An adaptive accelerated first-order method for convex optimization

Renato D.C. Monteiro [*]   Camilo Ortiz[†]   Benar F. Svaiter [‡]

July 30, 2012

Abstract

In this paper, we present a new accelerated variant of Nesterov's method for solving a class of convex optimization problems, in which certain acceleration parameters are adaptively (and aggressively) chosen so as to: preserve the theoretical iteration-complexity of the original method, and; substantially improve its practical performance in comparison to the other existing variants. Computational results are presented to demonstrate that the proposed adaptive accelerated method performs quite well compared to other variants proposed earlier in the literature.

1 Introduction

In his seminal paper [11] (see also [13]), Nesterov presented a scheme for accelerating first-order methods, more specifically, the steepest descent method for unconstrained convex optimization and the projected gradient method for constrained convex optimization. He shows that the rate of convergence of these methods, namely $\mathcal{O}(1/k)$, where k denotes the iteration count, can be improved to $\mathcal{O}(1/k^2)$ by considering their accelerated variants. Due to its wide use for solving large-scale convex optimization problems arising in several applications, other accelerated variants of Nesterov's method (see for example [1, 3, 5, 6, 8, 11, 12, 14, 15, 18]) have been proposed and studied in the literature. In particular, there has been an increasing concern of improving the practical performance of these methods in the solution of large-scale convex optimization problems. *In this paper, we present a new accelerated variant for solving a class of convex optimization problems, in which certain acceleration parameters are adaptively (and aggressively) chosen so as to: 1) preserve the theoretical iteration-complexity of the original method, and; 2) substantially improve its practical performance in comparison to the other existing variants.*

Our paper is organized as follows. In Section 2, we present a generic framework for analyzing adaptive accelerated methods for convex minimization problems and derive two bounds on the functional gap, namely: one in terms of the sequence of acceleration parameters and another in terms of the sequence of (possibly variable) stepsizes. We then propose a specific adaptive choice of the sequence of acceleration parameters whose goal is greedily minimize the derived bound on the functional gap in terms of these parameters. In Section 3, we consider a special instance of the generic framework for a special class of convex programming problems whose objective function f has the property that, at each point $\bar{x}$, one can find an easy proper closed convex $g_{\bar{x}}$ such that $g_{\bar{x}} \leq f \leq g_{\bar{x}} + L\| \cdot -\bar{x}\|^2/2$, where $L \geq 0$ is a constant dependent on f only. Example of such

a function f consists of the sum of a function with Lipschitz continuous gradient and an easy nonsmooth proper closed convex function. Another example consists of the composition of a convex non-decreasing function on $\mathbb{R}^m$ with a tuple of m functions as above. Finally, Section 4 presents computational results demonstrating that the proposed adaptive accelerated method performs quite well compared to other existing methods.

1.1 Notation

Throughout this paper, $\mathcal{X}$ will denote a finite dimensional inner product real vector space with inner product and induced norm denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. We denote the set of real numbers by $\mathbb{R}$ and the set of extended real numbers, namely $\mathbb{R} \cup \{\pm\infty\}$ by $\bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$. The domain of a point-to-point function f is denoted by $\mathrm{Dom}\, f$. The effective domain of a function $f : \mathcal{X} \to \bar{R}$ is defined as $\mathrm{dom}\, f := \{x \in \mathcal{X} : f(x) < \infty\}$. The *indicator function* of a closed convex set $X \subseteq \mathcal{X}$ is the function $\delta_X : \mathcal{X} \to [0, \infty]$ defined as

$$\delta_X(x) = \begin{cases} 0, & x \in X, \\ \infty, & \text{otherwise.} \end{cases}$$

2 A generic framework of adaptive accelerated methods

In this paper, we are concerned with the problem

$$f_* := \min\{f(x) : x \in \mathcal{X}\}, \tag{1}$$

where the following conditions are assumed:

C.1) $f : \mathcal{X} \to \bar{\mathbb{R}}$ is a proper closed convex function;

C.2) the set X_* of optimal solutions of (1) is non-empty.

We now state a generic framework of accelerated methods for solving (1).

A-HPE Framework

0) Let $\tau \in [0, 1]$, $A_0 \geq 0$, points $y_0, u_0 \in \mathcal{X}$ and an affine function $\Gamma_0 : \mathcal{X} \to \mathbb{R}$ such that

$$A_0 \Gamma_0 \leq A_0 f, \quad A_0 f(y_0) \leq \inf A_0 \Gamma_0(x) + \frac{1}{2} \|x - u_0\|^2, \tag{2}$$

be given and set $k = 0$;

1) set

$$x_k = u_0 - A_k \nabla \Gamma_k = \operatorname{argmin}\left\{ A_k \Gamma_k(x) + \frac{1}{2} \|x - u_0\|^2 \right\}; \tag{3}$$

2) compute a stepsize $\lambda_{k+1} > 0$, a point $\tilde{y}_{k+1} \in \mathcal{X}$, and an affine function $\gamma_{k+1} : \mathcal{X} \to \mathbb{R}$ minorizing f such that

$$\lambda_{k+1} f(\tilde{y}_{k+1}) + \frac{1}{2}(1 - \tau)\|\tilde{y}_{k+1} - \tilde{x}_k\|^2 \leq \min\left\{ \lambda_{k+1} \gamma_{k+1}(x) + \frac{1}{2}\|x - \tilde{x}_k\|^2 \right\}, \tag{4}$$

where

$$\tilde{x}_k = \frac{A_k}{A_k + a_{k+1}} y_k + \frac{a_{k+1}}{A_k + a_{k+1}} x_k, \tag{5}$$

$$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}; \tag{6}$$

3) choose $y_{k+1} \in \mathcal{X}$ such that $f(y_{k+1}) \leq f(\tilde{y}_{k+1})$ and a pair $(A'_k, a'_{k+1}) \in \mathbb{R}^2$ such that

$$A'_k \geq 0, a_{k+1} \geq 0, \quad A'_k + a'_{k+1} \geq A_k + a_{k+1}, \tag{7}$$

$$(A'_k + a'_{k+1}) f(y_{k+1}) \leq \inf_{x \in \mathcal{X}} (A'_k \Gamma_k + a'_{k+1} \gamma_{k+1})(x) + \frac{1}{2} \|x - u_0\|^2, \tag{8}$$

with the safeguard that $A'_0 = A_0$ when $k = 0$;

4) set

$$A_{k+1} := A'_k + a'_{k+1}, \quad \Gamma_{k+1} := \frac{A'_k}{A'_k + a'_{k+1}} \Gamma_k + \frac{a'_{k+1}}{A'_k + a'_{k+1}} \gamma_{k+1}, \tag{9}$$

$k \leftarrow k + 1$, and go to step 1.

end

We now make some remarks about the A-HPE framework. First, for every $k \geq 0$ and $\lambda_{k+1} > 0$, the pair $(\tilde{y}_{k+1}, \gamma_{k+1}) = (\tilde{y}^*_{k+1}, \gamma^*_{k+1})$, where

$$\tilde{y}^*_{k+1} := (I + \lambda_{k+1} \partial f)^{-1}(\tilde{x}_k), \qquad \gamma^*_{k+1}(x) = f(\tilde{y}^*_{k+1}) + \frac{1}{\lambda_{k+1}} \langle x - \tilde{y}^*_{k+1}, \tilde{x}_k - \tilde{y}^*_{k+1} \rangle,$$

can be easily seen to satisfy (4) (see Lemma 3.2). Hence, it is always possible to choose λ_{k+1}, $\tilde{y}_{k+1}$ and γ_{k+1} as in step 2 of the A-HPE framework. However, the above choice is not practical since requires computation of the resolvent $(I + \lambda_{k+1} \partial f)^{-1}$ of f. We will discuss in Section 3 more practical ways of computing the above three entities so as to satisfy the requirements of step 2. Second, if no pair (A_0, Γ_0) as in step 1 such that $A_0 > 0$, and hence $\Gamma_0 \leq f$, is known, then we can always choose $A_0 = 0$ and $\Gamma_0 \equiv 0$.

3

Before discussing the convergence properties of the A-HPE framework, we discuss the connection between the A-HPE framework stated here and the A-HPE framework presented in [10]. We first state the two simple results, the first one without proof.

Lemma 2.1. *Let $\chi \in \mathbb{R}$, $u_0, y \in \mathcal{X}$ and an affine function $\Gamma : \mathcal{X} \to \mathbb{R}$ be given. Then, the following conditions are equivalent:*

   a) $\min\{\Gamma(x) + \|x - u_0\|^2/2 : x \in \mathcal{X}\} \geq \chi$;

   b) $\chi - \Gamma(u_0) + \|\nabla\Gamma\|^2/2 \leq 0$;

   c) $\|\nabla\Gamma + y - u_0\|^2 + 2[\chi - \Gamma(y)] \leq \|y - u_0\|^2$.

Lemma 2.2. *Let $\lambda > 0$, $\tau \in [0, 1]$, $\tilde{x}, \tilde{y} \in \mathcal{X}$ be given. For any $\gamma : \mathcal{X} \to \mathbb{R}$ affine linear functional the following conditions are equivalent:*

   a) $\gamma \leq f$ *and*

$$\lambda f(\tilde{y}) + \frac{1 - \tau}{2}\|\tilde{y} - \tilde{x}\|^2 \leq \inf_{x \in \mathcal{X}} \lambda\gamma(x) + \frac{1}{2}\|x - \tilde{x}\|^2; \tag{10}$$

   b) $\gamma(x) = f(\tilde{y}) + \langle v, x - \tilde{y}\rangle - \varepsilon,$

$$v \in \partial_\varepsilon f(\tilde{y}), \quad \|\lambda v + \tilde{y} - \tilde{x}\|^2 + 2\lambda\varepsilon \leq \tau\|\tilde{y} - \tilde{x}\|^2 \tag{11}$$

Proof. To prove the equivalence of a) and b) first observe that if $f(\tilde{y}) = \infty$ neither a) nor b) hold. Now assume that $f(\tilde{y}) \in \mathbb{R}$. Under this assumption, any given affine functional γ has an (unique) expression as

$$\gamma(x) = f(\tilde{y}) + \langle v, x - \tilde{y}\rangle - \varepsilon$$

and, conversely, any $v \in \mathcal{X}$, $\varepsilon \in \mathbb{R}$ defines an (unique) affine functional by the above expression. In both cases, $v = \nabla\gamma$, $\varepsilon = f(\tilde{y}) - \gamma(\tilde{y})$ and this γ minorizes f if and only is $v \in \partial_\varepsilon f(\tilde{y})$. Equivalence between (10) and (11) follows from the equivalence between a) and c) of Lemma 2.1 with $\Gamma = \lambda\gamma$, $u_0 = \tilde{x}$, $y = \tilde{y}$ and $\chi = \lambda f(\tilde{y}) + (1 - \tau)\|\tilde{y} - u_0\|^2/2$. $\qquad\square$

Hence, step 2) is equivalent to finding a triple $(\tilde{y}, v, \varepsilon) = (\tilde{y}_{k+1}, \nabla\gamma_{k+1}, \varepsilon_{k+1}) \in \mathcal{X} \times \mathcal{X} \times \mathbb{R}_+$ satisfying the error condition in Lemma 2.2(b):

$$v_{k+1} \in \partial_{\varepsilon_{k+1}} f(\tilde{y}_{k+1}), \quad \|\lambda_{k+1}v_{k+1} + \tilde{y}_{k+1} - \tilde{x}_k\|^2 + 2\lambda_{k+1}\varepsilon_{k+1} \leq \tau\|\tilde{y}_{k+1} - \tilde{x}_k\|^2. \tag{12}$$

Rather than using condition a) of Lemma 2.2 as in (4), the A-HPE framework studied in [10] uses the error condition in Lemma 2.2(b) as in (12). It can be easily shown that the A-HPE framework presented in this paper with $(A'_k, a'_{k+1}) = (A_k, a_{k+1})$ is the same as the one studied in [10].

We now turn our attention to derive a bound on the functional gap $f(y_k) - f_*$, in terms of the acceleration parameter A_k, which is further refined to yield one in terms of the sequence of stepsizes $\{\lambda_k\}$.

We start by stating the following technical result.

Lemma 2.3. *Let $A \geq 0$, $u_0 \in \mathcal{X}$ and a closed convex function $\Gamma : \mathcal{X} \to \mathbb{R}$ be given and let*

$$x = \arg\min\left\{A\Gamma(u) + \frac{1}{2}\|u - u_0\|^2\right\}, \qquad \Theta = \min\left\{A\Gamma(u) + \frac{1}{2}\|u - u_0\|^2\right\}. \tag{13}$$

4

Let $\lambda > 0$, $y \in \mathcal{X}$ and a proper closed convex function $\gamma : \mathcal{X} \to \mathbb{R}$ minorizing f be given, and define

$$a := \frac{\lambda + \sqrt{\lambda^2 + 4\lambda A}}{2}, \qquad \tilde{x} := \frac{A}{A+a}y + \frac{a}{A+a}x, \tag{14}$$

and

$$\theta := \min\left\{\lambda\gamma(u) + \frac{1}{2}\|u - \tilde{x}\|^2\right\}. \tag{15}$$

Then,

$$\operatorname{argmin}\left\{a\gamma(u) + A\Gamma(u) + \frac{1}{2}\|u - u_0\|^2\right\} \geq [\Theta - Af(y)] + (A+a)\frac{\theta}{\lambda}.$$

Proof. Let an arbitrary $u \in \mathcal{X}$ be given. Define

$$\tilde{u} := \frac{Ay + au}{A + a}$$

and note that

$$\tilde{u} - \tilde{x} = \left(\frac{a}{A+a}\right)^2(u - x).$$

Note also that the definition of a in (14) implies that

$$\lambda = \frac{a^2}{A + a}.$$

In view of the two relations in (13), the fact that $\gamma \leq f$, the functions Γ and γ are convex, and the last three relations , we conclude that

$$
\begin{aligned}
a\gamma(u) &+ A\Gamma(u) + \frac{1}{2}\|u - u_0\|^2 \\
&\geq a\gamma(u) + \frac{1}{2}\|u - x\|^2 + \Theta \\
&\geq a\gamma(u) + \frac{1}{2}\|u - x\|^2 + A\gamma(y) + [\Theta - Af(y)] \\
&\geq (A+a)\gamma\left(\frac{Ay + au}{A+a}\right) + \frac{1}{2}\|u - x\|^2 + [\Theta - Af(y)] \\
&= (A+a)\left[\gamma(\tilde{u}) + \frac{(A+a)}{2a^2}\|\tilde{u} - \tilde{x}\|^2\right] + [\Theta - Af(y)] \\
&\geq (A+a)\min\left\{\gamma(u') + \frac{1}{2\lambda}\|u' - \tilde{x}\|^2\right\} + [\Theta - Af(y)].
\end{aligned}
$$

The result now follows from (15) and the fact that the above inequality holds for any $u \in \mathcal{X}$. $\quad\square$

We now use the above result to establish the following result.

