Comput Optim Appl (2016) 64:31–73
DOI 10.1007/s10589-015-9802-0

An adaptive accelerated first-order method for convex optimization

Renato D. C. Monteiro[1] · **Camilo Ortiz**[1] ·
Benar F. Svaiter[2]

Received: 28 May 2014 / Published online: 11 November 2015

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

✉ Camilo Ortiz
camiort@gatech.edu

Renato D. C. Monteiro
monteiro@isye.gatech.edu

Benar F. Svaiter
benar@impa.br

1 School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA

2 IMPA, Estrada Dona Castorina 110, 22460-320 Rio de Janeiro, Brazil

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 [1,3,5,6,8,10–12,14,15,18]). 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. More specifically, the new accelerated variant chooses the acceleration parameters at every iteration by solving a simple two-variable convex quadratically constrained linear program. Moreover, its iteration cost is comparable to the aforementioned accelerated methods since it only computes a resolvent (or proximal subproblem) and possibly a projection 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, Sect. 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.

In what follows, we refer to convex functions as 0-strongly convex functions. This terminology has the benefit of allowing us to treat both the convex and strongly convex case simultaneously. 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 for some $\mu_h \geq 0$, $h : \mathcal{X} \to \bar{\mathbb{R}}$ is a proper closed μ_h-strongly convex (possibly nonsmooth) 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 ∇f is L-Lipschitz continuous on Ω for some $L > 0$, i.e.,

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

C.4 for some $\mu_f \geq 0$, function f is μ_f-strongly convex on Ω, and hence satisfies

$$f(x') \geq f(x) + \langle \nabla f(x), x' - x \rangle + \frac{\mu_f}{2}\|x' - x\|^2 \quad \forall x, x' \in \Omega;$$

C.5 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.5. 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 Ω defined as

$$\Pi_\Omega(x) = \arg\min \{\|x - u\| : u \in \Omega\} \qquad \forall x \in \mathcal{X}.$$

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

0) Let $\lambda \in (0, 1/(L-\mu_f)]$ and $x^0 \in \mathcal{X}$ be given, and set $\mu := \mu_f + \mu_h$, $A_0 = 0$, $y^0 = x^0$, 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 function $\gamma_k : \mathcal{X} \to \mathbb{R}$ as

$$\begin{aligned} a_k &:= \frac{\lambda(A_{k-1}\mu+1) + \sqrt{\lambda^2(A_{k-1}\mu+1)^2 + 4\lambda(A_{k-1}\mu+1)A_{k-1}}}{2}, \\ \tilde{x}^k &:= \frac{A_{k-1}}{A_{k-1}+a_k}\, y^{k-1} + \frac{a_k}{A_{k-1}+a_k}\, x^{k-1}, \qquad \tilde{x}^k_\Omega = \Pi_\Omega(\tilde{x}^k), \\ y^k &:= \arg\min_{u\in\mathcal{X}} \left\{ p_k(u) + \frac{1}{2\lambda}\|u-\tilde{x}^k\|^2 \right\}, \\ \gamma_k(u) &:= p_k(y^k) + \frac{1}{\lambda}\langle \tilde{x}^k - y^k, u - y^k\rangle + \frac{\mu}{2}\|u-y^k\|^2 \quad \forall u \in \mathcal{X}; \end{aligned}$$

where

$$p_k(u) := f(\tilde{x}^k_\Omega) + \langle \nabla f(\tilde{x}^k_\Omega), u - \tilde{x}^k_\Omega\rangle + \frac{\mu_f}{2}\|u-\tilde{x}^k_\Omega\|^2 + h(u) \quad \forall u \in \mathcal{X};$$

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

$$A' \ge 0,\ a' \ge 0, \quad A' + a' \ge A_{k-1} + a_k, \tag{2}$$

$$(A'+a')\phi(y^k) \le \inf_{u\in\mathcal{X}} \left\{ (A'\Gamma_{k-1} + a'\gamma_k)(u) + \frac{1}{2}\|u-x^0\|^2 \right\}, \tag{3}$$

with the safeguard that $A'_0 = A_0 = 0$ when $k = 1$;
3) compute A_k and x^k as

$$A_k = A'_{k-1} + a'_k, \tag{4}$$

$$x^k = \arg\min_{u\in\mathcal{X}} \left\{ A_k\Gamma_k(u) + \frac{1}{2}\|u-x^0\|^2 \right\} \tag{5}$$

where Γ_k is the function defined recursively as

$$\Gamma_k(u) = \frac{A'_{k-1}}{A'_{k-1}+a'_k}\Gamma_{k-1}(u) + \frac{a'_k}{A'_{k-1}+a'_k}\gamma_k(u) \quad \forall u \in \mathcal{X}; \tag{6}$$

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

Observe that (3), (4) and (6) imply

$$A_k\phi(y^k) \le \inf_{u\in\mathcal{X}} \left\{ A_k\Gamma_k(u) + \frac{1}{2}\|u-x^0\|^2 \right\}. \tag{7}$$

Also, note that the Accelerated Class does not specify how to choose $(A'_{k-1}, a'_k) \in \mathbb{R}\times\mathbb{R}$ satisfying (2) and (3). 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 (33) of Lemma 4, 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 (4) and (7).

The following result not only shows that the pair $(A'_{k-1}, a'_k) = (A_{k-1}, a_k)$ satisfies (2) and (3), but that any instance of the above class has optimal iteration-complexity.

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

(a) $(A'_{k-1}, a'_k) = (A_{k-1}, a_k)$ *satisfies* (2) *and* (3), *and hence the Accelerated Class is well-defined;*
(b) Γ_k *is a quadratic function with Hessian μI which minorizes ϕ and* (7) *holds;*
(c) *there holds*

$$A_k \geq \lambda \max\left\{\frac{k^2}{4}, \left(1 + \frac{\sqrt{\lambda\mu}}{2}\right)^{2(k-1)}\right\};$$

(d) *there holds*

$$\phi(y^k) - \phi^* \leq \frac{d_0^2}{2A_k} \tag{8}$$

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

Proof Statements (a), (b) and (c) follow from Lemma 4 in Appendix 1. Letting x^* denote the projection of x^0 onto X^*, (d) follows from (c) and the fact that (7) 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 1(d) implies that any instance of the Accelerated Class with $\lambda = 1/(L - \mu_f)$ has the optimal iteration-complexity

$$\mathcal{O}\left((L - \mu_f)d_0^2 \min\left\{\frac{1}{k^2}, \left(1 + \frac{1}{2}\sqrt{\frac{\mu}{L - \mu_f}}\right)^{-2(k-1)}\right\}\right).$$

Moreover, based on the fact that (4) and (8) 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

$$(A'_{k-1}, a'_k) \in \arg\max_{A',a'} \{A' + a' : A' \geq 0, a' \geq 0 \text{ and } (A', a') \text{ satisfies (3)}\} \tag{9}$$

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

We will now establish that (3) is equivalent to a convex quadratic constraint, and hence that (9) is a simple two-variable convex quadratically constrained linear program. We first state the following simple technical result whose proof is straightforward.

Lemma 1 *Assume that $q : \mathcal{X} \to \mathbb{R}$ is a quadratic function whose Hessian $Q : \mathcal{X} \to \mathcal{X}$ is a positive definite operator. Then, for any given $u^0 \in \mathcal{X}$ and $\chi \in \mathbb{R}$, the following conditions are equivalent:*

(a) $\min\{q(u) : u \in \mathcal{X}\} \geq \chi$;
(b) $\langle \nabla q(u^0), Q^{-1} \nabla q(u^0) \rangle + 2[\chi - q(u^0)] \leq 0$.

The following result gives the aforementioned characterization for condition (3).

Proposition 2 *For every $k \geq 1$, condition (3) holds if and only if*

$$\begin{aligned} &2A'[\phi(y^k) - \Gamma_{k-1}(x^0)] + 2a'[\phi(y^k) - \gamma_k(x^0)] \\ &\quad + \frac{\|A' \nabla \Gamma_{k-1}(x^0) + a' \nabla \gamma_k(x^0)\|^2}{1 + \mu(A' + a')} \leq 0. \end{aligned} \tag{10}$$

Proof This result follows from Lemma 1 with $q = A'\Gamma_{k-1} + a'\gamma_k + \|u - x_0\|^2/2$, $\chi = (A' + a')\phi(y^k)$ and $u_0 = x_0$, and the observation that the Hessian of q is $[1 + \mu(A' + a')]I$ in this case. □

Note that (10) is a simple convex constraint on the scalars A' and a'. Note also that (10) reduces to a convex quadratic constraint when $\mu = 0$. Hence, the new accelerated method based on (9) chooses the pair (A'_{k-1}, a'_k) as an optimal solution of the simple two-variable convex constrained program

$$\begin{aligned} &\max \ A' + a' \\ &\text{s.t.} \ \ 2A'[\phi(y^k) - \Gamma_{k-1}(x^0)] + 2a'[\phi(y^k) - \gamma_k(x^0)] + \frac{\|A' \nabla \Gamma_{k-1}(x^0) + a' \nabla \gamma_k(x^0)\|^2}{1 + \mu(A' + a')} \leq 0, \\ &\qquad A', a' \geq 0. \end{aligned} \tag{11}$$

The remaining part of this section discusses the case where problem (11) is unbounded. It will be seen that this case implies that $y^k \in X^*$, in which case any instance of the accelerated class may be successfully stopped. The following result, whose proof is given in Appendix 2, considers a slightly more general problem which contains (9) as a special case.

Proposition 3 *Suppose that $x^0, y \in \mathcal{X}$ and $\gamma_1, \ldots, \gamma_m : \mathcal{X} \to \mathbb{R}$ are convex quadratic functions such that, for every $i = 1, \ldots, m$, $\gamma_i \le \phi$ and either $\nabla^2 \gamma_i$ is zero or is positive definite, and consider the problem*

$$\begin{array}{ll} \max & \sum_{i=1}^m \alpha_i \\ \text{s.t.} & \inf_{u \in \mathcal{X}} \left\{ \sum_{i=1}^m \alpha_i \gamma_i(u) + \frac{1}{2}\|u - x^0\|^2 \right\} \ge \sum_{i=1}^m \alpha_i \phi(y), \\ & \alpha_1 \ge 0, \ldots, \alpha_m \ge 0. \end{array} \tag{12}$$

Then, the following conditions are equivalent:

(a) *problem (12) is unbounded;*
(b) *there exist not all zero nonnegative scalars $\bar{\alpha}_1, \ldots, \bar{\alpha}_m$ such that*

$$\sum_{i=1}^m \bar{\alpha}_i \nabla \gamma_i(y) = 0, \qquad \bar{\alpha}_i[\phi(y) - \gamma_i(y)] = 0, \quad i = 1, \ldots, m. \tag{13}$$

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

It follows from Proposition 3 that problem (9) (or equivalently, (11)) is bounded whenever $y^k \notin X^*$. In such case, (11) has an optimal solution since its feasible set is compact.

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 (11) 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];
(iv) 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);
(c) random nonnegative least squares (NNLSs) (see Subsection 3.3);
(d) random ℓ_1 norm regularized least squares (L1LSs) (see Subsection 3.4).

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

$$\left\| y^k - \left(I + \frac{1}{L}\partial h\right)^{-1} \left(y^k - \frac{1}{L}\nabla f(y^k)\right)\right\| \leq 10^{-6} \tag{14}$$

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 (14) 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., $\phi(y^{k-1}) < \phi(y^k)$, 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 kth 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.

All AA variants considered in this benchmark use $\lambda = 0.99/L$. Even though theoretically we can choose $\lambda = 1/L$, our numerical experience have shown us that $\lambda = 0.99/L$ leads to more stable implementations.

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, 5 and 7 report the times and Tables 2, 4, 6 and 8 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 Figs. 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.

