# An adaptive accelerated first-order method for convex optimization

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

July 30, 2012 (Revised: May 14, 2014)

**Abstract**

This paper presents a new accelerated variant of Nesterov's method for solving composite convex optimization problems in which certain acceleration parameters are adaptively (and aggressively) chosen so as to substantially improve its practical performance compared to existing accelerated variants while at the same time preserve the optimal iteration-complexity shared by these methods. Computational results are presented to demonstrate that the proposed adaptive accelerated method endowed with a restarting scheme outperforms other existing accelerated variants.

## 1 Introduction

The methods of choice for solving large-scale convex optimization problems, specially when high accuracy is not needed, are first-order methods due to their cheap iteration cost (time and memory). In [11] (see also [13]), Nesterov presents a scheme for accelerating first-order methods, more specifically, the steepest descent method for unconstrained convex optimization and , more generally, 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 corresponding 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 for the aforementioned problems, and more generally the composite convex optimization problem, have been proposed and studied in the literature (see for example [1, 3, 5, 6, 8, 10, 11, 12, 14, 15, 18]). Moreover, there has been an increasing effort towards improving the practical performance of these methods (see for example [6, 16]) in the solution of large-scale convex optimization problems.

All of the accelerated variants mentioned above use certain accelerated parameters which are obtained by using well-known update formulas. This paper presents a new accelerated variant for composite convex optimization in which the acceleration parameters are adaptively (and aggressively) chosen so as to substantially improve its practical performance in comparison to these aforementioned variants while at the same time preserve the optimal iteration-complexity shared by these methods (with the exemption of [16]). More specifically, the new accelerated variant chooses

*School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205 (email:monteiro@isye.gatech.edu). The work of this author was partially supported by NSF Grants CMMI-0900094 and CMMI-1300221, and ONR Grant ONR N00014-11-1-0062.

†School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205 (email: camiort@gatech.edu).

‡IMPA, Estrada Dona Castorina 110, 22460-320 Rio de Janeiro, Brazil (email:benar@impa.br). The work of this author was partially supported by CNPq grants no. 303583/2008-8, 302962/2011-5, 480101/2008-6, 474944/2010-7, FAPERJ grants E-26/102.821/2008, E-26/102.940/2011.

the acceleration parameters at every iteration by solving a simple two-variable convex quadratically constrained linear program. Moreover, its iteration cost is comparable to or better than that of the aforementioned accelerated methods since it only computes a single resolvent (or proximal subproblem) at every iteration.

For the purpose of our computational experiments, we have implemented the new accelerated variant endowed with two restarting schemes in order to improve its practical performance. The first restarting scheme is an aggressive one which restarts the method from the last iterate whenever the objective function increases (also proposed in [16]). The second restarting scheme is a more conservative (and more robust) one which restarts the method whenever the objective function increases and the number of iterations at that point is sufficiently large. Finally, computational results are presented on several conic quadratic programming instances showing that the new accelerated variant endowed with either one of the restarting schemes is faster and more robust than other existing Nesterov's variants.

Our paper is organized as follows. Section 2 presents a class of accelerated first-order methods, and corresponding iteration-complexity results, for solving composite convex optimization problems. Moreover, this section proposes a specific instance of the latter class which adaptively chooses the sequence of acceleration parameters so as to greedily minimize an upper bound on the primal gap. Section 3 presents computational results comparing the latter instance (endowed with one of the aforementioned restart schemes) with other existing variants of Nesterov's method. Finally, Section 4 presents some final remarks.

## 2 An accelerated first-order method

This section introduces an accelerated first-order method presented in [10] for solving composite convex optimization problems. First, we propose a method with general acceleration parameters satisfying certain conditions which still guarantee an optimal iteration-complexity. Next, for the purpose of improving the practical performance of the method, we propose a specific way for choosing these parameters that greedily minimizes an upper bound on the primal gap.

Let $\mathbb{R}$ denote the set of real numbers and define $\bar{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. Also, let $\mathcal{X}$ denote a finite dimensional real inner product space with inner product and induced norm denoted by $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$, respectively. Our problem of interest is the composite convex optimization problem

$$\phi^* := \min_{x \in \mathcal{X}} \phi(x) := f(x) + h(x) \tag{1}$$

where:

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

**C.2** $f$ is a real-valued function which is differentiable and convex on a closed convex set

$$\Omega \supseteq \operatorname{dom} h := \{x \in \mathcal{X} : h(x) < \infty\};$$

**C.3** $\nabla f$ is $L$-Lipschitz continuous on $\Omega$ for some $L > 0$, i.e.,

$$\|\nabla f(x') - \nabla f(x)\| \leq L\|x' - x\| \qquad \forall x, x' \in \Omega;$$

**C.4** the set $X^*$ of optimal solutions of (1) is non-empty.

There are many accelerated variants of Nesterov's method for solving (1) under assumptions C.1–C.4. In this paper we are interested in studying the properties of the following accelerated variant of Nesterov's method whose statement uses the projection operator $\Pi_\Omega : \mathcal{X} \to \Omega$ onto $\Omega$ defined as

$$\Pi_\Omega(x) = \arg\min_{\tilde{x} \in \Omega} \{\|x - \tilde{x}\|\} \qquad \forall x \in \mathcal{X}.$$

**Accelerated Class**: A class of accelerated algorithms for (1)

0) Let $\lambda \in (0, 1/L]$ and $x^0 \in \mathcal{X}$ be given, and set $A_0 = 0$, $y^0 = x^0$, the affine function $\Gamma_0 : \mathcal{X} \to \mathbb{R}$ as $\Gamma_0 \equiv 0$, and $k = 1$;

1) compute $a_k$, $\tilde{x}^k$, $\tilde{x}^k_\Omega$, $y^k$ and the affine function $\gamma_k : \mathcal{X} \to \mathbb{R}$ as

$$a_k := \frac{\lambda + \sqrt{\lambda^2 + 4\lambda A_{k-1}}}{2}, \tag{2}$$

$$\tilde{x}^k := \frac{A_{k-1}}{A_{k-1} + a_k}\, y^{k-1} + \frac{a_k}{A_{k-1} + a_k}\, x^{k-1}, \qquad \tilde{x}^k_\Omega = \Pi_\Omega(\tilde{x}^k), \tag{3}$$

$$y^k := \arg\min_{x \in \mathcal{X}} \left\{ \lambda h(x) + \frac{1}{2}\|x - \tilde{x}^k + \lambda \nabla f(\tilde{x}^k_\Omega)\|^2 \right\}, \tag{4}$$

$$\gamma_k(x) := f(\tilde{x}^k_\Omega) + \langle \nabla f(\tilde{x}^k_\Omega), y^k - \tilde{x}^k_\Omega \rangle + h(y^k) + \frac{1}{\lambda}\langle \tilde{x}^k - y^k, x - y^k \rangle \qquad \forall x \in \mathcal{X}; \tag{5}$$