Lemma 2.4. *The following statements hold for every $k \geq 0$:*

a) *Γ_k is affine and $A_k\Gamma_k \leq A_k f$;*

b) *the pair $(A'_k, a'_{k+1}) = (A_k, a_{k+1})$ satisfies (8);*

c) *there holds*

$$A_k f(y_k) \leq \min\left\{A_k\Gamma_k(u) + \frac{1}{2}\|u - u_0\|^2\right\};$$

5

Proof. We first claim that a) and c) imply b). Indeed, assume that a) and c) hold. Using c) and Lemma 2.3 with $\Gamma = \Gamma_k$, $\gamma = \gamma_{k+1}$, $\lambda = \lambda_{k+1}$, $y = y_k$, $A = A_k$, we conclude that

$$\min\left\{a_{k+1}\gamma_{k+1}(u) + A_k\Gamma_k(u) + \frac{1}{2}\|u - u_0\|^2\right\} \geq \frac{A_k + a_{k+1}}{\lambda_{k+1}} \min\left\{\lambda_{k+1}\gamma_{k+1}(u) + \frac{1}{2}\|u - \tilde{x}_k\|^2\right\}$$
$$\geq (A_k + a_{k+1})f(\tilde{y}_{k+1}) \geq (A_k + a_{k+1})f(y_{k+1}),$$

where the last two inequalities are due to (4) and step 3 of the A-HPE framework, respectively. We have thus shown that b) holds.

In view of the above claim, it suffices to show that a) and c) holds for every $k \geq 0$. The proof is by induction for k. They certainly hold for $k = 0$ due to (2) and the assumption that Γ_0 is affine. Assume now that a) and c) hold for k. By the above claim, b) also holds for k. This implies that k-th step of the A-HPE framework is well-defined, and hence that c) holds for $k + 1$, due to (8) and the first identity in (9). Also, statement a) for k, relation (9), and the fact that γ_{k+1} is an affine function, imply that Γ_{k+1} is affine and

$$A_{k+1}\Gamma_{k+1} = A'_k\Gamma_k + a'_{k+1}\gamma_{k+1} \leq A'_k f + a'_{k+1}f = A_{k+1}f,$$

thereby showing that a) holds for $k + 1$. Thus, the result follows. $\qquad\square$

The main result about the A-HPE method is as follows.

Theorem 2.5. *Let x_* be the projection of u_0 onto X_* and d_0 be the distance of u_0 to X_*. Then, for every $k \geq 1$,*

$$\frac{1}{2}\|x_k - x_*\|^2 + A_k\left[f(y_k) - f_*\right] \leq \frac{1}{2}\,d_0^2. \tag{16}$$

and

$$A_k \geq \frac{1}{4}\left(\sum_{j=1}^{k}\sqrt{\lambda_j}\right)^2. \tag{17}$$

As a consequence, for every integer $k \geq 1$,

$$f(y_k) - f_* \leq \frac{d_0^2}{2A_k} \leq 2d_0^2\left(\sum_{j=1}^{k}\sqrt{\lambda_j}\right)^{-2}, \qquad \|x_k - x_*\| \leq d_0. \tag{18}$$

Proof. The fact that Γ_k is an affine function minorizing f, the definition of x_* and (3) imply that

$$A_k f_* + \frac{1}{2}d_0^2 = A_k\Gamma_k(x_*) + \frac{1}{2}\|x_* - u_0\|^2 = \min\left\{A_k\Gamma_k(x) + \frac{1}{2}\|x - u_0\|^2\right\} + \frac{1}{2}\|x_* - x_k\|^2$$
$$\geq A_k f(y_k) + \frac{1}{2}\|x_* - x_k\|^2,$$

where the last inequality is due to Lemma 2.4(c). Hence, (16) follows. Now, by (6), we have

$$a_{k+1} \geq \frac{\lambda_{k+1}}{2} + \sqrt{\lambda_{k+1}A_k},$$

which, together with (7) and (9), imply that

$$A_{k+1} = A'_k + a'_{k+1} \geq A_k + a_{k+1} \geq A_k + \left(\frac{\lambda_{k+1}}{2} + \sqrt{\lambda_{k+1}A_k}\right) \geq \left(\sqrt{A_k} + \frac{1}{2}\sqrt{\lambda_{k+1}}\right)^2.$$

6

This clearly implies that

$$\sqrt{A_k} - \sqrt{A_0} \geq \frac{1}{2} \left(\sum_{j=1}^{k} \sqrt{\lambda_j} \right),$$

and hence that (17) holds. The two inequalities in (18) follow immediately from (16) and (17). $\square$

We will now discuss an alternative characterization for condition (8).

Proposition 2.6. *For every $k \geq 0$, condition (8) holds if, and only if,*

$$A_k'[f(y_{k+1}) - \Gamma_k(u_0)] + a_{k+1}'[f(y_{k+1}) - \gamma_{k+1}(u_0)] + \frac{1}{2}\|A_k'\nabla\Gamma_k + a_{k+1}'\nabla\gamma_{k+1}\|^2 \leq 0. \quad (19)$$

Proof. This result follows from the equivalence between a) and b) of Lemma 2.1 with $\Gamma = A_k'\Gamma_k + a_{k+1}'\gamma_{k+1}$, $\chi = (A_k' + a_{k+1}')f(y_{k+1})$ and $y = y_{k+1}$. $\square$

Note that (19) is a a single convex quadratic constraint on the scalars A_k' and a_{k+1}'. In view of the first inequality in (18), which suggests that the rate of convergence of $\{f(y_k) - f_*\}$ to zero is faster the larger A_k is chosen, it seems natural to choose the pair (A_k', a_{k+1}') as the optimal solution of the two-variable convex quadratic constrained problem

$$\begin{aligned}
\max \quad & A' + a' \\
\text{s.t.} \quad & A_k'[f(y_{k+1}) - \Gamma_k(u_0)] + a_{k+1}'[f(y_{k+1}) - \gamma_{k+1}(u_0)] + \tfrac{1}{2}\|A_k'\nabla\Gamma_k + a_{k+1}'\nabla\gamma_{k+1}\|^2 \leq 0, \quad (20) \\
& A', a' \geq 0.
\end{aligned}$$

Since (20) contains only two variables, an optimal solution of (20), when it exists (see Proposition 2.7 below), can be easily computed.

In the remaining part of this section, we discuss the case when problem (20) is unbounded. We will see that this case implies that $y_{k+1} \in X_*$, in which case we may successfully stop the A-HPE framework. In fact, we consider a slightly more general problem which covers problem (20) as a special case.

Proposition 2.7. *Suppose that $u_0, y \in X$ and $\gamma_1, \ldots, \gamma_m : X \to \mathbb{R}$ are affine functions minorizing f and consider the problem*

$$\begin{aligned}
\max \quad & \sum_{i=1}^{m} \alpha_i \\
\text{s.t.} \quad & \inf_{x \in X} \left\{ \sum_{i=1}^{m} \alpha_i\gamma_i(x) + \tfrac{1}{2}\|x - u_0\|^2 \right\} \geq \sum_{i=1}^{m} \alpha_i f(y), \quad (21) \\
& \alpha_1 \geq 0, \ldots, \alpha_m \geq 0.
\end{aligned}$$

Then, the following conditions are equivalent:

a) problem (21) is unbounded;

b) there exist not all zero nonnegative scalars $\bar{\alpha}_1, \ldots, \bar{\alpha}_m$ such that

$$\sum_{i=1}^{m} \bar{\alpha}_i \nabla\gamma_i = 0, \qquad \bar{\alpha}_i[f(y) - \gamma_i(y)] = 0, \quad i = 1, \ldots, m, \quad (22)$$

in which case $y \in X_*$.

Proof. First, note that the equivalence between a) and c) of Lemma 2.1 with $\Gamma = \sum_{i=1}^{m} \alpha_i \gamma_i$ and $\chi = \sum_{i=1}^{m} \alpha_i f(y)$ imply that the hard constraint in problem (21) is equivalent to

$$\left\| \sum_{i=1}^{m} \alpha_i \nabla \gamma_i + y - u_0 \right\|^2 + 2 \sum_{i=1}^{m} \alpha_i [f(y) - \gamma_i(y)] \leq \|y - u_0\|^2. \tag{23}$$

Now, assume that b) holds. Then, in view of (22), the point $\alpha(t) := (t\bar{\alpha}_1, \ldots, t\bar{\alpha}_m)$ clearly satisfies (23), and hence is feasible for (21), for every $t > 0$. Hence, for a fixed $x_* \in X_*$, this conclusion implies that

$$\sum_{i=1}^{m} (t\bar{\alpha}_i) f(y) \leq \sum_{i=1}^{m} (t\bar{\alpha}_i) \gamma_i(x_*) + \frac{1}{2} \|x_* - u_0\|^2 \leq t \left(\sum_{i=1}^{m} \bar{\alpha}_i \right) f(x_*) + \frac{1}{2} \|x_* - u_0\|^2,$$

where the last inequality follows from the assumption that $\gamma_i \leq f$ and the nonnegativity of $t\bar{\alpha}_1, \ldots, t\bar{\alpha}_m$. Dividing this expression by $t(\sum_{i=1}^{m} \bar{\alpha}_i) > 0$, and letting $t \to \infty$, we then conclude that $f(y) - f(x_*) \leq 0$, and hence that $y \in X_*$. Also, the objective function of (21) at the point $\alpha(t)$ converges to infinity as $t \to \infty$, showing that b) implies a).

Now assume that a) holds. Then, there exists a sequence $\{\alpha^k = (\alpha_1^k, \ldots, \alpha_m^k) : k \geq 0\}$ of feasible solutions of problem (21) along which its objective function converges to infinity. Consider the sequence $\{\bar{\alpha}^k := \alpha^k/(e^T \alpha^k) : k \geq 0\}$, where e is the vector of all ones. Clearly, $\{\bar{\alpha}^k\}$ has an accumulation point $\bar{\alpha} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_m)$, where $0 \neq \bar{\alpha} \geq 0$. Moreover, using the fact that α^k satisfies (23) and the fact that $\gamma_i(y) \leq f(y)$, we easily see that (22) holds. $\qquad\square$

3 An adaptive accelerated method for a class of convex functions

In this section, we discuss an instance of the A-HPE framework for solving convex functions that, at every point $\bar{x} \in \mathcal{X}$, can be well-approximated by another convex function with an easily computable resolvent.

More specifically, in this section, we further assume that:

C.3) *there exists $L \geq 0$ such that, for every $\bar{x} \in \mathcal{X}$, we can construct a proper closed convex function $g_{\bar{x}} : \mathcal{X} \to \bar{\mathbb{R}}$ such that the resolvent $(I + \lambda \partial \bar{g}_{\bar{x}})^{-1}(x)$ is easily computable for any $x \in \mathcal{X}$ and $\lambda > 0$, and*

$$g_{\bar{x}}(x) \leq f(x) \leq g_{\bar{x}}(x) + \frac{L}{2} \|x - \bar{x}\|^2, \quad \forall x \in \mathcal{X}. \tag{24}$$

We now make a remark about condition C.3. Clearly, (24) is equivalent to require that $\operatorname{dom} g_{\bar{x}} = \operatorname{dom} f$ and that the inequality in (24) holds for every $x \in \operatorname{dom} f$.

Before stating the method for functions f satisfying C.1-C.3, we discuss some important examples of such functions.

First example: Assume that $\phi : \operatorname{Dom} \phi \subseteq \mathcal{X} \to \mathbb{R}$ is a function which is differentiable and convex on a closed convex set $\emptyset \neq \Omega \subseteq \operatorname{Dom} \phi$, and its gradient $\nabla \phi$ is L-Lipschitz continuous on Ω, i.e.,

$$\|\nabla \phi(x') - \nabla \phi(x)\| \leq L \|x' - x\|, \quad \forall x, x' \in \Omega.$$

Assume also that $h : \mathcal{X} \to \bar{\mathbb{R}}$ is a closed proper convex function such that $\operatorname{dom} h \subseteq \Omega$. Finally, assume that $f : \mathcal{X} \to \bar{\mathbb{R}}$ is defined as

$$f(x) = \begin{cases} \phi(x) + h(x), & \text{if } x \in \operatorname{dom} h; \\ +\infty, & \text{otherwise.} \end{cases} \tag{25}$$

Clearly, f is a proper closed convex function. Moreover, the following result shows that f satisfies condition C.3.

Proposition 3.1. *For any $\bar{x} \in \mathcal{X}$, the function $g_{\bar{x}} : \mathcal{X} \to \bar{\mathbb{R}}$ defined as*

$$g_{\bar{x}}(x) = \phi(\bar{x}_\Omega) + \langle \nabla\phi(\bar{x}_\Omega), x - \bar{x}_\Omega \rangle + h(x), \quad \forall x \in \mathcal{X},$$

where $\bar{x}_\Omega := P_\Omega(\bar{x})$, satisfies (24).