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	AA	FISTA-R	FISTA-R	GKR-2	GKR-3
CQP_n5_p2	5	0.1	0.1	0.1	0.1	0.1	9.9	0.5
CQP_n5_p3	5	0.1	0.1	0	0.1	0.1	0.4	0.3
CQP_n5_p4	5	0.1	0.1	0.3	0.2	0.3	16.3	2.4
CQP_n5_p5	5	0.2	0.1	0.2	0.2	0.3	7.5	0.9
CQP_n10_p2	10	0.1	0.1	0.2	0.1	0.1	4.7	0.5
CQP_n10_p3	10	0.1	0.1	0.2	0.1	0.2	0.2	0.5
CQP_n10_p4	10	0.2	0.2	0.3	0.2	0.3	99.5*(1.2−6)	1.3
CQP_n10_p5	10	0.5	0.6	0.4	0.7	0.4	20.1	7.9
CQP_n20_p2	20	0.1	0.1	0.2	0.1	0.1	24.8	0.3
CQP_n20_p3	20	0.1	0.1	0.1	0.1	0.1	1.8	0.4
CQP_n20_p4	20	0.2	0.1	0.5	0.2	0.3	80.1	4.2
CQP_n20_p5	20	0.4	0.5	0.5	0.6	0.4	76.6	12.9
CQP_n50_p2	50	0.1	0.1	0.1	0.1	0.1	44.8	0.3
CQP_n50_p3	50	0.1	0.2	0.4	0.2	0.4	29.5	1.1
CQP_n50_p4	50	0.2	0.2	0.4	0.1	0.2	5.8	3
CQP_n50_p5	50	0.2	0.2	0.3	0.3	0.4	27.4*(1.3−6)	6.1
CQP_n100_p2	100	0.1	0.1	0.2	0.1	0.2	46.2	0.5
CQP_n100_p3	100	0.2	0.2	0.4	0.2	0.4	7.5	1.7
CQP_n100_p4	100	0.2	0.2	0.6	0.2	0.8	121.5	2.5
CQP_n100_p5	100	0.3	0.3	0.3	0.3	0.7	138.2*(2.1−6)	8.5
CQP_n200_p2	200	0.1	0.1	0.2	0.1	0.2	28.2*(1.6−6)	0.8
CQP_n200_p3	200	0.2	0.2	0.4	0.2	0.5	40.3	1.1
CQP_n200_p4	200	0.2	0.2	0.7	0.3	0.7	3.3	2.5
CQP_n200_p5	200	0.3	0.4	0.6	0.4	0.7	47.9	7.4
CQP_n500_p2	500	0.1	0.2	0.4	0.1	0.4	145.4	0.9
CQP_n500_p3	500	0.3	0.3	0.8	0.3	1.2	10.8	2.4
CQP_n500_p4	500	0.3	0.4	1.1	0.6	1.4	14.7	4.3
CQP_n500_p5	500	0.4	0.5	1.5	1	2.1	23.7	12.8
CQP_n700_p2	700	0.2	0.2	0.4	0.2	0.6	119.9	1.1
CQP_n700_p3	700	0.3	0.4	1.1	0.6	1.9	115.9	3.1
CQP_n700_p4	700	0.4	0.4	1.4	1.1	3.1	18.3	7.6
CQP_n700_p5	700	1.4	1.1	1.7	1.1	5.4	19.1	19.6
CQP_n900_p2	900	0.3	0.3	0.8	0.7	0.8	120.0*(2.2−6)	1.7
CQP_n900_p3	900	0.6	0.6	2	0.9	2.9	28.5	4
CQP_n900_p4	900	0.9	1	2.9	1.4	3.8	26.4	9.7
CQP_n900_p5	900	1.6	1.9	4.4	2.8	6.9	31.3	24.9
CQP_n1K_p2	1000	0.4	0.4	1.5	0.6	1.3	84.3*(4.0−6)	4.3
CQP_n1K_p3	1000	0.7	0.7	2.6	1.2	3.1	31.5	5
CQP_n1K_p4	1000	1.1	1.5	3.3	1.6	5.6	20	8.4

Table 1 continued

Problem		Time in seconds (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA-R	GKR-2	GKR-3
CQP_n1K_p5	1000	2.5	1.8	7.3	3	8.7	63.8	63.1
CQP_n2K_p2	2000	1	1.2	4.1	2.8	4.3	112.0*(3.1−6)	3.9
CQP_n2K_p3	2000	1.8	2.9	7.4	3.7	10.3	88.6	11.2
CQP_n2K_p4	2000	3	3.1	14.6	6	17.8	120.8	23.4
CQP_n2K_p5	2000	9.5	9.6	20.4	10.4	23.6	114	68.9
CQP_n5K_p2	5000	8.7	7.1	23.2	14.1	17.9	863.0*(3.2−6)	20.2
CQP_n5K_p3	5000	20.3	19	44.3	29.1	48.5	400.7	42.6
CQP_n5K_p4	5000	23.6	37.8	86.6	46.7	129.1	471.1	161.4
CQP_n5K_p5	5000	47.8	40.3	126.8	91	146.6	597.6	513.9
CQP_n7K_p2	7000	18.9	25.1	37.8	46	70.5	1544.8*(4.4−6)	65.2
CQP_n7K_p3	7000	31.4	37.8	82.2	45.9	128.1	923.9	85.5
CQP_n7K_p4	7000	25.9	29.2	141.7	63.7	167	1082.3*(1.2−6)	264.2
CQP_n7K_p5	7000	64.4	102.6	193.4	115.8	338.5	1307.1	734
CQP_n9K_p2	9000	25.4	40.3	64.6	68.3	58.9	3071.3*(5.5−6)	100.6
CQP_n9K_p3	9000	64.1	54.2	181.1	120	219	2046.5	287.3
CQP_n9K_p4	9000	95	86	344.1	129.2	483.1	1992	669.5
CQP_n9K_p5	9000	128.8	163.3	424.8	292.7	530.7	2165.3	1203.9
CQP_n10K_p2	10000	107.9	81.1	89.2	240.4	84.2	3519.7*(5.5−6)	176.1
CQP_n10K_p3	10000	84	64.3	227	113	251.8	1911.6	250.1
CQP_n10K_p4	10000	173.8	106.9	322.9	236.9	365.4	2089.7	733.5
CQP_n10K_p5	10000	141.8	202.6	596.1	451.3	688.5	2190.9	1784

3.1 Numerical results for random CQPs

This subsection compares the performance of our methods AA, 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

$$\min_{x \in \mathbb{R}^n} \left\{ x^T Q x + b^T x \ : \ l \leq x \leq u \right\}. \tag{15}$$

Letting f and h be defined as

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

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	AA	FISTA-R	FISTA	GKR-2	GKR-3
CQP_n5_p2	5	26	25	105	35	59	913	48
CQP_n5_p3	5	41	38	53	35	53	18	20
CQP_n5_p4	5	89	92	300	104	314	1491	195
CQP_n5_p5	5	176	94	179	152	215	397	75
CQP_n10_p2	10	39	44	152	40	94	315	37
CQP_n10_p3	10	41	46	178	58	140	16	44
CQP_n10_p4	10	130	95	239	146	254	2000*(1.2−6)	114
CQP_n10_p5	10	462	483	391	536	425	579	907
CQP_n20_p2	20	23	32	129	33	65	483	18
CQP_n20_p3	20	47	40	65	46	61	49	36
CQP_n20_p4	20	118	89	417	114	289	1132	407
CQP_n20_p5	20	374	360	504	459	423	1041	1422
CQP_n50_p2	50	27	36	95	33	58	525	29
CQP_n50_p3	50	64	93	305	90	289	334	95
CQP_n50_p4	50	126	132	316	164	353	873	416
CQP_n50_p5	50	183	165	251	247	308	2000*(1.3−6)	398
CQP_n100_p2	100	37	48	129	49	124	618	46
CQP_n100_p3	100	88	93	383	118	338	948	176
CQP_n100_p4	100	116	130	538	186	576	1726	256
CQP_n100_p5	100	232	235	499	321	659	2000*(2.1−6)	861
CQP_n200_p2	200	41	52	160	61	140	2000*(1.6−6)	62
CQP_n200_p3	200	69	80	359	96	326	525	104
CQP_n200_p4	200	134	139	574	183	584	391	219
CQP_n200_p5	200	232	259	541	298	645	628	803
CQP_n500_p2	500	57	74	199	77	151	1926	58
CQP_n500_p3	500	110	134	461	125	406	939	155
CQP_n500_p4	500	164	163	786	241	710	1384	307
CQP_n500_p5	500	293	294	954	377	1017	1374	1063
CQP_n700_p2	700	51	72	182	111	163	1598	63
CQP_n700_p3	700	95	134	474	130	421	1567	178
CQP_n700_p4	700	169	189	712	238	686	801	346
CQP_n700_p5	700	336	346	874	462	929	1168	1008
CQP_n900_p2	900	52	72	196	120	179	2000*(2.2−6)	62
CQP_n900_p3	900	105	131	455	143	398	1148	137
CQP_n900_p4	900	155	168	693	225	641	889	350
CQP_n900_p5	900	320	372	958	431	998	1443	1086
CQP_n1K_p2	1000	58	68	232	83	197	2000*(4.0−6)	76
CQP_n1K_p3	1000	111	125	486	169	433	1206	170

Table 2 continued

Problem		# of iterations (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA	GKR-2	GKR-3
CQP_n1K_p4	1000	164	192	791	258	721	857	379
CQP_n1K_p5	1000	329	377	1226	432	1274	996	1289
CQP_n2K_p2	2000	63	76	226	85	187	2000*(3.1−6)	70
CQP_n2K_p3	2000	133	159	498	179	428	1113	172
CQP_n2K_p4	2000	160	185	851	277	769	1540	414
CQP_n2K_p5	2000	313	341	1100	438	1102	1021	1076
CQP_n5K_p2	5000	83	87	260	103	215	2000*(3.2−6)	79
CQP_n5K_p3	5000	194	199	540	263	453	1240	169
CQP_n5K_p4	5000	290	299	934	388	825	1066	412
CQP_n5K_p5	5000	356	447	1284	522	1265	1391	1161
CQP_n7K_p2	7000	85	94	261	102	219	2000*(4.4−6)	77
CQP_n7K_p3	7000	136	138	555	195	473	1341	176
CQP_n7K_p4	7000	190	226	976	275	871	2000*(1.2−6)	388
CQP_n7K_p5	7000	369	539	1404	611	1385	1392	1264
CQP_n9K_p2	9000	85	93	259	221	225	2000*(5.5−6)	78
CQP_n9K_p3	9000	152	225	538	204	463	1378	176
CQP_n9K_p4	9000	299	343	1051	337	916	1203	523
CQP_n9K_p5	9000	397	537	1378	673	1379	1333	1259
CQP_n10K_p2	10000	204	210	261	225	225	2000*(5.5−6)	80
CQP_n10K_p3	10000	183	198	565	264	487	1422	186
CQP_n10K_p4	10000	236	258	1022	393	918	1338	408
CQP_n10K_p5	10000	459	541	1509	727	1473	1318	1316

where $B = \{x \in \mathbb{R}^n : l \le x \le u\}$, we can easily see that (15) 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, AA-R1 and AA-R2 outperform the other methods on most of the random sparse CQP instances. From Figs. 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, 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.

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	AA	FISTA-R	FISTA	GKR-2	GKR-3
BIQ_n101_m5252	101	0.9	1.5	3.9	1.4	3.2	360.4*(1.4−6)	56.7
BIQ_n121_m7502	121	1.1	1.1	2.2	1.7	6.3	695.4*(1.4−6)	44.6
BIQ_n151_m11627	151	1.3	1.5	4.8	1.8	4.5	939.9*(1.4−6)	64.5
BIQ_n201_m20502	201	2.7	2.1	4.4	3.1	6.5	707.5*(1.4−6)	157.9
BIQ_n251_m31877	251	2.7	3.1	10.2	4	15.9	1073.8*(1.4−6)	144.8
BIQ_n51_m1377	51	1	1	2.2	1.1	2.8	285.3*(1.4−6)	44.4
BIQ_n501_m126252	501	16.3	13.9	20.7	17.7	52.3	6386.8*(1.4−6)	706
BIQ_n21_m252	21	1	0.7	1.9	1	2.5	402.9*(1.4−6)	46.8
BIQ_n41_m902	41	1	0.8	2	1	2.7	481.6*(1.4−6)	47.8
BIQ_n71_m2627	71	0.8	1.2	3.6	1.2	3.1	548.0*(1.4−6)	52.8
BIQ_n81_m3402	81	1	1.1	3.2	1.5	4.2	266.3*(1.4−6)	40.4
BIQ_n61_m1952	61	0.9	1.1	3	1.2	3.7	537.2*(1.4−6)	42.1
BIQ_n31_m527	31	1	0.8	2.2	1.2	4	248.6*(1.4−6)	47.6
BIQ_n91_m4277	91	1	1.2	4.3	1.5	3.2	279.0*(1.4−6)	48.7
FAP_n52_m1378	52	0.4	0.5	5.2	0.4	0.5	5.6	3.7
FAP_n61_m1866	61	0.4	0.5	5.7	0.5	0.5	6.5	3.5
FAP_n65_m2145	65	0.5	0.6	6	0.5	0.6	11.2	3.9
FAP_n81_m3321	81	0.5	0.5	8.9	0.6	0.7	5.4	4.7
FAP_n84_m3570	84	0.5	0.6	9.5	0.5	0.6	5.9	4.4
FAP_n93_m4371	93	0.5	0.7	10.6	0.7	0.8	15.3	4.3
FAP_n98_m4851	98	0.5	0.7	6.5	0.6	0.8	8	4.4
FAP_n120_m7260	120	0.7	0.9	11.4	0.8	1	17	5.6