2) choose a pair $(A'_{k-1}, a'_k) = (A', a') \in \mathbb{R} \times \mathbb{R}$ such that

$$A' \geq 0,\ a' \geq 0, \quad A' + a' \geq A_{k-1} + a_k, \tag{6}$$

$$(A' + a')\phi(y^k) \leq \inf_{x \in \mathcal{X}} \left\{ (A'\Gamma_{k-1} + a'\gamma_k)(x) + \frac{1}{2}\|x - x^0\|^2 \right\}, \tag{7}$$

with the safeguard that $A'_0 = A_0 = 0$ when $k = 1$;

3) compute

$$A_k = A'_{k-1} + a'_k, \tag{8}$$

$$\Gamma_k = \frac{A'_{k-1}}{A'_{k-1} + a'_k}\Gamma_{k-1} + \frac{a'_k}{A'_{k-1} + a'_k}\gamma_k, \tag{9}$$

$$x^k = x^0 - A_k \nabla \Gamma_k; \tag{10}$$

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

---

Observe that (7), (8) and (9) imply

$$A_k \phi(y^k) \leq \inf_{x \in \mathcal{X}} \left\{ A_k \Gamma_k(x) + \frac{1}{2}\|x - x^0\|^2 \right\}. \tag{11}$$

Also, note that the Accelerated Class does not specify how to choose $(A'_{k-1}, a'_k) \in \mathbb{R} \times \mathbb{R}$ satisfying (6) and (7). Different choices of the pair $(A'_{k-1}, a'_k)$ determines different instances of the class. For example, if the pair $(A'_{k-1}, a'_k)$ is chosen as $(A'_{k-1}, a'_k) = (A_{k-1}, a_k)$ then, using equation (30) of Lemma A.3, it can be shown that the Accelerated Class reduces to Algorithm I of [10]. Moreover, if $(A'_{k-1}, a'_k) = (A_{k-1}, a_k)$ and $\Omega = \mathcal{X}$ (and hence the gradient of $f$ is defined and is $L$-Lipschitz continuous on the whole $\mathcal{X}$), then the Accelerated Class reduces to a well-known Nesterov's accelerated variant, namely FISTA [1]. On the other hand, this paper is concerned with the instance of the above class which chooses the pair $(A'_{k-1}, a'_k)$ which maximizes $A_k$ subject to conditions (8) and (11).

The following result not only shows that the pair $(A'_{k-1}, a'_k) = (A_{k-1}, a_k)$ satisfies (6) and (7), but that any instance of the above class has optimal iteration-complexity.

**Proposition 2.1.** *The following statements hold for every $k \geq 1$ :*

*a) $(A'_{k-1}, a'_k) = (A_{k-1}, a_k)$ satisfies (6) and (7), and hence the Accelerated Class is well-defined;*

*b) $\Gamma_k$ is affine and $\Gamma_k \leq \phi$;*

*c) $A_k \geq \lambda k^2/4$;*

*d) there holds*

$$\phi(y^k) - \phi^* \leq \frac{d_0^2}{2A_k} \leq \frac{2d_0^2}{\lambda k^2} \tag{12}$$

*where $d_0$ is the distance of $x^0$ to $X^*$.*

*Proof.* Statements a), b) and c) follow from Lemma A.3. Letting $x^*$ denote the projection of $x^0$ onto $X^*$, d) follows from c) and the fact that (11) and b) imply that

$$A_k\phi(y^k) \leq A_k\Gamma_k(x^*) + \frac{1}{2}\|x^* - x^0\|^2 \leq A_k\phi^* + \frac{1}{2}d_0^2.$$

□

Observe that Proposition 2.1(d) implies that any instance of the Accelerated Class with $\lambda = 1/L$ has the optimal iteration-complexity $\mathcal{O}(Ld_0^2/k^2)$. Moreover, based on the fact that (8) and (12) imply that

$$\phi(y^k) - \phi^* = \mathcal{O}\left(\frac{1}{A'_{k-1} + a'_k}\right),$$

our new accelerated variant chooses the pair $(A'_{k-1}, a'_k)$ as

$$(A'_{k-1}, a'_k) \in \arg\max_{A', a'}\{A' + a' : A' \geq 0, a' \geq 0 \text{ and } (A', a') \text{ satisfies (7)}\} \tag{13}$$

so as to greedily reduce the primal gap $\phi(y^k) - \phi^*$ as much as possible.

We will now establish that (7) is equivalent to a convex quadratic constraint, and hence that (13) is a simple two-variable convex quadratically constrained linear program. We first state the following simple technical result.

**Lemma 2.2.** *Let $\chi \in \mathbb{R}$, $x^0, y \in \mathcal{X}$ and an affine function $\Gamma : \mathcal{X} \to \mathbb{R}$ be given. Then, the following conditions are equivalent:*

a) $\min\{\Gamma(x) + \|x - x^0\|^2/2 : x \in \mathcal{X}\} \geq \chi$;

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

*Proof.* The result follows from the optimality conditions of the unconstrained optimization problem in a). □

The following result gives the aforementioned characterization for condition (7).

**Proposition 2.3.** *For every $k \geq 1$, condition (7) holds if and only if*

$$A'[\phi(y^k) - \Gamma_{k-1}(x^0)] + a'[\phi(y^k) - \gamma_k(x^0)] + \frac{1}{2}\|A'\nabla\Gamma_{k-1} + a'\nabla\gamma_k\|^2 \leq 0. \tag{14}$$

*Proof.* This result follows from Lemma 2.2 with $\Gamma = A'\Gamma_{k-1} + a'\gamma_k$, $\chi = (A' + a')\phi(y^k)$ and $y = y^k$. □

Note that (14) is a single convex quadratic constraint on the scalars $A'$ and $a'$. Hence, the new accelerated method based on (13) chooses the pair $(A'_{k-1}, a'_k)$ as an optimal solution of the simple two-variable convex quadratically constrained linear program

$$\begin{array}{ll} \max & A' + a' \\ \text{s.t.} & A'[\phi(y^{k+1}) - \Gamma_k(x^0)] + a'[\phi(y^{k+1}) - \gamma_{k+1}(x^0)] + \frac{1}{2}\|A'\nabla\Gamma_k + a'\nabla\gamma_{k+1}\|^2 \leq 0, \\ & A', a' \geq 0. \end{array} \tag{15}$$

The remaining part of this section discusses the case where problem (15) is unbounded. It will be seen that this case implies that $y^k \in X^*$, in which case the new accelerated method may be successfully stopped. The following result, whose proof is given in Appendix B, considers a slightly more general problem which contains (15) as a special case.

