Gradient based method for cone programming with application to large-scale compressed sensing

Zhaosong Lu*

September 3, 2008 (Revised: September 17, 2008)

Abstract

In this paper, we study a gradient based method for general cone programming (CP) problems. In particular, we first consider four natural primal-dual convex smooth minimization reformulations for them, and then discuss a variant of Nesterov's smooth (VNS) method recently proposed by Auslender and Teboulle [1] for solving these reformulations. The associated worst-case major arithmetic operations costs of the VNS method for them are estimated and compared. We show that for a class of CP problems, the VNS method based on the last reformulation generally outperforms that applied to the others. Finally, we discuss the application of the VNS method [1] to some large-scale CP problems arising in compressed sensing, which are highly challenging to simplex and/or interior point (IP) methods. The performance of this method is compared with the IP method [5] that is specially implemented for these problems. Our computational results demonstrate that for these CP problems, the VNS method [1] applied to the mostly suitable reformulation mentioned above substantially outperforms the IP method [5].

Keywords: Cone programming, variant of Nesterov's smooth method, compressed sensing

AMS 2000 subject classification: 65K05, 65K10, 90C05, 90C22, 90C25

1 Introduction

In [26, 27], Nesterov proposed an optimal algorithm for solving convex programming problems of the form

$$\bar{f} \equiv \inf\{f(u) : u \in \mathcal{U}\}, \tag{1}$$

*Department of Mathematics, Simon Fraser University, Burnaby, BC V5A 1S6, Canada (Email: zhaosong@sfu.ca). This author was supported in part by SFU President's Research Grant and NSERC Discovery Grant.

where f is a convex function with Lipschitz continuous gradient and $\mathcal{U}$ is a sufficiently simple closed convex set. It is shown that his method has $\mathcal{O}(\sqrt{L/\epsilon})$ iteration-complexity bound, where L is the Lipschitz constant for the gradient of f and $\epsilon > 0$ is the absolute precision of the final objective function value. In his method, each iterate needs to solve two proximal subproblems or one proximal subproblem plus one projection subproblem. Recently, Auslender and Teboulle [1] proposed a variant of Nesterov's smooth (VNS) method that has the same complexity as Nesterov's smooth method [26, 27], but it requires solving only one proximal subproblem per iteration. More recently, Tseng [31] extended the VNS method [1] to a broader class of convex optimization problems.

Lan et al. [23] recently studied first-order methods for general CP problems. In particular, they proposed a class of primal-dual convex (smooth and/or nonsmooth) minimization reformulations for them, and discussed suitable first-order methods for solving these reformulations such as Nesterov's optimal method [26, 27], Nesterov's smooth approximation scheme [27], Nemirovski's prox-method [25] and a variant of Nesterov's optimal method [23]. They also provided iteration-complexity bounds for these first-order methods when applied to the aforementioned reformulations of CP problems. Nevertheless, it is not clear which reformulation is mostly suitable for these methods since the performance of first-order methods generally vary with the different reformulations.

In this paper we study the VNS method [1] for general cone programming (CP) problems. In particular, we first consider four natural primal-dual convex smooth minimization reformulations (11)-(14) for them, and then discuss the VNS method [1] for solving these reformulations. The associated worst-case major arithmetic operations costs of the VNS method for them are established and compared. We show that for a class of CP problems, the VNS method based on reformulation (14) generally outperforms that applied to the others, namely, (11)-(13). Finally, we discuss the application of the VNS method [1] to some large-scale CP problems arising in compressed sensing, which are highly challenging to simplex and/or interior point (IP) methods. The performance of this method is compared with the IP method [5] that is specially implemented for these problems. Our computational results demonstrate that for these CP problems, the VNS method [1] applied to the mostly suitable reformulation mentioned above substantially outperforms the IP method [5].

The rest of paper is organized as follows. In Section 2, we introduce four natural primal-dual convex smooth minimization reformulations for cone programming problems. In Section 3, we discuss the VNS method for solving these reformulations. In Section 4, we discuss the application of the VNS method to some large-scale CP problems arising in compressed sensing. In Section 5, the performance of the VNS method is compared with the IP method [5] on a set of randomly generated compressed sensing instances. Finally, we present some concluding remarks in Section 6.

1.1 Notation

The following notation is used throughout our paper. All vector spaces given in this paper are assumed to be finite dimensional. The symbol $\Re^n$ denotes the n-dimensional Euclidean

space. The 1-norm, Euclidean norm and infinity-norm in $\Re^n$ are denoted by $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$, respectively. We denote by $\mathbf{1}$ the vector of all ones whose dimension should be clear from the context. Given a sequence of vectors $x^i \in \Re^{n_i}$ for $i = 1, \ldots, m$, let $(x^1; \cdots; x^m)$ denote the vector in $\Re^n$ with $n = \sum_{i=1}^m n_i$ obtained by stacking the vectors x^i one by one for $i = 1, \ldots, m$. We denote n-dimensional quadratic (or second-order) cone by $\mathcal{Q}^n$, that is,

$$\mathcal{Q}^n = \left\{ x \in \Re^n : x_1 \geq \sqrt{x_2^2 + \ldots + x_n^2} \right\}.$$

The space of all $m \times n$ matrices with real entries is denoted by $\Re^{m\times n}$. We denote by I the identity matrix whose size should be clear from the context. Given any $X \in \Re^{m\times n}$, we denote its operator norm by $\|X\|_2$, i.e., $\|X\|_2 = \max\{\|Xu\|_2 : \|u\|_2 \leq 1\}$. We define the condition number of a real nonsingular matrix X as $\kappa(X) = \|X\|_2\|X^{-1}\|_2$. Given an $n \times n$ real symmetric matrix X, let $\lambda_i(X)$ denote its ith largest eigenvalue for $i = 1, ..., n$, and $\lambda_{\min}(X)$ (resp., $\lambda_{\max}(X)$) denote its minimal (resp., maximal) eigenvalue.

Given a real Hilbert space U, we denote its inner product by $\langle\cdot,\cdot\rangle_U$, which gives rise to the inner product norm $\|\cdot\|_U$ on U, that is, $\|\cdot\|_U = \sqrt{\langle\cdot,\cdot\rangle_U}$. If V denotes another Hilbert space, and $\mathcal{E} : U \to V$ is a linear operator, the adjoint of $\mathcal{E}$ is the linear operator $\mathcal{E}^* : V \to U$ defined by

$$\langle \mathcal{E}^* v, u\rangle_U = \langle \mathcal{E}u, v\rangle_V, \quad \forall u \in U,\ v \in V.$$

With a slight abuse of notation, we sometimes write $\mathcal{E}$ and $\mathcal{E}^*$ as their representation matrices with respect to some natural (standard) basis. Further, the operator norm of $\mathcal{E}$ is defined as

$$\|\mathcal{E}\| = \max_u\{\|\mathcal{E}u\|_V : \|u\|_U \leq 1\}. \tag{2}$$

Also, if $\mathcal{E}$ is invertible, its condition number is defined as

$$\kappa(\mathcal{E}) = \|\mathcal{E}\|\|\mathcal{E}^{-1}\|. \tag{3}$$

In addition, for an operator $\mathcal{E}$, $\mathrm{Im}(\mathcal{E})$ denotes its range space.

Let U be a normed vector space whose norm is denoted by $\|\cdot\|$. The dual space of U, denoted by U^*, is the normed vector space consisting of all linear functionals of $u^* : U \to \Re$, endowed with the dual norm $\|\cdot\|^*$ defined as

$$\|u^*\|^* = \max_u\{u^*(u) :\ \|u\| \leq 1\}, \quad \forall u^* \in U^*. \tag{4}$$

A function $f : \Omega \subseteq U \to \Re$ is said to be L-Lipschitz-differentiable with respect to $\|\cdot\|$ if it is differentiable and

$$\|f'(u) - f'(\tilde{u})\|^* \leq L\|u - \tilde{u}\| \quad \forall u, \tilde{u} \in \Omega.$$

Given a closed convex set $\mathcal{C} \subseteq U$ and an arbitrary norm $\|\cdot\|$ on U, let $d_{\mathcal{C}} : U \to \Re$ denote the distance function for $\mathcal{C}$ measured in terms of $\|\cdot\|$, namely,

$$d_{\mathcal{C}}(u) := \inf_{\tilde{u}\in\mathcal{C}} \|u - \tilde{u}\|, \quad \forall u \in U. \tag{5}$$

2 Reformulations of cone programming

In [23], a class of primal-dual convex minimization reformulations for cone programming (CP) problems were proposed. In this section, we consider four natural reformulations among them.

Assume X and Y are two real Hilbert spaces. Given a linear operator $\mathcal{A} : X \to Y$, $c \in X$ and $b \in Y$ and a closed convex cone $\mathcal{L} \subseteq X$, consider the CP problem

$$\begin{array}{ll} \min\limits_{x} & \langle c, x\rangle_X \\ \text{s.t.} & \mathcal{A}x = b, \quad x \in \mathcal{L}, \end{array} \tag{6}$$

and its associated dual problem

$$\begin{array}{ll} \max\limits_{(y,s)} & \langle b, y\rangle_Y \\ \text{s.t.} & \mathcal{A}^* y + s = c, \quad s \in \mathcal{L}^*, \end{array} \tag{7}$$

where $\mathcal{L}^* := \{s \in X : \langle s, x\rangle_X \geq 0, \, \forall x \in \mathcal{L}\}$ is the dual cone of $\mathcal{L}$. We make the following assumption throughout the paper.

Assumption 1 *The pair of CP problems (6) and (7) have optimal solutions and their associated duality gap is zero.*

In view of the above assumption, a primal-dual optimal solution of (6) and (7) can be found by solving the following constrained system of linear equations:

$$\begin{array}{rcll} \mathcal{A}x - b & = & 0, & \\ \mathcal{A}^* y + s - c & = & 0, & (x, s, y) \in \mathcal{L} \times \mathcal{L}^* \times Y. \\ \langle c, x\rangle_X - \langle b, y\rangle_Y & = & 0, & \end{array} \tag{8}$$

The primal-dual system (8) is a special case of the cone linear system (CLS) described as follows. Let U and V denote two real Hilbert spaces. Given a linear operator $\mathcal{E} : U \to V$, a vector $e \in V$, and a closed convex cone $\mathcal{K} \subseteq U$, the general CLS consists of finding a vector u such that

$$\mathcal{E}u - e = 0, \quad u \in \mathcal{K}. \tag{9}$$

Indeed, (8) can be viewed as a special case of (9) by letting $U = X \times X \times Y$, $V = Y \times X \times \Re$,

$$\mathcal{K} = \mathcal{L} \times \mathcal{L}^* \times Y, \quad \mathcal{E} = \begin{pmatrix} \mathcal{A} & 0 & 0 \\ 0 & \mathcal{I} & \mathcal{A}^* \\ c & 0 & -b \end{pmatrix}, \quad u = \begin{bmatrix} x \\ s \\ y \end{bmatrix}, \text{ and } e = \begin{bmatrix} b \\ c \\ 0 \end{bmatrix},$$

where $\mathcal{I}$ denotes the identity operator.