Proof. Let $\bar{x} \in \mathcal{X}$ be given and note that $\bar{x}_\Omega := P_\Omega(\bar{x}) \in \Omega$. It is well-known that the assumptions on ϕ imply that

$$0 \leq \phi(x) - \phi(\bar{x}_\Omega) - \langle \nabla\phi(\bar{x}_\Omega), x - \bar{x}_\Omega \rangle \leq \frac{L}{2}\|x - \bar{x}_\Omega\|^2 \leq \frac{L}{2}\|x - \bar{x}\|^2, \quad \forall x \in \Omega,$$

where the last inequality is due to the non-expansiveness property of P_Ω. Hence, if $x \in \operatorname{dom} f$, we conclude by adding $\phi(\bar{x}_\Omega) + \langle \nabla\phi(\bar{x}_\Omega), x - \bar{x}_\Omega \rangle + h(x)$ to all terms of the above relation that $g_{\bar{x}}(x) \leq f(x) \leq g_{\bar{x}}(x) + (L/2)\|x - \bar{x}\|^2$. Since $\operatorname{dom} f = \operatorname{dom} g_{\bar{x}}$, the proposition now follows from the remark following condition C.3. $\qquad\square$

Finally, note that the resolvents of $g_{\bar{x}}$ can be computed in terms of the resolvents of h. Hence, as long as the latter can be easily computed, the function f and the family of functions $\{g_{\bar{x}} : \bar{x} \in \mathcal{X}\}$ satisfies condition C.3.

Second example: Let $\Psi : \mathbb{R}^m \to \mathbb{R}$ be a closed convex function which, as a function of each variable, is nondecreasing. Assume also that Ψ is M-Lipschitz continuous with respect to the Euclidean norm on $\mathbb{R}^m$. Let $f_i : \mathcal{X} \to \Re$ together with the family $\{g_{i,\bar{x}} : \bar{x} \in \mathcal{X}\}$ satisfy C.3 with $L = L_i$, for every $i = 1, \ldots, m$. Then, it is easy to see that the function $f := \Psi(f_1, \ldots, f_m)$ together with the family $\{g_{\bar{x}} := \Psi(g_{1,\bar{x}}, \ldots, g_{m,\bar{x}}) : \bar{x} \in \mathcal{X}\}$ satisfy (24) with $L = M(L_1^2 + \ldots + L_m^2)^{1/2}$. Hence, as long as the resolvent of $g_{\bar{x}} := \Psi(g_{1,\bar{x}}, \ldots, g_{m,\bar{x}})$ can be computed exactly, the function f and the family $\{g_{\bar{x}} : \bar{x} \in \mathcal{X}\}$ will satisfy C.3.

We will now present an instance of the A-HPE framework for functions f satisfying conditions C.1-C.3. The actual proof that the method below is an instance of the A-HPE framework will given in Proposition 3.3.

Adaptive accelerated (AA) method

0) Let $L \geq 0$ be as in condition C.3 above and $\tau \in (0,1]$ be given. Let $A_0 \geq 0$, points $y_0, u_0 \in X$ and an affine function $\Gamma_0 : X \to \mathbb{R}$ such that

$$A_0 \Gamma_0 \leq A_0 f, \quad A_0 f(y_0) \leq \inf A_0 \Gamma_0(x) + \frac{1}{2}\|x - u_0\|^2,$$

be given, and set $\lambda := \tau/L$, $\Gamma_0^0 = \Gamma_0(u_0)$, $V_0 = \nabla \Gamma_0$ and $k = 1$;

1) set $x_k = u_0 - A_k V_k$,

$$a_{k+1} = \frac{\lambda + \sqrt{\lambda^2 + 4\lambda A_k}}{2}, \tag{26}$$

$$\tilde{x}_k = \frac{A_k}{A_k + a_{k+1}} y_k + \frac{a_{k+1}}{A_k + a_{k+1}} x_k,$$

and solve the problem

$$\tilde{y}_{k+1} = \operatorname{argmin}\left\{\lambda g_{\tilde{x}_k}(\tilde{x}) + \frac{1}{2}\|\tilde{x} - \tilde{x}_k\|^2\right\}, \tag{27}$$

where $g_{\tilde{x}_k}$ is as in condition C.3 above.

2) set

$$v_{k+1} := \frac{1}{\lambda}(\tilde{x}_k - \tilde{y}_{k+1}), \qquad \gamma_{k+1}^0 = g_{\tilde{x}_k}(\tilde{y}_{k+1}) + \langle u_0 - \tilde{y}_{k+1}, v_{k+1}\rangle; \tag{28}$$

3) choose $y_{k+1} \in X$ such that $f(y_{k+1}) \leq f(\tilde{y}_{k+1})$ and the pair $(A_k', a_{k+1}') \in \mathbb{R}^2$ as an optimal solution of

$$\begin{aligned}
\max \quad & A' + a' \\
\text{s.t.} \quad & \tfrac{1}{2}\|A'V_k + a'v_{k+1}\|^2 + A'[f(y_{k+1}) - \Gamma_k^0] + a'[f(y_{k+1}) - \gamma_{k+1}^0] \leq 0, \\
& A', a' \geq 0.
\end{aligned} \tag{29}$$

with the safeguard that $A_0 = A_0'$ when $k = 0$; if the above problem is unbounded, then **stop** by declaring y_{k+1} to be an optimal solution of (1);

4) set $A_{k+1} = A_k' + a_{k+1}'$,

$$\Gamma_{k+1}^0 = \frac{A_k'}{A_k' + a_{k+1}'}\Gamma_k^0 + \frac{a_{k+1}'}{A_k' + a_{k+1}'}\gamma_{k+1}^0, \tag{30}$$

$$V_{k+1} = \frac{A_k'}{A_k' + a_{k+1}'}V_k + \frac{a_{k+1}'}{A_k' + a_{k+1}'}v_{k+1}, \tag{31}$$

$k \leftarrow k + 1$, and go to step 1.

end

We now make a remark about the AA method for the case where f and $g_{\tilde{x}}$ are as in the first example at the beginning of this section. For this case, it can be shown that when $\Omega = X$, and hence $\nabla\phi$ is defined and is L-Lipschitz continuous on the whole X, the AA method reduces to a well-known variant of Nesterov's method, namely FISTA [1] (see also Algorithm 2 of [18]).

Before establishing the convergence properties of the AA method, we state and proof two tech-

nical results.

Lemma 3.2. *Let $\lambda > 0$, $\xi \in \mathcal{X}$ and a proper closed convex function $\phi : \mathcal{X} \to (-\infty, \infty]$ be given and denote the optimal solution and optimal value of*

$$\min \left\{ \lambda \phi(x) + \frac{1}{2} \|x - \xi\|^2 \right\}. \tag{32}$$

by $\bar{x}$ and $\bar{\phi}$, respectively. Then, the affine function $\mathcal{A} : \mathcal{X} \to \mathbb{R}$ defined as

$$\mathcal{A}(x) = \phi(\bar{x}) + \frac{1}{\lambda} \langle \xi - \bar{x}, x - \bar{x} \rangle, \quad \forall x \in \mathcal{X},$$

has the property that $\mathcal{A} \leq \phi$ and the optimal solution and optimal value of

$$\min \left\{ \lambda \mathcal{A}(x) + \frac{1}{2} \|x - \xi\|^2 \right\} \tag{33}$$

is $\bar{x}$ and $\bar{\phi}$, respectively.

Proof. The optimality condition for (32) says that $0 \in \lambda \partial \phi(\bar{x}) + \bar{x} - \xi$, or equivalently

$$\frac{1}{\lambda} (\xi - \bar{x}) \in \partial \phi(\bar{x}),$$

and hence that $\mathcal{A} \leq \phi$, by the definition of $\mathcal{A}$ and the subgradient of a function. Moreover, $\bar{x}$ clearly satisfies the optimality condition of (33). Hence, the last claim of the proposition follows. $\square$

The following result shows that the AA method is a special instance of the A-HPE framework.

Proposition 3.3. *Consider the AA method and, for every $k \geq 0$, define*

$$\lambda_{k+1} = \lambda := \frac{\tau}{L}, \qquad \gamma_{k+1}(x) = g_{\tilde{x}_k}(\tilde{y}_{k+1}) + \langle x - \tilde{y}_{k+1}, v_{k+1} \rangle, \tag{34}$$

where v_{k+1} is defined in (28). Also, define the sequence of functions $\{\Gamma_k\}$ recursively as in (9). Then, for every $k \geq 0$, (4), (6) and (8) hold. In particular, the AA method is a special case of the A-HPE framework.

Proof. Using (26) and the definition of λ_{k+1} in (34), it follows that (6) trivially holds. Clearly, in view of (34) and (28), we have $v_{k+1} = \nabla \gamma_{k+1}$ and $\gamma_{k+1}^0 = \gamma_{k+1}(u_0)$ for every $k \geq 0$. Hence, the initialization in step 0 of the AA method, and the recursive definition of $\{\Gamma_k\}$ in (9) together with (30) and (31), imply that $\Gamma_k^0 = \Gamma_k(u_0)$ and $V_k = \nabla \Gamma_k$ for all $k \geq 0$. The above two conclusions then imply that the convex quadratic constraint in (29) is equivalent to (19), which in turn is equivalent to (8), in view of Proposition 2.6. We will now show that (4) holds for every $k \geq 0$. Indeed, in view of (27), the definition of λ_{k+1} in (34), and condition C.3 with $\bar{x} = \tilde{x}_k$ and $x = \tilde{y}_{k+1}$, we have

$$\min \left\{ \lambda_{k+1} g_{\tilde{x}_k}(\tilde{x}) + \frac{1}{2} \|\tilde{x} - \tilde{x}_k\|^2 \right\} = \lambda_{k+1} g_{\tilde{x}_k}(\tilde{y}_{k+1}) + \frac{1}{2} \|\tilde{y}_{k+1} - \tilde{x}_k\|^2$$

$$= \lambda_{k+1} \left[g_{\tilde{x}_k}(\tilde{y}_{k+1}) + \frac{L}{2} \|\tilde{y}_{k+1} - \tilde{x}_k\|^2 \right] + \frac{1}{2} (1 - \lambda_{k+1} L) \|\tilde{y}_{k+1} - \tilde{x}_k\|^2$$

$$\geq \lambda f(\tilde{y}_{k+1}) + \frac{1}{2} (1 - \tau) \|\tilde{y}_{k+1} - \tilde{x}_k\|^2,$$

Now, using the definition of γ_{k+1} in (34), Lemma 3.2 with $\phi = g_{\tilde{x}_k}$ and $\xi = \tilde{x}_k$, and noting that $\bar{x} = \tilde{y}_{k+1}$ in this case, we have

$$\min \left\{ \lambda_{k+1} g_{\tilde{x}_k}(\tilde{x}) + \frac{1}{2} \|\tilde{x} - \tilde{x}_k\|^2 \right\} = \min \left\{ \lambda_{k+1} \gamma_{k+1}(x) + \frac{1}{2} \|\tilde{x} - \tilde{x}_k\|^2 \right\}.$$

Combining the above two relations, we then conclude that (4) holds. $\square$

11

The following convergence result follows as an immediate consequence of Theorem 2.5, Proposition (3.3), and the fact that $\lambda_k = \tau L^{-1}$ for every $k \geq 1$.

Theorem 3.4. *Let x_* be the projection of u_0 onto X_* and d_0 be the distance of u_0 to X_*. Then, for every $k \geq 1$,*
$$f(y_k) - f_* \leq \frac{d_0^2}{2A_k} \leq \frac{2Ld_0^2}{\tau k^2}, \qquad \|x_k - x_*\| \leq d_0.$$

4 Numerical results

In this section, we describe two restarting variants of the the AA method and report numerical results comparing them to the following variants of Nesterov's method:

  i) FISTA (fast iterative shrinkage-thresholding algorithm) of [1] (see also Algorithm 2 of [18]);

  ii) FISTA-R: restarting variant of FISTA;

  iii) GKR-2: Algorithm 2 of [7];

  iii) GKR-3: Algorithm 3 of [7].

More specifically, we compare the performance of these methods using three classes of conic quadratic programming instances, namely:

  a) random convex quadratic programs (CQPs) (see Subsection 4.1);

  b) semidefinite least squares (SDLSs) (see Subsection 4.2); and

  c) random nonnegative least squares (NNLSs) (see Subsection 4.3).

We observe that we have implemented our own code for FISTA-R. Although we have recently learned that a similar variant was implemented in [16], we have not had the opportunity to include their code in our computational benchmark. Nevertheless, we believe that our implementation should be very similar to the method in [16], and hence should reflect the actual performance of the latter algorithm.

We stop all methods whenever an iterate y_k is found such that

$$\left\| y_k - (I + \partial h)^{-1} \left(y_k - \frac{1}{L} \nabla g(y_k) \right) \right\| \leq 10^{-6} \tag{35}$$

or when 2000 iterations have been performed. Note that for the case where h is an indicator function of a closed convex set, the left hand side of (35) is exactly the norm of the projected gradient with stepsize $1/L$.

We now briefly describe the two restarting versions of the AA method. In the first version, if the function value at the end of the kth iteration increases, i.e., $f(y_k) < f(y_{k+1})$, then the method is restarted at step 0 with $u_0 = y_k$, $y_0 = y_k$, $A_0 = 0$, $\Gamma_0^0 = 0$ and $V_0 = 0$. We observe that we have implemented FISTA-R by incorporating this restarting scheme to FISTA.

In contrast to the first restarting version above, the second one allows some iterations to increase the function value, and hence is more conservative than the first one. More specifically, we restart this variant at the kth iteration whenever $f(y_k) < f(y_{k+1})$ and no more than $\lceil \log_2(k - l_k) \rceil$ restarts have been performed so far, where l_k is the iteration where the last restart was performed.

We will refer to the first and second restarting variants of the AA method as AA-R1 and AA-R2, respectively.

The codes for all the benchmarked variants tested are written in Matlab. All the computational results were obtained on a single core of a server with 2 Xeon X5520 processors at 2.27GHz and 48GB RAM.

We now make some general remarks about how the results are reported on the tables given below. Tables 1, 3 and 5 report the times and Tables 2, 4 and 6 report the number of iterations for all instances of the three problem classes. Note that each problem class is associated with two tables, one reporting the times and the other one the number of iterations required by each benchmarked method to solve all instances of the class. We display the time or number of iterations that a variant takes on an instance in red, and also with an asterisk (*), whenever it can not solve the instance to the required accuracy. In such a case, the accuracy obtained at the last iteration of the variant is also displayed in parentheses.