**Proposition 2.4.** *Suppose that $x^0, y \in \mathcal{X}$ and $\gamma_1, \ldots, \gamma_m : \mathcal{X} \to \mathbb{R}$ are affine functions minorizing $\phi$ and consider the problem*

$$\begin{array}{ll} \max & \sum_{i=1}^m \alpha_i \\ \text{s.t.} & \inf_{x\in\mathcal{X}} \left\{\sum_{i=1}^m \alpha_i\gamma_i(x) + \frac{1}{2}\|x - x^0\|^2\right\} \geq \sum_{i=1}^m \alpha_i\phi(y), \\ & \alpha_1 \geq 0, \ldots, \alpha_m \geq 0. \end{array} \tag{16}$$

*Then, the following conditions are equivalent:*

*a) problem* (16) *is unbounded;*

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

$$\sum_{i=1}^m \bar{\alpha}_i\nabla\gamma_i = 0, \qquad \bar{\alpha}_i[\phi(y) - \gamma_i(y)] = 0, \quad i = 1, \ldots, m. \tag{17}$$

*In both cases, $y \in X^*$.*

Proposition 2.4 deals with the case where (15) is unbounded. On the other hand, when (15) is bounded, its feasible set is compact, and hence (15) has an optimal solution.

In our computational experiments, we will refer to the instance of the Accelerated Class which chooses the pair $(A'_{k-1}, a'_k)$ as an optimal solution of (15) as the *adaptive accelerated* (AA) method.

# 3 Numerical results

In this section, we describe two restarting variants of 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 3.1);

b) semidefinite least squares (SDLSs) (see Subsection 3.2); and

c) random nonnegative least squares (NNLSs) (see Subsection 3.3).

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

$$\left\| y^k - (I + \partial h)^{-1} \left( y^k - \frac{1}{L} \nabla f(y^k) \right) \right\| \leq 10^{-6} \tag{18}$$

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 (18) 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 $k$th iteration increases, i.e., $\phi(y^{k-1}) < \phi(y^k)$, then the method is restarted at step 0 with $x^0 = y^{k-1}$ and $y^0 = y^{k-1}$. FISTA-R refers to the variant of FISTA which incorporates this restarting scheme. Moreover, FISTA-R is equivalent to one of the restarting variants of FISTA discussed in [16].

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 $k$th iteration whenever $\phi(y^{k-1}) < \phi(y^k)$ 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. 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 cannot 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.

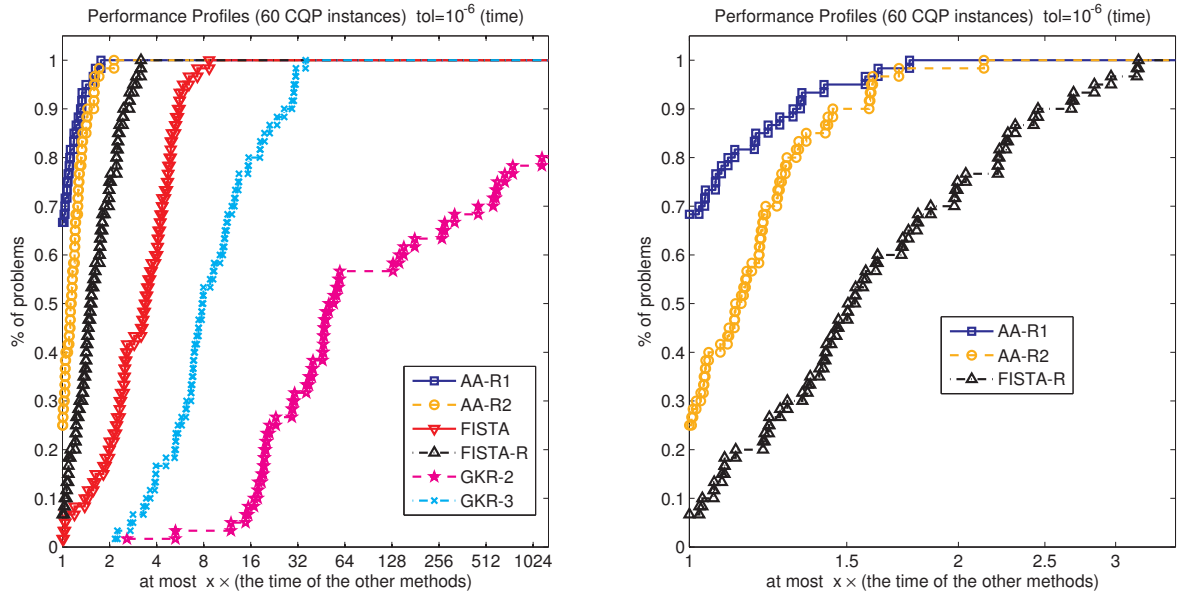


Figure 1: Time performance profiles for solving CQP instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

### 3.1 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].

Let $\mathbb{R}^n$ denote the $n$-dimensional Euclidean space, $\mathcal{S}^n$ denote the set of all $n \times n$ symmetric matrices and $\mathcal{S}^n_+$ denote the cone of $n \times n$ symmetric positive semidefinite matrices. Given $Q \in \mathcal{S}^n_+$, $b \in \mathbb{R}^n$, $l \in \{\mathbb{R} \cup \{-\infty\}\}^n$ and $u \in \{\mathbb{R} \cup \{\infty\}\}^n$ such that $l \leq u$, the box constrained convex quadratic programming problem is defined as

$$\min_{x \in \mathbb{R}^n} \left\{ x^T Q x + b^T x \ : \ l \leq x \leq u \right\}. \tag{19}$$

Letting $f$ and $h$ be defined as

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

where $B = \{x \in \mathbb{R}^n : l \leq x \leq u\}$, we can easily see that (19) is a special case of (1) with $\Omega = \mathcal{X} = \mathbb{R}^n$

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

Note that AA-R1 and AA-R2 outperform the other methods on most of the random sparse CQP instances. From Figures 1 and 2 we can see that the aggressive restart scheme of AA-R1 performs slight better than the conservative one of AA-R2 on these instances.