Table 3 continued

Problem		Time in seconds (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA	GKR-2	GKR-3
FAP_n174_m15225	174	0.9	1.2	27.9	1.1	2.3	21.1	8.7
FAP_n183_m14479	183	1	1.4	28	1.5	1.9	38.9	10.4
FAP_n252_m24292	252	2.1	2.6	37.6	2.3	3.2	39.8	20.9
FAP_n369_m26462	369	3.7	4.3	137.8	4	6.1	78.6	44.8
FAP_n2118_m322924	2118	354.3	466.9	19603.6	550.5	1169	9788	5722.2
FAP_n4110_m1154467	4110	2518.5	3039.2	137651.3	3187	5151.6	66187	29440.6
QAP_n676_m229877	676	212.3	263.3	834.3	247.6	543.1	16715.0*(2.7−6)	3566.5
QAP_n144_m10672	144	6.8	7.5	44.9	5.6	44.2*(1.1−6)	807.3*(2.5−6)	544.1
QAP_n225_m25783	225	25.7	13	102.8	14	18.6	689.2	186.8
QAP_n324_m53161	324	34	36.1	131.9	27.6	98.5	3220.6*(1.9−6)	3124.4
QAP_n400_m80828	400	58.5	53.4	316.2	52.6	440.8	4832.9*(2.4−6)	5571.4*(1.3−6)
QAP_n484_m118127	484	95.3	113.9	421.3	87.1	434.4	5387	1133.6
QAP_n625_m196598	625	282.9	210.7	825.3	159.4	1420.3	14179.8*(2.8−6)	2809
QAP_n196_m19619	196	12	10.7	63.8	11.1	24.1	1150.6*(3.3−6)	1299.8*(2.3−6)
QAP_n256_m33302	256	21.5	21.7	105.1	15.7	24.8	2265.5*(1.7−6)	3289.2*(1.2−6)
QAP_n289_m42362	289	28	25.9	87	22.9	53.7	2496.2*(4.0−6)	1727.5
QAP_n441_m98152	441	68.2	76.3	371.7	59	339.9	4207.4	12851.7*(1.8−6)
QAP_n729_m267217	729	243	270.5	1251.9	236.9	376.6	20852.3*(1.8−6)	20097.9
QAP_n784_m308936	784	333	328.7	1196.3	295.4	1384.4	24197.1*(2.0−6)	23049.4
QAP_n900_m406843	900	480.7	516.7	2151.4	464.1	1678.3	23520.9	6558.9
QAP_n1225_m752813	1225	1258	1170.2	4423	1121	2873.9	99556.0*(1.3−6)	15090.9
QAP_n1600_m1283258	1600	2512.7	2811.8	9551	3014.3	3610.1	207778.0*(1.5−6)	171500.0*(1.1−6)
RAND_n1K_m100K_p3	1000	568	532.4	3840.7*(9.7−6)	1683.9	2074.3	57775.2	43090

Table 3 continued

Problem		Time in seconds (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA	GKR-2	GKR-3
RAND_n1K_m150K_p3	1000	768.4	777.7	3604.1*(1.4−5)	1809.5	2810.2	63396.8*(1.2−6)	61719.5
RAND_n300_m10K_p4	300	13	12.3	120.7	16.1	31.9	810.3	490.3
RAND_n300_m20K_p3	300	43.5	46.7	199.7*(1.2−5)	68.1	139.6	2918.1*(1.1−6)	2987.4
RAND_n300_m25K_p3	300	41.6	48.9	198.6*(1.1−5)	143.3	157.7	2981.1*(1.5−6)	2682.2*(1.0−6)
RAND_n400_m15K_p4	400	15.7	27.2	344.3	27.7	96.8	851.8	794.4
RAND_n400_m30K_p3	400	82.3	94.4	380.9*(9.4−6)	168	362.2	4951.4	8962.2
RAND_n400_m40K_p3	400	80.1	84.7	380.4*(7.7−6)	178	273.2	9705.8*(1.5−6)	5465.2*(1.3−6)
RAND_n500_m20K_p4	500	38.4	58.3	581.4	47.2	117.9	2274.4	2292.9
RAND_n500_m30K_p3	500	100.1	104.1	632.6*(9.5−6)	141.3	306.4	6594.5	5263
RAND_n500_m40K_p3	500	113.7	119.6	612.0*(5.2−6)	247.7	576.4	8313	7979.3
RAND_n500_m50K_p3	500	192.9	148.3	634.0*(1.3−5)	647.7	411.9	10854.6*(1.1−6)	10096.9*(1.3−6)
RAND_n600_m20K_p4	600	36.8	37.6	281.2	73.6	120.8	2175.8	2349.3
RAND_n600_m40K_p3	600	233.1	199.5	949.9*(8.6−6)	321.3	816.2	12218	12037.4
RAND_n600_m50K_p3	600	168.9	166.9	981.8*(9.6−6)	445	571.8	13565.8	11012.5
RAND_n600_m60K_p3	600	218.8	265	992.8*(1.4−5)	411.6	792.8	17428.4*(1.6−6)	15356.9*(1.2−6)
RAND_n700_m50K_p3	700	236	239.5	1497.6*(7.8−6)	402.1	1124.4	20615.5	17822.2
RAND_n700_m70K_p3	700	258.8	241	1404.4*(1.1−5)	592.2	1383	19660.1	16415.1
RAND_n700_m90K_p3	700	352.7	317.7	1486.5*(1.5−5)	948.2	1075.1	21967.6*(1.3−6)	22553.8*(1.1−6)
RAND_n800_m100K_p3	800	422.3	641.1	2097.2*(1.7−5)	1024	1514.7	30362.9	25120.1
RAND_n800_m110K_p3	800	403	405.1	2126.5*(1.5−5)	1152.5	2206.4	30609.5*(1.2−6)	30426.3*(1.1−6)
RAND_n800_m70K_p3	800	289.9	310.8	1970.2*(9.3−6)	728.4	1803	45568.8	21851.4
RAND_n900_m100K_p3	900	589.3	587.5	2809.7*(8.5−6)	1278.9	1666.9	48650.9	39438.5
RAND_n900_m140K_p3	900	552.5	577	2819.6*(1.5−5)	1465.5	2767.4	46800.9*(1.3−6)	39116.8

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	AA	FISTA-R	FISTA	GKR-2	GKR-3
BIQ_n101_m5252	101	67	72	271	90	251	2000*(1.4−6)	285
BIQ_n121_m7502	121	65	67	136	90	251	2000*(1.4−6)	285
BIQ_n151_m11627	151	68	73	243	90	251	2000*(1.4−6)	285
BIQ_n201_m20502	201	79	67	173	90	251	2000*(1.4−6)	285
BIQ_n251_m31877	251	64	66	257	90	251	2000*(1.4−6)	285
BIQ_n51_m1377	51	79	67	173	90	251	2000*(1.4−6)	285
BIQ_n501_m126252	501	65	67	105	90	251	2000*(1.4−6)	285
BIQ_n21_m252	21	79	67	173	90	251	2000*(1.4−6)	285
BIQ_n41_m902	41	79	67	173	90	251	2000*(1.4−6)	285
BIQ_n71_m2627	71	65	68	285	90	251	2000*(1.4−6)	285
BIQ_n81_m3402	81	68	73	243	90	251	2000*(1.4−6)	285
BIQ_n61_m1952	61	68	73	243	90	251	2000*(1.4−6)	285
BIQ_n31_m527	31	79	67	173	90	251	2000*(1.4−6)	285
BIQ_n91_m4277	91	65	68	285	90	251	2000*(1.4−6)	285
FAP_n52_m1378	52	25	27	389	26	37	29	18
FAP_n61_m1866	61	21	25	389	26	38	32	18
FAP_n65_m2145	65	25	27	398	26	38	46	18
FAP_n81_m3321	81	23	23	479	26	37	23	19
FAP_n84_m3570	84	21	26	530	26	38	22	19
FAP_n93_m4371	93	20	29	558	26	39	43	18
FAP_n98_m4851	98	21	27	342	26	43	36	18
FAP_n120_m7260	120	24	28	488	26	43	38	18

Table 4 continued

Problem		# of iterations (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA	GKR-2	GKR-3
FAP_n174_m15225	174	23	29	773	26	46	43	18
FAP_n183_m14479	183	21	26	648	26	44	41	19
FAP_n252_m24292	252	26	31	515	26	45	35	20
FAP_n369_m26462	369	24	27	883	26	47	40	20
FAP_n2118_m322924	2118	25	30	1405	27	54	43	22
FAP_n4110_m1154467	4110	26	30	1558	29	54	47	22
QAP_n676_m229877	676	283	297	1091	323	885	2000*(2.7−6)	342
QAP_n144_m10672	144	202	223	1424	233	2000*(1.1−6)	2000*(2.5−6)	1688
QAP_n225_m25783	225	375	228	1630	251	385	968	227
QAP_n324_m53161	324	240	262	908	275	1001	2000*(1.9−6)	1939
QAP_n400_m80828	400	263	258	1533	286	1744	2000*(2.4−6)	2000*(1.3−6)
QAP_n484_m118127	484	271	290	1307	299	1763	1462	284
QAP_n625_m196598	625	274	304	1384	316	1740	2000*(2.8−6)	387
QAP_n196_m19619	196	207	220	1307	244	674	2000*(3.3−6)	2000*(2.3−6)
QAP_n256_m33302	256	236	253	1167	259	382	2000*(1.7−6)	2000*(1.2−6)
QAP_n289_m42362	289	231	248	870	266	609	2000*(4.0−6)	1357
QAP_n441_m98152	441	254	269	1270	293	1611	1455	2000*(1.8−6)
QAP_n729_m267217	729	283	305	1428	327	527	2000*(1.8−6)	1646
QAP_n784_m308936	784	279	314	1116	333	1626	2000*(2.0−6)	1587
QAP_n900_m406843	900	302	323	1474	343	1055	1393	330
QAP_n1225_m752813	1225	321	333	1224	368	931	2000*(1.3−6)	303
QAP_n1600_m1283258	1600	333	346	1297	388	583	2000*(1.5−6)	2000*(1.1−6)
RAND_n1K_m100K_p3	1000	280	282	2000*(9.7−6)	840	1070	1657	1572

Table 4 continued

Problem		# of iterations (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA	GKR-2	GKR-3
RAND_n1K_m150K_p3	1000	368	370	2000*(1.4−5)	895	1351	2000*(1.2−6)	1972
RAND_n300_m10K_p4	300	102	118	1286	167	337	387	323
RAND_n300_m20K_p3	300	405	407	2000*(1.2−5)	690	1411	2000*(1.1−6)	2000
RAND_n300_m25K_p3	300	409	411	2000*(1.1−5)	1244	1579	2000*(1.5−6)	2000*(1.0−6)
RAND_n400_m15K_p4	400	86	151	1855	160	319	315	304
RAND_n400_m30K_p3	400	440	442	2000*(9.4−6)	939	1176	1841	1866
RAND_n400_m40K_p3	400	405	407	2000*(7.7−6)	1090	1515	2000*(1.5−6)	2000*(1.3−6)
RAND_n500_m20K_p4	500	109	149	1834	153	372	485	459
RAND_n500_m30K_p3	500	313	262	2000*(9.5−6)	449	944	1379	1104
RAND_n500_m40K_p3	500	353	355	2000*(5.2−6)	810	1184	1747	1618
RAND_n500_m50K_p3	500	442	423	2000*(1.3−5)	1179	1297	2000*(1.1−6)	2000*(1.3−6)
RAND_n600_m20K_p4	600	65	74	551	150	243	274	187
RAND_n600_m40K_p3	600	366	368	2000*(8.6−6)	522	1011	1508	1542
RAND_n600_m50K_p3	600	323	325	2000*(9.6−6)	533	1150	1755	1476
RAND_n600_m60K_p3	600	401	403	2000*(1.4−5)	851	1327	2000*(1.6−6)	2000*(1.2−6)
RAND_n700_m50K_p3	700	327	330	2000*(7.8−6)	558	989	1830	1570
RAND_n700_m70K_p3	700	356	358	2000*(1.1−5)	788	1187	1843	1521
RAND_n700_m90K_p3	700	392	394	2000*(1.5−5)	1420	1420	2000*(1.3−6)	2000*(1.1−6)
RAND_n800_m100K_p3	800	383	385	2000*(1.7−5)	971	1285	1995	1688
RAND_n800_m110K_p3	800	378	380	2000*(1.5−5)	1182	1378	2000*(1.2−6)	2000*(1.1−6)
RAND_n800_m70K_p3	800	274	276	2000*(9.3−6)	739	1073	1911	1453
RAND_n900_m100K_p3	900	381	383	2000*(8.5−6)	849	1190	1885	1831
RAND_n900_m140K_p3	900	368	379	2000*(1.5−5)	893	1429	2000*(1.3−6)	1712

