An accelerated first-order method with complexity analysis for solving cubic regularization subproblems

Rujun Jiang* Man-Chung Yue† Zhishuo Zhou‡

November 28, 2019

Abstract

We propose a novel first-order method to solve the cubic regularization subproblem (CRS) based on a novel reformulation. The reformulation is a constrained convex optimization problem whose feasible region admits an easily computable projection. Our reformulation requires computing the minimum eigenvalue of the Hessian. To avoid the expensive computation of the exact minimum eigenvalue, we develop a surrogate problem to the reformulation where the exact minimum eigenvalue is replaced with an approximate one. We then apply first-order methods such as the Nesterov's accelerated projected gradient method (APG) and projected Barzilai-Borwein method to solve the surrogate problem. As our main theoretical contribution, we show that when an ϵ-approximate minimum eigenvalue is computed by the Lanczos method and the surrogate problem is approximately solved by APG, our approach returns an ϵ-optimal solution to CRS in $\tilde{O}(\epsilon^{-1/2})$ matrix-vector multiplications. Numerical experiments show that our methods are comparable to and outperform the Krylov subspace method in the easy and hard cases, respectively.

1 Introduction

Motivated by applications in machine learning and signal processing, optimization problems of the following form have attracted significant attention:

$$\min_{x\in\mathbb{R}^n} F(x), \tag{1}$$

*School of Data Science, Fudan University, Shanghai, China, rjjiang@fudan.edu.cn

†Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, manchung.yue@polyu.edu.hk

‡School of Data Science, Fudan University, Shanghai, China, zhouzs18@fudan.edu.cn

where F is a twice continuously differentiable function that is possibly non-convex. The cubic regularization method [16, 8] is among the most successful algorithms for solving problem (1). At each iteration of the cubic regularization method, the subproblem takes the form

$$\min_{x\in\mathbb{R}^n} f_1(x) := \frac{1}{2}x^T Ax + b^T x + \frac{\rho}{3}\|x\|^3, \tag{CRS}$$

where $\|\cdot\|$ denotes the Euclidean norm, A is an $n \times n$ symmetric matrix (not necessarily positive semidefinite) and ρ is a regularization parameter. In particular, A and b represent the Hessian and gradient of the function F at the current iterate, respectively. It was first proved by Nesterov and Polyak [16] that the cubic regularization method enjoys an iteration complexity of $O(\epsilon^{-3/2})$ if each subproblem is solved exactly. Cartis et al. [8] developed a generalization of the cubic regularization method, called ARC, which allows the subproblems to be solved inexactly and the regularization parameter $\rho > 0$ to be chosen adaptively. In the same paper, they showed that the iteration complexity of ARC is again $O(\epsilon^{-3/2})$. Complementing to these global complexity results, Yue et al. [22] showed that the cubic regularization method enjoys a local quadratic convergence rate under an error bound-type condition.

Despite the above strong theoretical guarantees, the practical performance of the cubic regularization method depends critically on the efficiency of solving its subproblems. As such, there have been considerable endeavors on developing fast algorithms for solving (CRS). One of the most successful algorithms for solving large-scale instances of (CRS) in practice is the Krylov subspace method [8]. Carmon and Duchi [6] provided the first the convergence rate analysis of the Krylov subspace method. In particular, they showed that the Krylov subspace method achieves an ϵ-approximate optimal solution in $O(\epsilon^{-1/2})$ or $O(\sqrt{\kappa}\log 1/\epsilon)$ operations (matrix-vector multiplications) in the easy case[1], where κ is the condition number of (CRS). Unfortunately, the Krylov subspace method may fail to converge to the optimal solution when the problem (CRS) is in the hard case or close to being in the hard case [6]. Carmon and Duchi also showed in another paper [5] that the gradient descent method is able to converge to the global minimizer if the step size is sufficiently small, and the convergence rate is $\tilde{O}(1/\epsilon)$. Although, for the problem (CRS), the convergence rate of the gradient descent method is worse than that of the Krylov subspace method, it works in both the easy and hard cases. On the other hand, based on the cubic regularization method, Agarwal et al. [1] derived an algorithm with $\tilde{O}(\epsilon^{-7/4})$ operations (where $\tilde{O}(\cdot)$ hides the logarithmic factors) for finding an approximate local minimum of problem (1), i.e., a point satisfies $x \in \mathbb{R}^n$ satisfying

$$\|\nabla F(x)\| \le \epsilon \quad \text{and} \quad \nabla^2 F(x) \succeq -\sqrt{\epsilon} I.$$

[1]For the problem (CRS), it is said to be in the easy if the optimal solution x^* satisfies $\rho\|x^*\| > -\lambda_1$ and hard case otherwise.