### 3.2 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.

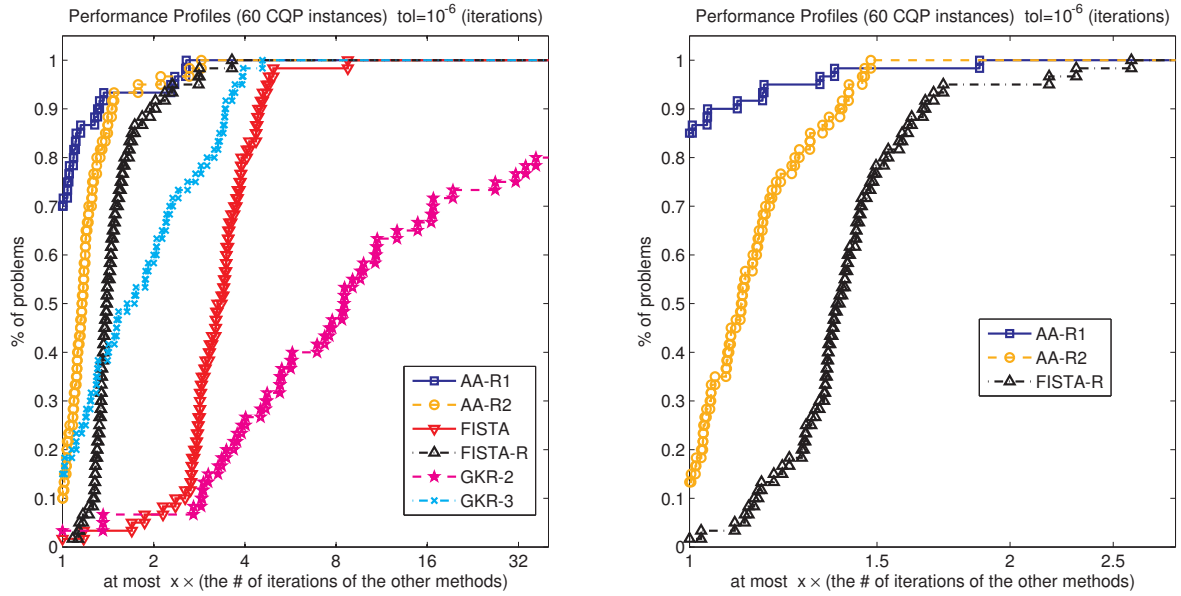


Figure 2: Iteration performance profiles for solving CQP instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

Given a linear map $\mathcal{A} \in \mathcal{S}^n \to \mathbb{R}^m$ and $b \in \mathbb{R}^m$, the semidefinite programming (SDP) feasibility problem consists of finding $x$ such that

$$\mathcal{A}x = b, \qquad x \in \mathcal{S}^n_+.$$

We can solve the above problem by considering the SDLS reformulation

$$\min_{x \in \mathcal{S}^n} \left\{ \frac{1}{2} \|\mathcal{A}x - b\|^2 \ : \ x \in \mathcal{S}^n_+ \right\}. \tag{20}$$

Letting $f$ and $h$ be defined as

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

where $\|\cdot\|$ denotes the Euclidean norm, we can easily see that (20) is a special case of (1) with $\Omega = \mathcal{X} = \mathcal{S}^n$.

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.

Note that AA-R1 and AA-R2 outperform the other methods on most of the SDLS instances. From Figures 3 and 4 we can see that the aggressive restart scheme of AA-R1 performs slight better than the conservative one of AA-R2 on these instances.

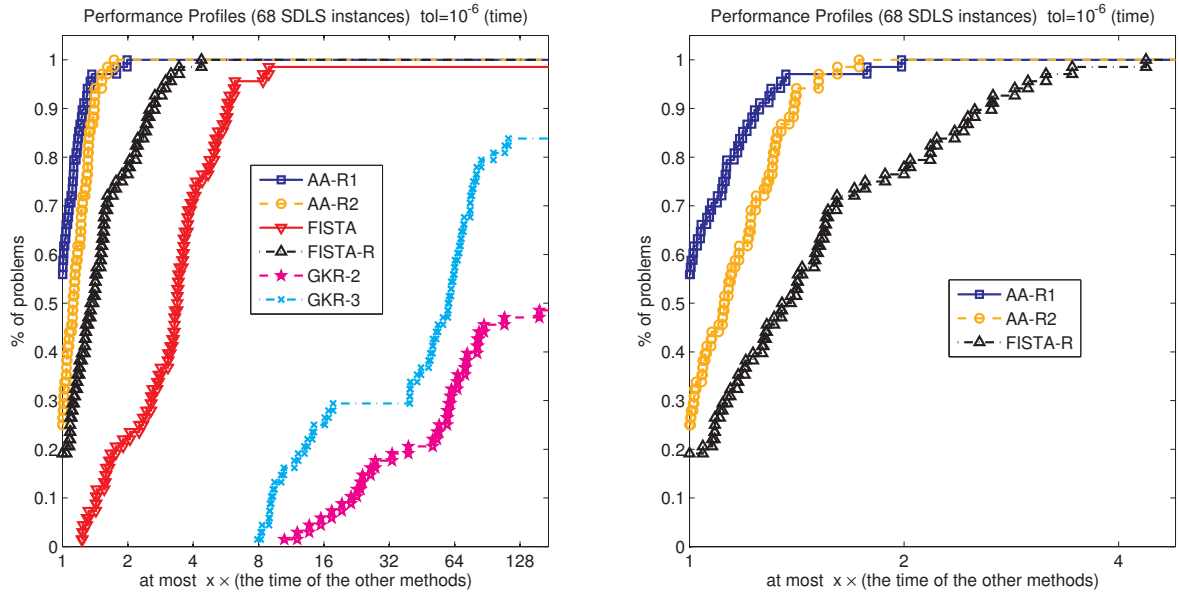


Figure 3: Time performance profiles for solving SDLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