Figures 1, 3 and 5 plot the time performance profiles (see [4]), and Figures 2, 4 and 6 plot the iteration performance profiles for each of the three problem classes. We recall the following definition of a performance profile. For a given instance, a method A is said to be at most x times slower than method B, if the time taken (resp. number of iterations performed) by method A is at most x times the time taken (resp. number of iterations performed) by method B. A point (x, y) is in the performance profile curve of a method if it can solve exactly $(100y)\%$ of all the tested instances x times slower than any other competing method.

4.1 Numerical results for random CQPs

This subsection compares the performance of our methods AA-R1 and AA-R2 with the variants of Nesterov's method listed at the beginning of this section on a class of randomly generated sparseCQP instances. These instances were also used to report the performance of GKR-2 and GKR-3 in [7].

Given $Q \in S_+^n$, $b \in \mathbb{R}^n$, $l \in \{\mathbb{R} \cup \{-\infty\}\}^n$ and $u \in \{\mathbb{R} \cup \{\infty\}\}^n$ such that $l \leq u$, the box constrained convex quadratic programming problem is defined as

$$\min_{x \in \mathbb{R}^n} \left\{ x^T Q x + b^T x \ : \ l \leq x \leq u \right\}. \tag{36}$$

Letting ϕ and h be defined as

$$\phi(x) = x^T Q x + b^T x, \qquad h(x) = \delta_B(x), \qquad \forall x \in \mathbb{R}^n,$$

where $B = \{x \in \mathbb{R}^n : l \leq x \leq u\}$, we can easily see that (36) is a special case of (1) with $\mathcal{X} = \mathbb{R}^n$ and f given by (25).

Figures 1 and 2 plot time and iteration performance profiles of all variants of Nesterov's method for solving this collection of random sparseCQP instances, respectively. Tables 1 and 2 report the time and number of iterations taken by each method, respectively.

4.2 Numerical results for SDLSs

This subsection compares the performance of our methods AA-R1 and AA-R2 with the variants of Nesterov's method listed at the beginning of this section on a class of SDLS instances.

Let S^n be the set of all $n \times n$ symmetric matrices and S_+^n be the cone of $n \times n$ symmetric positive semidefinite matrices. Given a linear map $\mathcal{A} \in S^n \to \mathbb{R}^m$ and $b \in \mathbb{R}^m$, the semidefinite programming (SDP) feasibility problem consists of finding x such that

$$\mathcal{A}x = b, \qquad x \in S_+^n.$$

We can solve the above problem by considering the SDLS reformulation

$$\min_{x \in S^n} \left\{ \frac{1}{2} \|\mathcal{A}x - b\|^2 \ : \ x \in S_+^n \right\}. \tag{37}$$

13
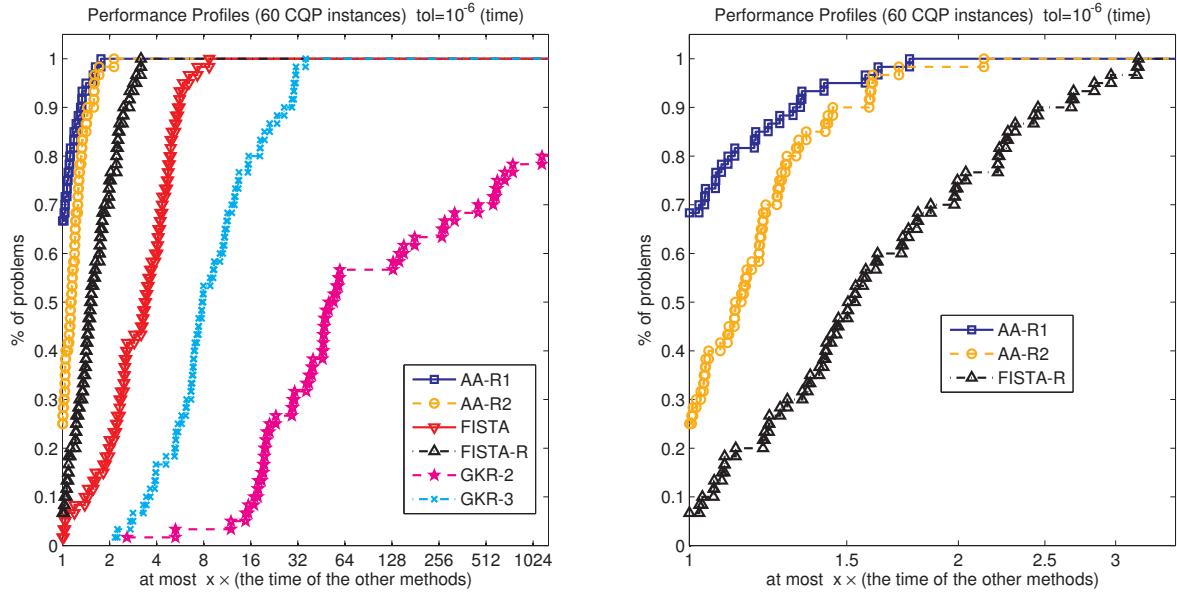
Figure 1: Time performance profiles for solving CQP instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.
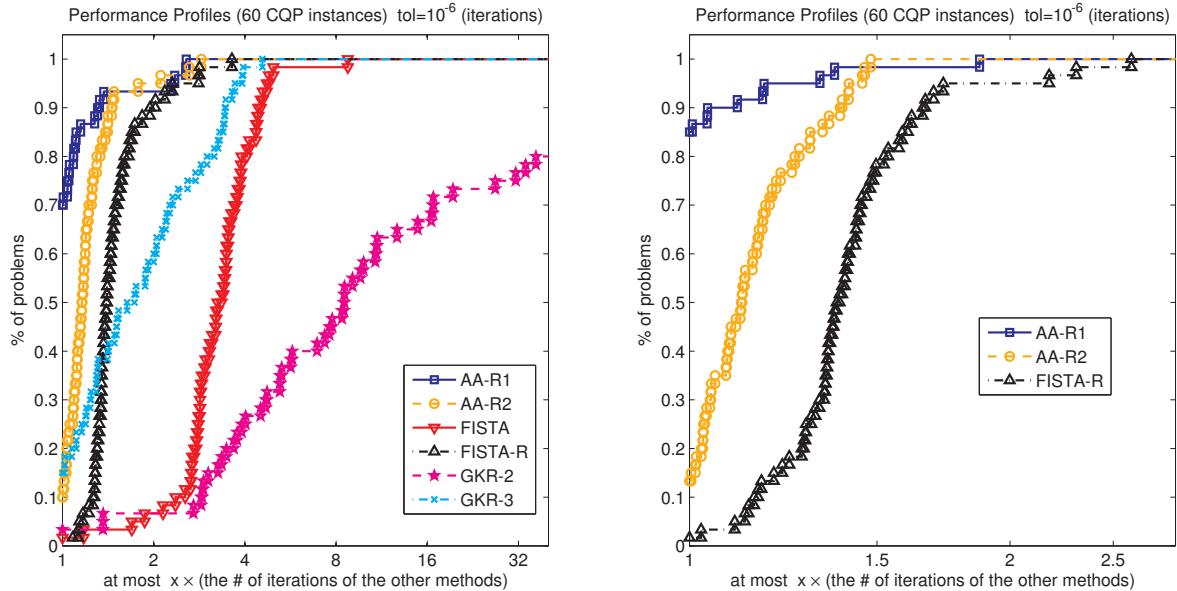
Figure 2: Iteration performance profiles for solving CQP instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.
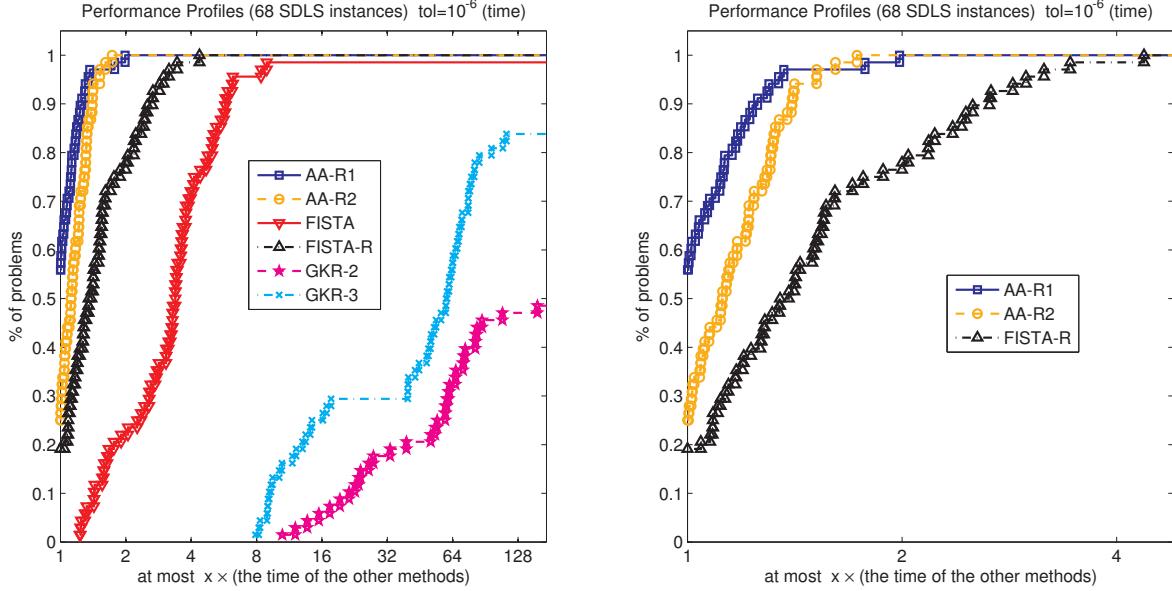
Figure 3: Time performance profiles for solving SDLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

Letting ϕ and h be defined as

$$\phi(x) = \frac{1}{2} \|\mathcal{A}x - b\|^2, \qquad h(x) = \delta_{S_+^n}(x), \qquad \forall x \in \mathcal{S}^n,$$

where $\|\cdot\|$ denotes the Euclidean norm, we can easily see that (37) is a special case of (1) with $\mathcal{X} = \mathcal{S}^n$ and f given by (25).

The SDLS instances included in this comparison are obtained via the above construction from the feasibility sets (after bringing them into standard form) of four classes of SDPs , namely: i) randomly generated SDPs as in [17]; ii) SDP relaxations of frequency assignment problems (see for example Subsection 2.4 in [2]); iii) SDP relaxations of binary integer quadratic problems (see for example Section 7 in [19]); and iv) SDP relaxations of quadratic assignment problems (see for example Section 7 in [19]).

Figures 3 and 4 plot time and iteration performance profiles of all variants of Nesterov's method for solving this collection of SDLS instances, respectively. Tables 3 and 4 report the time and number of iterations taken by each method, respectively.

4.3 Numerical results for NNLSs

This subsection compares the performance of our methods AA-R1 and AA-R2 with the variants of Nesterov's method listed at the beginning of this section on a class of NNLS instances randomly generated as in [9].

Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$, the NNLS problem is defined as

$$\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \|Ax - b\|^2 \ : \ x \geq 0 \right\}, \tag{38}$$

where $\|\cdot\|$ denotes the Euclidean norm.

Letting ϕ and h be defined as

$$\phi(x) = \frac{1}{2} \|Ax - b\|^2, \qquad h(x) = \delta_{\mathbb{R}_+^n}(x), \qquad \forall x \in \mathbb{R}^n,$$

15
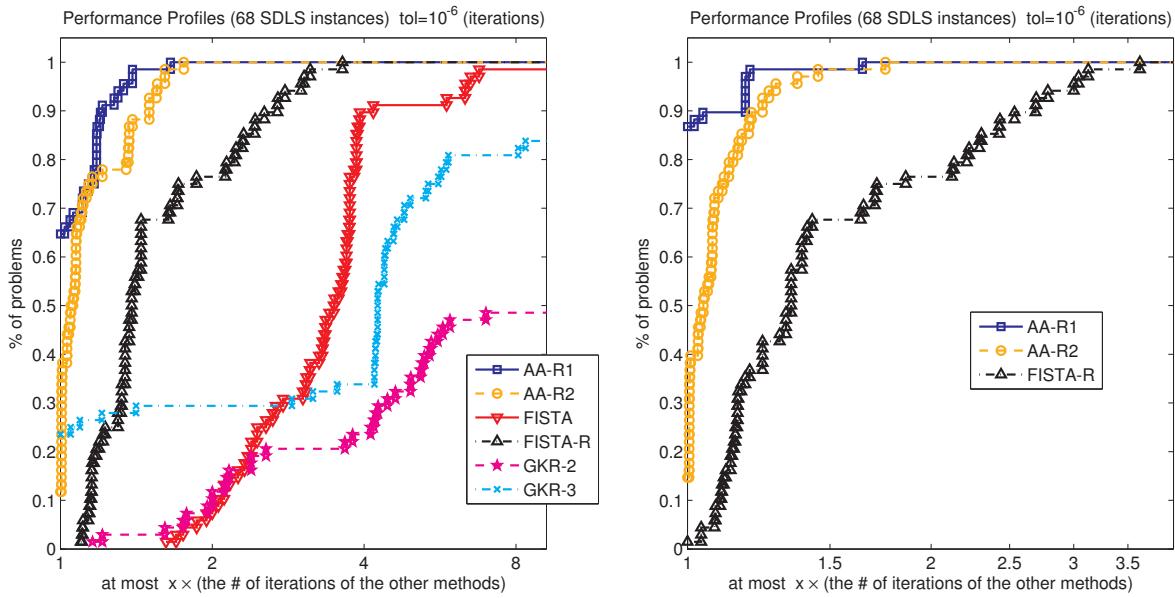
Figure 4: Number of iterations performance profiles for solving SDLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

where $\mathbb{R}_+^n$ is the cone of nonnegative vectors in $\mathbb{R}^n$, we can easily see that (38) is a special case of (1) with $\mathcal{X} = \mathbb{R}^n$ and f given by (25).