Henceforth, we will consider the more general problem (9), for which we assume a solution exists. Our approach to solving this problem will be to reformulate it as a smooth convex minimization problem.

Let $\mathcal{M}$ be the linear manifold defined as

$$\mathcal{M} = \{u \in U : \mathcal{E}u - e = 0\}. \tag{10}$$

Let $d_{\mathcal{K}}(\cdot)$ and $d_{\mathcal{M}}(\cdot)$ denote the distance functions for $\mathcal{K}$ and $\mathcal{M}$ measured in terms of the norm $\|\cdot\|_U$, respectively. We easily see that problem (9) can be reformulated as the following minimization problems:

$$\min\{f_1(u) := \|\mathcal{E}u - e\|_V^2 : \ u \in \mathcal{K}\}, \tag{11}$$

$$\min\{f_2(u) := (d_{\mathcal{M}}(u))^2 + (d_{\mathcal{K}}(u))^2 : \ u \in U\}, \tag{12}$$

$$\min\{f_3(u) := (d_{\mathcal{M}}(u))^2 : \ u \in \mathcal{K}\}, \tag{13}$$

$$\min\{f_4(u) := (d_{\mathcal{K}}(u))^2 : \ u \in \mathcal{M}\}. \tag{14}$$

We next show that the objective functions of problems (11)-(14) are convex differentiable and their gradients are Lipschitz continuous, and hence (11)-(14) are smooth convex minimization reformulations of problem (9).

Proposition 1 *The functions $f_1(u)$, $f_2(u)$, $f_3(u)$ and $f_4(u)$ are convex and $2\|\mathcal{E}\|^2$, 4, 2, 2-Lipschitz differentiable with respect to the norm $\|\cdot\|_U$, respectively, where $\|\mathcal{E}\|$ is defined in (2).*

Proof. The conclusion immediately follows from Propositions 1 and 15 of [23]. ∎

We shall remark that (11)-(14) are special cases of a class of smooth convex minimization reformulations proposed in [23] for problem (9). In particular, (11) has been well studied in [23], and Nemirovski's prox-method [25], Nesterov's smooth method [26, 27] and its variant [23] were applied to solve it. In addition, (12) has been recently studied in Jarre and Rendl [21], where nonlinear conjugate gradient methods were applied to solve it.

In this paper, we discuss the VNS method [1] for solving CP problems (6) and (7) based on (11)-(14), and estimate the associated worst-case major arithmetic operation costs for them. We show that for a broad class of CP problems including Lovász capacity [24], the semidefinite programming relaxation of MAXCUT [18] and those arising in compressed sensing [30, 12, 5, 6, 7, 8, 10, 11, 9], it is computationally cheaper to solve them by applying the VNS method [1] to (14) than (11)-(13).

3 A variant of Nesterov's smooth method

In this section, we review a variant Nesterov's smooth (VNS) method recently proposed by Auslender and Teboulle [1] for solving a class of smooth convex programming problems. We also discuss its convergence results when applied to problems (11)-(14).

Let U be a normed vector space with the norm denoted by $\|\cdot\|$, and let $\mathcal{U} \subseteq U$ be a closed convex set. Assume that $f : \mathcal{U} \to \Re$ is a differentiable convex function such that for some $L \geq 0$,

$$\|\nabla f(u) - \nabla f(\tilde{u})\|^* \leq L\|u - \tilde{u}\|, \quad \forall u, \tilde{u} \in \mathcal{U}. \tag{15}$$

Our problem of interest in this section is the convex programming problem (1).

We assume throughout our discussion that the optimal value f^* of problem (1) is finite and that its set of optimal solutions is nonempty. Let $h_U : \mathcal{U} \to \Re$ be a differentiable strongly convex function with modulus $\sigma_u > 0$ with respect to $\|\cdot\|_U$, i.e.,

$$h_U(u) \geq h_U(\tilde{u}) + \langle \nabla h_U(\tilde{u}), u - \tilde{u}\rangle + \frac{\sigma_u}{2}\|u - \tilde{u}\|^2, \quad \forall u, \tilde{u} \in \mathcal{U}. \tag{16}$$

The Bregman distance $d_{h_U} : \mathcal{U} \times \mathcal{U} \to \Re$ associated with h_U is defined as

$$d_{h_U}(u; \tilde{u}) = h_U(u) - l_{h_U}(u; \tilde{u}), \quad \forall u, \tilde{u} \in \mathcal{U}, \tag{17}$$

where $l_{h_U} : U \times \mathcal{U} \to \Re$ is the "linear approximation" of h_U defined as

$$l_{h_U}(u; \tilde{u}) = h_U(\tilde{u}) + \langle \nabla h_U(\tilde{u}), u - \tilde{u}\rangle, \quad \forall (u, \tilde{u}) \in U \times \mathcal{U}.$$

We are now ready to state the VNS method proposed by Auslender and Teboulle [1] for solving (1).

Variant of Nesterov's Smooth (VNS) Method:

0) Let $\bar{u}_0 = \tilde{u}_0 \in \mathcal{U}$ be given and set $k = 0$.

1) Set $u_k = \frac{2}{k+2}\bar{u}_k + \frac{k}{k+2}\tilde{u}_k$ and compute $f(u_k)$ and $\nabla f(u_k)$.

2) Compute $(\tilde{u}_{k+1}, \bar{u}_{k+1}) \in \mathcal{U} \times \mathcal{U}$ as

$$\bar{u}_{k+1} = \operatorname{argmin}\left\{\frac{k+2}{2} l_f(u; u_k) + \frac{L}{\sigma_u} d_{h_U}(u; \bar{u}_k) : u \in \mathcal{U}\right\}, \tag{18}$$

$$\tilde{u}_{k+1} = \frac{2}{k+2}\bar{u}_{k+1} + \frac{k}{k+2}\tilde{u}_k. \tag{19}$$

3) Set $k \leftarrow k + 1$ and go to step 1.

end

The main convergence result established by Auslender and Teboulle [1] regarding the above algorithm is summarized in the following theorem (see also Tseng [31]).

Theorem 2 *The sequence $\{\tilde{u}_k\}$ generated by the above variant of Nesterov's algorithm satisfies*

$$f(\tilde{u}_k) - f^* \leq \frac{4L\, d_{h_U}(u^*; \tilde{u}_0)}{\sigma_u\,(k+1)^2}, \quad \forall k \geq 1, \tag{20}$$

where u^* is an optimal solution of (1).

We observe the above VNS method requires solving only one proximal subproblem per iteration, but Nesterov's smooth algorithm [27] needs to solve two proximal subproblems or one projection subproblem plus one proximal subproblem per iteration. Also, the above algorithm shares the same worst-case iteration complexity with Nesterov's smooth algorithm [27]. Therefore, the VNS method [1] is generally more computationally efficient.

The following results are immediate consequences of Theorem 2 regarding the behavior of the VNS method applied to (11)-(14) for solving the general CLS (9).

Corollary 3 *Let $\{\tilde{u}_k\}$ be the sequence generated by the VNS method applied to (11). Given any $\epsilon > 0$, an iterate $\tilde{u}_k \in \mathcal{K}$ satisfying $\|\mathcal{E}\tilde{u}_k - e\|_V \leq \epsilon$ can be found in no more than*

$$\left\lceil \frac{2\sqrt{2}\|\mathcal{E}\|}{\epsilon}\sqrt{\frac{d_{h_U}(u^*;\tilde{u}_0)}{\sigma_u}} \right\rceil$$

iterations, where u^* is a solution of (9) and $\|\mathcal{E}\|$ is defined in (2).

Proof. Note that $f(u) = \|\mathcal{E}u - e\|_V^2$ for all $u \in \mathcal{K}$, $f^* = 0$, and that any solution u^* of (9) is also an optimal solution of (11). Further, in view of Proposition 1, we know that the function f is $2\|\mathcal{E}\|^2$-Lipschitz-differentiable with respect to $\|\cdot\|_U$. Using these facts and Theorem 2, we see that

$$\|\mathcal{E}\tilde{u}_k - e\|_V^2 \leq \frac{8\|\mathcal{E}\|^2\, d_{h_U}(u^*;\tilde{u}_0)}{\sigma_u\,(k+1)^2}, \quad \forall k \geq 1.$$

The corollary then follows immediately from the above relation. ∎

We can similarly show that the following three results hold.

Corollary 4 *Let $\{\tilde{u}_k\}$ be the sequence generated by the VNS method applied to (12). Given any $\epsilon > 0$, an iterate $\tilde{u}_k \in U$ satisfying $\sqrt{(d_{\mathcal{M}}(\tilde{u}_k))^2 + (d_{\mathcal{K}}(\tilde{u}_k))^2} \leq \epsilon$ can be found in no more than*

$$\left\lceil \frac{4}{\epsilon}\sqrt{\frac{d_{h_U}(u^*;\tilde{u}_0)}{\sigma_u}} \right\rceil$$

iterations, where u^* is a solution of (9).

Corollary 5 *Let $\{\tilde{u}_k\}$ be the sequence generated by the VNS method applied to (13). Given any $\epsilon > 0$, an iterate $\tilde{u}_k \in \mathcal{K}$ satisfying $d_{\mathcal{M}}(\tilde{u}_k) \leq \epsilon$ can be found in no more than*

$$\left\lceil \frac{2\sqrt{2}}{\epsilon}\sqrt{\frac{d_{h_U}(u^*;\tilde{u}_0)}{\sigma_u}} \right\rceil$$

iterations, where u^* is a solution of (9).