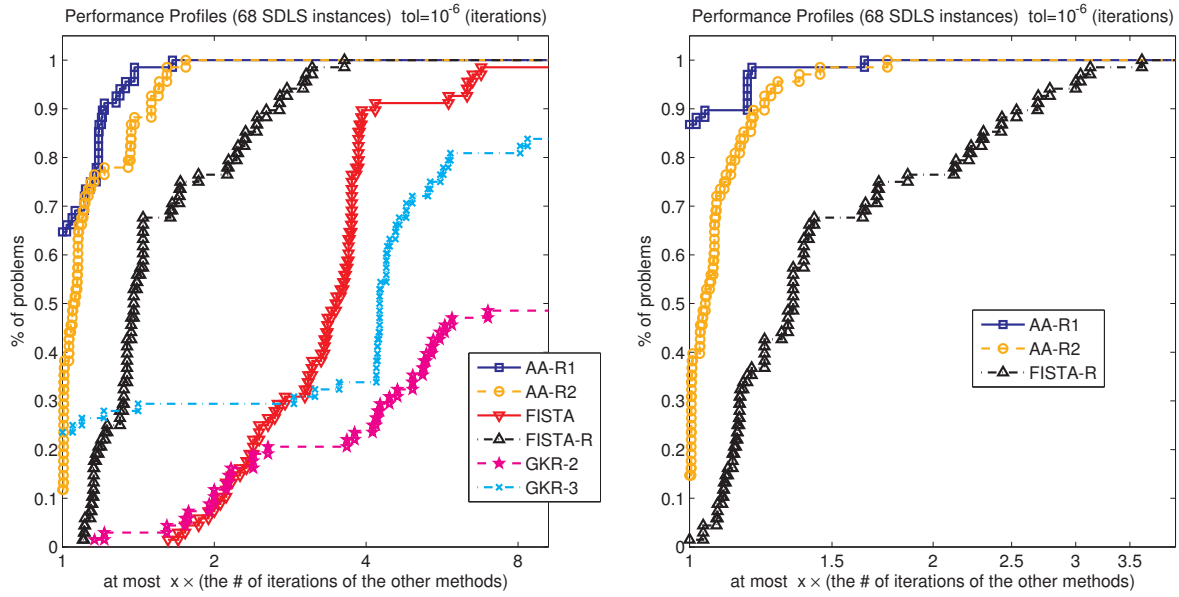


Figure 4: Number of iterations performance profiles for solving SDLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

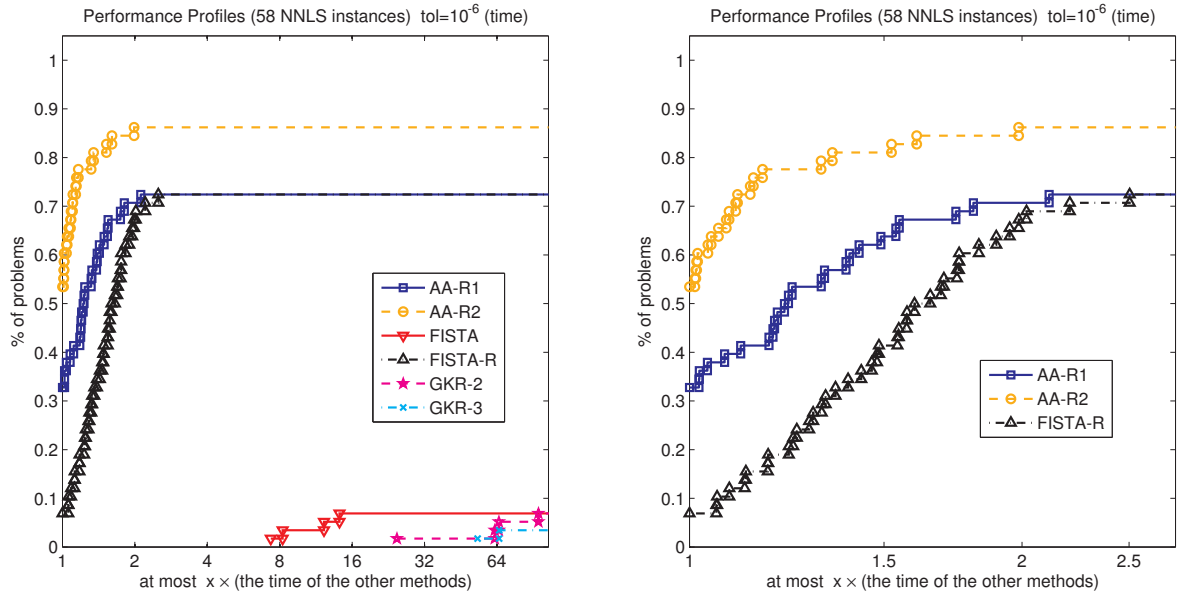


Figure 5: Time performance profiles for solving NNLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

### 3.3 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

$$\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\|Ax-b\|^2 \;:\; x \geq 0 \right\}, \tag{21}$$

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

Letting $f$ and $h$ be defined as

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

where $\mathbb{R}^n_+$ is the cone of nonnegative vectors in $\mathbb{R}^n$, we can easily see that (21) is a special case of (1) with $\Omega = \mathcal{X} = \mathbb{R}^n$.

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.

Note that AA-R2 outperforms the other methods on most of the NNLS instances where a solution with the required accuracy was found. From Figures 5 and 6 we can see that the conservative restart scheme of AA-R2 performs better and is more robust than the aggressive one of AA-R1 on these instances. Also, we can see that the methods that do not incorporate a restart scheme are only able to solve less than 10% of the NNLS instances, while the ones that do incorporate a restart scheme solve at least 70% of them.