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	AA	FISTA-R	FISTA	GKR-2	GKR-3
NNLS_n200_m1K	200	1	0.6	2.4*(8.8−6)	0.7	2.2*(1.2−5)	23.3*(1.9−6)	16.8*(1.7−5)
NNLS_n200_m10K	200	0.8	0.9	5.4*(6.4−5)	1.4	7.2*(2.3−5)	25.7*(3.8−5)	21.9*(3.3−5)
NNLS_n200_m2K	200	0.2	0.2	1.7	0.3	1.9	14.7	14.7
NNLS_n200_m20K	200	14.7*(2.2−6)	0.8	9.9*(7.1−5)	1.1	6.2	46.4*(3.2−6)	59.0*(4.8−6)
NNLS_n200_m400	200	0.2	0.2	2	0.3	2.8	31.1*(6.7−6)	12
NNLS_n200_m4K	200	1.7	1.4	3.2*(1.3−4)	1.8	3.0*(1.4−4)	33.0*(1.1−5)	40.5*(3.3−4)
NNLS_n200_m40K	200	12.8	10.7	28.8*(9.7−5)	11.3	52.3*(5.1−6)	100.9*(1.3−5)	87.5*(3.2−4)
NNLS_n200_m600	200	0.9	0.8	2.2*(4.2−5)	0.8	1.3*(1.7−5)	17.2*(6.2−6)	21.1*(5.6−5)
NNLS_n200_m6K	200	0.5	0.5	3.6*(1.0−4)	0.6	3.3*(1.7−5)	19.6*(1.3−5)	20.8*(5.0−6)
NNLS_n200_m800	200	0.8	0.8	1.6*(4.5−5)	1	2.4*(7.7−5)	20.1*(8.0−6)	23.7*(5.4−5)
NNLS_n200_m8K	200	1.8	5.4*(1.1−6)	4.6*(2.1−4)	3.2	5.0*(3.1−4)	36.5*(1.6−5)	31.7*(4.1−4)
NNLS_n400_m1K	400	0.4	0.4	2.7*(6.3−6)	0.5	2.7*(2.1−6)	26.4	23.0*(7.0−6)
NNLS_n400_m10K	400	3.9	2.6	9.3*(1.7−4)	4.8	13.7*(3.0−4)	35.8*(1.1−4)	59.7*(1.9−4)
NNLS_n400_m2K	400	1.4	1.5	3.3*(1.3−4)	1.5	4.0*(1.4−4)	41.2*(6.5−6)	41.0*(1.1−4)
NNLS_n400_m20K	400	8.5	7.7	19.1*(1.5−4)	7.6	29.3*(3.7−4)	118.3*(1.5−5)	105.6*(3.0−4)
NNLS_n400_m4K	400	0.9	0.8	4.1*(4.3−5)	0.8	7.1*(7.9−6)	44.2*(1.9−6)	44.3*(5.5−6)
NNLS_n400_m40K	400	61.7*(1.1−6)	11.9	38.9*(2.0−4)	23.2	55.7*(7.9−5)	284.7*(2.0−5)	101.7*(1.9−4)
NNLS_n400_m6K	400	8.7*(1.0−6)	1.2	6.0*(6.2−5)	1.9	6.6*(4.5−5)	24.7*(2.0−6)	35.2*(4.1−5)
NNLS_n400_m800	400	0.3	0.3	2.9*(2.8−5)	0.5	4.1*(1.2−6)	30.8	31.4*(4.6−6)
NNLS_n400_m8K	400	2.8	4.6	6.4*(2.2−4)	8.4*(1.1−6)	12.3*(2.6−4)	43.2*(1.2−4)	60.1*(2.9−4)
NNLS_n600_m10K	600	3.1	3.6	14.8*(1.2−4)	17.2*(1.9−6)	17.6*(8.4−5)	47.9*(6.8−6)	45.5*(5.6−5)
NNLS_n600_m2K	600	1.1	1.1	3.5*(3.6−5)	1.8	3.3*(6.1−6)	32.2*(2.6−6)	28.7*(5.0−5)

Table 5 continued

Problem		Time in seconds (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA	GKR-2	GKR-3
NNLS_n600_m20K	600	28.6*(2.2−6)	3.2	30.4*(4.3−6)	40.8*(1.5−6)	45.6	122.8*(1.8−5)	93.0*(4.9−6)
NNLS_n600_m4K	600	1.8	1.2	5.3*(4.6−5)	1.3	5.4*(6.6−6)	41.0*(2.0−6)	31.0*(3.5−5)
NNLS_n600_m40K	600	79.5*(4.6−6)	65.1*(2.6−6)	66.8*(1.4−4)	10.5	88.2*(5.8−5)	207.4*(4.5−5)	263.6*(3.3−5)
NNLS_n600_m6K	600	1.2	1.1	7.4*(3.0−4)	1.7	13.5*(7.5−5)	39.5*(4.6−5)	55.5*(6.0−5)
NNLS_n600_m8K	600	1.6	1.8	8.5*(9.5−5)	2.5	18.0*(3.0−5)	44.8*(2.8−5)	51.2*(1.8−5)
NNLS_n800_m10K	800	2.4	3.2	15.1*(9.1−5)	3.5	33.9*(2.2−5)	105.8*(2.4−6)	101.1*(1.3−5)
NNLS_n800_m2K	800	1.2	1.2	4.0*(6.3−5)	1.7	4.3*(1.2−4)	29.1*(4.2−5)	28.5*(7.8−5)
NNLS_n800_m20K	800	9.5	6.8	32.5*(1.8−4)	12.9	38.8*(6.6−5)	104.8*(1.8−5)	91.9*(2.1−4)
NNLS_n800_m4K	800	3.1	2.3	6.0*(9.5−5)	5.9	12.2*(8.3−5)	29.0*(5.3−6)	63.1*(7.1−5)
NNLS_n800_m40K	800	70	33.1	79.1*(2.2−4)	72.0*(5.6−6)	93.7*(1.4−4)	360.0*(3.4−5)	222.6*(3.9−4)
NNLS_n800_m6K	800	1.5	1.7	8.7*(1.9−5)	2.5	11.2*(1.2−5)	49.1*(4.0−6)	44.3*(1.4−5)
NNLS_n800_m8K	800	2.7	5.4	14.4*(9.3−6)	22.4*(1.1−6)	29.7*(3.1−6)	67.1	91.9*(1.3−5)
NNLS_n1K_m10K	1000	23.3*(1.8−6)	3.5	24.2*(4.1−5)	5.1	31.2*(4.7−5)	132.0*(4.0−5)	97.1*(3.9−5)
NNLS_n1K_m2K	1000	1.7	1.2	4.7*(8.1−5)	2.1	4.3*(2.0−4)	45.1*(1.8−5)	26.2*(1.3−4)
NNLS_n1K_m20K	1000	11.4	11.2	43.7*(1.9−4)	50.0*(1.1−6)	59.7*(7.5−5)	244.5*(6.3−5)	166.8*(7.1−5)
NNLS_n1K_m4K	1000	1.9	1.6	7.4*(4.7−5)	2.2	7.6*(2.5−5)	59.4*(4.7−6)	35.5*(3.5−5)
NNLS_n1K_m40K	1000	103.6*(3.5−6)	95.3*(1.1−6)	84.5*(5.2−4)	137.5*(1.8−6)	143.5*(3.1−4)	331.9*(3.1−5)	276.9*(4.0−4)
NNLS_n1K_m6K	1000	1.9	2.8	10.0*(5.7−5)	2.1	17.5*(1.1−5)	56.6*(4.6−6)	50.9*(5.3−6)
NNLS_n1K_m8K	1000	7.2	5.2	16.2*(4.2−5)	6.5	18.7*(5.2−5)	82.3*(1.3−5)	74.2*(1.4−4)
NNLS_n2K_m10K	2000	14.2	11.5	45.9*(1.1−4)	18.1	71.8*(5.1−5)	139.7*(7.8−6)	133.3*(1.0−4)
NNLS_n2K_m20K	2000	89.2*(1.2−6)	24.7	83.4*(3.4−5)	54.7	108.4*(2.3−5)	371.2*(2.3−5)	236.4*(2.4−5)
NNLS_n2K_m4K	2000	7.5	6.9	14.7*(6.7−5)	6.4	23.9*(3.1−5)	79.2*(2.4−5)	91.5*(2.6−5)
NNLS_n2K_m40K	2000	209.7*(5.5−6)	29.7	196.8*(2.7−5)	60	221.7*(2.2−5)	573.2*(1.2−5)	800.8*(1.2−5)

Table 5 continued

Problem		Time in seconds (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA	GKR-2	GKR-3
NNLS_n2K_m6K	2000	19.7	12.8	28.5*(8.8−5)	19.8	27.2*(8.4−5)	124.6*(3.7−5)	131.4*(6.2−5)
NNLS_n2K_m8K	2000	3.6	4.9	28.1*(4.2−5)	37.3*(1.6−6)	31.7*(1.3−5)	127.8*(8.2−6)	82.6*(1.1−5)
NNLS_n4K_m10K	4000	94.8*(1.6−6)	23.7	75.0*(6.9−5)	31.5	88.1*(4.9−5)	468.4*(4.1−5)	351.1*(3.6−5)
NNLS_n4K_m20K	4000	53.6	157.3*(1.2−6)	157.7*(1.5−4)	185.9*(1.3−6)	180.8*(8.1−5)	561.3*(4.7−5)	381.3*(5.8−5)
NNLS_n4K_m40K	4000	472.0*(1.6−6)	452.2*(1.6−6)	476.1*(8.8−5)	533.3*(2.9−6)	435.9*(8.3−5)	1558.5*(1.3−5)	1221.4*(1.4−4)
NNLS_n4K_m8K	4000	21.9	16.5	58.4*(1.0−4)	28.9	67.3*(8.8−5)	209.8*(2.1−5)	145.3*(3.8−5)
NNLS_n6K_m20K	6000	311.9*(2.5−6)	79.7	219.0*(4.0−5)	267.9*(1.2−6)	370.5*(7.6−5)	1095.7*(9.9−6)	1064.1*(6.5−5)
NNLS_n6K_m40K	6000	643.4*(3.0−6)	222	507.2*(2.1−4)	620.1*(1.2−6)	603.6*(1.2−4)	2254.7*(5.5−5)	2521.3*(9.2−5)
NNLS_n8K_m20K	8000	174.5	100.1	344.0*(1.6−4)	198.9	356.7*(6.6−5)	1477.6*(3.7−5)	1106.5*(7.3−5)
NNLS_n8K_m40K	8000	109.6	592.2*(2.3−6)	632.5*(1.3−5)	637.7*(1.4−6)	1068.5*(1.0−5)	3209.2*(6.2−6)	2556.4*(2.0−5)
NNLS_n10K_m20K	10000	469.2*(2.1−6)	162.2	360.8*(1.8−4)	478.3*(3.0−6)	530.7*(1.1−4)	1623.3*(5.9−5)	1410.5*(8.2−5)
NNLS_n10K_m40K	10000	708.7*(2.6−6)	712.7*(2.4−6)	864.6*(1.3−4)	785.6*(2.8−6)	1039.4*(7.0−5)	3231.6*(2.6−5)	2604.1*(1.3−4)
NNLS_n20K_m40K	20000	1952.1*(1.2−6)	2194.0*(1.8−6)	1803.4*(1.3−4)	2044.2*(3.4−6)	1792.2*(7.5−5)	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	AA	FISTA-R	FISTA	GKR-2	GKR-3
NNLS_n200_m1K	200	836	447	2000*(8.8−6)	689	2000*(1.2−5)	2000*(1.9−6)	2000*(1.7−5)
NNLS_n200_m10K	200	325	330	2000*(6.4−5)	553	2000*(2.3−5)	2000*(3.8−5)	2000*(3.3−5)
NNLS_n200_m2K	200	132	141	1264	199	1240	1046	885
NNLS_n200_m20K	200	2000*(2.2−6)	138	2000*(7.1−5)	181	1246	2000*(3.2−6)	2000*(4.8−6)
NNLS_n200_m400	200	168	171	1783	251	1866	2000*(6.7−6)	1360
NNLS_n200_m4K	200	979	758	2000*(1.3−4)	1147	2000*(1.4−4)	2000*(1.1−5)	2000*(3.3−4)
NNLS_n200_m40K	200	936	738	2000*(9.7−5)	824	2000*(5.1−6)	2000*(1.3−5)	2000*(3.2−4)
NNLS_n200_m600	200	809	641	2000*(4.2−5)	800	2000*(1.7−5)	2000*(6.2−6)	2000*(5.6−5)
NNLS_n200_m6K	200	230	246	2000*(1.0−4)	323	2000*(1.7−5)	2000*(1.3−5)	2000*(5.0−6)
NNLS_n200_m800	200	623	627	2000*(4.5−5)	808	2000*(7.7−5)	2000*(8.0−6)	2000*(5.4−5)
NNLS_n200_m8K	200	720	2000*(1.1−6)	2000*(2.1−4)	1320	2000*(3.1−4)	2000*(1.6−5)	2000*(4.1−4)
NNLS_n400_m1K	400	291	275	2000*(6.3−6)	370	2000*(2.1−6)	1702	2000*(7.0−6)
NNLS_n400_m10K	400	647	669	2000*(1.7−4)	943	2000*(3.0−4)	2000*(1.1−4)	2000*(1.9−4)
NNLS_n400_m2K	400	860	865	2000*(1.3−4)	1049	2000*(1.4−4)	2000*(6.5−6)	2000*(1.1−4)
NNLS_n400_m20K	400	775	775	2000*(1.5−4)	866	2000*(3.7−4)	2000*(1.5−5)	2000*(3.0−4)
NNLS_n400_m4K	400	344	363	2000*(4.3−5)	404	2000*(7.9−6)	2000*(1.9−6)	2000*(5.5−6)
NNLS_n400_m40K	400	2000*(1.1−6)	391	2000*(2.0−4)	556	2000*(7.9−5)	2000*(2.0−5)	2000*(1.9−4)
NNLS_n400_m6K	400	2000*(1.0−6)	387	2000*(6.2−5)	673	2000*(4.5−5)	2000*(2.0−6)	2000*(4.1−5)
NNLS_n400_m800	400	224	217	2000*(2.8−5)	321	2000*(1.2−6)	1908	2000*(4.6−6)
NNLS_n400_m8K	400	744	706	2000*(2.2−4)	2000*(1.1−6)	2000*(2.6−4)	2000*(1.2−4)	2000*(2.9−4)
NNLS_n600_m10K	600	432	444	2000*(1.2−4)	2000*(1.9−6)	2000*(8.4−5)	2000*(6.8−6)	2000*(5.6−5)
NNLS_n600_m2K	600	612	555	2000*(3.6−5)	718	2000*(6.1−6)	2000*(2.6−6)	2000*(5.0−5)