Corollary 6 *Let $\{\tilde{u}_k\}$ be the sequence generated by the VNS method applied to (14). Given any $\epsilon > 0$, an iterate $\tilde{u}_k \in \mathcal{M}$ satisfying $d_{\mathcal{K}}(\tilde{u}_k) \leq \epsilon$ can be found in no more than*

$$\left\lceil \frac{2\sqrt{2}}{\epsilon}\sqrt{\frac{d_{h_U}(u^*;\tilde{u}_0)}{\sigma_u}} \right\rceil$$

iterations, where u^* is a solution of (9).

From Corollaries 3-6, we see that the ϵ-optimal solutions $\tilde{u}_k$ obtained by the VNS method for (11)-(14) satisfy

$$u \in U, \quad \sqrt{(d_{\mathcal{M}}(u))^2 + (d_{\mathcal{K}}(u))^2} \leq \epsilon, \tag{21}$$

and hence, with respect to this termination criterion, they are also ϵ-optimal solutions of (9). However, we shall notice that the worst-case iteration complexity of the VNS method when applied to (11)-(14) can differ from each other. In next section, we will discuss which one of (11)-(14) is mostly suitable for the VNS method [1] to solve (9) in the context of cone programming.

4 Variant of Nesterov's smooth method for cone programming

In this section, we discuss the variant of Nesterov's smooth (VNS) method [1] described in Section 3 for solving cone programming (CP) problems (6) and (7). In particular, the worst-case major arithmetic operations costs of the VNS method when applied to (11)-(14) in the context of CP are estimated and compared.

Throughout this section, we focus on the VNS method for solving CP problems (6) and (7), or equivalently, a special class of CLS (9) with $U \equiv X \times X \times Y$, $V \equiv Y \times X \times \Re$,

$$\mathcal{K} \equiv \mathcal{L} \times \mathcal{L}^* \times Y, \quad \mathcal{E} \equiv \begin{pmatrix} \mathcal{A} & 0 & 0 \\ 0 & \mathcal{I} & \mathcal{A}^* \\ c & 0 & -b \end{pmatrix}, \quad u \equiv \begin{bmatrix} x \\ s \\ y \end{bmatrix}, \text{ and } e \equiv \begin{bmatrix} b \\ c \\ 0 \end{bmatrix}. \tag{22}$$

The norms $\|\cdot\|_U$ and $\|\cdot\|_V$ on the vector spaces U and V are specialized as follows:

$$\|u\|_U = \sqrt{\|u_x\|_X^2 + \|u_s\|_X^2 + \|u_y\|_Y^2}, \quad \forall u = (u_x; u_s; u_y) \in U, \tag{23}$$

$$\|v\|_V = \sqrt{\|v_p\|_Y^2 + \|v_d\|_X^2 + v_o^2}, \quad \forall v = (v_p; v_d; v_o) \in V, \tag{24}$$

and the strongly convex function $h_U(\cdot)$ for the VNS method is chosen as

$$h_U(u) = \frac{1}{2}\|u\|_U^2, \quad \forall u \in U. \tag{25}$$

Given a closed set $C \subseteq U$, let $\Pi_C : U \to C$ be the projection map with respect to the norm $\|\cdot\|_U$, that is,

$$\Pi_C(u) = \operatorname{argmin}\{\|u - \tilde{u}\|_U : \ \tilde{u} \in C\}, \quad \forall u \in U. \tag{26}$$

We adopt the same convention as in Golub and Van Loan [19] for counting flops, that is, a flop is a floating point operation (e.g., the inner product of two n-vectors involves $2n$ flops).

Definition 1 *For any map $\xi : Z \to W$ between two sets Z and W, and for each $z \in Z$, we denote by $\mathrm{fps}(\xi(z))$ the number of flops required to compute $\xi(z) \in W$. Moreover, we define $\mathrm{fps}(\xi) = \sup\{\mathrm{fps}(\xi(z)) : z \in Z\}$.*

We clearly see that $\mathrm{fps}(\mathcal{A})$, $\mathrm{fps}(\mathcal{A}^*)$, $\mathrm{fps}(b)$ and $\mathrm{fps}(c)$ are finite. In general, $\mathrm{fps}(\mathcal{A})$ and $\mathrm{fps}(\mathcal{A}^*)$ are much larger than $\mathrm{fps}(b)$ and $\mathrm{fps}(c)$. Therefore for simplicity, we do not count $\mathrm{fps}(b)$ and $\mathrm{fps}(c)$ in any worst-case major arithmetic operations cost given in this section.

The worst-case major arithmetic operations costs of the VNS method when applied to (11)-(14) in the context of CP are estimated as follows.

Theorem 7 *Let $\mathcal{M}$, $\mathcal{E}$ and $\mathcal{K}$ be defined in (10) and (22), respectively. The worst-case major arithmetic operations costs of the VNS method when applied to (11)-(14) for finding an ϵ-optimal solution of CP problems (6) and (7) based on the termination criterion (21) are*

$$\mathcal{C}_1(\epsilon) := [2(\mathrm{fps}(\mathcal{A}) + \mathrm{fps}(\mathcal{A}^*)) + \mathrm{fps}(\Pi_{\mathcal{K}})] \left\lceil \frac{2\sqrt{2}\|\mathcal{E}\| \, \|u^* - \tilde{u}_0^1\|_U}{\epsilon} \right\rceil, \tag{27}$$

$$\mathcal{C}_2(\epsilon) := [\mathrm{fps}(\Pi_{\mathcal{M}}) + \mathrm{fps}(\Pi_{\mathcal{K}})] \left\lceil \frac{4\|u^* - \tilde{u}_0^2\|_U}{\epsilon} \right\rceil, \tag{28}$$

$$\mathcal{C}_3(\epsilon) := [\mathrm{fps}(\Pi_{\mathcal{M}}) + \mathrm{fps}(\Pi_{\mathcal{K}})] \left\lceil \frac{2\sqrt{2}\|u^* - \tilde{u}_0^3\|_U}{\epsilon} \right\rceil, \tag{29}$$

$$\mathcal{C}_4(\epsilon) := [\mathrm{fps}(\Pi_{\mathcal{M}}) + \mathrm{fps}(\Pi_{\mathcal{K}})] \left\lceil \frac{2\sqrt{2}\|u^* - \tilde{u}_0^4\|_U}{\epsilon} \right\rceil, \tag{30}$$

respectively, where u^* is an optimal solution of CP problems (6) and (7), and $\tilde{u}_0^1 \in \mathcal{K}$, $\tilde{u}_0^2 \in U$, $\tilde{u}_0^3 \in \mathcal{K}$ and $\tilde{u}_0^4 \in \mathcal{M}$ are the initial points for the VNS method for (11)-(14), respectively.

Proof. In view of (11)-(14), (5), (26) and Proposition 15 of [23], we have

$$\begin{array}{ll} \nabla f_1(u) = 2\mathcal{E}^*(\mathcal{E}u - e), & \nabla f_2(u) = 2(2u - \Pi_{\mathcal{M}}(u) - \Pi_{\mathcal{K}}(u)), \\ \nabla f_3(u) = 2(u - \Pi_{\mathcal{M}}(u)), & \nabla f_4(u) = 2(u - \Pi_{\mathcal{K}}(u)). \end{array}$$

Using (25) and the first relation above, we see that when applied to (11) for solving CP problems (6) and (7), the worst-case major arithmetic operations cost per iteration of the

VNS method includes: $2(\mathrm{fps}(\mathcal{A}) + \mathrm{fps}(\mathcal{A}^*))$ for computing $\nabla f_1(\cdot)$ in step 1); and $\mathrm{fps}(\Pi_{\mathcal{K}})$ for solving the proximal subproblem in step 2) with $h_U(\cdot)$ given in (25). Together with Corollary 3, we see that the worst-case major arithmetic operations cost of the VNS method when applied to (11) for finding an ϵ-optimal solution of CP problems (6) and (7) is given by (27). Similarly, we can show the remaining conclusions hold. ■

As observed in our computational experiment, the actual number of iterations performed by the VNS method is generally proportional to its worst-case iteration complexity given in Theorem 2. Thus, the worst-case major arithmetic operations costs estimated in Theorem 7 can be used to compare the performance of the VNS method when applied to (11)-(14). In order to compare $\mathcal{C}_1(\epsilon)$, $\mathcal{C}_2(\epsilon)$, $\mathcal{C}_3(\epsilon)$ and $\mathcal{C}_4(\epsilon)$, we assume throughout this section that $\tilde{u}_0^1 = \tilde{u}_0^3 = \Pi_{\mathcal{K}}(\tilde{u}_0^2)$ and $\tilde{u}_0^4 = \Pi_{\mathcal{M}}(\tilde{u}_0^1)$ for some $\tilde{u}_0^2 \in U$. These together with $u^* \in \mathcal{K} \cap \mathcal{M}$ immediately imply that

$$\|u^* - \tilde{u}_0^4\|_U \;\le\; \|u^* - \tilde{u}_0^3\|_U \;=\; \|u^* - \tilde{u}_0^1\|_U \;\le\; \|u^* - \tilde{u}_0^2\|_U.$$

In view of this relation, (28) and (29), we conclude that $\mathcal{C}_4(\epsilon) \le \mathcal{C}_3(\epsilon) \le \mathcal{C}_2(\epsilon)$, and hence it is more efficient to solve (14) than (12) and (13) by the VNS method for finding an ϵ-optimal solution of CP problems (6) and (7) based on the termination criterion (21).

In the rest of this section, our goal is to compare $\mathcal{C}_4(\epsilon)$ with $\mathcal{C}_1(\epsilon)$. Before proceeding, we make another assumption throughout this section regarding CP problems (6) and (7).

Assumption 2 $\mathrm{Im}(\mathcal{A}) = Y$ *and* $b \neq 0$.

The assumption $\mathrm{Im}(\mathcal{A}) = Y$ is quite standard, which has often been imposed in the literature. Also, the assumption $b \neq 0$ is fairly mild. Indeed, for the case where $b = 0$, the CP problems (6) and (7) become trivial, and their solutions can be obtained straightforwardly.

We now establish some useful properties for the operators $\mathcal{A}\mathcal{A}^*$ and $\mathcal{E}\mathcal{E}^*$.