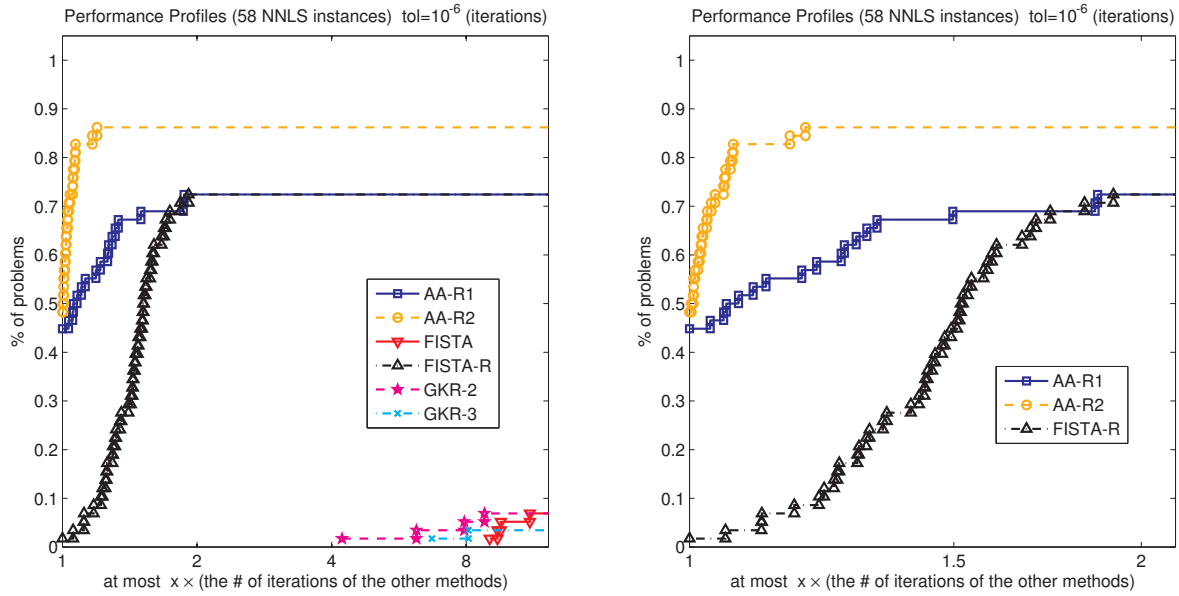


Figure 6: Number of iterations performance profiles for solving NNLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

# 4 Concluding remarks

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 sections. 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. 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.

Finally, our implementation of the AA method and its restarting variants can be found at `http://www.isye.gatech.edu/~cod3/COrtiz/software/`.


## 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`, `doi:10.1137/080716542`.

[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`.

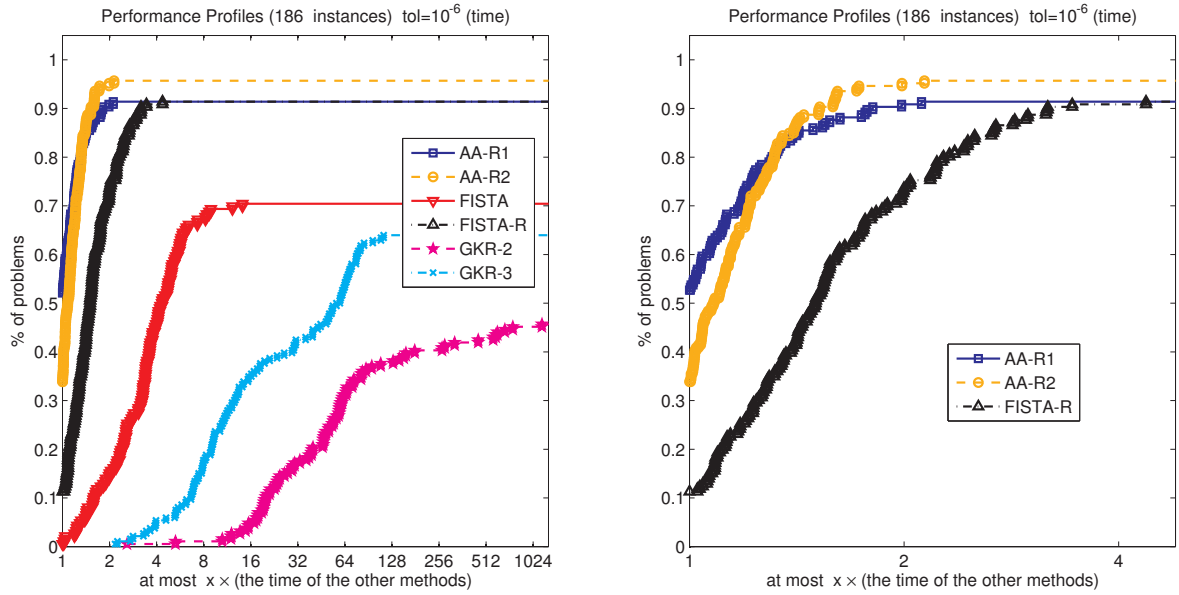


Figure 7: Time performance profiles for solving all the three problem classes with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

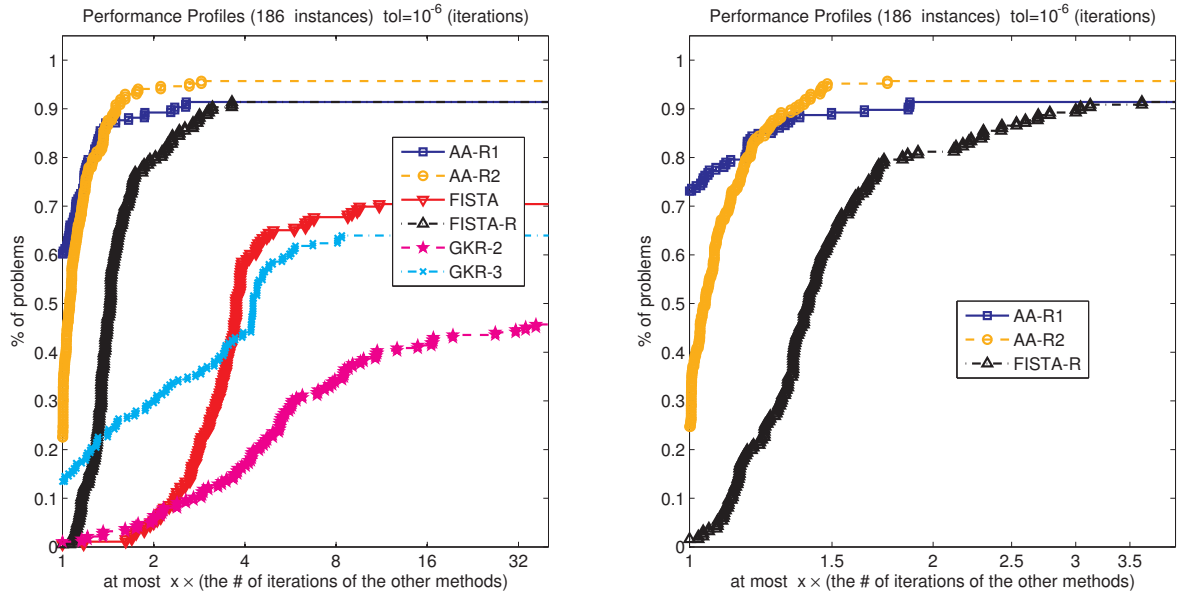


Figure 8: Iterations performance profiles for solving all the three problem classes with accuracy $\bar{\epsilon} = 10^{-6}$. On the left we include all the methods and on the right we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R.

[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, 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, doi:10.1007/s10107-012-0541-z.

[7] Clóvis C. Gonzaga, Elizabeth W. Karas, and Diane 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. *SIAM Journal on Optimization*, 23(2):1092–1125, 2013. URL: http://epubs.siam.org/doi/abs/10.1137/110833786, arXiv:http://epubs.siam.org/doi/pdf/10.1137/110833786, doi:10.1137/110833786.