Figures 5 and 6 plot the time and iteration performance profiles of all variants of Nesterov's method for solving this collection of random NNLS instances, respectively. Tables 5 and 6 report the time and number of iterations taken by each method, respectively.

4.4 Conclusions

We have observed in our computational experiments that AA-R2 quickly stops performing restarts, while AA-R1 periodically continues to perform restarts.

Figures 7 and 8 plot the time and iteration performance profiles of all variants of Nesterov's method on the collection of instances obtained by combining all the three instance classes described in the previous subsections. From these plots we can see that the overall performances of AA-R1 and AA-R2 are very close to one another, with AA-R2 turning out to be more robust, as it solves more problems to the specified accuracy 10^{-6} than any other of the variants tested. Finally, we can see that AA-R1 and AA-R2 are the two fastest variants among the six ones used in this benchmark in terms of both time and number of iterations.

Acknowledgements

We want to thank Elizabeth W. Karas for generously providing us with the code of the algorithms in [7].

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, March 2009. URL: http://dx.doi.org/10.1137/080716542, http://dx.doi.org/10.1137/080716542 doi:10.1137/080716542.
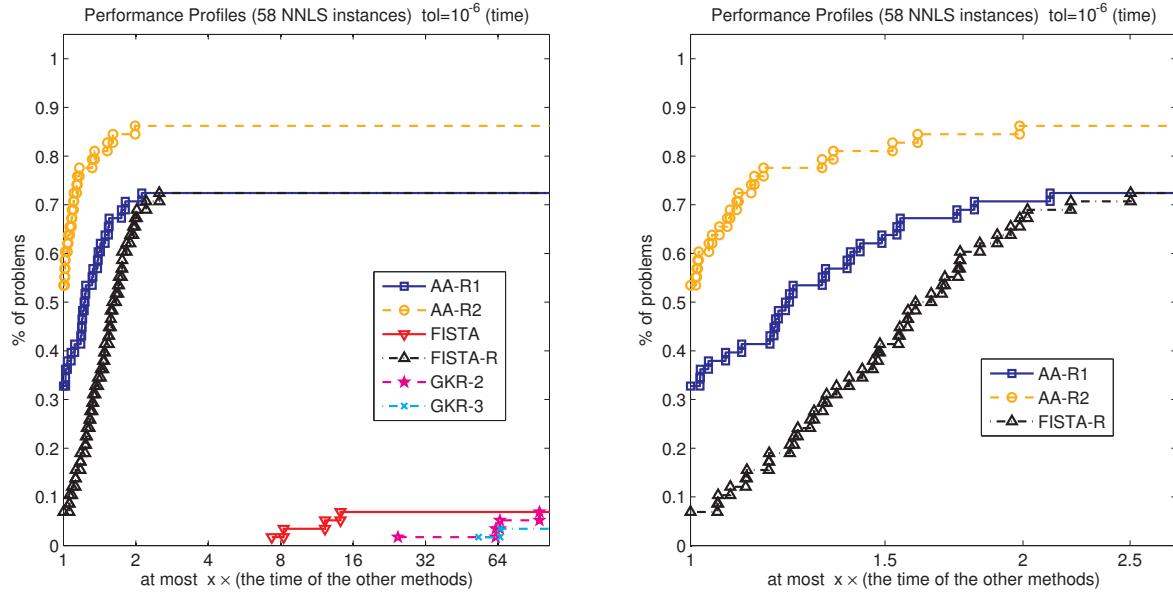
Figure 5: Time performance profiles for solving NNLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.
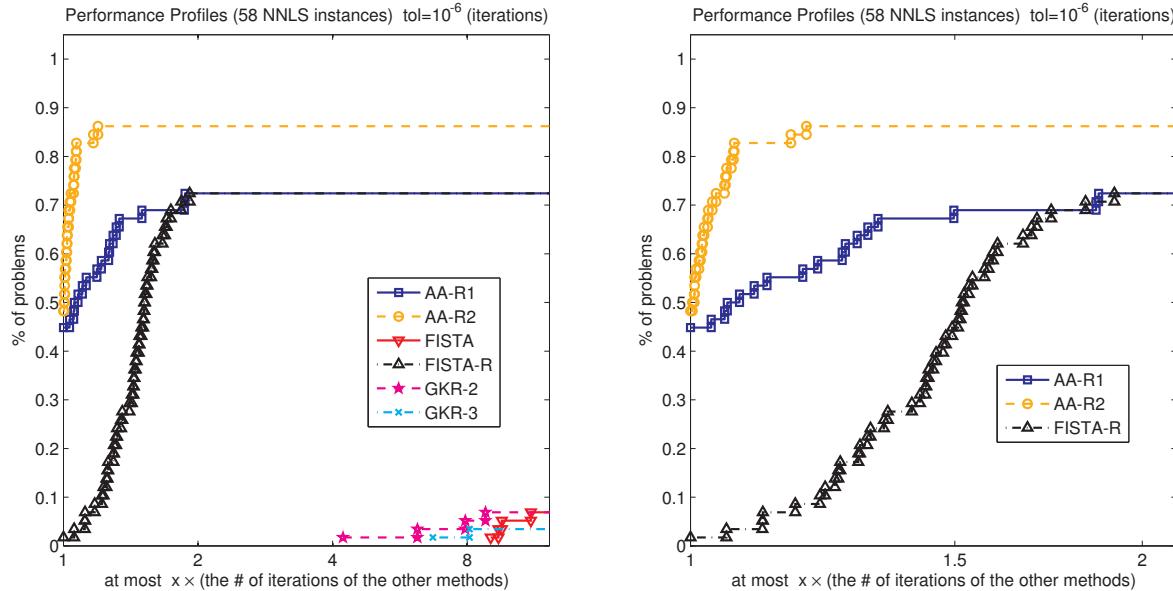

Figure 6: Number of iterations performance profiles for solving NNLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.
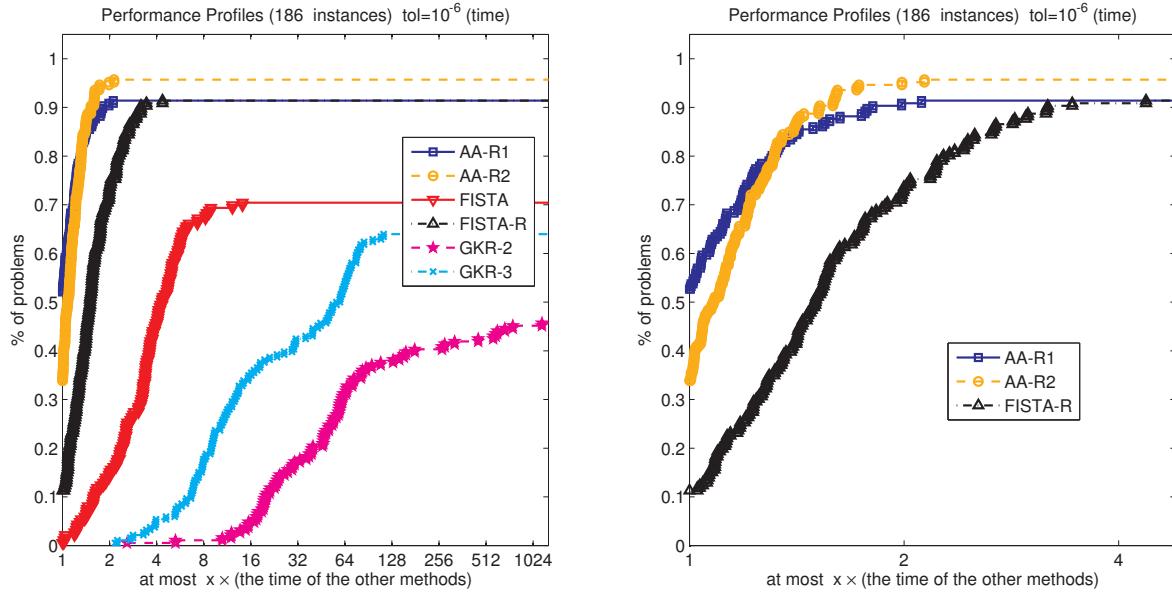
Figure 7: Time performance profiles for solving all the three problem classes with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.
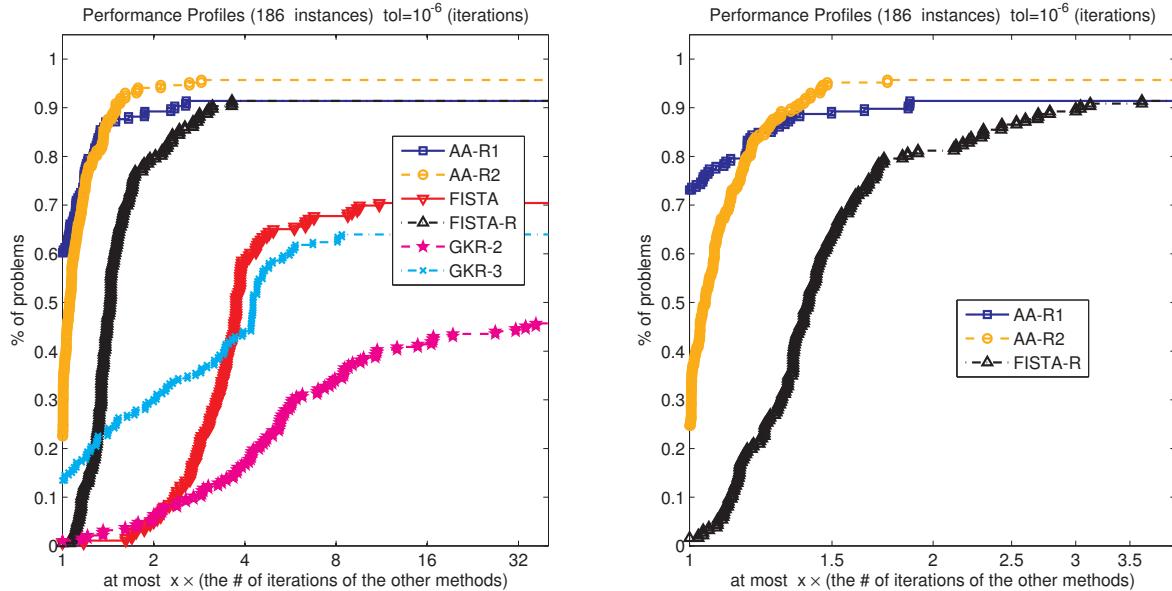
Figure 8: Iterations performance profiles for solving all the three problem classes with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

[2] S. Burer, R. D. C. Monteiro, and Y. Zhang. A computational study of a gradient-based log-barrier algorithm for a class of large-scale SDPs. *Mathematical Programming*, 95:359–379, 2003. 10.1007/s10107-002-0353-7. URL: `http://dx.doi.org/10.1007/s10107-002-0353-7`.

[3] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. Core discussion paper 2011/02, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, December 2010.

[4] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles, January 2002. URL: `http://dx.doi.org/http://dx.doi.org/doi:10.1007/s101070100263`, http://dx.doi.org/http://dx.doi.org/doi:10.1007/s101070100263 `doi:http://dx.doi.org/doi:10.1007/s101070100263`.

[5] D. Goldfarb and K. Scheinberg. Fast first-order methods for composite convex optimization with line search. *Optimization-online preprint 3004*, 2011. URL: `http://www.optimization-online.org/DB_HTML/2011/04/3004.html`.

[6] C. C. Gonzaga and E. W. Karas. Fine tuning nesterov's steepest descent algorithm for differentiable convex programming. *Mathematical Programming*, pages 1–26, 2012. URL: `http://dx.doi.org/10.1007/s10107-012-0541-z`, http://dx.doi.org/10.1007/s10107-012-0541-z `doi:10.1007/s10107-012-0541-z`.

[7] C. C. Gonzaga, E. W. Karas, and D. R. Rossetto. An optimal algorithm for constrained differentiable convex optimization. *Optimization-online preprint 3053*, pages 1–16, 2011. URL: `http://www.optimization-online.org/DB_HTML/2011/06/3053.html`.

[8] G. Lan, Z. Lu, and R.D.C. Monteiro. Primal-dual first-order methods with iteration-complexity for cone programming. *Mathematical Programming*, 126(1):1–29, 2011.

[9] M. Merritt and Y. Zhang. Interior-point gradient method for large-scale totally nonnegative least squares problems. *Journal of Optimization Theory and Applications*, 126:191–202, 2005. 10.1007/s10957-005-2668-z. URL: `http://dx.doi.org/10.1007/s10957-005-2668-z`.

[10] R. D. C. Monteiro and B. F. Svaiter. An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. Manuscript, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA, April 2011.

[11] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o $(1/k2)$. *Doklady AN SSSR*, 269(3):543–547, 1983.

[12] Y. Nesterov. *Introductory Lectures on Convex Optimization: a basic course*. Kluwer Academic Publishers, Massachusetts, 2004.

[13] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005. 10.1007/s10107-004-0552-5. URL: `http://dx.doi.org/10.1007/s10107-004-0552-5`.

[14] Y. Nesterov. *Gradient methods for minimizing composite objective function*. CORE, 2007.

[15] Yurii Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Math. Program.*, 109(2-3, Ser. B):319–344, 2007.

[16] B. O'Donoghue and E. Candès. Adaptive Restart for Accelerated Gradient Schemes. 2012. `arXiv:1204.3982v1`.

Table 1: Time comparison of the methods on CQP instances.