Proposition 8 *Let $\mathcal{E}$ be defined in (22). Under Assumption 2, the following statements hold:*

i) The operator $\mathcal{A}\mathcal{A}^*$ is invertible;

ii) The operator $\mathcal{E}\mathcal{E}^*$ is invertible, and moreover,

$$(\mathcal{E}\mathcal{E}^*)^{-1} = \mathcal{H}^*\mathcal{G}\mathcal{H}, \tag{31}$$

where

$$\mathcal{H} \;=\; \begin{pmatrix} \mathcal{I} & 0 & 0 \\ 0 & \mathcal{I} & 0 \\ -c\mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^{-1} & b\mathcal{A}(I + \mathcal{A}^*\mathcal{A})^{-1} & \mathcal{I} \end{pmatrix}, \tag{32}$$

$$\mathcal{G} \;=\; \begin{pmatrix} (\mathcal{A}\mathcal{A}^*)^{-1} & 0 & 0 \\ 0 & (\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1} & 0 \\ 0 & 0 & \xi^{-1} \end{pmatrix}, \tag{33}$$

$$\xi \;=\; cc^* + bb^* - c\mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}c^* - b\mathcal{A}(I + \mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*b^*. \tag{34}$$

Proof. Let $y \in Y$ be such that $\mathcal{A}\mathcal{A}^* y = 0$. Then we have

$$\|\mathcal{A}^* y\|_X^2 = \langle \mathcal{A}^* y,\ \mathcal{A}^* y\rangle_X = \langle y,\ \mathcal{A}\mathcal{A}^* y\rangle_Y = 0,$$

and hence $\mathcal{A}^* y = 0$. It leads to

$$\langle \mathcal{A}x,\ y\rangle_Y = \langle x,\ \mathcal{A}^* y\rangle_X = 0, \quad \forall x \in X,$$

which together with the assumption $\mathrm{Im}(\mathcal{A}) = Y$, implies that $\langle y,\ y\rangle_Y = 0$, and hence $y = 0$. It follows that $\mathcal{A}\mathcal{A}^*$ is injective. Now let $\{y^i\}_{i=1}^m$ be a basis for the space Y. As shown above, $\mathcal{A}\mathcal{A}^*(\sum_{i=1}^m \alpha_i y^i) = 0$ for some $\{\alpha_i\}_{i=1}^m$ yields $\sum_{i=1}^m \alpha_i y^i = 0$. The latter relation implies $\alpha_i = 0$ for $i = 1, \ldots, m$. Thus, we immediately see that $\{\mathcal{A}\mathcal{A}^* y^i\}_{i=1}^m$ is linearly independent, and $\{\mathcal{A}\mathcal{A}^* y^i\}_{i=1}^m$ is a basis for Y, which leads to $\mathcal{A}\mathcal{A}^* Y = Y$, and hence $\mathcal{A}\mathcal{A}^*$ is surjective. The statement i) immediately follows.

In view of the definition of $\mathcal{E}$ (see (22)), we obtain that

$$\mathcal{E}u = \begin{pmatrix} \mathcal{A}u_x \\ u_s + \mathcal{A}^* u_y \\ cu_x - bu_y \end{pmatrix}, \quad \forall u = \begin{pmatrix} u_x \\ u_s \\ u_y \end{pmatrix} \in U.$$

The above relation together with Assumption 2 immediately implies that $\mathrm{Im}(\mathcal{E}) = V$. Using this result and following a similar proof as in statement i), we conclude that $\mathcal{E}\mathcal{E}^*$ is invertible. Further, we see from (22) that

$$\mathcal{E}^* = \begin{pmatrix} \mathcal{A}^* & 0 & c^* \\ 0 & \mathcal{I} & 0 \\ 0 & \mathcal{A} & -b^* \end{pmatrix}, \quad \mathcal{E}\mathcal{E}^* = \begin{pmatrix} \mathcal{A}\mathcal{A}^* & 0 & \mathcal{A}c^* \\ 0 & \mathcal{I} + \mathcal{A}^*\mathcal{A} & -\mathcal{A}^* b^* \\ c\mathcal{A}^* & -b\mathcal{A} & cc^* + bb^* \end{pmatrix}.$$

In view of this relation and (32), it is easy to verify that

$$\mathcal{H}\mathcal{E}\mathcal{E}^*\mathcal{H}^* = \begin{pmatrix} \mathcal{A}\mathcal{A}^* & 0 & 0 \\ 0 & \mathcal{I} + \mathcal{A}^*\mathcal{A} & 0 \\ 0 & 0 & \xi \end{pmatrix}. \tag{35}$$

Noticing that $\mathcal{H}$, $\mathcal{H}^*$ and $\mathcal{E}\mathcal{E}^*$ are invertible, we conclude from (35) that ξ^{-1} exists. By taking inverse on both sides of (35), we immediately see that (31) holds. ∎

In order to compare $\mathcal{C}_4(\epsilon)$ with $\mathcal{C}_1(\epsilon)$, it is clear from (27) and (30) that we need to relate $\mathrm{fps}(\Pi_{\mathcal{M}})$ with $\mathrm{fps}(\mathcal{A})$ and $\mathrm{fps}(\mathcal{A}^*)$. We next aim to build a relation between them.

In view of (26), (10) and (23), we can easily show that

$$\Pi_{\mathcal{M}}(u) = u + \mathcal{E}^*(\mathcal{E}\mathcal{E}^*)^{-1}(e - \mathcal{E}u),$$

which together with the definition of $\mathcal{E}$ implies that

$$\mathrm{fps}(\Pi_{\mathcal{M}}) \approx 2\,(\mathrm{fps}(\mathcal{A}) + \mathrm{fps}(\mathcal{A}^*)) + \mathrm{fps}((\mathcal{E}\mathcal{E}^*)^{-1}). \tag{36}$$

We now need to estimate $\mathrm{fps}((\mathcal{E}\mathcal{E}^*)^{-1})$. First, we note from (34) that ξ only depends on $\mathcal{A}$, b, c and their adjoint operators, and thus it needs to be computed only once. The major work for obtaining ξ is to compute:

$$\tau := (\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}c^*, \quad \delta := (\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*b^*, \tag{37}$$

$\mathcal{A}^*\tau$ and $\mathcal{A}\delta$. We see that the major arithmetic operations cost for computing ξ, τ and δ is

$$2\left(\mathrm{fps}(\mathcal{A}) + \mathrm{fps}(\mathcal{A}^*)\right) + \mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1}) + \mathrm{fps}((\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1}).$$

From now on, we assume that ξ, τ and δ are computed beforehand as above. We are now ready to estimate $\mathrm{fps}((\mathcal{E}\mathcal{E}^*)^{-1})$ as follows.

Lemma 9 *The following holds:*

$$\mathrm{fps}((\mathcal{E}\mathcal{E}^*)^{-1}) \approx \mathrm{fps}(\mathcal{A}) + \mathrm{fps}(\mathcal{A}^*) + \mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1}) + \mathrm{fps}((\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1}). \tag{38}$$

Proof. In view of (31), we see that for any $v = (v_p; v_d; v_o) \in V$, $z = (\mathcal{E}\mathcal{E}^*)^{-1}v$ can be computed according to the following steps: 1) compute $h = \mathcal{H}v$; 2) compute $w = \mathcal{G}h$; and 3) compute $z = \mathcal{H}^*w$. We now analyze the main computation for them in details. For step 1), using (32) and the relation $h = \mathcal{H}v$, we obtain $h = (v_p; v_d; h_o)$, where

$$h_o = -c\mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^{-1}v_p + b\mathcal{A}(\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1}v_d + v_o.$$

We clearly see that the main computation for step 1) is to compute:

$$\varrho := (\mathcal{A}\mathcal{A}^*)^{-1}v_p, \quad \vartheta := (\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1}v_d, \tag{39}$$

$\mathcal{A}^*\varrho$ and $\mathcal{A}\vartheta$. Hence, the worst-case major arithmetic operations cost for step 1) is

$$\mathrm{fps}(\mathcal{A}) + \mathrm{fps}(\mathcal{A}^*) + \mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1}) + \mathrm{fps}((\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1}).$$

For step 2), using (33), (39) and the relations $h = (v_p; v_d; h_o)$ and $w = \mathcal{G}h$, we obtain $w = (w_p; w_d; w_o)$, where

$$w_p = (\mathcal{A}\mathcal{A}^*)^{-1}v_p = \varrho, \quad w_d = (\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1}v_d = \vartheta, \quad w_o = \xi^{-1}h_o.$$

Since ϱ, ϑ and h_o are computed in step 1), w is readily available, and so there is almost no computational cost for step 2). Finally, for step 3), using (32), (37) and the relation $z = \mathcal{H}^*w$, we obtain $z = (z_p; z_d; w_o)$, where

$$\begin{aligned} z_p &= w_p - (\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}c^*w_o = w_p - \tau w_o, \\ z_d &= w_d + (\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1}\mathcal{A}^*b^*w_o = w_d + \delta w_o. \end{aligned}$$

Noticing that w_p, w_d and w_o are obtained in step 2), there is almost no computational cost for step 3). Summing up the computational costs for steps 1)-3), we immediately see the conclusion holds. ∎

The following result provides an estimate of $\mathrm{fps}((\mathcal{I} + \mathcal{A}^*\mathcal{A})^{-1})$.

Lemma 10 *The following holds:*

$$\mathrm{fps}((\mathcal{I}+\mathcal{A}^*\mathcal{A})^{-1}) \approx \mathrm{fps}(\mathcal{A}) + \mathrm{fps}(\mathcal{A}^*) + \mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1}). \tag{40}$$

Proof. By applying Sherman-Morrison-Woodbury formula in the context of operators, we have

$$(\mathcal{I}+\mathcal{A}^*\mathcal{A})^{-1} = \mathcal{I} - \mathcal{A}^*(\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A},$$

which immediately leads to the conclusion. ∎

In view of (36), (38) and (40), we now obtain an estimate of $\mathrm{fps}(\Pi_{\mathcal{M}})$ in terms of $\mathcal{A}$ and $\mathcal{A}^*$ as follows.

Lemma 11 *The following holds:*

$$\mathrm{fps}(\Pi_{\mathcal{M}}) \approx 4(\mathrm{fps}(\mathcal{A}) + \mathrm{fps}(\mathcal{A}^*)) + \mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1}) + \mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1}). \tag{41}$$

Using (30) and (41), we finally obtain an estimate of $\mathcal{C}_4(\epsilon)$ as follows.

