSNR):

$$PSNR\,[\text{dB}] := 10\log_{10}\left(n \times 255^2/\|\mathbf{x}-\tilde{\mathbf{x}}\|_2^2\right), \qquad (18)$$

where $\tilde{\mathbf{x}}$ is an original image and is the recovered image.

In Fig. 8, we illustrate the original greyscale images of size 512 × 512 with a problem dimension $n = 262144$. We have also showed the images recovered by each method and their PSNRs. These PSNR values were the averages of results from 10 trials where $m = \lfloor 0.15n \rfloor$ and $k = \lfloor 0.05n \rfloor$.

It is immediately observed that ADM-MIQP recovers images with higher quality than MDAL in terms of PSNR. ADM-MIQP then preserves the detailed information in the original images. For example, let us consider the text part "*US AIR Force*" of the recovered airplane image. As shown in Fig. 9, we clearly see this text in (b), recovered by ADM-MIQP, we cannot make out it in (c), recovered by MDAL. This result shows that ADM-MIQP surpasses MDAL in this image recovery example.

V. CONCLUSION

We proposed a fast method referred to as ADM-MIQP to solve the mixed integer quadratic programming problem (3) formulated in [14] from the l_0-norm problem (1). We derived ADM-MIQP using the alternating direction method, which has been recently used to solve integer programming problems in [20]–[22]. We then showed that ADM-MIQP is a first-order-type method. That is, matrix-vector products are only used to implement ADM-MIQP. We selected MDAL [13] and YALL1 [6] as competitors to ADM-MIQP because these methods are based on ADM to solve the l_0-norm and the l_1-norm problems, respectively. We also compared ADM-MIQP with ORACLE, an approach which involved *a priori* knowledge of the support set. We used both *support set error* (SSE) (4) and *mean square error* (MSE) (5) to assess the quality of a solution obtained by each method.

We empirically demonstrated that ADM-MIQP could achieve a significantly better performance than MDAL and YALL1 in terms of both SSE and MSE. We also showed that ADM-MIQP eventually achieved a performance close to that of ORACLE in terms of MSE. We showed that ADM-MIQP is computationally tractable for solving (3) up to the order of one million in the problem dimension. We confirmed that the computational cost of ADM-MIQP is $\mathrm{O}\left(n^{1.3}\right)$ in our simulations. We concluded that ADM-MIQP is efficient in finding an accurate solution to (3) when the problem dimension n is large.

The next step is to conduct convergence analysis for ADM-MIQP. Specifically, it will be interesting to prove that a solution of ADM-MIQP is convergent. Also, this work can be extended to determine the appropriate penalty value that would guarantee the convergence of ADM-MIQP.

REFERENCES

[1] E. J. Candés, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[2] S.-J. Kim, K. Koh, M. Lustig, S. Boyd and D. Gorinevsky, "An interior point method for large-scale l1-regularized least squares," *IEEE. J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[3] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[4] L. Xiao and T. Zhang, "A proximal-gradient homotopy method for the sparse least-squares problem," *SIAM J. Optim.*, vol. 23, no. 2, pp. 1062–1091, 2013.

[5] T. Goldstein and S. Osher, "The split Bregman method for ℓ_1-regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[6] F. Yang and Y. Zhang, "Alternating direction algorithms for ℓ_1-problems in compressive sensing," *SIAM J. Sci. Comput.*, vol. 33, no. 1, pp. 250–278, 2011.

[7] F. Wen, L. Pei, Y. Yang, W. Yu, and P. Liu, "Efficient and robust recovery of sparse signal and image using generalized nonconvex regularization," *IEEE Trans. Comput. Imag.*, vol. 3, no. 3, pp. 566–579, Dec. 2017.

[8] R. Chartrand and V. Staneva, "Restricted isometry properties and nonconvex compressive sensing," *Inverse Problems*, vol. 24, no. 3, p. 035020, May 2008.

[9] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, Nov. 2009.

[10] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 298–309, Apr. 2010.

[11] T. Blumensath, "Accelerated iterative hard thresholding," *Signal Process.*, vol. 92, no. 3, pp. 752–756, Mar. 2012.

[12] Z. Dong and W. Zhu, "Homotopy methods based on ℓ_0-norm for compressed sensing," *IEEE Trans. Neural Netw. Learn Syst.*, vol. 29, no. 4, pp. 1132–1146, Apr. 2018.