Table 6 continued

Problem		# of iterations (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA	GKR-2	GKR-3
NNLS_n600_m20K	600	2000*(2.2−6)	194	2000*(4.3−6)	2000*(1.5−6)	1855	2000*(1.8−5)	2000*(4.9−6)
NNLS_n600_m4K	600	379	387	2000*(4.6−5)	466	2000*(6.6−6)	2000*(2.0−6)	2000*(3.5−5)
NNLS_n600_m40K	600	2000*(4.6−6)	2000*(2.6−6)	2000*(1.4−4)	392	2000*(5.8−5)	2000*(4.5−5)	2000*(3.3−5)
NNLS_n600_m6K	600	304	308	2000*(3.0−4)	461	2000*(7.5−5)	2000*(4.6−5)	2000*(6.0−5)
NNLS_n600_m8K	600	288	336	2000*(9.5−5)	486	2000*(3.0−5)	2000*(2.8−5)	2000*(1.8−5)
NNLS_n800_m10K	800	250	260	2000*(9.1−5)	395	2000*(2.2−5)	2000*(2.4−6)	2000*(1.3−5)
NNLS_n800_m2K	800	572	577	2000*(6.3−5)	830	2000*(1.2−4)	2000*(4.2−5)	2000*(7.8−5)
NNLS_n800_m20K	800	518	436	2000*(1.8−4)	626	2000*(6.6−5)	2000*(1.8−5)	2000*(2.1−4)
NNLS_n800_m4K	800	769	774	2000*(9.5−5)	1133	2000*(8.3−5)	2000*(5.3−6)	2000*(7.1−5)
NNLS_n800_m40K	800	1431	768	2000*(2.2−4)	2000*(5.6−6)	2000*(1.4−4)	2000*(3.4−5)	2000*(3.9−4)
NNLS_n800_m6K	800	348	349	2000*(1.9−5)	499	2000*(1.2−5)	2000*(4.0−6)	2000*(1.4−5)
NNLS_n800_m8K	800	417	444	2000*(9.3−6)	2000*(1.1−6)	2000*(3.1−6)	1760	2000*(1.3−5)
NNLS_n1K_m10K	1000	2000*(1.8−6)	352	2000*(4.1−5)	372	2000*(4.7−5)	2000*(4.0−5)	2000*(3.9−5)
NNLS_n1K_m2K	1000	749	500	2000*(8.1−5)	958	2000*(2.0−4)	2000*(1.8−5)	2000*(1.3−4)
NNLS_n1K_m20K	1000	411	433	2000*(1.9−4)	2000*(1.1−6)	2000*(7.5−5)	2000*(6.3−5)	2000*(7.1−5)
NNLS_n1K_m4K	1000	345	320	2000*(4.7−5)	533	2000*(2.5−5)	2000*(4.7−6)	2000*(3.5−5)
NNLS_n1K_m40K	1000	2000*(3.5−6)	2000*(1.1−6)	2000*(5.2−4)	2000*(1.8−6)	2000*(3.1−4)	2000*(3.1−5)	2000*(4.0−4)
NNLS_n1K_m6K	1000	263	278	2000*(5.7−5)	356	2000*(1.1−5)	2000*(4.6−6)	2000*(5.3−6)
NNLS_n1K_m8K	1000	718	547	2000*(4.2−5)	870	2000*(5.2−5)	2000*(1.3−5)	2000*(1.4−4)
NNLS_n2K_m10K	2000	705	580	2000*(1.1−4)	889	2000*(5.1−5)	2000*(7.8−6)	2000*(1.0−4)
NNLS_n2K_m20K	2000	2000*(1.2−6)	414	2000*(3.4−5)	521	2000*(2.3−5)	2000*(2.3−5)	2000*(2.4−5)
NNLS_n2K_m4K	2000	804	715	2000*(6.7−5)	897	2000*(3.1−5)	2000*(2.4−5)	2000*(2.6−5)
NNLS_n2K_m40K	2000	2000*(5.5−6)	286	2000*(2.7−5)	449	2000*(2.2−5)	2000*(1.2−5)	2000*(1.2−5)

Table 6 continued

Problem		# of iterations (accuracy less that 10^{-6})						
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA	GKR-2	GKR-3
NNLS_n2K_m6K	2000	828	850	2000*(8.8−5)	1193	2000*(8.4−5)	2000*(3.7−5)	2000*(6.2−5)
NNLS_n2K_m8K	2000	262	313	2000*(4.2−5)	2000*(1.6−6)	2000*(1.3−5)	2000*(8.2−6)	2000*(1.1−5)
NNLS_n4K_m10K	4000	2000*(1.6−6)	543	2000*(6.9−5)	826	2000*(4.9−5)	2000*(4.1−5)	2000*(3.6−5)
NNLS_n4K_m20K	4000	668	2000*(1.2−6)	2000*(1.5−4)	2000*(1.3−6)	2000*(8.1−5)	2000*(4.7−5)	2000*(5.8−5)
NNLS_n4K_m40K	4000	2000*(1.6−6)	2000*(1.6−6)	2000*(8.8−5)	2000*(2.9−6)	2000*(8.3−5)	2000*(1.3−5)	2000*(1.4−4)
NNLS_n4K_m8K	4000	580	591	2000*(1.0−4)	929	2000*(8.8−5)	2000*(2.1−5)	2000*(3.8−5)
NNLS_n6K_m20K	6000	2000*(2.5−6)	490	2000*(4.0−5)	2000*(1.2−6)	2000*(7.6−5)	2000*(9.9−6)	2000*(6.5−5)
NNLS_n6K_m40K	6000	2000*(3.0−6)	695	2000*(2.1−4)	2000*(1.2−6)	2000*(1.2−4)	2000*(5.5−5)	2000*(9.2−5)
NNLS_n8K_m20K	8000	964	723	2000*(1.6−4)	953	2000*(6.6−5)	2000*(3.7−5)	2000*(7.3−5)
NNLS_n8K_m40K	8000	334	2000*(2.3−6)	2000*(1.3−5)	2000*(1.4−6)	2000*(1.0−5)	2000*(6.2−6)	2000*(2.0−5)
NNLS_n10K_m20K	10000	2000*(2.1−6)	744	2000*(1.8−4)	2000*(3.0−6)	2000*(1.1−4)	2000*(5.9−5)	2000*(8.2−5)
NNLS_n10K_m40K	10000	2000*(2.6−6)	2000*(2.4−6)	2000*(1.3−4)	2000*(2.8−6)	2000*(7.0−5)	2000*(2.6−5)	2000*(1.3−4)
NNLS_n20K_m40K	20000	2000*(1.2−6)	2000*(1.8−6)	2000*(1.3−4)	2000*(3.4−6)	2000*(7.5−5)	2000*(3.9−5)	2000*(6.9−5)

Table 7 Time comparison of the methods on L1LS instances

Problem		Time in seconds (accuracy less that 10^{-5})				
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA
Ell1RAND_n_200_m_1000	200	2	2	6.9*(3.9−5)	2.3	5.4*(2.3−5)
Ell1RAND_n_200_m_10000	200	6.9	2.4	23.3*(4.8−4)	18.8	40.3*(5.6−4)
Ell1RAND_n_200_m_2000	200	2.2	4	7.5*(2.0−4)	6.8	19.1*(3.1−4)
Ell1RAND_n_200_m_20000	200	16.7	18.2	21.2*(8.1−4)	22.5	84.4*(4.8−4)
Ell1RAND_n_200_m_400	200	2	0.8	6.4*(6.7−5)	6.7	5.1*(3.8−5)
Ell1RAND_n_200_m_4000	200	15.3	6.4	11.1*(8.3−4)	9	11.2*(5.0−4)
Ell1RAND_n_200_m_40000	200	39.1	29.8	83.5*(8.6−4)	64.1	84.4*(4.6−4)
Ell1RAND_n_200_m_600	200	1.3	3.5	6.6*(8.6−5)	3.4	6.7*(5.4−5)
Ell1RAND_n_200_m_6000	200	13.4	3.5	10.3*(5.0−4)	6.3	12.2*(3.3−4)
Ell1RAND_n_200_m_800	200	3.2	2.7	4.7*(1.2−4)	2.4	17.1*(2.0−4)
Ell1RAND_n_200_m_8000	200	5.7	6.8	14.6*(2.9−4)	9.3	17.8*(5.5−4)
Ell1RAND_n_400_m_1000	400	1.8	1.9	7.2*(1.4−4)	6	20.8*(1.1−4)
Ell1RAND_n_400_m_10000	400	21.3	7	40.6*(7.2−4)	18.8	23.1*(4.7−4)
Ell1RAND_n_400_m_2000	400	3	3.4	8.3*(2.3−4)	6.7	5.9*(2.3−4)
Ell1RAND_n_400_m_20000	400	38.3	39.3	85.7*(1.2−3)	60.4	82.0*(9.3−4)
Ell1RAND_n_400_m_4000	400	11.8	3.8	9.0*(2.1−4)	7.4	13.9*(8.2−5)
Ell1RAND_n_400_m_40000	400	38.7	35.8	154.0*(1.1−3)	33.6	258.5*(1.0−3)
Ell1RAND_n_400_m_6000	400	18.1	5.4	17.3*(4.0−4)	11.9	7.0*(3.4−4)
Ell1RAND_n_400_m_800	400	3.3	2	7.3*(1.8−4)	6.1	6.9*(1.4−4)
Ell1RAND_n_400_m_8000	400	8.5	8	21.3*(7.6−4)	12.5	28.0*(4.4−4)
Ell1RAND_n_600_m_10000	600	18.8	9.5	36.5*(7.1−4)	37.3	38.8*(5.5−4)
Ell1RAND_n_600_m_2000	600	12.9	2.5	11.6*(2.5−4)	17.4	11.3*(1.9−4)

Table 7 continued

Problem		Time in seconds (accuracy less that 10^{-5})				
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA
Ell1RAND_n_600_m_20000	600	50.5	46	104.0*(1.2−3)	87.1	48.8*(5.2−4)
Ell1RAND_n_600_m_4000	600	18.8	7.3	8.5*(3.3−4)	24.2	22.0*(2.6−4)
Ell1RAND_n_600_m_40000	600	129.2	114.9	212.2*(1.5−3)	183.7	131.2*(9.2−4)
Ell1RAND_n_600_m_6000	600	21.7	10.5	9.3*(4.8−4)	33.1	23.6*(2.4−4)
Ell1RAND_n_600_m_8000	600	13.8	9.4	26.4*(4.3−4)	8.3	69.9*(3.3−4)
Ell1RAND_n_800_m_10000	800	12.2	32.1	45.6*(4.6−4)	34.3	50.7*(3.2−4)
Ell1RAND_n_800_m_2000	800	15.4	1.7	12.0*(2.9−4)	6.7	9.3*(2.8−4)
Ell1RAND_n_800_m_20000	800	54.4	55.2	127.8*(8.8−4)	40.9	150.1*(6.0−4)
Ell1RAND_n_800_m_4000	800	20.1	9.4	27.0*(3.9−4)	6.8	7.9*(2.2−4)
Ell1RAND_n_800_m_40000	800	72.5	104	233.8*(1.5−3)	126.3	138.2*(9.3−4)
Ell1RAND_n_800_m_6000	800	15.6	10.3	43.8*(4.6−4)	8.9	63.8*(3.6−4)
Ell1RAND_n_800_m_8000	800	20.4	6.5	56.7*(5.0−4)	51.7	55.1*(4.0−4)
Ell1RAND_n_1000_m_10000	1000	28.9	16.8	65.1*(5.8−4)	79.7	72.4*(2.7−4)
Ell1RAND_n_1000_m_2000	1000	13.4	2.6	10.5*(2.6−4)	11.6	37.0*(2.5−4)
Ell1RAND_n_1000_m_20000	1000	101.4	91.2	104.5*(8.9−4)	164.8	111.3*(7.3−4)
Ell1RAND_n_1000_m_4000	1000	21.2	9.6	30.4*(4.0−4)	12.5	64.0*(2.1−4)
Ell1RAND_n_1000_m_40000	1000	99.3	75.7	183.0*(1.4−3)	152.6	255.5*(8.0−4)
Ell1RAND_n_1000_m_6000	1000	14.9	18.3	40.5*(3.9−4)	28.5	62.3*(2.5−4)
Ell1RAND_n_1000_m_8000	1000	19.7	11.8	76.2*(4.0−4)	50.2	34.4*(3.4−4)
Ell1RAND_n_2000_m_10000	2000	61.5	65.7	107.3*(5.3−4)	144.5	118.5*(3.3−4)
Ell1RAND_n_2000_m_20000	2000	103.6	95.9	144.7*(7.8−4)	74.2	257.7*(5.0−4)
Ell1RAND_n_2000_m_4000	2000	35.5	24.1	46.2*(4.1−4)	51.4	52.4*(2.5−4)

Table 7 continued

Problem		Time in seconds (accuracy less that 10^{-5})				
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA
Ell1RAND_n_2000_m_40000	2000	238.4	174.7	227.6*(1.1−3)	149.9	294.5*(7.6−4)
Ell1RAND_n_2000_m_6000	2000	34.1	21.6	91.4*(3.6−4)	58.6	75.2*(3.0−4)
Ell1RAND_n_2000_m_8000	2000	38.1	22.2	87.0*(4.4−4)	145.9	80.6*(2.6−4)
Ell1RAND_n_4000_m_10000	4000	39.7	47.9	75.8*(5.5−4)	48.6	107.5*(3.4−4)
Ell1RAND_n_4000_m_20000	4000	116.9	128.1	175.4*(7.6−4)	184.7	244.4*(4.5−4)
Ell1RAND_n_4000_m_40000	4000	301.5	247.4	533.3*(1.1−3)	499.1*(1.4−5)	523.8*(7.3−4)
Ell1RAND_n_4000_m_8000	4000	104.7	25.7	80.8*(5.5−4)	129.0*(1.2−5)	135.8*(4.3−4)
Ell1RAND_n_6000_m_20000	6000	175.1	137	183.8*(8.2−4)	180.4	1212.6*(5.7−4)
Ell1RAND_n_6000_m_40000	6000	434.5	348	698.7*(9.1−4)	490.7	634.1*(5.1−4)
Ell1RAND_n_8000_m_20000	8000	141.3	168.8	437.6*(8.7−4)	246.8	511.5*(5.2−4)
Ell1RAND_n_8000_m_40000	8000	481	1033.6*(1.6−5)	897.4*(9.3−4)	883.6*(1.4−5)	763.6*(5.4−4)
Ell1RAND_n_10000_m_20000	10000	194.2	188.4	286.8*(1.1−3)	248.1	902.6*(6.1−4)
Ell1RAND_n_10000_m_40000	10000	462.1	1155.4*(1.4−5)	1079.3*(1.0−3)	1030.3*(1.1−5)	899.7*(6.0−4)
Ell1RAND_n_20000_m_40000	20000	1006	1637.7*(1.4−5)	1365.4*(1.3−3)	1911.0*(1.9−5)	2035.0*(8.2−4)