Problem		Time in seconds (accuracy less that 10^{-6})					
Instance	n	AA-R1	AA-R2	FISTA	FISTA-R	GKR-2	GKR-3
QP_n5_p2	5	0.1	0.1	0.1	0.1	9.9	0.5
QP_n5_p3	5	0.1	0.1	0.1	0.1	0.4	0.3
QP_n5_p4	5	0.1	0.1	0.3	0.2	16.3	2.4
QP_n5_p5	5	0.2	0.1	0.3	0.2	7.5	0.9
QP_n10_p2	10	0.1	0.1	0.1	0.1	4.7	0.5
QP_n10_p3	10	0.1	0.1	0.2	0.1	0.2	0.5
QP_n10_p4	10	0.2	0.2	0.3	0.2	99.5*(1.2 -6)	1.3
QP_n10_p5	10	0.5	0.6	0.4	0.7	20.1	7.9
QP_n20_p2	20	0.1	0.1	0.1	0.1	24.8	0.3
QP_n20_p3	20	0.1	0.1	0.1	0.1	1.8	0.4
QP_n20_p4	20	0.2	0.1	0.3	0.2	80.1	4.2
QP_n20_p5	20	0.4	0.5	0.4	0.6	76.6	12.9
QP_n50_p2	50	0.1	0.1	0.1	0.1	44.8	0.3
QP_n50_p3	50	0.1	0.2	0.4	0.2	29.5	1.1
QP_n50_p4	50	0.2	0.2	0.2	0.1	5.8	3
QP_n50_p5	50	0.2	0.2	0.4	0.3	27.4*(1.3 -6)	6.1
QP_n100_p2	100	0.1	0.1	0.2	0.1	46.2	0.5
QP_n100_p3	100	0.2	0.2	0.4	0.2	7.5	1.7
QP_n100_p4	100	0.2	0.2	0.8	0.2	121.5	2.5
QP_n100_p5	100	0.3	0.3	0.7	0.3	138.2*(2.1 -6)	8.5
QP_n200_p2	200	0.1	0.1	0.2	0.1	28.2*(1.6 -6)	0.8
QP_n200_p3	200	0.2	0.2	0.5	0.2	40.3	1.1
QP_n200_p4	200	0.2	0.2	0.7	0.3	3.3	2.5
QP_n200_p5	200	0.3	0.4	0.7	0.4	47.9	7.4
QP_n500_p2	500	0.1	0.2	0.4	0.1	145.4	0.9
QP_n500_p3	500	0.3	0.3	1.2	0.3	10.8	2.4
QP_n500_p4	500	0.3	0.4	1.4	0.6	14.7	4.3
QP_n500_p5	500	0.4	0.5	2.1	1	23.7	12.8
QP_n700_p2	700	0.2	0.2	0.6	0.2	119.9	1.1
QP_n700_p3	700	0.3	0.4	1.9	0.6	115.9	3.1
QP_n700_p4	700	0.4	0.4	3.1	1.1	18.3	7.6
QP_n700_p5	700	1.4	1.1	5.4	1.1	19.1	19.6
QP_n900_p2	900	0.3	0.3	0.8	0.7	120.0*(2.2 -6)	1.7
QP_n900_p3	900	0.6	0.6	2.9	0.9	28.5	4
QP_n900_p4	900	0.9	1	3.8	1.4	26.4	9.7
QP_n900_p5	900	1.6	1.9	6.9	2.8	31.3	24.9
QP_n1K_p2	1000	0.4	0.4	1.3	0.6	84.3*(4.0 -6)	4.3
QP_n1K_p3	1000	0.7	0.7	3.1	1.2	31.5	5
QP_n1K_p4	1000	1.1	1.5	5.6	1.6	20	8.4
QP_n1K_p5	1000	2.5	1.8	8.7	3	63.8	63.1
QP_n2K_p2	2000	1	1.2	4.3	2.8	112.0*(3.1 -6)	3.9
QP_n2K_p3	2000	1.8	2.9	10.3	3.7	88.6	11.2
QP_n2K_p4	2000	3	3.1	17.8	6	120.8	23.4
QP_n2K_p5	2000	9.5	9.6	23.6	10.4	114	68.9
QP_n5K_p2	5000	8.7	7.1	17.9	14.1	863.0*(3.2 -6)	20.2
QP_n5K_p3	5000	20.3	19	48.5	29.1	400.7	42.6
QP_n5K_p4	5000	23.6	37.8	129.1	46.7	471.1	161.4
QP_n5K_p5	5000	47.8	40.3	146.6	91	597.6	513.9
QP_n7K_p2	7000	18.9	25.1	70.5	46	1544.8*(4.4 -6)	65.2
QP_n7K_p3	7000	31.4	37.8	128.1	45.9	923.9	85.5
QP_n7K_p4	7000	25.9	29.2	167	63.7	1082.3*(1.2 -6)	264.2
QP_n7K_p5	7000	64.4	102.6	338.5	115.8	1307.1	734
QP_n9K_p2	9000	25.4	40.3	58.9	68.3	3071.3*(5.5 -6)	100.6
QP_n9K_p3	9000	64.1	54.2	219	120	2046.5	287.3
QP_n9K_p4	9000	95	86	483.1	129.2	1992	669.5
QP_n9K_p5	9000	128.8	163.3	530.7	292.7	2165.3	1203.9
QP_n10K_p2	10000	107.9	81.1	84.2	240.4	3519.7*(5.5 -6)	176.1
QP_n10K_p3	10000	84	64.3	251.8	113	1911.6	250.1
QP_n10K_p4	10000	173.8	106.9	365.4	236.9	2089.7	733.5
QP_n10K_p5	10000	141.8	202.6	688.5	451.3	2190.9	1784

[17] J. Povh, F. Rendl, and A. Wiegele. A boundary point method to solve semidefinite programs. *Computing*, 78:277–286, 2006. 10.1007/s00607-006-0182-2. URL: `http://dx.doi.org/10.1007/s00607-006-0182-2`.

[18] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2008.

[19] X.-Y. Zhao, D. Sun, and K.-C. Toh. A Newton-CG augmented lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4):1737–1765, 2010. URL: `http://link.aip.org/link/?SJE/20/1737/1`, http://dx.doi.org/10.1137/080718206 `doi:10.1137/080718206`.

Table 2: Number of iterations comparison of the methods onCQP instances.

Problem Instance	n	# of iterations (accuracy less that 10^{-6})					
		AA-R1	AA-R2	FISTA	FISTA-R	GKR-2	GKR-3
QP_n5_p2	5	26	25	59	35	913	48
QP_n5_p3	5	41	38	53	35	18	20
QP_n5_p4	5	89	92	314	104	1491	195
QP_n5_p5	5	176	94	215	152	397	75
QP_n10_p2	10	39	44	94	40	315	37
QP_n10_p3	10	41	46	140	58	16	44
QP_n10_p4	10	130	95	254	146	2000*(1.2 -6)	114
QP_n10_p5	10	462	483	425	536	579	907
QP_n20_p2	20	23	32	65	33	483	18
QP_n20_p3	20	47	40	61	46	49	36
QP_n20_p4	20	118	89	289	114	1132	407
QP_n20_p5	20	374	360	423	459	1041	1422
QP_n50_p2	50	27	36	58	33	525	29
QP_n50_p3	50	64	93	289	90	334	95
QP_n50_p4	50	126	132	353	164	873	416
QP_n50_p5	50	183	165	308	247	2000*(1.3 -6)	398
QP_n100_p2	100	37	48	124	49	618	46
QP_n100_p3	100	88	93	338	118	948	176
QP_n100_p4	100	116	130	576	186	1726	256
QP_n100_p5	100	232	235	659	321	2000*(2.1 -6)	861
QP_n200_p2	200	41	52	140	61	2000*(1.6 -6)	62
QP_n200_p3	200	69	80	326	96	525	104
QP_n200_p4	200	134	139	584	183	391	219
QP_n200_p5	200	232	259	645	298	628	803
QP_n500_p2	500	57	74	151	77	1926	58
QP_n500_p3	500	110	134	406	125	939	155
QP_n500_p4	500	164	163	710	241	1384	307
QP_n500_p5	500	293	294	1017	377	1374	1063
QP_n700_p2	700	51	72	163	111	1598	63
QP_n700_p3	700	95	134	421	130	1567	178
QP_n700_p4	700	169	189	686	238	801	346
QP_n700_p5	700	336	346	929	462	1168	1008
QP_n900_p2	900	52	72	179	120	2000*(2.2 -6)	62
QP_n900_p3	900	105	131	398	143	1148	137
QP_n900_p4	900	155	168	641	225	889	350
QP_n900_p5	900	320	372	998	431	1443	1086
QP_n1K_p2	1000	58	68	197	83	2000*(4.0 -6)	76
QP_n1K_p3	1000	111	125	433	169	1206	170
QP_n1K_p4	1000	164	192	721	258	857	379
QP_n1K_p5	1000	329	377	1274	432	996	1289
QP_n2K_p2	2000	63	76	187	85	2000*(3.1 -6)	70
QP_n2K_p3	2000	133	159	428	179	1113	172
QP_n2K_p4	2000	160	185	769	277	1540	414
QP_n2K_p5	2000	313	341	1102	438	1021	1076
QP_n5K_p2	5000	83	87	215	103	2000*(3.2 -6)	79
QP_n5K_p3	5000	194	199	453	263	1240	169
QP_n5K_p4	5000	290	299	825	388	1066	412
QP_n5K_p5	5000	356	447	1265	522	1391	1161
QP_n7K_p2	7000	85	94	219	102	2000*(4.4 -6)	77
QP_n7K_p3	7000	136	138	473	195	1341	176
QP_n7K_p4	7000	190	226	871	275	2000*(1.2 -6)	388
QP_n7K_p5	7000	369	539	1385	611	1392	1264
QP_n9K_p2	9000	85	93	225	221	2000*(5.5 -6)	78
QP_n9K_p3	9000	152	225	463	204	1378	176
QP_n9K_p4	9000	299	343	916	337	1203	523
QP_n9K_p5	9000	397	537	1379	673	1333	1259
QP_n10K_p2	10000	204	210	225	225	2000*(5.5 -6)	80
QP_n10K_p3	10000	183	198	487	264	1422	186
QP_n10K_p4	10000	236	258	918	393	1338	408
QP_n10K_p5	10000	459	541	1473	727	1318	1316

Table 3: Time comparison of the methods on SDLS instances.

Problem		Time in seconds (accuracy less that 10^{-6})					
Instance	n	AA-R1	AA-R2	FISTA	FISTA-R	GKR-2	GKR-3
BIQ_n21_m252	21	1	0.7	2.5	1	402.9*(1.4 -6)	46.8
BIQ_n31_m527	31	1	0.8	4	1.2	248.6*(1.4 -6)	47.6
BIQ_n41_m902	41	1	0.8	2.7	1	481.6*(1.4 -6)	47.8
BIQ_n51_m1377	51	1	1	2.8	1.1	285.3*(1.4 -6)	44.4
BIQ_n61_m1952	61	0.9	1.1	3.7	1.2	537.2*(1.4 -6)	42.1
BIQ_n71_m2627	71	0.8	1.2	3.1	1.2	548.0*(1.4 -6)	52.8
BIQ_n81_m3402	81	1	1.1	4.2	1.5	266.3*(1.4 -6)	40.4
BIQ_n91_m4277	91	1	1.2	3.2	1.5	279.0*(1.4 -6)	48.7
BIQ_n101_m5252	101	0.9	1.5	3.2	1.4	360.4*(1.4 -6)	56.7
BIQ_n121_m7502	121	1.1	1.1	6.3	1.7	695.4*(1.4 -6)	44.6
BIQ_n151_m11627	151	1.3	1.5	4.5	1.8	939.9*(1.4 -6)	64.5
BIQ_n201_m20502	201	2.7	2.1	6.5	3.1	707.5*(1.4 -6)	157.9
BIQ_n251_m31877	251	2.7	3.1	15.9	4	1073.8*(1.4 -6)	144.8
BIQ_n501_m126252	501	16.3	13.9	52.3	17.7	6386.8*(1.4 -6)	706
FAP_n52_m1378	52	0.4	0.5	0.5	0.4	5.6	3.7
FAP_n61_m1866	61	0.4	0.5	0.5	0.5	6.5	3.5
FAP_n65_m2145	65	0.5	0.6	0.6	0.5	11.2	3.9
FAP_n81_m3321	81	0.5	0.5	0.7	0.6	5.4	4.7
FAP_n84_m3570	84	0.5	0.6	0.6	0.5	5.9	4.4
FAP_n93_m4371	93	0.5	0.7	0.8	0.7	15.3	4.3
FAP_n98_m4851	98	0.5	0.7	0.8	0.6	8	4.4
FAP_n120_m7260	120	0.7	0.9	1	0.8	17	5.6
FAP_n174_m15225	174	0.9	1.2	2.3	1.1	21.1	8.7
FAP_n183_m14479	183	1	1.4	1.9	1.5	38.9	10.4
FAP_n252_m24292	252	2.1	2.6	3.2	2.3	39.8	20.9
FAP_n369_m26462	369	3.7	4.3	6.1	4	78.6	44.8
FAP_n2118_m322924	2118	354.3	466.9	1169	550.5	9788	5722.2
FAP_n4110_m1154467	4110	2518.5	3039.2	5151.6	3187	66187	29440.6
QAP_n144_m10672	144	6.8	7.5	44.2*(1.1 -6)	5.6	807.3*(2.5 -6)	544.1
QAP_n196_m19619	196	12	10.7	24.1	11.1	1150.6*(3.3 -6)	1299.8*(2.3 -6)
QAP_n225_m25783	225	25.7	13	18.6	14	689.2	186.8
QAP_n256_m33302	256	21.5	21.7	24.8	15.7	2265.5*(1.7 -6)	3289.2*(1.2 -6)
QAP_n289_m42362	289	28	25.9	53.7	22.9	2496.2*(4.0 -6)	1727.5
QAP_n324_m53161	324	34	36.1	98.5	27.6	3220.6*(1.9 -6)	3124.4
QAP_n400_m80828	400	58.5	53.4	440.8	52.6	4832.9*(2.4 -6)	5571.4*(1.3 -6)
QAP_n441_m98152	441	68.2	76.3	339.9	59	4207.4	12851.7*(1.8 -6)
QAP_n484_m118127	484	95.3	113.9	434.4	87.1	5387	1133.6
QAP_n625_m196598	625	282.9	210.7	1420.3	159.4	14179.8*(2.8 -6)	2809
QAP_n676_m229877	676	212.3	263.3	543.1	247.6	16715.0*(2.7 -6)	3566.5
QAP_n729_m267217	729	243	270.5	376.6	236.9	20852.3*(1.8 -6)	20097.9
QAP_n784_m308936	784	333	328.7	1384.4	295.4	24197.1*(2.0 -6)	23049.4
QAP_n900_m406843	900	480.7	516.7	1678.3	464.1	23520.9	6558.9
QAP_n1225_m752813	1225	1258	1170.2	2873.9	1121	99556.0*(1.3 -6)	15090.9
QAP_n1600_m1283258	1600	2512.7	2811.8	3610.1	3014.3	207778.0*(2.3 -6)	171500.0*(1.1 -6)
RAND_n300_m10K_p4	300	13	12.3	31.9	16.1	810.3	490.3
RAND_n300_m20K_p3	300	43.5	46.7	139.6	68.1	2918.1*(1.1 -6)	2987.4
RAND_n300_m25K_p3	300	41.6	48.9	157.7	143.3	2981.1*(1.5 -6)	2682.2*(1.0 -6)
RAND_n400_m15K_p4	400	15.7	27.2	96.8	27.7	851.8	794.4
RAND_n400_m30K_p3	400	82.3	94.4	362.2	168	4951.4	8962.2
RAND_n400_m40K_p3	400	80.1	84.7	273.2	178	9705.8*(1.5 -6)	5465.2*(1.3 -6)
RAND_n500_m20K_p4	500	38.4	58.3	117.9	47.2	2274.4	2292.9
RAND_n500_m30K_p3	500	100.1	104.1	306.4	141.3	6594.5	5263
RAND_n500_m40K_p3	500	113.7	119.6	576.4	247.7	8313	7979.3
RAND_n500_m50K_p3	500	192.9	148.3	411.9	647.7	10854.6*(1.1 -6)	10096.9*(1.3 -6)
RAND_n600_m20K_p4	600	36.8	37.6	120.8	73.6	2175.8	2349.3
RAND_n600_m40K_p3	600	233.1	199.5	816.2	321.3	12218	12037.4
RAND_n600_m50K_p3	600	168.9	166.9	571.8	445	13565.8	11012.5
RAND_n600_m60K_p3	600	218.8	265	792.8	411.6	17428.4*(1.6 -6)	15356.9*(1.2 -6)
RAND_n700_m50K_p3	700	236	239.5	1124.4	402.1	20615.5	17822.2
RAND_n700_m70K_p3	700	258.8	241	1383	592.2	19660.1	16415.1
RAND_n700_m90K_p3	700	352.7	317.7	1075.1	948.2	21967.6*(1.3 -6)	22553.8*(1.1 -6)
RAND_n800_m100K_p3	800	422.3	641.1	1514.7	1024	30362.9	25120.1
RAND_n800_m110K_p3	800	403	405.1	2206.4	1152.5	30609.5*(1.2 -6)	30426.3*(1.1 -6)
RAND_n800_m70K_p3	800	289.9	310.8	1803	728.4	45568.8	21851.4
RAND_n900_m100K_p3	900	589.3	587.5	1666.9	1278.9	48650.9	39438.5
RAND_n900_m140K_p3	900	552.5	577	2767.4	1465.5	46800.9*(1.3 -6)	39116.8
RAND_n1K_m100K_p3	1000	568	532.4	2074.3	1683.9	57775.2	43090
RAND_n1K_m150K_p3	1000	768.4	777.7	2810.2	1809.5	63396.8*(1.2 -6)	61719.5