Theorem 12 *Let $\mathcal{C}_4(\epsilon)$ be defined in (30). We have*

$$\mathcal{C}_4(\epsilon) \approx [4(\mathrm{fps}(\mathcal{A})+\mathrm{fps}(\mathcal{A}^*)) + \mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1}) + \mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1}) + \mathrm{fps}(\Pi_{\mathcal{K}})] \left\lceil \frac{2\sqrt{2}\|u^* - \tilde{u}_0^4\|_U}{\epsilon} \right\rceil \tag{42}$$

for some $\tilde{u}_0^4 \in \mathcal{M}$.

We now compare $\mathcal{C}_1(\epsilon)$ and $\mathcal{C}_4(\epsilon)$ that are given in (27) and (42), respectively. First, we recall from the above assumption that $\tilde{u}_0^4 = \Pi_{\mathcal{M}}(\tilde{u}_0^1)$, and hence $\|u^* - \tilde{u}_0^4\|_U \leq \|u^* - \tilde{u}_0^1\|_U$. Also, we obtain from the definition of $\mathcal{E}$ (see (22)) that $\|\mathcal{E}\| \geq 1$, and moreover, in view of (23) and (24), we have

$$\begin{aligned}
\|\mathcal{E}\| &= \max\left\{ \left\| \begin{pmatrix} \mathcal{A} & 0 & 0 \\ 0 & \mathcal{I} & \mathcal{A}^* \\ c & 0 & -b \end{pmatrix} \begin{pmatrix} x \\ s \\ y \end{pmatrix} \right\|_V : \left\| \begin{pmatrix} x \\ s \\ y \end{pmatrix} \right\|_U \leq 1 \right\}, \\
&\leq \max\left\{ \left\| \begin{pmatrix} \mathcal{A} \\ 0 \\ c \end{pmatrix} x \right\|_V + \left\| \begin{pmatrix} 0 \\ \mathcal{I} \\ 0 \end{pmatrix} s \right\|_V + \left\| \begin{pmatrix} 0 \\ \mathcal{A}^* \\ -b \end{pmatrix} y \right\|_V : \left\| \begin{pmatrix} x \\ s \\ y \end{pmatrix} \right\|_U \leq 1 \right\}, \\
&= \max\left\{ \sqrt{\|\mathcal{A}x\|_Y^2 + \langle c, x\rangle_X^2} + \|s\|_X + \sqrt{\|\mathcal{A}^* y\|_X^2 + \langle -b, y\rangle_Y^2} : \left\| \begin{pmatrix} x \\ s \\ y \end{pmatrix} \right\|_U \leq 1 \right\}, \\
&\leq \max\left\{ \sqrt{\|\mathcal{A}\|^2 + \|c\|^2}\ \|x\|_X + \|s\|_X + \sqrt{\|\mathcal{A}^*\|^2 + \|b\|^2}\ \|y\|_Y : \left\| \begin{pmatrix} x \\ s \\ y \end{pmatrix} \right\|_U \leq 1 \right\}, \\
&\leq \sqrt{\|\mathcal{A}\|^2 + \|c\|^2 + 1 + \|\mathcal{A}^*\|^2 + \|b\|^2}.
\end{aligned}$$

Thus, we see that $\|\mathcal{E}\|$ can be as large as the quantity given on the right-hand side of the above inequality, which can be much larger than 1. Based on the above discussion, we conclude from (27) and (42) that $\mathcal{C}_4(\epsilon)$ can be much less than $\mathcal{C}_1(\epsilon)$ provided that $\mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1})$ and $\mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1})$ are comparable to $\mathrm{fps}(\mathcal{A})+\mathrm{fps}(\mathcal{A}^*)$. Roughly speaking, $\mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1})$ and $\mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1})$ are the arithmetic operations cost for solving linear systems:

$$\mathcal{A}\mathcal{A}^*v = h, \quad (\mathcal{I}+\mathcal{A}\mathcal{A}^*)v = h \tag{43}$$

for any $h \in Y$, respectively. We now look into two cases where $\mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1})$ and $\mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1})$ are $\mathcal{O}(\mathrm{fps}(\mathcal{A})+\mathrm{fps}(\mathcal{A}^*))$.

1) $\kappa(\mathcal{A}\mathcal{A}^*)$ is small. Noting that $\kappa(\mathcal{I}+\mathcal{A}\mathcal{A}^*) \le \kappa(\mathcal{A}\mathcal{A}^*)$, so $\kappa(\mathcal{I}+\mathcal{A}\mathcal{A}^*)$ is also small. Then the number of iterations performed by the conjugate gradient (CG) method for solving linear systems (43) is also relatively small. Moreover, we easily see that the arithmetic operations cost of CG method per iteration is $\mathcal{O}(\mathrm{fps}(\mathcal{A})+\mathrm{fps}(\mathcal{A}^*))$. Therefore, in this case, $\mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1})$ and $\mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1})$ are $\mathcal{O}(\mathrm{fps}(\mathcal{A})+\mathrm{fps}(\mathcal{A}^*))$ when CG method is applied to solve linear systems (43).

2) $\mathcal{A}\mathcal{A}^*$ is a diagonal operator. In this case, linear systems (43) can be trivially solved, and $\mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1})$ and $\mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1})$ are even much less than $\mathrm{fps}(\mathcal{A})+\mathrm{fps}(\mathcal{A}^*)$.

Based on the above discussion, we have the following main conclusion.

Proposition 13 *When* $\mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1})$ *and* $\mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1})$ *are comparable to* $\mathrm{fps}(\mathcal{A})+\mathrm{fps}(\mathcal{A}^*)$*, it can be much more efficient to solve (14) than (11)-(13) by the VNS method for finding an* ϵ*-optimal solution of CP problems (6) and (7) based on the termination criterion (21). Especially, when* $\mathcal{A}\mathcal{A}^*$ *is a block diagonal operator in which each diagonal block is either a diagonal operator or an operator with a small conditional number,* $\mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1})$ *and* $\mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1})$ *are comparable to* $\mathrm{fps}(\mathcal{A})+\mathrm{fps}(\mathcal{A}^*)$.

We remark that there are some important CP problems from application for which $\mathrm{fps}((\mathcal{A}\mathcal{A}^*)^{-1})$ and $\mathrm{fps}((\mathcal{I}+\mathcal{A}\mathcal{A}^*)^{-1})$ are comparable to $\mathrm{fps}(\mathcal{A})+\mathrm{fps}(\mathcal{A}^*)$. For example, for Lovász capacity [24], the semidefinite programming relaxation of MAXCUT [18], $\mathcal{A}\mathcal{A}^*$ is a diagonal operator. In next section, we will see that several CP problems arising in compressed sensing also fall into the aforementioned category.

5 Application to compressed sensing

In this section, we discuss the application of the VNS method [1] described in Section 3 to several important problems arising in compressed sensing.

5.1 Problems arising in compressed sensing

In this subsection, we describe several important problems arising in compressed sensing that are of our interest.

One of the central problems in compressed sensing is how to decode a large sparse signal using a relatively small number of linear measurements or observations. Given the data consisting of the matrix $A \in \Re^{m\times n}$ and the vector $\mathtt{b}$, and a nonnegative parameter λ, the following models except the last one were recently proposed in the literature [6, 7, 8, 10, 11, 9, 30, 12] for recovering signal from highly incomplete information. We name them similarly as in [5, 30, 12].

1. Min-l_1 bounded residual correlation:
$$\min_x\{\|x\|_1 :\ \|A^T(Ax-\mathtt{b})\|_\infty \le \lambda\}. \tag{44}$$

2. Min-l_1 with equality constraints:
$$\min_x\{\|x\|_1 :\ Ax=\mathtt{b}\}. \tag{45}$$

3. Minimum l_1 error approximation:
$$\min_x \|A^Tx-\mathtt{b}\|_1. \tag{46}$$

4. Min-l_1 with quadratic constraints:
$$\min_x\{\|x\|_1 :\ \|Ax-\mathtt{b}\|_2 \le \lambda\}. \tag{47}$$

5. LASSO:
$$\min_x \left\{\frac{1}{2}\|Ax-\mathtt{b}\|_2^2 :\ \|x\|_1 \le \lambda\right\}. \tag{48}$$

6. Basis pursuit de-noising:
$$\min_x \lambda\|x\|_1 + \frac{1}{2}\|Ax-\mathtt{b}\|_2^2. \tag{49}$$

7. LASSO Lagrangian relaxation:
$$\min_x \lambda\|x\|_1 + \|Ax-\mathtt{b}\|_2. \tag{50}$$

We now provide a brief interpretation for the above models. For the details, see [5, 30, 12] and the references therein. The first model, also referred to as the *Dantzig Selector*, was proposed by Candès and Tao [11], which provides a sparse vector x^* such that the residual $Ax^*-\mathtt{b}$ is not too correlated with any of the columns of A. The second model, also known as *basis pursuit*, was studied in [7, 10, 12]. Candès et al. [7] and Candès and Tao [10] show that

if there exist a sufficiently sparse x^* such that $Ax^* = \mathtt{b}$, then model (45) will find it. The third model arises in the context of channel coding (see Candès and Tao [9]), which provides a vector $x^* \in \Re^m$ such that the difference between $A^T x^*$ and $\mathtt{b}$ is sparse. The fourth model was studied in Candès et al. [8], which finds a sparse vector for closely explaining the observations. The fifth model, often referred to as *Least Absolute Shrinkage and Selection Operator* (LASSO), was proposed by Tibshirani [30], and the sixth model was studied in Chen et al. [12]. The motivation of the last three models is similar to that of the fourth model. It is not hard to observe that the last four models are equivalent to each other in the sense that a solution of one model is also that of another one for some parameter λ, and vice versa.

For the above models, the data matrix A is usually either a Gaussian random matrix (that is, its elements are independently drawn from the standard normal distribution) or has all rows randomly chosen from an orthonormal matrix such as the fast Fourier transform (FFT), discrete cosine transform (DCT) or wavelets transform matrix. It is often fully dense and large-scale in applications of interest. Moreover, the matrix A usually has full row rank, and $\kappa(AA^T)$ is usually small. Indeed, when A is a Gaussian random matrix, it follows from a well-known random matrix theory (see [14], for example) that A has full row rank with probability one, and

$$n\left(1-\sqrt{\frac{m}{n}}\right)^2 \le \lambda_i(AA^T) \le n\left(1+\sqrt{\frac{m}{n}}\right)^2,\ i=1,\ldots,m,$$

and hence,

$$\kappa(AA^T) = \frac{\lambda_{\max}(AA^T)}{\lambda_{\min}(AA^T)} \le \left(\frac{1+\sqrt{m/n}}{1-\sqrt{m/n}}\right)^2$$

with prevailing probability for large n. In applications, usually $m \ll n$, and thus, $\kappa(AA^T)$ is often nearly one. When A is a partial orthonormal matrix mentioned above, we have $AA^T = I$ and hence, $\kappa(AA^T) = 1$.