[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*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

[13] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

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

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

[16] Brendan O'Donoghue and Emmanuel Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, pages 1–18, 2013. URL: http://dx.doi.org/10.1007/s10208-013-9150-3, doi:10.1007/s10208-013-9150-3.

[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`, `doi:10.1137/080718206`.

# A Technical results use in Section 2

This section presents some technical results used in Section 2. In particular, Lemma A.3 shows that the Accelerated Class satisfies statements a)–c) of Proposition 2.1 at every iteration.

Assume that the functions $\phi$, $f$, and $h$, and the set $\Omega$ satisfy C.1–C.4 of Section 2 in the following results. We start by showing two technical lemmas that will be used to show the main result of this appendix in Lemma A.3.

**Lemma A.1.** *Let $A \geq 0$, $\lambda > 0$, $x^0, y \in \mathcal{X}$ and a closed convex function $\Gamma : \mathcal{X} \to \mathbb{R}$ be given, and assume that the triple $(y, A, \Gamma)$ satisfies*

$$A\phi(y) \quad \leq \quad \min_{u \in \mathcal{X}} \{A\Gamma(u) + \frac{1}{2}\|u - x^0\|^2\}, \qquad A\Gamma \leq A\phi. \tag{22}$$

*Also, define $x \in \mathcal{X}$, $a > 0$ and $\tilde{x} \in \mathcal{X}$ as*

$$x = \arg\min_{u \in \mathcal{X}} \left\{ A\Gamma(u) + \frac{1}{2}\|u - x^0\|^2 \right\}, \tag{23}$$

$$a := \frac{\lambda + \sqrt{\lambda^2 + 4\lambda A}}{2}, \qquad \tilde{x} := \frac{A}{A+a}y + \frac{a}{A+a}x. \tag{24}$$

*Then, for any proper closed convex function $\gamma : \mathcal{X} \to \mathbb{R}$ minorizing $\phi$ we have*

$$\min_{u \in \mathcal{X}} \left\{ a\gamma(u) + A\Gamma(u) + \frac{1}{2}\|u - u_0\|^2 \right\} \geq (A+a)\frac{\theta}{\lambda},$$

*where*

$$\theta := \min_{u \in \mathcal{X}} \left\{ \lambda\gamma(u) + \frac{1}{2}\|u - \tilde{x}\|^2 \right\}. \tag{25}$$

*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 (24) implies that

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

In view of (23), 22, the fact that $\gamma \leq \phi$, the functions $\Gamma$ and $\gamma$ are convex, and the last three relations, we conclude that

$$\begin{aligned}
a\gamma(u) + A\Gamma(u) + \frac{1}{2}\|u - x^0\|^2 & \\
&\geq a\gamma(u) + \frac{1}{2}\|u - x\|^2 + \min_{u\in\mathcal{X}} \left\{ A\Gamma(u) + \frac{1}{2}\|u - x^0\|^2 \right\} \\
&\geq a\gamma(u) + \frac{1}{2}\|u - x\|^2 + A\gamma(y) \\
&\geq (A+a)\gamma\left(\frac{Ay + au}{A+a}\right) + \frac{1}{2}\|u - x\|^2 \\
&= (A+a)\left[\gamma(\tilde{u}) + \frac{(A+a)}{2a^2}\|\tilde{u} - \tilde{x}\|^2\right] \\
&\geq (A+a)\min_{u'\in\mathcal{X}} \left\{ \gamma(u') + \frac{1}{2\lambda}\|u' - \tilde{x}\|^2 \right\}.
\end{aligned}$$

The result now follows from (25) and the fact that the above inequality holds for any $u \in \mathcal{X}$. □

**Lemma A.2.** *Let $\lambda > 0$, $\xi \in \mathcal{X}$ and a proper closed convex function $g : \mathcal{X} \to (-\infty, \infty]$ be given and denote the optimal solution and optimal value of*

$$\min_{x\in\mathcal{X}} \left\{ \lambda g(x) + \frac{1}{2}\|x - \xi\|^2 \right\}. \tag{26}$$

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

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

*has the property that $\gamma \leq g$ and the optimal solution and optimal value of*

$$\min_{x\in\mathcal{X}} \left\{ \lambda\gamma(x) + \frac{1}{2}\|x - \xi\|^2 \right\} \tag{27}$$

*is $\bar{x}$ and $\bar{g}$, respectively.*

*Proof.* The optimality conditions for (26) imply that $0 \in \lambda\partial g(\bar{x}) + \bar{x} - \xi$, or equivalently

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

and hence that $\gamma \leq g$, by the definition of $\gamma$ and the subgradient of a function. Moreover, $\bar{x}$ clearly satisfies the optimality condition of (27). Hence, the last claim of the proposition follows. □

The main result of this appendix is as follows.

**Lemma A.3.** *Let $A \geq 0$, $0 < \lambda \leq 1/L$, $x^0, y \in \mathcal{X}$ and an affine function $\Gamma : \mathcal{X} \to \mathbb{R}$ be given, and assume that the triple $(y, A, \Gamma)$ satisfies (22). Compute $x$ as in (23), $a$ and $\tilde{x}$ as in (24),*

$$\tilde{x}_\Omega := \Pi_\Omega(\tilde{x}), \qquad A_+ := A + a$$

*and*

$$y^+ \;:=\; (I + \lambda\partial h)^{-1}(\tilde{x} - \lambda\nabla f(\tilde{x}_\Omega)) = \arg\min_{u\in\mathcal{X}} \left\{ \lambda h(u) + \frac{1}{2}\|u - \tilde{x} + \lambda\nabla f(\tilde{x}_\Omega)\|^2 \right\},$$

*and define the affine functions* $\gamma, \Gamma_+ : \mathcal{X} \to \mathbb{R}$ *as*

$$\begin{aligned} \gamma(u) &:= f(\tilde{x}_\Omega) + \langle \nabla f(\tilde{x}_\Omega), y^+ - \tilde{x}_\Omega \rangle + h(y^+) + \frac{1}{\lambda}\langle \tilde{x} - y^+, u - y^+ \rangle \qquad \forall u \in \mathcal{X}, \\ \Gamma_+ &:= \frac{A}{A+a}\Gamma + \frac{a}{A+a}\gamma. \end{aligned}$$