Table 4: Number of iterations comparison of the methods on SDLS instances.

Problem		# of iterations (accuracy less that 10^{-6})					
Instance	n	AA-R1	AA-R2	FISTA	FISTA-R	GKR-2	GKR-3
BIQ_n21_m252	21	79	67	251	90	2000*(1.4 -6)	285
BIQ_n31_m527	31	79	67	251	90	2000*(1.4 -6)	285
BIQ_n41_m902	41	79	67	251	90	2000*(1.4 -6)	285
BIQ_n51_m1377	51	79	67	251	90	2000*(1.4 -6)	285
BIQ_n61_m1952	61	68	73	251	90	2000*(1.4 -6)	285
BIQ_n71_m2627	71	65	68	251	90	2000*(1.4 -6)	285
BIQ_n81_m3402	81	68	73	251	90	2000*(1.4 -6)	285
BIQ_n91_m4277	91	65	68	251	90	2000*(1.4 -6)	285
BIQ_n101_m5252	101	67	72	251	90	2000*(1.4 -6)	285
BIQ_n121_m7502	121	65	67	251	90	2000*(1.4 -6)	285
BIQ_n151_m11627	151	68	73	251	90	2000*(1.4 -6)	285
BIQ_n201_m20502	201	79	67	251	90	2000*(1.4 -6)	285
BIQ_n251_m31877	251	64	66	251	90	2000*(1.4 -6)	285
BIQ_n501_m126252	501	65	67	251	90	2000*(1.4 -6)	285
FAP_n52_m1378	52	25	27	37	26	29	18
FAP_n61_m1866	61	21	25	38	26	32	18
FAP_n65_m2145	65	25	27	38	26	46	18
FAP_n81_m3321	81	23	23	37	26	23	19
FAP_n84_m3570	84	21	26	38	26	22	19
FAP_n93_m4371	93	20	29	39	26	43	18
FAP_n98_m4851	98	21	27	43	26	36	18
FAP_n120_m7260	120	24	28	43	26	38	18
FAP_n174_m15225	174	23	29	46	26	43	18
FAP_n183_m14479	183	21	26	44	26	41	19
FAP_n252_m24292	252	26	31	45	26	35	20
FAP_n369_m26462	369	24	27	47	26	40	20
FAP_n2118_m322924	2118	25	30	54	27	43	22
FAP_n4110_m1154467	4110	26	30	54	29	47	22
QAP_n144_m10672	144	202	223	2000*(1.1 -6)	233	2000*(2.5 -6)	1688
QAP_n196_m19619	196	207	220	674	244	2000*(3.3 -6)	2000*(2.3 -6)
QAP_n225_m25783	225	375	228	385	251	968	227
QAP_n256_m33302	256	236	253	382	259	2000*(1.7 -6)	2000*(1.2 -6)
QAP_n289_m42362	289	231	248	609	266	2000*(4.0 -6)	1357
QAP_n324_m53161	324	240	262	1001	275	2000*(1.9 -6)	1939
QAP_n400_m80828	400	263	258	1744	286	2000*(2.4 -6)	2000*(1.3 -6)
QAP_n441_m98152	441	254	269	1611	293	1455	2000*(1.8 -6)
QAP_n484_m118127	484	271	290	1763	299	1462	284
QAP_n625_m196598	625	274	304	1740	316	2000*(2.8 -6)	387
QAP_n676_m229877	676	283	297	885	323	2000*(2.7 -6)	342
QAP_n729_m267217	729	283	305	527	327	2000*(1.8 -6)	1646
QAP_n784_m308936	784	279	314	1626	333	2000*(2.0 -6)	1587
QAP_n900_m406843	900	302	323	1055	343	1393	330
QAP_n1225_m752813	1225	321	333	931	368	2000*(1.3 -6)	303
QAP_n1600_m1283258	1600	333	346	583	388	2000*(2.3 -6)	2000*(1.1 -6)
RAND_n300_m10K_p4	300	102	118	337	167	387	323
RAND_n300_m20K_p3	300	405	407	1411	690	2000*(1.1 -6)	2000
RAND_n300_m25K_p3	300	409	411	1579	1244	2000*(1.5 -6)	2000*(1.0 -6)
RAND_n400_m15K_p4	400	86	151	319	160	315	304
RAND_n400_m30K_p3	400	440	442	1176	939	1841	1866
RAND_n400_m40K_p3	400	405	407	1515	1090	2000*(1.5 -6)	2000*(1.3 -6)
RAND_n500_m20K_p4	500	109	149	372	153	485	459
RAND_n500_m30K_p3	500	313	262	944	449	1379	1104
RAND_n500_m40K_p3	500	353	355	1184	810	1747	1618
RAND_n500_m50K_p3	500	442	423	1297	1179	2000*(1.1 -6)	2000*(1.3 -6)
RAND_n600_m20K_p4	600	65	74	243	150	274	187
RAND_n600_m40K_p3	600	366	368	1011	522	1508	1542
RAND_n600_m50K_p3	600	323	325	1150	533	1755	1476
RAND_n600_m60K_p3	600	401	403	1327	851	2000*(1.6 -6)	2000*(1.2 -6)
RAND_n700_m50K_p3	700	327	330	989	558	1830	1570
RAND_n700_m70K_p3	700	356	358	1187	788	1843	1521
RAND_n700_m90K_p3	700	392	394	1420	1420	2000*(1.3 -6)	2000*(1.1 -6)
RAND_n800_m100K_p3	800	383	385	1285	971	1995	1688
RAND_n800_m110K_p3	800	378	380	1378	1182	2000*(1.2 -6)	2000*(1.1 -6)
RAND_n800_m70K_p3	800	274	276	1073	739	1911	1453
RAND_n900_m100K_p3	900	381	383	1190	849	1885	1831
RAND_n900_m140K_p3	900	368	379	1429	893	2000*(1.3 -6)	1712
RAND_n1K_m100K_p3	1000	280	282	1070	840	1657	1572
RAND_n1K_m150K_p3	1000	368	370	1351	895	2000*(1.2 -6)	1972

Table 5: Time comparison of the methods on NNLS instances.