5.2 Variant of Nesterov's smooth method for compressed sensing

In this subsection, we discuss the application of the variant of Nesterov's smooth (VNS) method [1] described in Section 3 to problems (45)-(50). In particular, we first reformulate them into CP problems in the form of either (6) or (7), and then apply the VNS method to solve the latter problems based on a suitable reformulation.

Throughout this subsection, all vector spaces are in fact Euclidean spaces and their inner products are standard ones. As seen from Section 4, the nontrivial computational parts of the VNS method when applied to CP problems (6) and (7) based on formulation (14) are possibly two projections: one is projection of a point into $\mathcal{K}$, and another one is projection of a point into $\mathcal{M}$, where $\mathcal{M}$ and $\mathcal{K}$ are given in (10) and (22), respectively. Also, as shown in Section 4, the latter projection can be reduced to two linear systems given in (43). For these reasons, when we discuss the VNS method for CP problems (6) and (7) based on formulation (14), we only focus on how to efficiently solve the first projection mentioned above and two linear systems given in (43).

5.2.1 Variant of Nesterov's smooth method for problem (44)

We can show that problem (44) can be reformulated as

$$\begin{array}{rrcrcl} \max\limits_{x^+,x^-} & -\mathbf{1}^T x^+ & - & \mathbf{1}^T x^- & & \\ \text{s.t.} & A^T A x^+ & - & A^T A x^- & \leq & \lambda \mathbf{1} + A^T \mathtt{b}, \\ & -A^T A x^+ & + & A^T A x^- & \leq & \lambda \mathbf{1} - A^T \mathtt{b}, \\ & -x^+ & & & \leq & 0, \\ & & & -x^- & \leq & 0. \end{array} \tag{51}$$

Clearly, (51) is a CP problem in the form of (7) with $\mathcal{L} = \Re^{4n}_+$, $X = \Re^{4n}$, $Y = \Re^{2n}$, and

$$\mathcal{A} = \begin{bmatrix} A^T A & -A^T A & -I & 0 \\ -A^T A & A^T A & 0 & -I \end{bmatrix}, \qquad b = \begin{pmatrix} -\mathbf{1} \\ -\mathbf{1} \end{pmatrix}, \quad c = (\lambda \mathbf{1} + A^T \mathtt{b}; \lambda \mathbf{1} - A^T \mathtt{b}; 0; 0).$$

Note that $\mathcal{K} = \mathcal{L} \times \mathcal{L}^* \times Y = \Re^{4n}_+ \times \Re^{4n}_+ \times \Re^{2n}$. Hence, projection of a point into $\mathcal{K}$ is trivial and cheap.

We next show how the linear systems in (43) can be efficiently solved. First, we easily see that

$$\mathcal{A}\mathcal{A}^* = I + \begin{bmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{bmatrix}^2 = I + 2 \begin{bmatrix} A^T A A^T \\ -A^T A A^T \end{bmatrix} [A \quad -A].$$

By applying Sherman-Morrison-Woodbury formula for the above identity, we obtain that

$$\begin{aligned} (\mathcal{A}\mathcal{A}^*)^{-1} &= I - 2 \begin{bmatrix} A^T A A^T \\ -A^T A A^T \end{bmatrix} \left(I + 2[A \quad -A] \begin{bmatrix} A^T A A^T \\ -A^T A A^T \end{bmatrix} \right)^{-1} [A \quad -A], \\ &= I - 2 \begin{bmatrix} A^T A A^T \\ -A^T A A^T \end{bmatrix} \left(I + 4(AA^T)^2 \right)^{-1} [A \quad -A]. \end{aligned} \tag{52}$$

Similarly, we have

$$(\mathcal{I} + \mathcal{A}\mathcal{A}^*)^{-1} = \frac{1}{2} I - \frac{1}{2} \begin{bmatrix} A^T A A^T \\ -A^T A A^T \end{bmatrix} \left(I + 2(AA^T)^2 \right)^{-1} [A \quad -A]. \tag{53}$$

When the matrix A has all rows randomly chosen from an orthonormal matrix such as FFT, DCT or wavelets transform matrix, we have $AA^T = I$. Using this fact, (52) and (53), we see that

$$\begin{aligned} (\mathcal{A}\mathcal{A}^*)^{-1} &= I - \frac{2}{5} \begin{bmatrix} A^T \\ -A^T \end{bmatrix} [A \quad -A], \\ (\mathcal{I} + \mathcal{A}\mathcal{A}^*)^{-1} &= \frac{1}{2} I - \frac{1}{6} \begin{bmatrix} A^T \\ -A^T \end{bmatrix} [A \quad -A], \end{aligned}$$

and hence, the linear systems in (43) can be trivially solved in this case. When A is a Gaussian random matrix, we know from Subsection 5.1 that $\kappa(AA^T)$ is often small, and so $\kappa(I + 4(AA^T)^2)$ and $\kappa(I + 2(AA^T)^2)$ are also often small. Using this fact, (52) and (53), we observe that the linear systems in (43) can be reduced to the ones with the coefficient matrices $I + 4(AA^T)^2$ and $I + 2(AA^T)^2$, and the latter linear systems can be efficiently solved by CG method. Further, the arithmetic operations cost of CG method per iteration is $\mathcal{O}(\text{fps}(\mathcal{A}) + \text{fps}(\mathcal{A}^*))$. Thus, $\text{fps}((\mathcal{A}\mathcal{A}^*)^{-1})$ and $\text{fps}((\mathcal{I} + \mathcal{A}\mathcal{A}^*)^{-1})$ are comparable to $\text{fps}(\mathcal{A}) + \text{fps}(\mathcal{A}^*)$. In view of Proposition 13, CP problem (51) (or equivalently (44)) can be more efficiently solved by the VNS method based on (14) than (11)-(13).

5.2.2 Variant of Nesterov's smooth method for problem (45)

We observe that problem (45) can be reformulated as

$$\begin{array}{ll} \min\limits_{x^+,x^-} & \mathbf{1}^T x^+ + \mathbf{1}^T x^- \\ \text{s.t.} & Ax^+ - Ax^- = \mathtt{b}, \\ & x^+ \geq 0,\ x^- \geq 0. \end{array} \tag{54}$$

Clearly, (54) is a CP problem in the form of (6) with

$$\mathcal{A} = [A\ -A], \quad b = \mathtt{b}, \quad c = (\mathbf{1}; \mathbf{1}), \quad \mathcal{L} = \Re^{2n}_+, \quad X = \Re^{2n}, \quad Y = \Re^m.$$

Note that $\mathcal{K} = \mathcal{L} \times \mathcal{L}^* \times Y = \Re^{2n}_+ \times \Re^{2n}_+ \times \Re^m$. Hence, projection of a point into $\mathcal{K}$ is trivial and cheap. Also, we easily see that $\mathcal{A}\mathcal{A}^* = 2AA^T$. When the matrix A has all rows randomly chosen from an orthonormal matrix such as FFT, DCT or wavelets transform matrix, we immediately have $\mathcal{A}\mathcal{A}^* = 2I$. When A is a Gaussian random matrix, we know from Subsection 5.1 that $\kappa(AA^T)$ is often small, which together with the relation $\mathcal{A}\mathcal{A}^* = 2AA^T$ implies that $\kappa(\mathcal{A}\mathcal{A}^*)$ is also often small. Therefore, in view of Proposition 13, CP problem (54) (or equivalently (45)) can be more efficiently solved by the VNS method based on (14) than (11)-(13). Also, the linear systems in (43) can be efficiently solved directly or by CG method.

5.2.3 Variant of Nesterov's smooth method for problem (46)

We observe that problem (46) can be reformulated as

$$\begin{array}{lll} \max\limits_{x,u} & -\mathbf{1}^T u & \\ \text{s.t.} & A^T x - u & \leq \mathtt{b}, \\ & -A^T x - u & \leq -\mathtt{b}. \end{array} \tag{55}$$

Clearly, (55) is a CP problem in the form of (7) with

$$\mathcal{A} = \begin{bmatrix} A & -A \\ -I & -I \end{bmatrix}, \quad b = \begin{pmatrix} 0 \\ -\mathbf{1} \end{pmatrix}, \quad c = \begin{pmatrix} \mathtt{b} \\ -\mathtt{b} \end{pmatrix}, \quad \mathcal{L} = \Re^{2n}_+, \quad X = \Re^{2n}, \quad Y = \Re^{m+n}.$$

Note that $\mathcal{K} = \mathcal{L} \times \mathcal{L}^* \times Y = \Re^{2n}_+ \times \Re^{2n}_+ \times \Re^{m+n}$. Hence, projection of a point into $\mathcal{K}$ is trivial and cheap. Also, we easily see that

$$\mathcal{A}\mathcal{A}^* = \begin{bmatrix} 2AA^T & 0 \\ 0 & 2AA^T \end{bmatrix}.$$

Using a similar argument as in Subsection 5.2.2, we have either $\mathcal{A}\mathcal{A}^* = 2I$ or $\kappa(\mathcal{A}\mathcal{A}^*)$ is often small. In view of Proposition 13, CP problem (55) (or equivalently (46)) can be more efficiently solved by the VNS method based on (14) than (11)-(13). Also, the linear systems in (43) can be efficiently solved directly or by CG method.

5.2.4 Variant of Nesterov's smooth method for problem (47)

We can show that problem (47) can be reformulated as

$$\begin{aligned} \min_{x^+,x^-,t,u} \quad & \mathbf{1}^T x^+ + \mathbf{1}^T x^- \\ \text{s.t.} \quad & \begin{bmatrix} 0 & 0 & -1 & 0 \\ A & -A & 0 & -I \end{bmatrix} \begin{pmatrix} x^+ \\ x^- \\ t \\ u \end{pmatrix} = \begin{pmatrix} -\lambda \\ \mathtt{b} \end{pmatrix}, \\ & x^+ \geq 0,\ x^- \geq 0,\ (t;u) \in \mathcal{Q}^{m+1}. \end{aligned} \tag{56}$$

Clearly, (56) is a CP problem in the form of (6) with $\mathcal{L} = \Re^{2n}_+ \times \mathcal{Q}^{m+1}$, $X = \Re^{m+2n+1}$, $Y = \Re^{m+1}$, and

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ A & -A & 0 & -I \end{bmatrix}, \quad b = \begin{pmatrix} -\lambda \\ \mathtt{b} \end{pmatrix}, \quad c = (\mathbf{1}; \mathbf{1}; 0; 0).$$

Note that $\mathcal{K} = \mathcal{L} \times \mathcal{L}^* \times Y = \Re^{2n}_+ \times \mathcal{Q}^{m+1} \times \Re^{2n}_+ \times \mathcal{Q}^{m+1} \times \Re^{m+1}$. Hence, projection of a point into $\mathcal{K}$ can be reduced to projection of a point into $\mathcal{Q}^{m+1}$, which can be cheaply computed according to the following proposition.