Table 8 Number of iterations comparison of the methods on L1LS instances

Problem		# of iterations (accuracy less that 10^{-5})				
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA
Ell1RAND_n_200_m_1000	200	462	469	2000*(3.9−5)	737	2000*(2.3−5)
Ell1RAND_n_200_m_10000	200	804	705	2000*(4.8−4)	1358	2000*(5.6−4)
Ell1RAND_n_200_m_2000	200	802	807	2000*(2.0−4)	1023	2000*(3.1−4)
Ell1RAND_n_200_m_20000	200	710	685	2000*(8.1−4)	883	2000*(4.8−4)
Ell1RAND_n_200_m_400	200	575	598	2000*(6.7−5)	655	2000*(3.8−5)
Ell1RAND_n_200_m_4000	200	938	864	2000*(8.3−4)	1296	2000*(5.0−4)
Ell1RAND_n_200_m_40000	200	705	656	2000*(8.6−4)	984	2000*(4.6−4)
Ell1RAND_n_200_m_600	200	864	729	2000*(8.6−5)	798	2000*(5.4−5)
Ell1RAND_n_200_m_6000	200	793	796	2000*(5.0−4)	1254	2000*(3.3−4)
Ell1RAND_n_200_m_800	200	802	645	2000*(1.2−4)	821	2000*(2.0−4)
Ell1RAND_n_200_m_8000	200	658	673	2000*(2.9−4)	914	2000*(5.5−4)
Ell1RAND_n_400_m_1000	400	701	742	2000*(1.4−4)	972	2000*(1.1−4)
Ell1RAND_n_400_m_10000	400	783	853	2000*(7.2−4)	1528	2000*(4.7−4)
Ell1RAND_n_400_m_2000	400	813	663	2000*(2.3−4)	980	2000*(2.3−4)
Ell1RAND_n_400_m_20000	400	990	780	2000*(1.2−3)	1898	2000*(9.3−4)
Ell1RAND_n_400_m_4000	400	584	575	2000*(2.1−4)	745	2000*(8.2−5)
Ell1RAND_n_400_m_40000	400	856	930	2000*(1.1−3)	1246	2000*(1.0−3)
Ell1RAND_n_400_m_6000	400	753	674	2000*(4.0−4)	1142	2000*(3.4−4)
Ell1RAND_n_400_m_800	400	712	760	2000*(1.8−4)	984	2000*(1.4−4)
Ell1RAND_n_400_m_8000	400	820	831	2000*(7.6−4)	1499	2000*(4.4−4)
Ell1RAND_n_600_m_10000	600	837	779	2000*(7.1−4)	1568	2000*(5.5−4)
Ell1RAND_n_600_m_2000	600	756	731	2000*(2.5−4)	1288	2000*(1.9−4)

Table 8 continued

Problem		# of iterations (accuracy less that 10^{-5})				
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA
Ell1RAND_n_600_m_20000	600	750	870	2000*(1.2−3)	1283	2000*(5.2−4)
Ell1RAND_n_600_m_4000	600	702	766	2000*(3.3−4)	1215	2000*(2.6−4)
Ell1RAND_n_600_m_40000	600	967	985	2000*(1.5−3)	1285	2000*(9.2−4)
Ell1RAND_n_600_m_6000	600	672	684	2000*(4.8−4)	1183	2000*(2.4−4)
Ell1RAND_n_600_m_8000	600	884	843	2000*(4.3−4)	1128	2000*(3.3−4)
Ell1RAND_n_800_m_10000	800	725	784	2000*(4.6−4)	1220	2000*(3.2−4)
Ell1RAND_n_800_m_2000	800	869	777	2000*(2.9−4)	1236	2000*(2.8−4)
Ell1RAND_n_800_m_20000	800	836	781	2000*(8.8−4)	1576	2000*(6.0−4)
Ell1RAND_n_800_m_4000	800	715	731	2000*(3.9−4)	1084	2000*(2.2−4)
Ell1RAND_n_800_m_40000	800	803	848	2000*(1.5−3)	1077	2000*(9.3−4)
Ell1RAND_n_800_m_6000	800	824	759	2000*(4.6−4)	1304	2000*(3.6−4)
Ell1RAND_n_800_m_8000	800	776	806	2000*(5.0−4)	1174	2000*(4.0−4)
Ell1RAND_n_1000_m_10000	1000	807	682	2000*(5.8−4)	1144	2000*(2.7−4)
Ell1RAND_n_1000_m_2000	1000	640	691	2000*(2.6−4)	1058	2000*(2.5−4)
Ell1RAND_n_1000_m_20000	1000	982	864	2000*(8.9−4)	1444	2000*(7.3−4)
Ell1RAND_n_1000_m_4000	1000	662	595	2000*(4.0−4)	987	2000*(2.1−4)
Ell1RAND_n_1000_m_40000	1000	841	758	2000*(1.4−3)	1386	2000*(8.0−4)
Ell1RAND_n_1000_m_6000	1000	693	729	2000*(3.9−4)	1090	2000*(2.5−4)
Ell1RAND_n_1000_m_8000	1000	947	842	2000*(4.0−4)	1856	2000*(3.4−4)
Ell1RAND_n_2000_m_10000	2000	858	730	2000*(5.3−4)	1551	2000*(3.3−4)
Ell1RAND_n_2000_m_20000	2000	807	829	2000*(7.8−4)	1453	2000*(5.0−4)
Ell1RAND_n_2000_m_4000	2000	729	745	2000*(4.1−4)	1312	2000*(2.5−4)

Table 8 continued

Problem		# of iterations (accuracy less that 10^{-5})				
Instance	n	AA-R1	AA-R2	AA	FISTA-R	FISTA
Ell1RAND_n_2000_m_40000	2000	1049	903	2000*(1.1−3)	1608	2000*(7.6−4)
Ell1RAND_n_2000_m_6000	2000	741	778	2000*(3.6−4)	1399	2000*(3.0−4)
Ell1RAND_n_2000_m_8000	2000	675	763	2000*(4.4−4)	1266	2000*(2.6−4)
Ell1RAND_n_4000_m_10000	4000	772	782	2000*(5.5−4)	1173	2000*(3.4−4)
Ell1RAND_n_4000_m_20000	4000	776	809	2000*(7.6−4)	1282	2000*(4.5−4)
Ell1RAND_n_4000_m_40000	4000	974	783	2000*(1.1−3)	2000*(1.4−5)	2000*(7.3−4)
Ell1RAND_n_4000_m_8000	4000	983	780	2000*(5.5−4)	2000*(1.2−5)	2000*(4.3−4)
Ell1RAND_n_6000_m_20000	6000	997	800	2000*(8.2−4)	1321	2000*(5.7−4)
Ell1RAND_n_6000_m_40000	6000	875	750	2000*(9.1−4)	1594	2000*(5.1−4)
Ell1RAND_n_8000_m_20000	8000	791	796	2000*(8.7−4)	1412	2000*(5.2−4)
Ell1RAND_n_8000_m_40000	8000	910	2000*(1.6−5)	2000*(9.3−4)	2000*(1.4−5)	2000*(5.4−4)
Ell1RAND_n_10000_m_20000	10000	835	864	2000*(1.1−3)	1242	2000*(6.1−4)
Ell1RAND_n_10000_m_40000	10000	832	2000*(1.4−5)	2000*(1.0−3)	2000*(1.1−5)	2000*(6.0−4)
Ell1RAND_n_20000_m_40000	20000	1009	2000*(1.4−5)	2000*(1.3−3)	2000*(1.9−5)	2000*(8.2−4)

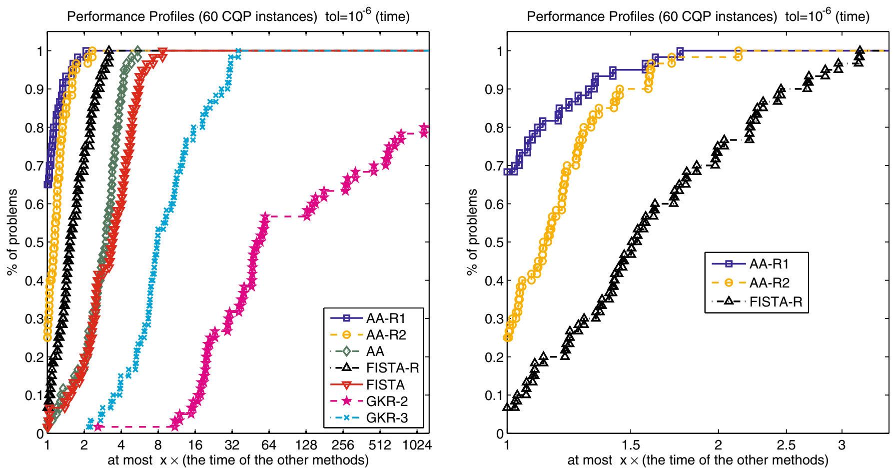


Fig. 1 Time performance profiles for solving CQP instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the *left* we include all the methods and on the *right* we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R

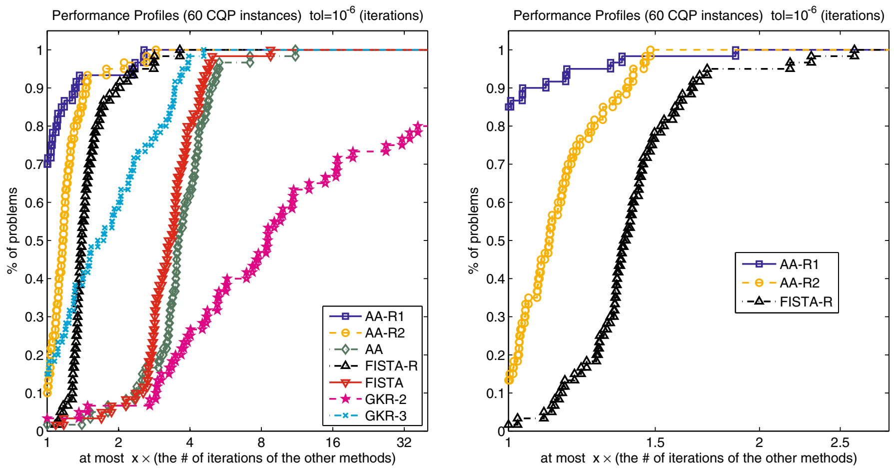


Fig. 2 Iteration performance profiles for solving CQP instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the *left* we include all the methods and on the *right* we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R

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

$$\min_{x \in \mathcal{S}^n} \left\{ \frac{1}{2} \|\mathcal{A}x - b\|^2 \ : \ x \in \mathcal{S}^n_+ \right\}. \tag{16}$$

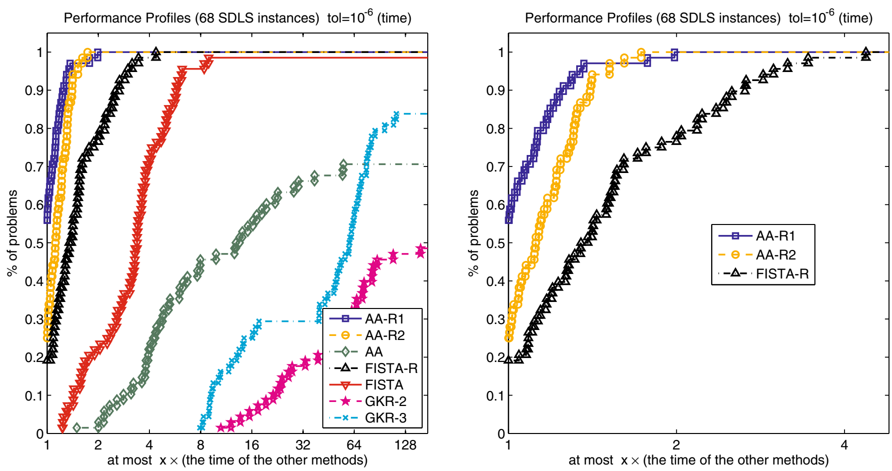


Fig. 3 Time performance profiles for solving SDLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the *left* we include all the methods and on the *right* we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R