Problem		Time in seconds (accuracy less that 10^{-6})					
Instance	n	AA-R1	AA-R2	FISTA	FISTA-R	GKR-2	GKR-3
NNLS_n200_m10K	200	0.8	0.9	7.2*(2.3 -5)	1.4	25.7*(3.8 -5)	21.9*(3.3 -5)
NNLS_n200_m1K	200	1	0.6	2.2*(1.2 -5)	0.7	23.3*(1.9 -6)	16.8*(1.7 -5)
NNLS_n200_m20K	200	14.7*(2.2 -6)	0.8	6.2	1.1	46.4*(3.2 -6)	59.0*(4.8 -6)
NNLS_n200_m2K	200	0.2	0.2	1.9	0.3	14.7	14.7
NNLS_n200_m400	200	0.2	0.2	2.8	0.3	31.1*(6.7 -6)	12
NNLS_n200_m40K	200	12.8	10.7	52.3*(5.1 -6)	11.3	100.9*(1.3 -5)	87.5*(3.2 -4)
NNLS_n200_m4K	200	1.7	1.4	3.0*(1.4 -4)	1.8	33.0*(1.1 -5)	40.5*(3.3 -4)
NNLS_n200_m600	200	0.9	0.8	1.3*(1.7 -5)	0.8	17.2*(6.2 -6)	21.1*(5.6 -5)
NNLS_n200_m6K	200	0.5	0.5	3.3*(1.7 -5)	0.6	19.6*(1.3 -5)	20.8*(5.0 -6)
NNLS_n200_m800	200	0.8	0.8	2.4*(7.7 -5)	1	20.1*(8.0 -6)	23.7*(5.4 -5)
NNLS_n200_m8K	200	1.8	5.4*(1.1 -6)	5.0*(3.1 -4)	3.2	36.5*(1.6 -5)	31.7*(4.1 -4)
NNLS_n400_m10K	400	3.9	2.6	13.7*(3.0 -4)	4.8	35.8*(1.1 -4)	59.7*(1.9 -4)
NNLS_n400_m1K	400	0.4	0.4	2.7*(2.1 -6)	0.5	26.4	23.0*(7.0 -6)
NNLS_n400_m20K	400	8.5	7.7	29.3*(3.7 -4)	7.6	118.3*(1.5 -5)	105.6*(3.0 -4)
NNLS_n400_m2K	400	1.4	1.5	4.0*(1.4 -4)	1.5	41.2*(6.5 -6)	41.0*(1.1 -4)
NNLS_n400_m40K	400	61.7*(1.1 -6)	11.9	55.7*(7.9 -5)	23.2	284.7*(2.0 -5)	101.7*(1.9 -4)
NNLS_n400_m4K	400	0.9	0.8	7.1*(7.9 -6)	0.8	44.2*(1.9 -6)	44.3*(5.5 -6)
NNLS_n400_m6K	400	8.7*(1.0 -6)	1.2	6.6*(4.5 -5)	1.9	24.7*(2.0 -6)	35.2*(4.1 -5)
NNLS_n400_m800	400	0.3	0.3	4.1*(1.2 -6)	0.5	30.8	31.4*(4.6 -6)
NNLS_n400_m8K	400	2.8	4.6	12.3*(2.6 -4)	8.4*(1.1 -6)	43.2*(1.2 -4)	60.1*(2.9 -4)
NNLS_n600_m10K	600	3.1	3.6	17.6*(8.4 -5)	17.2*(1.9 -6)	47.9*(6.8 -6)	45.5*(5.6 -5)
NNLS_n600_m20K	600	28.6*(2.2 -6)	3.2	45.6	40.8*(1.5 -6)	122.8*(1.8 -5)	93.0*(4.9 -6)
NNLS_n600_m2K	600	1.1	1.1	3.3*(6.1 -6)	1.8	32.2*(2.6 -6)	28.7*(5.0 -5)
NNLS_n600_m40K	600	79.5*(4.6 -6)	65.1*(2.6 -6)	88.2*(5.8 -5)	10.5	207.4*(4.5 -5)	263.6*(3.3 -5)
NNLS_n600_m4K	600	1.8	1.2	5.4*(6.6 -6)	1.3	41.0*(2.0 -6)	31.0*(3.5 -5)
NNLS_n600_m6K	600	1.2	1.1	13.5*(7.5 -5)	1.7	39.5*(4.6 -5)	55.5*(6.0 -5)
NNLS_n600_m8K	600	1.6	1.8	18.0*(3.0 -5)	2.5	44.8*(2.8 -5)	51.2*(1.8 -5)
NNLS_n800_m10K	800	2.4	3.2	33.9*(2.2 -5)	3.5	105.8*(2.4 -6)	101.1*(1.3 -5)
NNLS_n800_m20K	800	9.5	6.8	38.8*(6.6 -5)	12.9	104.8*(1.8 -5)	91.9*(2.1 -4)
NNLS_n800_m2K	800	1.2	1.2	4.3*(1.2 -4)	1.7	29.1*(4.2 -5)	28.5*(7.8 -5)
NNLS_n800_m40K	800	70	33.1	93.7*(1.4 -4)	72.0*(5.6 -6)	360.0*(3.4 -5)	222.6*(3.9 -4)
NNLS_n800_m4K	800	3.1	2.3	12.2*(8.3 -5)	5.9	29.0*(5.3 -6)	63.1*(7.1 -5)
NNLS_n800_m6K	800	1.5	1.7	11.2*(1.2 -5)	2.5	49.1*(4.0 -6)	44.3*(1.4 -5)
NNLS_n800_m8K	800	2.7	5.4	29.7*(3.1 -6)	22.4*(1.1 -6)	67.1	91.9*(1.3 -5)
NNLS_n1K_m10K	1000	23.3*(1.8 -6)	3.5	31.2*(4.7 -5)	5.1	132.0*(4.0 -5)	97.1*(3.9 -5)
NNLS_n1K_m20K	1000	11.4	11.2	59.7*(7.5 -5)	50.0*(1.1 -6)	244.5*(6.3 -5)	166.8*(7.1 -5)
NNLS_n1K_m2K	1000	1.7	1.2	4.3*(2.0 -4)	2.1	45.1*(1.8 -5)	26.2*(1.3 -4)
NNLS_n1K_m40K	1000	103.6*(3.5 -6)	95.3*(1.1 -6)	143.5*(3.1 -4)	137.5*(1.8 -6)	331.9*(3.1 -5)	276.9*(4.0 -4)
NNLS_n1K_m4K	1000	1.9	1.6	7.6*(2.5 -5)	2.2	59.4*(4.7 -6)	35.5*(3.5 -5)
NNLS_n1K_m6K	1000	1.9	2.8	17.5*(1.1 -5)	2.1	56.6*(4.6 -6)	50.9*(5.3 -6)
NNLS_n1K_m8K	1000	7.2	5.2	18.7*(5.2 -5)	6.5	82.3*(1.3 -5)	74.2*(1.4 -4)
NNLS_n2K_m10K	2000	14.2	11.5	71.8*(5.1 -5)	18.1	139.7*(7.8 -6)	133.3*(1.0 -4)
NNLS_n2K_m20K	2000	89.2*(1.2 -6)	24.7	108.4*(2.3 -5)	54.7	371.2*(2.3 -5)	236.4*(2.4 -5)
NNLS_n2K_m40K	2000	209.7*(5.5 -6)	29.7	221.7*(2.2 -5)	60	573.2*(1.2 -5)	800.8*(1.2 -5)
NNLS_n2K_m4K	2000	7.5	6.9	23.9*(3.1 -5)	6.4	79.2*(2.4 -5)	91.5*(2.6 -5)
NNLS_n2K_m6K	2000	19.7	12.8	27.2*(8.4 -5)	19.8	124.6*(3.7 -5)	131.4*(6.2 -5)
NNLS_n2K_m8K	2000	3.6	4.9	31.7*(1.3 -5)	37.3*(1.6 -6)	127.8*(8.2 -6)	82.6*(1.1 -5)
NNLS_n4K_m10K	4000	94.8*(1.6 -6)	23.7	88.1*(4.9 -5)	31.5	468.4*(4.1 -5)	351.1*(3.6 -5)
NNLS_n4K_m20K	4000	53.6	157.3*(1.2 -6)	180.8*(8.1 -5)	185.9*(1.3 -6)	561.3*(4.7 -5)	381.3*(5.8 -5)
NNLS_n4K_m40K	4000	472.0*(1.6 -6)	452.2*(1.6 -6)	435.9*(8.3 -5)	533.3*(2.9 -6)	1558.5*(1.3 -5)	1221.4*(1.4 -4)
NNLS_n4K_m8K	4000	21.9	16.5	67.3*(8.8 -5)	28.9	209.8*(2.1 -5)	145.3*(3.8 -5)
NNLS_n6K_m20K	6000	311.9*(2.5 -6)	79.7	370.5*(7.6 -5)	267.9*(1.2 -6)	1095.7*(9.9 -6)	1064.1*(6.5 -5)
NNLS_n6K_m40K	6000	643.4*(3.0 -6)	222	603.6*(1.2 -4)	620.1*(1.2 -6)	2254.7*(5.5 -5)	2521.3*(9.2 -5)
NNLS_n8K_m20K	8000	174.5	100.1	356.7*(6.6 -5)	198.9	1477.6*(3.7 -5)	1106.5*(7.3 -5)
NNLS_n8K_m40K	8000	109.6	592.2*(2.3 -6)	1068.5*(1.0 -5)	637.7*(1.4 -6)	3209.2*(6.2 -6)	2556.4*(2.0 -5)
NNLS_n10K_m20K	10000	469.2*(2.1 -6)	162.2	530.7*(1.1 -4)	478.3*(3.0 -6)	1623.3*(5.9 -5)	1410.5*(8.2 -5)
NNLS_n10K_m40K	10000	708.7*(2.6 -6)	712.7*(2.4 -6)	1039.4*(7.0 -5)	785.6*(2.8 -6)	3231.6*(2.6 -5)	2604.1*(1.3 -4)
NNLS_n20K_m40K	20000	1952.1*(1.2 -6)	2194.0*(1.8 -6)	1792.2*(7.5 -5)	2044.2*(3.4 -6)	6670.7*(3.9 -5)	4648.8*(6.9 -5)

Table 6: Number of iterations comparison of the methods on NNLS instances.

Problem Instance	n	AA-R1	AA-R2	FISTA	FISTA-R	GKR-2	GKR-3
NNLS_n200_m10K	200	325	330	2000*(2.3 -5)	553	2000*(3.8 -5)	2000*(3.3 -5)
NNLS_n200_m1K	200	836	447	2000*(1.2 -5)	689	2000*(1.9 -6)	2000*(1.7 -5)
NNLS_n200_m20K	200	2000*(2.2 -6)	138	1246	181	2000*(3.2 -6)	2000*(4.8 -6)
NNLS_n200_m2K	200	132	141	1240	199	1046	885
NNLS_n200_m400	200	168	171	1866	251	2000*(6.7 -6)	1360
NNLS_n200_m40K	200	936	738	2000*(5.1 -6)	824	2000*(1.3 -5)	2000*(3.2 -4)
NNLS_n200_m4K	200	979	758	2000*(1.4 -4)	1147	2000*(1.1 -5)	2000*(3.3 -4)
NNLS_n200_m600	200	809	641	2000*(1.7 -5)	800	2000*(6.2 -6)	2000*(5.6 -5)
NNLS_n200_m6K	200	230	246	2000*(1.7 -5)	323	2000*(1.3 -5)	2000*(5.0 -6)
NNLS_n200_m800	200	623	627	2000*(7.7 -5)	808	2000*(8.0 -6)	2000*(5.4 -5)
NNLS_n200_m8K	200	720	2000*(1.1 -6)	2000*(3.1 -4)	1320	2000*(1.6 -5)	2000*(4.1 -4)
NNLS_n400_m10K	400	647	669	2000*(3.0 -4)	943	2000*(1.1 -4)	2000*(1.9 -4)
NNLS_n400_m1K	400	291	275	2000*(2.1 -6)	370	1702	2000*(7.0 -6)
NNLS_n400_m20K	400	775	775	2000*(3.7 -4)	866	2000*(1.5 -5)	2000*(3.0 -4)
NNLS_n400_m2K	400	860	865	2000*(1.4 -4)	1049	2000*(6.5 -6)	2000*(1.1 -4)
NNLS_n400_m40K	400	2000*(1.1 -6)	391	2000*(7.9 -5)	556	2000*(2.0 -5)	2000*(1.9 -4)
NNLS_n400_m4K	400	344	363	2000*(7.9 -6)	404	2000*(1.9 -6)	2000*(5.5 -6)
NNLS_n400_m6K	400	2000*(1.0 -6)	387	2000*(4.5 -5)	673	2000*(2.0 -6)	2000*(4.1 -5)
NNLS_n400_m800	400	224	217	2000*(1.2 -6)	321	1908	2000*(4.6 -6)
NNLS_n400_m8K	400	744	706	2000*(2.6 -4)	2000*(1.1 -6)	2000*(1.2 -4)	2000*(2.9 -4)
NNLS_n600_m10K	600	432	444	2000*(8.4 -5)	2000*(1.9 -6)	2000*(6.8 -6)	2000*(5.6 -5)
NNLS_n600_m20K	600	2000*(2.2 -6)	194	1855	2000*(1.5 -6)	2000*(1.8 -5)	2000*(4.9 -6)
NNLS_n600_m2K	600	612	555	2000*(6.1 -6)	718	2000*(2.6 -6)	2000*(5.0 -5)
NNLS_n600_m40K	600	2000*(4.6 -6)	2000*(2.6 -6)	2000*(5.8 -5)	392	2000*(4.5 -5)	2000*(3.3 -5)
NNLS_n600_m4K	600	379	387	2000*(6.6 -6)	466	2000*(2.0 -6)	2000*(3.5 -5)
NNLS_n600_m6K	600	304	308	2000*(7.5 -5)	461	2000*(4.6 -5)	2000*(6.0 -5)
NNLS_n600_m8K	600	288	336	2000*(3.0 -5)	486	2000*(2.8 -5)	2000*(1.8 -5)
NNLS_n800_m10K	800	250	260	2000*(2.2 -5)	395	2000*(2.4 -6)	2000*(1.3 -5)
NNLS_n800_m20K	800	518	436	2000*(6.6 -5)	626	2000*(1.8 -5)	2000*(2.1 -4)
NNLS_n800_m2K	800	572	577	2000*(1.2 -4)	830	2000*(4.2 -5)	2000*(7.8 -5)
NNLS_n800_m40K	800	1431	768	2000*(1.4 -4)	2000*(5.6 -6)	2000*(3.4 -5)	2000*(3.9 -4)
NNLS_n800_m4K	800	769	774	2000*(8.3 -5)	1133	2000*(5.3 -6)	2000*(7.1 -5)
NNLS_n800_m6K	800	348	349	2000*(1.2 -5)	499	2000*(4.0 -6)	2000*(1.4 -5)
NNLS_n800_m8K	800	417	444	2000*(3.1 -6)	2000*(1.1 -6)	1760	2000*(1.3 -5)
NNLS_n1K_m10K	1000	2000*(1.8 -6)	352	2000*(4.7 -5)	372	2000*(4.0 -5)	2000*(3.9 -5)
NNLS_n1K_m20K	1000	411	433	2000*(7.5 -5)	2000*(1.1 -6)	2000*(6.3 -5)	2000*(7.1 -5)
NNLS_n1K_m2K	1000	749	500	2000*(2.0 -4)	958	2000*(1.8 -5)	2000*(1.3 -4)
NNLS_n1K_m40K	1000	2000*(3.5 -6)	2000*(1.1 -6)	2000*(3.1 -4)	2000*(1.8 -6)	2000*(3.1 -5)	2000*(4.0 -4)
NNLS_n1K_m4K	1000	345	320	2000*(2.5 -5)	533	2000*(4.7 -6)	2000*(3.5 -5)
NNLS_n1K_m6K	1000	263	278	2000*(1.1 -5)	356	2000*(4.6 -6)	2000*(5.3 -6)
NNLS_n1K_m8K	1000	718	547	2000*(5.2 -5)	870	2000*(1.3 -5)	2000*(1.4 -4)
NNLS_n2K_m10K	2000	705	580	2000*(5.1 -5)	889	2000*(7.8 -6)	2000*(1.0 -4)
NNLS_n2K_m20K	2000	2000*(1.2 -6)	414	2000*(2.3 -5)	521	2000*(2.3 -5)	2000*(2.4 -5)
NNLS_n2K_m40K	2000	2000*(5.5 -6)	286	2000*(2.2 -5)	449	2000*(1.2 -5)	2000*(1.2 -5)
NNLS_n2K_m4K	2000	804	715	2000*(3.1 -5)	897	2000*(2.4 -5)	2000*(2.6 -5)
NNLS_n2K_m6K	2000	828	850	2000*(8.4 -5)	1193	2000*(3.7 -5)	2000*(6.2 -5)
NNLS_n2K_m8K	2000	262	313	2000*(1.3 -5)	2000*(1.6 -6)	2000*(8.2 -6)	2000*(1.1 -5)
NNLS_n4K_m10K	4000	2000*(1.6 -6)	543	2000*(4.9 -5)	826	2000*(4.1 -5)	2000*(3.6 -5)
NNLS_n4K_m20K	4000	668	2000*(1.2 -6)	2000*(8.1 -5)	2000*(1.3 -6)	2000*(4.7 -5)	2000*(5.8 -5)
NNLS_n4K_m40K	4000	2000*(1.6 -6)	2000*(1.6 -6)	2000*(8.3 -5)	2000*(2.9 -6)	2000*(1.3 -5)	2000*(1.4 -4)
NNLS_n4K_m8K	4000	580	591	2000*(8.8 -5)	929	2000*(2.1 -5)	2000*(3.8 -5)
NNLS_n6K_m20K	6000	2000*(2.5 -6)	490	2000*(7.6 -5)	2000*(1.2 -6)	2000*(9.9 -6)	2000*(6.5 -5)
NNLS_n6K_m40K	6000	2000*(3.0 -6)	695	2000*(1.2 -4)	2000*(1.2 -6)	2000*(5.5 -5)	2000*(9.2 -5)
NNLS_n8K_m20K	8000	964	723	2000*(6.6 -5)	953	2000*(3.7 -5)	2000*(7.3 -5)
NNLS_n8K_m40K	8000	334	2000*(2.3 -6)	2000*(1.0 -5)	2000*(1.4 -6)	2000*(6.2 -6)	2000*(2.0 -5)
NNLS_n10K_m20K	10000	2000*(2.1 -6)	744	2000*(1.1 -4)	2000*(3.0 -6)	2000*(5.9 -5)	2000*(8.2 -5)
NNLS_n10K_m40K	10000	2000*(2.6 -6)	2000*(2.4 -6)	2000*(7.0 -5)	2000*(2.8 -6)	2000*(2.6 -5)	2000*(1.3 -4)
NNLS_n20K_m40K	20000	2000*(1.2 -6)	2000*(1.8 -6)	2000*(7.5 -5)	2000*(3.4 -6)	2000*(3.9 -5)	2000*(6.9 -5)