Proposition 14 *Let $\Pi_{\mathcal{Q}^{m+1}}(\cdot)$ be defined in (26). Given a point $u := (t; x) \in \Re^{m+1}$, we have*

$$\Pi_{\mathcal{Q}^{m+1}}(u) = \begin{cases} (t; x), & \text{if } t \geq \|x\|_2; \\ (0; 0), & \text{if } t \leq -\|x\|_2; \\ \frac{t+\|x\|_2}{2}\left(1; \frac{x}{\|x\|_2}\right), & \text{otherwise.} \end{cases}$$

Proof. The conclusion follows from Fact 2.3 of [2] by letting $\alpha = 1$. ∎

We also easily see that

$$\mathcal{A}\mathcal{A}^* = \begin{bmatrix} 1 & 0 \\ 0 & I + 2AA^T \end{bmatrix}. \tag{57}$$

When the matrix A has all rows randomly chosen from an orthonormal matrix such as FFT, DCT or wavelets transform matrix, we have $AA' = I$, and hence $\mathcal{A}\mathcal{A}^*$ becomes a diagonal operator. When A is a Gaussian random matrix, we know from Subsection 5.1 that $\kappa(AA^T)$ is often small, and so $\kappa(I + 2AA^T)$ is also often small. Using this fact and (57), we observe that $\mathcal{A}\mathcal{A}^*$ is a block diagonal operator consisting of one diagonal block and another block with a small conditional number. In view of Proposition 13, CP problem (56) (or equivalently (47)) can be more efficiently solved by the VNS method based on (14) than (11)-(13). Also, the linear systems in (43) can be efficiently solved directly or by CG method.

5.2.5 Variant of Nesterov's smooth method for problem (48)

We can show that problem (48) can be reformulated as

$$\begin{array}{cc} \min\limits_{x^+,x^-,r,t_1,t_2,u} & 2t_1 \\ \text{s.t.} & \begin{array}{rl} \mathbf{1}^T x^+ + \mathbf{1}^T x^- + r & = \lambda, \\ t_1 - t_2 & = 2, \\ Ax^+ - Ax^- - u & = \mathtt{b}, \\ x^+ \geq 0,\ x^- \geq 0, & r \geq 0,\ (t_1; t_2; u) \in \mathcal{Q}^{m+2}. \end{array} \end{array} \tag{58}$$

Clearly, (58) is a CP problem in the form of (6) with $\mathcal{L} = \Re_+^{2n+1} \times \mathcal{Q}^{m+2}$, $X = \Re^{m+2n+3}$, $Y = \Re^{m+2}$, and

$$\mathcal{A} = \begin{bmatrix} \mathbf{1}^T & \mathbf{1}^T & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ A & -A & 0 & 0 & 0 & -I \end{bmatrix}, \quad b = \begin{pmatrix} \lambda \\ 2 \\ \mathtt{b} \end{pmatrix}, \quad c = (0; 0; 0; 2; 0; 0).$$

Note that $\mathcal{K} = \mathcal{L} \times \mathcal{L}^* \times Y = \Re_+^{2n+1} \times \mathcal{Q}^{m+2} \times \Re_+^{2n+1} \times \mathcal{Q}^{m+2} \times \Re^{m+2}$. Hence, projection of a point into $\mathcal{K}$ can be reduced to projection of a point into $\mathcal{Q}^{m+2}$, which can be cheaply computed according to Proposition 14.

We also easily see that

$$\mathcal{A}\mathcal{A}^* = \begin{bmatrix} 2n+1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & I + 2AA^T \end{bmatrix}.$$

Using this identity and a similar argument as in Subsection 5.2.4, we observe that $\mathcal{A}\mathcal{A}^*$ is either a diagonal operator or a block diagonal operator consisting of one diagonal block and another block with a small conditional number. In view of Proposition 13, CP problem (58) (or equivalently (48)) can be more efficiently solved by the VNS method based on (14) than (11)-(13). Also, the linear systems in (43) can be efficiently solved directly or by CG method.

5.2.6 Variant of Nesterov's smooth method for problem (49)

We can show that problem (49) can be reformulated as

$$\begin{array}{ll} \min\limits_{x^+,x^-,t_1,t_2,u} & \lambda \mathbf{1}^T x^+ + \lambda \mathbf{1}^T x^- + 2t_1 \\ \text{s.t.} & \begin{array}{rl} t_1 - t_2 & = 2, \\ Ax^+ - Ax^- - u & = \mathtt{b}, \\ x^+ \geq 0,\ x^- \geq 0, & (t_1; t_2; u) \in \mathcal{Q}^{m+2}. \end{array} \end{array} \tag{59}$$

Clearly, (59) is a CP problem in the form of (6) with $\mathcal{L} = \Re^{2n}_+ \times \mathcal{Q}^{m+2}$, $X = \Re^{m+2n+2}$, $Y = \Re^{m+1}$, and

$$\mathcal{A} = \begin{bmatrix} 0 & 0 & 1 & -1 & 0 \\ A & -A & 0 & 0 & -I \end{bmatrix}, \quad b = \begin{pmatrix} 2 \\ \mathtt{b} \end{pmatrix}, \quad c = (\lambda \mathbf{1}; \lambda \mathbf{1}; 2; 0; 0).$$

Note that $\mathcal{K} = \mathcal{L} \times \mathcal{L}^* \times Y = \Re^{2n}_+ \times \mathcal{Q}^{m+2} \times \Re^{2n}_+ \times \mathcal{Q}^{m+2} \times \Re^{m+1}$. Hence, projection of a point into $\mathcal{K}$ can be reduced to projection of a point into $\mathcal{Q}^{m+2}$, which can be cheaply computed according to Proposition 14.

We also easily see that

$$\mathcal{A}\mathcal{A}^* = \begin{bmatrix} 2 & 0 \\ 0 & I + 2AA^T \end{bmatrix}.$$

Using this identity and a similar argument as in Subsection 5.2.4, we observe that $\mathcal{A}\mathcal{A}^*$ is either a diagonal operator or a block diagonal operator consisting of one diagonal block and another block with a small conditional number. In view of Proposition 13, CP problem (59) (or equivalently (49)) can be more efficiently solved by the VNS method based on (14) than (11)-(13). Also, the linear systems in (43) can be efficiently solved directly or by CG method.

5.2.7 Variant of Nesterov's smooth method for problem (50)

We can show that problem (50) can be reformulated as

$$\begin{array}{ll} \min\limits_{x^+,x^-,t_1,t_2,u} & \lambda \mathbf{1}^T x^+ + \lambda \mathbf{1}^T x^- + t \\ \text{s.t.} & \begin{array}{rl} Ax^+ - Ax^- - u & = \mathtt{b}, \\ x^+ \geq 0,\ x^- \geq 0, & (t; u) \in \mathcal{Q}^{m+1}. \end{array} \end{array} \tag{60}$$

Clearly, (60) is a CP problem in the form of (6) with $\mathcal{L} = \Re^{2n}_+ \times \mathcal{Q}^{m+1}$, and $X = \Re^{m+2n+1}$, $Y = \Re^m$, and

$$\mathcal{A} = [A \quad -A \quad 0 \quad I], \quad b = \mathtt{b}, \quad c = (\lambda \mathbf{1}; \lambda \mathbf{1}; 1; 0).$$

Note that $\mathcal{K} = \mathcal{L} \times \mathcal{L}^* \times Y = \Re^{2n}_+ \times \mathcal{Q}^{m+1} \times \Re^{2n}_+ \times \mathcal{Q}^{m+1} \times \Re^m$. Hence, projection of a point into $\mathcal{K}$ can be reduced to projection of a point into $\mathcal{Q}^{m+1}$, which can be cheaply computed according to Proposition 14.

We also easily see that $\mathcal{A}\mathcal{A}^* = I + 2AA^T$. Using this identities and a similar argument as in Subsection 5.2.2, we observe that $\mathcal{A}\mathcal{A}^*$ is either a diagonal operator or an operator with a small conditional number. In view of Proposition 13, CP problem (60) (or equivalently (50)) can be more efficiently solved by the VNS method based on (14) than (11)-(13). Also, the linear systems in (43) can be efficiently solved directly or by CG method.

6 Computational results

As indicated by Friedlander and Sauders [17], problem (44), namely, the *Dantzig selector* is extremely challenging to simplex and interior point (IP) methods because of its high dimension and full density. They also mentioned that the IP method implemented in l_1-magic [5] generally outperforms simplex methods. In this section, we compare the performance of the variant of Nesterov's smooth (VNS) method [1] described in Section 3 with the IP method [5] for solving problem (44) on a set of randomly generated instances.

As discussed in Subsection 5.2.1, the cone programming (CP) reformulation of problem (44) is a special case of (7), where X and Y are Euclidean spaces and their inner products are the natural ones. Thus, all norms used in the VNS method are just Euclidean norms. We choose $\tilde{u}_0^1 = \tilde{u}_0^2 = \tilde{u}_0^3 := (x_0; s_0; y_0) = (0; 0; 0)$ and $\tilde{u}_0^4 = \Pi_{\mathcal{M}}(\tilde{u}_0^1)$ as the initial points for the VNS method when applied to (11)-(14), respectively. The initial point for the IP method [5] is chosen by default. It shall be mentioned that for this IP method, conjugate gradient method is applied to solve the associated Newton system. The codes for these methods are written in Matlab.