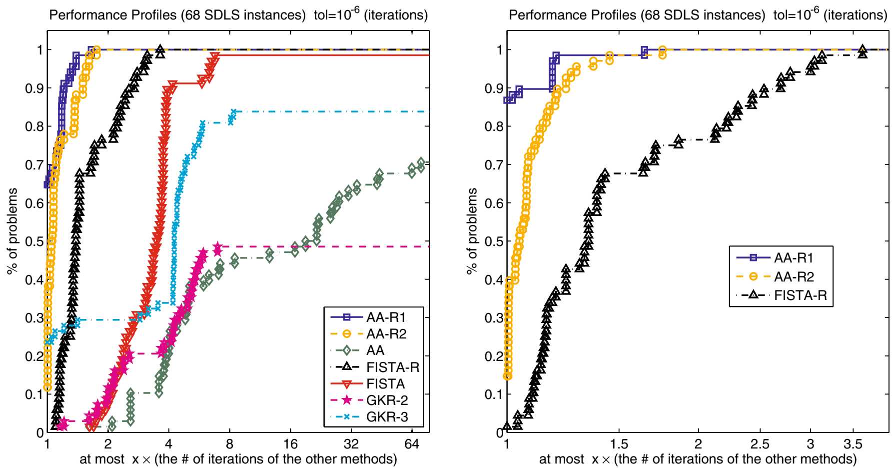


Fig. 4 Number of iterations performance profiles for solving SDLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the *left* we include all the methods and on the *right* we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R

Letting f and h be defined as

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

where $\|\cdot\|$ denotes the Euclidean norm, we can easily see that (16) 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

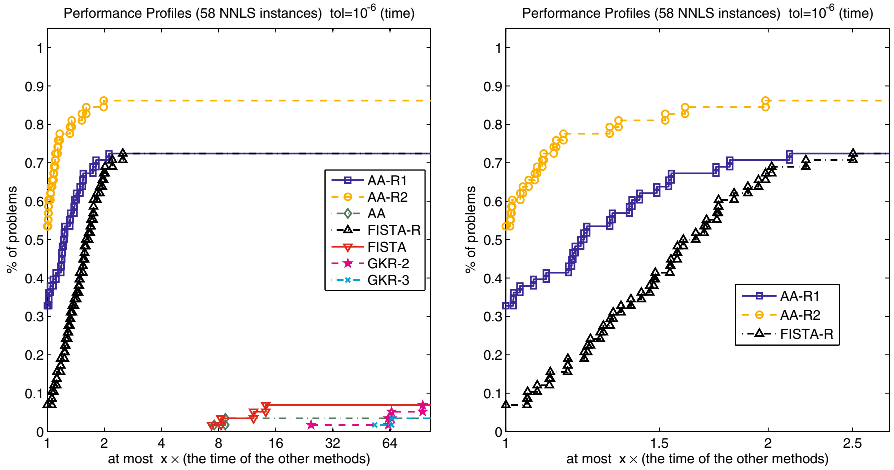


Fig. 5 Time performance profiles for solving NNLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the *left* we include all the methods and on the *right* we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R

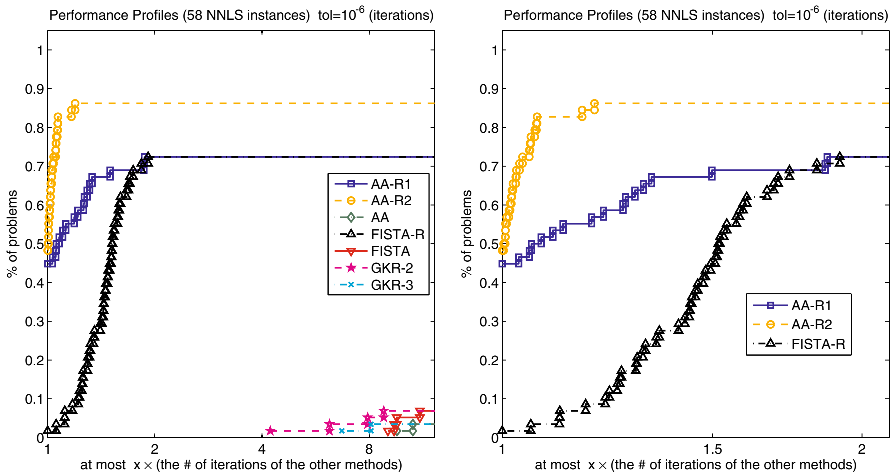


Fig. 6 Number of iterations performance profiles for solving NNLS instances with accuracy $\bar{\epsilon} = 10^{-6}$. On the *left* we include all the methods and on the *right* we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R

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 Sect. 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 Figs. 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.

3.3 Numerical results for NNLSs

This subsection compares the performance of our methods AA, 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

$$\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2} \|Ax - b\|^2 \; : \; x \geq 0 \right\}, \tag{17}$$

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 (17) 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 Figs. 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.

3.4 Numerical results for L1LSs

This subsection compares the performance of our methods AA, AA-R1 and AA-R2 with the variants of Nesterov's method FISTA and FISTA-R on a class of L1LS instances randomly generated according to [9].

Given a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$, the L1LS problem is defined as

$$\min_{x\in\mathbb{R}^n} \frac{1}{2} \|Ax - b\|^2 + \tau \|x\|_1 \tag{18}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\|\cdot\|_1$ denotes the ℓ_1 norm and $\tau > 0$. We set $\tau = 10^{-6}$ in our tests since larger values of this parameter would make the instances

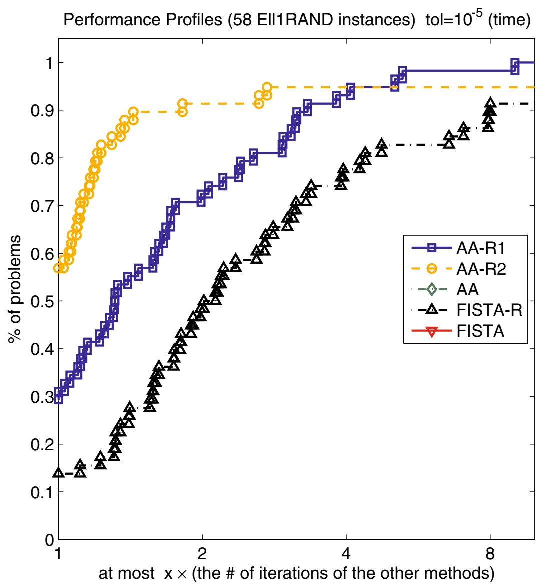


Fig. 7 Time performance profiles for solving L1LS instances with accuracy $\bar{\epsilon} = 10^{-5}$

harder for any of the benchmarked variants. Moreover, we set $\bar{\epsilon} = 10^{-5}$ for this comparison since for $\bar{\epsilon} = 10^{-6}$ only 30 % of the instances were solved by the best variant (in at most 2000 iterations).

Letting f and h be defined as

$$f(x) = \frac{1}{2}\|Ax - b\|^2, \qquad h(x) = \tau\|x\|_1, \qquad \forall x \in \mathbb{R}^n,$$

we can easily see that (18) is a special case of (1) with $\Omega = \mathcal{X} = \mathbb{R}^n$.

Figures 7 and 8 plot the time and iteration performance profiles of all variants of Nesterov's method for solving this collection of random L1LS instances, respectively. Tables 7 and 8 report the time and number of iterations taken by each method, respectively.

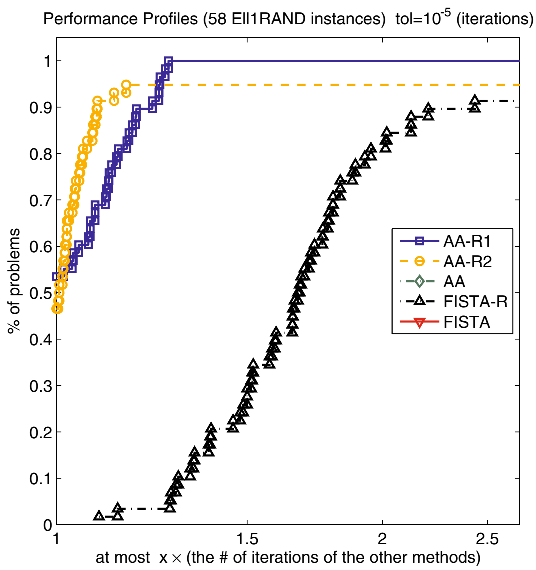
Note that AA-R1 and AA-R2 outperform the other methods on most of the L1LS instances where a solution with the required accuracy was found. Also the methods that do not incorporate a restart scheme are not able to solve any of the L1LS instances, while the ones that incorporate a restart scheme solve at least 90 % of them.

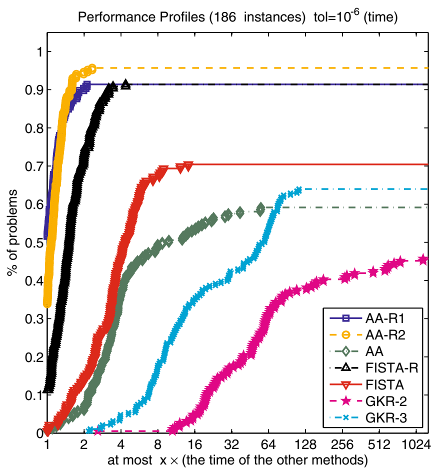
4 Concluding remarks

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

Figures 9 and 10 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 Sect. 3.1, 3.2 and 3.3. From these plots we can see that the

Fig. 8 Number of iterations performance profiles for solving L1LS instances with accuracy $\bar{\epsilon} = 10^{-5}$





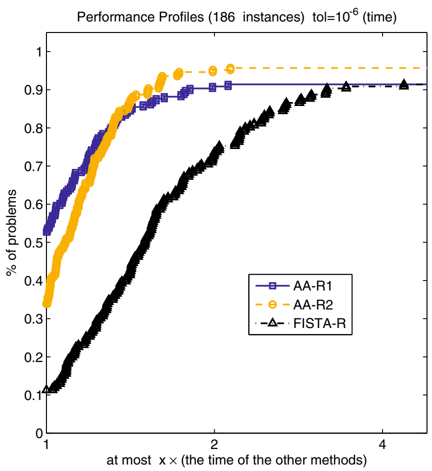


Fig. 9 Time performance profiles for solving all the three problem classes (CQPs, SDLSs and NNLSs) with accuracy $\bar{\epsilon} = 10^{-6}$. On the *left* we include all the methods and on the *right* we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R

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 seven 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/.

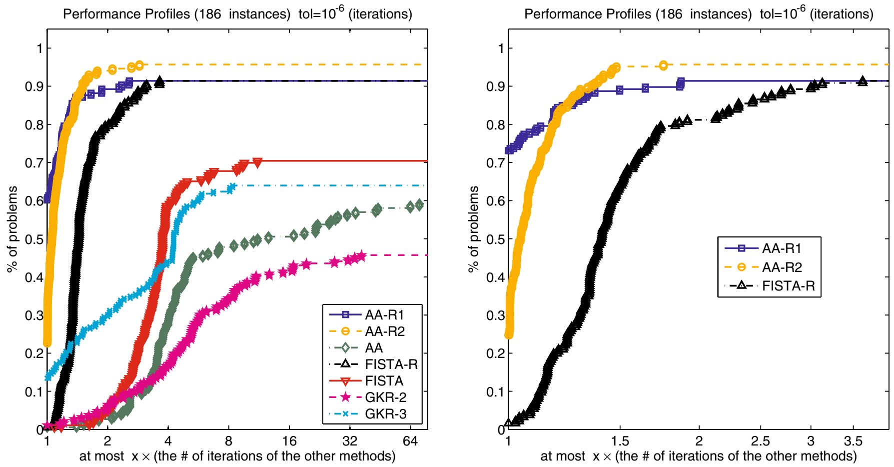


Fig. 10 Iterations performance profiles for solving all the three problem classes (CQPs, SDLSs and NNLSs) with accuracy $\bar{\epsilon} = 10^{-6}$. On the *left* we include all the methods and on the *right* we include the three fastest variants, namely: AA-R1, AA-R2 and FISTA-R

Acknowledgments We want to thank Elizabeth W. Karas for generously providing us with the code of the algorithms in [7]. Renato D.C Monteiro: The work of this author was partially supported by NSF Grants CMMI-0900094 and CMMI-1300221, and ONR Grant ONR N00014-11-1-0062. Benar F. Svaiter: 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.

Appendix 1: Technical results used in Section 2

This section establishes a technical result which is used in the proof of Proposition 1 of Sect. 2, namely, Lemma 4.

This section assumes that the functions ϕ, f and h, and the set Ω, satisfy conditions C.1–C.5 of Sect. 2. Before stating and giving the proof of Lemma 4, we establish two technical results.