[13] B. Dong and Y. Zhang, "An efficient algorithm for ℓ_0 minimization in wavelet frame based image restoration," *J. Sci. Comput.*, vol. 54, pp. 350–368, Feb. 2013.

[14] S. Bourguignon, J. Ninin, H. Carfantan, and M. Mongeau, "Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1405–1419, Oct. 2016.

[15] D. Bertsimas, A. King, and R. Mazumder, "Best subset selection via a modern optimization lens," *Ann. Statist.*, vol. 44, no. 2, pp. 813–852, 2016.

[16] *IBM ILOG CPLEX V12.8.0*. Accessed: Jun. 1, 2018. [Online]. Available: http://www.ibm.com/products/ilog-cplex-optimization-studio

[17] S. Boyd and J. Mattingley, "Branch and bound methods," Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, Course Notes EE364b, Mar. 2018. [Online] Available: https://web.stanford.edu/class/ee364b/lectures/bb_notes.pdf

[18] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 644–658, Mar. 2015.

[19] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *J. Sci. Comput.*, vol. 66, no. 3, pp. 889–916, Mar. 2016.

[20] A. K. Yadav, R. Ranjan, U. Mahbub, and M. C. Rotkowitz, "New methods for handling binary constraints," in *Proc. 54th Annu. Allerton Conf. Commun. Control Comput.*, Sep. 2016, pp. 1074–1080.

[21] N. Souto and R. Dinis, "MIMO detection and equalization for single-carrier systems using the alternating direction method of multipliers," *IEEE Signal Process. Lett.*, vol. 23, no. 12, pp. 1751–1755, Dec. 2016.

[22] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, "A simple effective heuristic for embedded mixed-integer quadratic programming," *Int. J. Control*, Apr. 2017, doi: 10.1080/00207179.2017.1316016.

[23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 1996.

[24] M. F. Duarte *et al.*, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, Mar. 2008.

[25] D. B. Phillips *et al.*, "Adaptive foveated single-pixel imaging with dynamic supersampling," *Sci. Adv.*, vol. 3, no. 4, p. e1601782, Apr. 2017.

[26] G. Satat, M. Tancik, and R. Raskar, "Lensless imaging with compressive ultrafast sensing," *IEEE Trans. Comput. Imag.*, vol. 3, no. 3, pp. 398–407, Sep. 2017.

[27] G. Huang, H. Jiang, K. Matthews, and P. Wilford, "Lensless imaging by compressive sensing," in *Proc. 20th IEEE Int. Conf. Image Process. (ICIP)*, Melbourne, VIC, Australia, Sep. 2013, pp. 2101–2105.

[28] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, "An overview of JPEG-2000," in *Proc. Data Compress. Conf.*, Snowbird, UT, USA, Mar. 2000, pp. 523–541.

[29] S. J. Park and H.-N Lee. (2018). *ADM-MIQP*. Accessed: Oct. 1, 2018. [Online]. Available: https://github.com/infonetGIST/infonetcompressedsensing

[30] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," in *Handbook of Applied Optimizations*. Oxford, U.K.: Oxford Univ. Press, 2000.

[31] S. Boyd and L. Vandenberghe, "Localization and cutting-plane method," Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, Course Notes EE364b, Apr. 2008. [Online] Available: https://see.stanford.edu/materials/lsocoee364b/05-localization_methods_notes.pdf

[32] M. V. Afonso, J.-M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

SANGJUN PARK received the B.S. degree in computer engineering from Chungnam National University, Daejeon, South Korea, in 2009. He is currently pursuing the Ph.D. degree with the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, South Korea. His research interests include information theory, numerical optimization, and compressed sensing.

HEUNG-NO LEE (SM'13) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California at Los Angeles, Los Angeles, CA, USA, in 1993, 1994, and 1999, respectively. From 1999 to 2002, he was with HRL Laboratories, LLC, Malibu, CA, USA, as a Research Staff Member. From 2002 to 2008, he was an Assistant Professor with the University of Pittsburgh, Pittsburgh, PA, USA. Since 2009, he has been with the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, South Korea. His areas of research include information theory, signal processing theory, communications/networking theory, and their application to wireless communications and networking, compressive sensing, future Internet, and brain–computer interface. He has received several prestigious national awards, including the Top 100 National Research and Development Award in 2012, the Top 50 Achievements of Fundamental Researches Award in 2013, and the Science/Engineer of the Month (January 2014).