In addition, the following termination criterion will be used for the VNS and IP methods when applied to CP problems (6) and (7). For a given tolerance $\epsilon > 0$, we want to find a point $(x, s, y) \in \mathcal{L} \times \mathcal{L}^* \times Y$ such that

$$\max\{\|\mathcal{A}^* y + s - c\|_X,\ \|\mathcal{A}x - b\|_Y,\ |\langle c, x\rangle - \langle b, y\rangle|\} \le \epsilon, \tag{61}$$

All instances for problem (44) are randomly generated in the same manner as described in l_1-magic [5]. Given $\sigma > 0$, and positive integers m, n, T with $m < n$ and $T < n$, we now describe how to generate data A and $\mathtt{b}$ for problem (44). We first generate a matrix $W \in \Re^{n\times m}$ with random entries chosen from a normal distribution with mean zero, variance one and standard deviation one. Then we compute an orthonormal basis, denoted by B, for the range space of W, and set $A = B^T$. We also randomly generate a vector $\tilde{x} \in \Re^n$ with only T nonzero components that are ± 1, and generate a vector $v \in \Re^m$ with random entries chosen from a normal distribution with mean zero, variance one and standard deviation one. Finally, set $\mathtt{b} = A\tilde{x} + \sigma v$. Especially, we choose $\sigma = 0.005$ for our experiment. All computations are performed on an Intel Xeon 5320 CPU (1.86GHz) and 12GB RAM running Red Hat Enterprise Linux 4 (kernel 2.6.9).

In the first experiment, we apply the VNS method to solve problem (51), namely, the CP reformulation of problem (44). In particular, we compare the performance of the VNS method for (51) based on reformulations (11)-(14) (labeled as VNS1, VNS2, VNS3 and VNS4, respectively). The termination criterion (61) with $\epsilon = 0.1$ is used for them. More specifically, suppose that $\tilde{u}^k$ is an approximate solution obtained at the kth iteration by these four approaches. Then $\tilde{u}^k$ is used to check the termination criterion for VNS1 and VNS3 while $u_k = \Pi_{\mathcal{K}}(\tilde{u}^k)$ is used for VNS2 and VNS4. The performance of VNS1-VNS4 for randomly generated relatively small-scale instances of problem (44) are presented in Table 1. The parameters m, n and T of the instances are listed in columns one to three of Table 1, respectively. The numbers of iterations of VNS1-VNS4 are given in columns four to seven, and CPU times (in seconds) are

Table 1: Comparison of VNS1–VNS4

Problem			Iter				Time			
m	n	T	VNS1	VNS2	VNS3	VNS4	VNS1	VNS2	VNS3	VNS4
120	512	20	3967	227	121	109	4.6	0.6	0.3	0.2
240	1024	40	6577	233	142	112	18.7	1.5	1.0	0.6
360	1536	60	9713	289	172	138	101.0	7.3	5.0	3.1
480	2048	80	12608	337	193	160	301.4	19.3	13.1	8.2
600	2560	100	14492	334	198	161	540.0	30.1	21.0	13.2
720	3072	120	18318	387	230	185	962.1	49.7	34.9	21.5
840	3584	140	18518	358	214	168	1301.2	61.8	43.7	26.4
960	4096	160	20629	353	223	167	1897.3	79.6	59.7	34.0
1080	4608	180	21809	364	222	169	2466.7	101.4	73.3	42.8
1200	5120	200	25494	402	247	189	3547.0	137.4	100.9	58.4

given in the last four columns, respectively. From Table 1, we conclude that VNS4, that is, the VNS method based on formulation (14), is the most efficient one among these four approaches for solving problem (51) (or equivalently (44)). Indeed, this phenomenon is consistent with the theoretical prediction given in Proposition 13.

From the above experiment, we have already seen that VNS4 outperforms VNS1-VNS3. In the second experiment, we compare the performance of VNS4 with the IP method implemented in l_1-magic [5] (labeled as IP). The termination criterion (61) with $\epsilon = 0.1$ is used for both methods. The performance of VNS4 and IP for randomly generated relatively large-scale instances of problem (44) are presented in Table 2. The parameters m, n and T of the instances are listed in columns one to three of Table 2, respectively. The numbers of iterations of VNS4 and IP are given in columns four to five, and CPU times (in seconds) are given in the last two columns, respectively. From Table 2, we conclude that VNS4, that is, the VNS method based on formulation (14) substantially outperforms the IP method [5] for solving problem (44) (or equivalently (44)). We shall also mention that the former method requires much less memory than the latter one.

7 Concluding remarks

In this paper, we studied the variant of Nesterov's smooth (VNS) method [1] for cone programming (CP) problems based on four natural primal-dual convex smooth minimization reformulations (11)-(14). We showed that for a class of CP problems, the VNS method based on (14) is most efficient among these reformulations. Also, we applied the VNS method to solve large-scale CP problems arising in compressed sensing, which are highly challenging to simplex and/or interior point (IP) methods. The computational results showed that the VNS method based on (14) substantially outperforms the IP method [5].

The Matlab codes for the VNS method based on formulation (14) for problems (44)-(47) are available online at www.math.sfu.ca/∼zhaosong. Since problems (48)-(50) are similar to

Table 2: Comparison of VNS4 and IP

Problem			Iter		Time	
m	n	T	VNS4	IP	VNS4	IP
1560	6656	260	199	23	103.3	1999.0
1680	7168	280	196	23	116.1	2483.7
1800	7680	300	209	24	142.9	3173.7
1920	8192	320	212	24	165.2	3839.7
2040	8704	340	219	23	193.8	4395.7
2160	9216	360	217	23	208.1	5207.1
2280	9728	380	212	24	227.9	6359.9
2400	10240	400	227	24	270.1	7413.8
3600	15360	600	247	27	670.4	27598.2
4800	20480	800	282	28	1326.9	67734.5

problem (47), the codes for them can be directly obtained by modifying that of (47). Regarding problems (48) and (49), there are quite a few efficient methods (see [29, 15, 32, 22, 16, 20, 13, 28, 3, 33] and the references therein) that greatly utilize their special structures. In contrast with them, the VNS method [1] discussed in this paper is a more general one. It would be interesting to compare the VNS method with these methods for problems (48) and (49). In addition, our current Matlab codes can only apply to the problems for which A is a matrix, but they can be easily generalized to handle the case where A is an operator.

Spectral projected gradient (SPG) methods (see [4] and the references therein) are alternative gradient based methods for smooth convex minimization problems. They have recently been successfully applied to solve problem (49) (see [16] for example) though their iteration complexity bounds are unknown so far. Clearly, the SPG methods can be suitably applied to solve CP problems (6) and (7) based on formulations (11)-(14). We would like to see how the performance of the SPG methods vary with these formulations, and also compare their performance with that of the VNS method [1] for CP problems.

References

[1] A. Auslender and M. Teboulle. Interior gradient and proximal methods for convex and conic optimization. *SIAM Journal on Optimization*, 16:697–725, 2006.

[2] H. Bauschke and S. Kruk. The method of reflection-projection for convex feasibility problems with an obtuse cone. *Journal of Optimization Theory and Applications*, 120(3):503–531, 2004.

[3] J. Bioucas-Dias and M. Figueiredo. Two-step algorithms for linear inverse problems with non-quadratic regularization, 2007. IEEE International Conference on Image Processing C ICIP 2007.

[4] E. Birgin, J. Martínez, and M. Raydon. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4), 1999.

[5] E. Candès and J. Romberg. 1_1-magic : Recovery of sparse signals via convex programming. User's guide, Applied & Computational Mathematics, California Institute of Technology, Pasadena, CA 91125, USA, October 2005.

[6] E. Candes and J. Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Foundations of Computational Mathematics*, 6(2):227–254, 2006.

[7] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[8] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2006.

[9] E. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[10] E. Candès and T. Tao. Near-optimal signal recovery from random projections and universal encoding strategies. *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[11] E. Candès and T. Tao. The dantzig selector: statistical estimation when p is much smaller than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

[12] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1999.

[13] I. Daubechies, M. Fornasier, and I. Loris. Accelerated projected gradient method for linear inverse problems with sparsity constraints. Manuscript, Princeton University, August 2007. Preprint on arXiv:0706:4297.

[14] A. Edelman. Eigenvalues and condition numbers of random matrices. *SIAM Journal on Matrix Analysis and Applications*, 9(4):543–560, 1988.

[15] B. Efron, T. Hastie, I.Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[16] 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 Journal on Selected Topics in Signal Processing*, 1(4), 2007.

[17] M. P. Friedlander and M. A. Saunders. Discussion: The dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2385–2391, 2007.

[18] M. X. Goemans and D. P. Williamson. .878-approximation algorithms for MAX CUT and MAX 2SAT. *Proceedings of the Symposium of Theoretical Computer Science*, pages 422–431, 1994.

[19] G. H. Golub and C. E. Van Loan. *Matrix Computations: Second Edition*. The John Hopkins University Press, Baltimore, MA 21211, 1989.

[20] E. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for l1-minimization: methodology and convergence. Manuscript, Department of Computational and Applied Mathematics, Rice University, 2007.

[21] F. Jarre and F. Rendl. An augmented primal-dual method for linear conic programs. *SIAM Journal on Optimization*, 19(2), 2008.

[22] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. A method for large-scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):606–617, 2007.

[23] G. Lan, Z. Lu, and R. D. C. Monteiro. Primal-dual first-order methods for cone programming. Manuscript, Department of Mathematics, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada, December 2006. Conditionally accepted in *Mathematical Programming*.

[24] L. Lovász. On the Shannon Capacity of a graph. *IEEE Transactions of Information Theory*, IT-25(1):1–7, January 1979.

[25] A. Nemirovski. Prox-method with rate of convergence $o(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15:229–251, 2005.

[26] Y. E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR*, 269:543–547, 1983. translated as Soviet Math. Docl.

[27] Y. E. Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103:127–152, 2005.

[28] Y. E. Nesterov. Gradient methods for minimizing composite objective function. Core discussion paper 2007/76, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, September 2007.

[29] M. Osborne, B. Presnell, and B. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20:389–403, 2000.

[30] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistics Society B*, 58(1):267–288, 1996.

[31] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Manuscript, Department of Mathematics, University of Washington, Seattle, WA, 98195, USA, May 2008. Submitted to SIAM Journal on Optimization.

[32] E. Van den Berg and M. P. Friedlander. In pursuit of a root. Manuscript, University of British Columbia, 2007. UBC Computer Science Technical Report TR-2007-16.

[33] S. Yun and K.-C. Toh. A coordinate gradient descent method for l1-regularized convex minimization. Preprint, National University of Singapore, May 2008.