Lemma 2 *Let $A \geq 0$, $\lambda > 0$ and x^0, $y \in \mathcal{X}$ be given and assume that $\Gamma : \mathcal{X} \to \bar{\mathbb{R}}$ is a proper closed convex function $\Gamma : \mathcal{X} \to \bar{\mathbb{R}}$ such that $A\Gamma$ is $A\mu$-strongly convex. Assume also that the triple (y, A, Γ) satisfies*

$$A\Gamma \leq A\phi, \quad A\phi(y) \leq \min_{u \in \mathcal{X}} \left\{ A\Gamma(u) + \frac{1}{2}\|u - x^0\|^2 \right\}. \tag{19}$$

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

$$x := \arg\min_{u \in \mathcal{X}} \left\{ A\Gamma(u) + \frac{1}{2}\|u - x^0\|^2 \right\}, \tag{20}$$

$$a := \frac{\lambda(A\mu+1)+\sqrt{\lambda^2(A\mu+1)^2+4\lambda(A\mu+1)A}}{2}, \qquad \tilde{x} := \frac{A}{A+a}y + \frac{a}{A+a}x. \tag{21}$$

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

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

where

$$\theta := \min_{u\in\mathcal{X}} \left\{ \gamma(u) + \frac{1}{2\lambda}\|u-\tilde{x}\|^2 \right\}. \tag{22}$$

Proof Note also that the definition of a in (21) implies that

$$\lambda = \frac{a^2}{(A+a)(A\mu+1)}. \tag{23}$$

Now, let an arbitrary $u \in \mathcal{X}$ be given and define

$$\tilde{u} := \frac{A}{A+a}y + \frac{a}{A+a}u.$$

Clearly, in view of the second identity in (21), we have

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

In view of (19), (20), the last two relations, and the fact that γ is a convex function minorizing ϕ and the function $A\Gamma(\cdot) + \|\cdot - x^0\|^2/2$ is $(A\mu+1)$-strongly convex, we conclude that

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

where the last inequality is due to (23) and the definition of θ in (22). Since the above inequality holds for every $u \in \mathcal{X}$, the conclusion of the lemma follows. □

Lemma 3 *Let $\lambda > 0$, $\xi \in \mathcal{X}$ and a proper closed μ-strongly convex function $g : \mathcal{X} \to \bar{\mathbb{R}}$ be given and denote the optimal solution of*

$$\min_{x \in \mathcal{X}} \left\{ g(x) + \frac{1}{2\lambda} \|x - \xi\|^2 \right\} \tag{24}$$

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

$$\gamma(u) := g(\bar{x}) + \frac{1}{\lambda} \langle \xi - \bar{x}, u - \bar{x} \rangle + \frac{\mu}{2} \|u - \bar{x}\|^2 \quad \forall u \in \mathcal{X}$$

has the property that $\gamma \le g$ and $\bar{x}$ is the optimal solution of

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

As a consequence, the optimal value of (25) is the same as that of (24).

Proof The optimality condition for (24) implies that $(\xi - \bar{x})/\lambda \in \partial g(\bar{x})$, and hence that $\gamma \le g$ in view of the definition of γ and the fact that g is a proper closed μ-strongly convex function. Moreover, $\bar{x}$ clearly satisfies the optimality condition for (25), from which the second claim of the lemma follows. Finally, the latter conclusion of the lemma follows immediately from its second claim. □

The main result of this appendix is as follows.

Lemma 4 *Let $A \ge 0$, $0 < \lambda \le 1/(L - \mu_f)$, $x^0, y \in \mathcal{X}$ and a proper closed μ-strongly convex function $\Gamma : \mathcal{X} \to \mathbb{R}$ be given, and assume that the triple (y, A, Γ) satisfies (19). Let x, a and $\tilde{x}$ be as in (20) and (21), and define*

$$\tilde{x}_\Omega := \Pi_\Omega(\tilde{x}), \qquad A^+ := A + a, \tag{26}$$

$$y^+ := \arg\min_{u \in \mathcal{X}} \left\{ p(u) + \frac{1}{2\lambda} \|u - \tilde{x}\|^2 \right\} \tag{27}$$

where

$$p(u) := f(\tilde{x}_\Omega) + \langle \nabla f(\tilde{x}_\Omega), u - \tilde{x}_\Omega \rangle + \frac{\mu_f}{2} \|u - \tilde{x}_\Omega\|^2 + h(u) \quad \forall u \in \mathcal{X}. \tag{28}$$

Also, define the functions $\gamma, \Gamma^+ : \mathcal{X} \to \bar{\mathbb{R}}$ as

$$\gamma(u) := p(y^+) + \frac{1}{\lambda} \langle \tilde{x} - y^+, u - y^+ \rangle + \frac{\mu}{2} \|u - y^+\|^2 \quad \forall u \in \mathcal{X}, \tag{29}$$

$$\Gamma^+ := \frac{A}{A + a} \Gamma + \frac{a}{A + a} \gamma \tag{30}$$

Then, Γ^+ *is a proper closed* μ*-strongly convex function and the triple* (y^+, A^+, Γ^+) *satisfies*

$$A^+\phi(y^+) \le \min_{u\in\mathcal{X}} \left\{ A^+\Gamma^+(u) + \frac{1}{2}\|u - x^0\|^2 \right\}, \qquad A^+\Gamma^+ \le A^+\phi, \tag{31}$$

$$\sqrt{A^+} \ge \sqrt{A} + \frac{1}{2}\sqrt{\lambda(A\mu + 1)}. \tag{32}$$

Moreover, if Γ *is a quadratic function with Hessian equal to* μI*, then the (unique) optimal solution* x^+ *of the minimization problem in* (31) *can be obtained as*

$$x^+ = \frac{1}{1 + \mu(A + a)} \left[x - \frac{a}{\lambda}(\tilde{x} - y^+) + \mu(Ax + ay^+) \right]. \tag{33}$$

Proof We first claim that

$$\gamma \le \phi, \qquad \phi(y^+) \le \min_{u\in\mathcal{X}} \left\{ \gamma(u) + \frac{1}{2\lambda}\|u - \tilde{x}\|^2 \right\}. \tag{34}$$

To prove the above claim, note that conditions C.2 and C.3, the non-expansiveness property of Π_Ω, and the assumption that $\lambda \in (0, 1/(L - \mu_f)]$, imply that

$$p(u) \le \phi(u) \le p(u) + \frac{L - \mu_f}{2}\|u - \tilde{x}_\Omega\|^2 \le p(u) + \frac{1}{2\lambda}\|u - \tilde{x}\|^2 \quad \forall u \in \Omega. \tag{35}$$

Then, in view of (27) and the above relation with $u = y^+$, we have

$$\min_{u\in\mathcal{X}} \left\{ p(u) + \frac{1}{2\lambda}\|u - \tilde{x}\|^2 \right\} = p(y^+) + \frac{1}{2\lambda}\|y^+ - \tilde{x}\|^2 \ge \phi(y^+).$$

The claim now follows from the first inequality in (35), the above relation, the definitions of γ and p, and Lemma 3 with $g = p$ and $\xi = \tilde{x}$ (and hence $\bar{x} = y^+$).

Now, using the assumption that (19) holds, the second relation in (34) and Lemma 2, we conclude that

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

and hence that the first inequality in (31) holds in view of (26) and (30). Moreover, the first inequality in (34), relations (26) and (30), and the first inequality of (19), clearly imply the second inequality in (31).

To show (32), let $\lambda_\mu := \lambda(A\mu + 1)$ and note that the definition of a in (21) implies that

$$a \geq \frac{\lambda_\mu}{2} + \sqrt{\lambda_\mu A}.$$

This inequality together with the definition of A^+ in (26) then yields

$$A^+ \geq A + \left(\frac{\lambda_\mu}{2} + \sqrt{\lambda_\mu A}\right) \geq \left(\sqrt{A} + \frac{1}{2}\sqrt{\lambda_\mu}\right)^2,$$

showing that (32) holds. Finally, to show (33), first observe that the optimality conditions for (20) and (31) imply that $x = x_0 - A\nabla\Gamma(x)$ and $x^+ = x_0 - A^+\nabla\Gamma^+(x^+)$. These two identities together with relations (26), (29) and (30), and the assumption that Γ is a quadratic function with Hessian equal to μI, then imply that

$$\begin{aligned} x^+ &= x_0 - A\nabla\Gamma(x^+) - a\nabla\gamma(x^+) \\ &= x_0 - A\nabla\Gamma(x) - a\nabla\gamma(x^+) + A[\nabla\Gamma(x) - \nabla\Gamma(x^+)] \\ &= x - a\left[\frac{1}{\lambda}(\tilde{x} - y^+) + \mu(x^+ - y^+)\right] + A[\mu(x - x^+)], \end{aligned}$$

and hence that (33) holds. □

Appendix 2: Proof of Proposition 3

First note that the equivalence between (a) and (b) of Lemma 1 with $\chi = \sum_{i=1}^m \alpha_i\phi(y)$, $u_0 = y$ and $q(\cdot) = \sum_{i=1}^m \alpha_i\gamma_i(\cdot) + \|\cdot - x_0\|^2/2$ (and hence $Q = I + \sum_{i=1}^m \alpha_i\nabla^2\gamma_i$) imply that the hard constraint in problem (12) is equivalent to

$$\begin{aligned} &\left\langle \left(\sum_{i=1}^m \alpha_i\nabla\gamma_i(y) + y - x^0\right), \left(I + \sum_{i=1}^m \alpha_i\nabla^2\gamma_i\right)^{-1}\left(\sum_{i=1}^m \alpha_i\nabla\gamma_i(y) + y - x^0\right)\right\rangle \\ &\quad + 2\sum_{i=1}^m \alpha_i[\phi(y) - \gamma_i(y)] \leq \|y - x^0\|^2. \end{aligned} \tag{36}$$

Now, assume that (b) holds. Then, in view of (13), the point $\alpha(t) := (t\bar{\alpha}_1, \ldots, t\bar{\alpha}_m)$ clearly satisfies (36), and hence is feasible for (12) for every $t > 0$. Hence, for a fixed $x^* \in X^*$, this conclusion implies that

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

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 (12) at the point $\alpha(t)$ converges to infinity as $t \to \infty$. We have thus shown that (b) implies (a).

Now assume that (a) holds. Since the feasible set of (12) is closed convex, it must have a nonzero direction of recession $\bar{\alpha} := (\bar{\alpha}_1, \ldots, \bar{\alpha}_m)$. Moreover, since this set obviously contains the zero vector, it follows that $\alpha(t) := (t\bar{\alpha}_1, \ldots, t\bar{\alpha}_m)$ is feasible for (12) for every $t > 0$. This implies that $\alpha(t)$ satisfies (36) for every $t > 0$ and $\bar{\alpha}_i \geq 0$ for every i. Using the latter fact, the assumption that $\gamma_i \leq \phi$ (and hence $\gamma_i(y) \leq \phi(y)$) for every i, and the assumption that either $\nabla^2\gamma_i$ is zero or is positive definite for every i (and hence, $\sum_{i=1}^m \bar{\alpha}_i \nabla^2\gamma_i$ is zero or is positive definite), we easily see that (13) holds. □

References

1. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Img. Sci. **2**(1), 183–202 (2009). http://dx.doi.org/10.1137/080716542
2. Burer, S., Monteiro, R.D.C., Zhang, Y.: A computational study of a gradient-based log-barrier algorithm for a class of large-scale SDPs. Math. Program. **95**, 359–379 (2003). http://dx.doi.org/10.1007/s10107-002-0353-7
3. Devolder, O., Glineur, F., Nesterov, Y.: 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, Dec 2010
4. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002). http://dx.doi.org/doi:10.1007/s101070100263
5. Goldfarb, D., Scheinberg, K.: Fast first-order methods for composite convex optimization with line search. Optim.ization-online Prepr. 3004 (2011). http://www.optimization-online.org/DB_HTML/2011/04/3004.html
6. Gonzaga, C.C., Karas, E.W.: Fine tuning Nesterov's steepest descent algorithm for differentiable convex programming. Math. Program., 1–26 (2012). http://dx.doi.org/10.1007/s10107-012-0541-z
7. Gonzaga, Clóvis C., Karas, Elizabeth W., Rossetto, Diane R.: An optimal algorithm for constrained differentiable convex optimization. SIAM J. Optim. **23**(4), 1939–1955 (2013). http://dx.doi.org/10.1137/110836602
8. Lan, G., Lu, Z., Monteiro, R.D.C.: Primal-dual first-order methods with iteration-complexity for cone programming. Math. Program. **126**(1), 1–29 (2011)
9. Merritt, M., Zhang, Y.: Interior-point gradient method for large-scale totally nonnegative least squares problems. J. Optim. Theory Appl. **126**, 191–202 (2005). http://dx.doi.org/10.1007/s10957-005-2668-z
10. Monteiro, R.D.C., Svaiter, B.F.: An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. SIAM J. Optim. **23**(2), 1092–1125 (2013). http://epubs.siam.org/doi/abs/10.1137/110833786
11. Nesterov, Y.: A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. Doklady AN SSSR **269**(3), 543–547 (1983)
12. Nesterov, Y.: Introductory lectures on convex optimization: a basic course of applied optimization, vol. 87. Kluwer Academic Publishers, Boston (2004)
13. Nesterov, Y.: Smooth minimization of non-smooth functions. Math. Program. **103**(1), 127–152 (2005)
14. Nesterov, Y.: Dual extrapolation and its applications to solving variational inequalities and related problems. Math. Program. **109**(2–3, Ser. B), 319–344 (2007)
15. Nesterov, Y.: Gradient methods for minimizing composite objective function. CORE, Leuven (2007)
16. O'Donoghue, B., Candès, E.: Adaptive restart for accelerated gradient schemes. *Found. Comput. Math.*, 1–18 (2013). http://dx.doi.org/10.1007/s10208-013-9150-3
17. Povh, J., Rendl, F., Wiegele, A.: A boundary point method to solve semidefinite programs. Computing **78**, 277–286 (2006). http://dx.doi.org/10.1007/s00607-006-0182-2

18. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. submitted to SIAM Journal. SIAM J Optim. (2008)
19. Zhao, X.-Y., Sun, D., Toh, K.-C.: A Newton-CG augmented lagrangian method for semidefinite programming. SIAM J. Optim. **20**(4), 1737–1765 (2010). http://link.aip.org/link/?SJE/20/1737/1