*Then, the triple* $(y^+, A_+, \Gamma_+)$ *satisfies*

$$A_+\phi(y^+) \leq \min_{u\in\mathcal{X}}\{A_+\Gamma_+(u) + \frac{1}{2}\|u - x^0\|^2\}, \qquad A_+\Gamma_+ \leq A_+\phi, \tag{28}$$

$$\sqrt{A_+} \geq \sqrt{A} + \frac{1}{2}\sqrt{\lambda}. \tag{29}$$

*Moreover,*

$$x^0 - A_+\nabla\Gamma_+ = x - \frac{a}{\lambda}(\tilde{x} - y^+). \tag{30}$$

*Proof.* We first claim that $\gamma \leq \phi$ and

$$\phi(y^+) \leq \min_{u\in\mathcal{X}}\left\{\gamma(u) + \frac{1}{2\lambda}\|u - \tilde{x}\|^2\right\}. \tag{31}$$

Define the function $g : \mathcal{X} \to \bar{\mathbb{R}}$ as

$$g(u) = f(\tilde{x}_\Omega) + \langle \nabla f(\tilde{x}_\Omega), u - \tilde{x}_\Omega \rangle + h(u) \qquad \forall u \in \mathcal{X},$$

and note that assumption C.3 implies

$$g(u) \leq \phi(u) \leq g(u) + \frac{L}{2}\|u - \tilde{x}_\Omega\|^2 \leq g(u) + \frac{1}{2\lambda}\|u - \tilde{x}_\Omega\|^2 \qquad \forall u \in \mathcal{X}. \tag{32}$$

Then, in view of the definition of $y^+$ and $\lambda$, and the above relation with $u = y^+$, we have

$$\begin{aligned} \min_{u\in\mathcal{X}}\left\{\lambda g(u) + \frac{1}{2}\|u - \tilde{x}\|^2\right\} &= \min_{u\in\mathcal{X}}\left\{\lambda h(u) + \frac{1}{2}\|u - \tilde{x} + \lambda\nabla f(\tilde{x}_\Omega)\|^2\right\} - \frac{\lambda^2}{2}\|\nabla f(\tilde{x}_\Omega)\|^2 + \lambda f(\tilde{x}_\Omega) \\ &= \lambda h(y^+) + \frac{1}{2}\|y^+ - \tilde{x} + \lambda\nabla f(\tilde{x}_\Omega)\|^2 - \frac{\lambda^2}{2}\|\nabla f(\tilde{x}_\Omega)\|^2 + \lambda f(\tilde{x}_\Omega) \\ &\geq \lambda h(y^+) + \frac{1}{2}\|y^+ - \tilde{x}_\Omega + \lambda\nabla f(\tilde{x}_\Omega)\|^2 - \frac{\lambda^2}{2}\|\nabla f(\tilde{x}_\Omega)\|^2 + \lambda f(\tilde{x}_\Omega) \\ &= \lambda\left[g(y^+) + \frac{1}{2\lambda}\|y^+ - \tilde{x}_\Omega\|^2\right] \\ &\geq \lambda\phi(y^+), \end{aligned}$$

where the first inequality is due to the non-expansiveness property of $\Pi_\Omega$. The claim follows from the above relation, the definitions of $\gamma$ and $g$, and Lemma A.2 with $\xi = \tilde{x}$ and noting that in this case $\bar{x} = y^+$.

Using the first inequality of (22), Lemma A.1 and (31), we conclude that

$$\begin{aligned} \min_{u\in\mathcal{X}}\left\{a\gamma(u) + A\Gamma(u) + \frac{1}{2}\|u - x^0\|^2\right\} &\geq \frac{A+a}{\lambda}\min_{u\in\mathcal{X}}\left\{\lambda\gamma(u) + \frac{1}{2}\|u - \tilde{x}\|^2\right\}. \\ &\geq (A+a)\phi(y^+). \end{aligned}$$

Hence, (28) follows from the previous relation, the definitions of $A_+$ and $\Gamma_+$, the second inequality in (22), and the previous claim. Relation (29) can be shown by noting that the definition of $a$ implies

$$a \geq \frac{\lambda}{2} + \sqrt{\lambda A}$$

which, together with the definition of $A_+$, implies that

$$A_+ \geq A + \left(\frac{\lambda}{2} + \sqrt{\lambda A}\right) \geq \left(\sqrt{A} + \frac{1}{2}\sqrt{\lambda}\right)^2.$$

Finally, (30) follows immediately from the definition of $\gamma$, $A_+$ and $\Gamma_+$, and the fact that $x = x^0 - A\nabla\Gamma$. □

# B Proof of Proposition 2.4

First, note that the equivalence between a) and c) of Lemma 2.2 with $\Gamma = \sum_{i=1}^m \alpha_i\gamma_i$ and $\chi = \sum_{i=1}^m \alpha_i\phi(y)$ imply that the hard constraint in problem (16) is equivalent to

$$\left\|\sum_{i=1}^m \alpha_i\nabla\gamma_i + y - x^0\right\|^2 + 2\sum_{i=1}^m \alpha_i[\phi(y) - \gamma_i(y)] \leq \|y - x^0\|^2. \tag{33}$$

Now, assume that b) holds. Then, in view of (17), the point $\alpha(t) := (t\bar{\alpha}_1, \ldots, t\bar{\alpha}_m)$ clearly satisfies (33), and hence is feasible for (16), for every $t > 0$. Hence, for a fixed $x^* \in X^*$, this conclusion implies that

$$\sum_{i=1}^m (t\bar{\alpha}_i)\phi(y) \leq \sum_{i=1}^m (t\bar{\alpha}_i)\gamma_i(x^*) + \frac{1}{2}\|x^* - x^0\|^2 \leq t\left(\sum_{i=1}^m \bar{\alpha}_i\right)\phi(x^*) + \frac{1}{2}\|x^* - x^0\|^2,$$

where the last inequality follows from the assumption that $\gamma_i \leq \phi$ and the non-negativity 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 $\phi(y) - \phi(x^*) \leq 0$, and hence that $y \in X^*$. Also, the objective function of (16) 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 (16) 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 $\alpha^k$ satisfies (33) and the fact that $\gamma_i(y) \leq \phi(y)$, we easily see that (17) holds. □

# C Tables

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 |

Table 2: Number of iterations comparison of the methods on CQP instances.

| Problem | | # of iterations (accuracy less that $10^{-6}$) | | | | | |
|---|---|---|---|---|---|---|---|
| **Instance** | $n$ | **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*(1.1 -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 | | # of iterations (accuracy less that $10^{-6}$) | | | | | |
|---|---|---|---|---|---|---|---|
| **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) |