A key component of their result is an algorithm for computing an approximate solution to the problem (CRS) in $\tilde{O}(\epsilon^{-1/4})$ operations. However, the approximate solution returned by this algorithm is not an ϵ-approximate global minimizer of the problem (CRS) in the traditional sense (see [1, Theorem 2] for details). Furthermore, the algorithm in [1] for solving (CRS) requires sophisticated parameter tuning, and no numerical results had been provided in the paper. Finally, a Newton-like method for solving problems of the form (1) had been developed by Birgin and Martínez recently [4] based on the Bunch-Parlett-Kaufman factorization [11], a matrix factorization whose computational cost is similar to that of the Cholesky factorization. The subproblem of this algorithm is of the same form as (CRS) except that the matrix A is diagonal (see [4, Equation (8)]), and the authors developed a specialized algorithm for solving such subproblems, which is not applicable to general instances of (CRS). From the above discussion, it is desirable to have an algorithm for solving the problem (CRS) that works efficiently in practice for both the hard and easy cases and enjoys theoretical guarantees. In this paper, we achieve this goal by developing a first-order method for solving arbitrary instances of (CRS) with $\tilde{O}(\epsilon^{-1/2})$ matrix-vector multiplications.

Our approach is based on a novel reformulation of the problem (CRS), which is a constrained convex optimization problem built using the minimum eigenvalue of the matrix A. The feasible region of the reformulation admits an efficient, closed-form projection. Therefore, when the exact computation of the minimum eigenvalue is viable, we can apply any algorithm for solving constrained convex optimization problems to solve the reformulation to global optimality. The optimal solution to the problem (CRS) can then be constructed by using the optimal solution of the reformulation. In practice, it is often prohibitively expensive to compute the exact minimum eigenvalue of the matrix A, if not impossible. We circumvent this limitation by developing a surrogate problem to the reformulation. The surrogate problem is again a constrained convex optimization problem with an easily computable projection onto its feasible region. More importantly, the surrogate problem requires only an approximate minimum eigenvalue, which can be computed efficiently by using, e.g., the Lanczos method [11]. Similarly, an ϵ-approximate optimal of the problem (CRS) can be constructed from an ϵ-approximate solution of the surrogate problem.

The said bound $\tilde{O}(\epsilon^{-1/2})$ on the number of operations is proved by combining the follow two ideas. First, the computation of an ϵ-approximate minimum eigenvalue by using the Lanczos method requires $O(\epsilon^{-1/2}\log(n/\delta))$ matrix-vector multiplications. Second, solving the surrogate problem by the Nesterov's accelerated projected gradient descent method [17, 3] (APG) requires $O(\epsilon^{-1/2})$ iterations, where each iteration consists of one gradient and Hessian evaluations and one matrix-vector multiplication. Therefore, the total number of operations of our method is bounded by $O(\epsilon^{-1/2}\log(n/\delta))$ (see Theorem 3.4). This bound is similar

to the sublinear bound for the Krylov subspace method proved in [6] in the easy case and better than that of the gradient descent method in [5]. Note also that our bound is for the subproblem and hence not directly comparable with that of [1]. Besides, our algorithm has the advantage that it is easily implementable. Furthermore, as we shall see in our numerical section, the proposed algorithm works efficiently in practice for high-dimensional problems—our algorithm shows a comparable performance to the Krylov subspace method in the easy case. An another advantage of our algorithm is that, unlike the Krylov subspace method, it works in both the easy and hard cases. This saves us from the computational overhead due to the need of detecting the hard case.

We remark that our approach is inspired by the recent line of research [12, 21] on linear-time algorithms for the trust region subproblem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \frac{1}{2}x^T Ax + b^T x \\ \text{subject to} \quad & \|x\|^2 \le 1, \end{aligned} \tag{TRS}$$

and the close resemblance between the problems (CRS) and (TRS). More specifically, the algorithms in [12, 21] are based on a convex reformulation for the (TRS) derived in [9]. Motivated by the works [12, 21], Jiang and Li [14] recently derived a novel convex reformulation for the generalized trust region subproblem, which further inspires us to explore hidden convexity for (CRS) in this paper. It should also be pointed out that our reformulation and its surrogate problem offer great potential and flexibility for the design of fast algorithms to solve the problem (CRS). Indeed, one can apply any algorithm for constrained convex optimization problems to solve these two optimization problems. Proving theoretical guarantees for other algorithms for solving these two models is left as a future research.

The remaining of this paper is organized as follows. In Section 2, we derive our convex reformulation based on the minimum eigenvalue of matrix A and discuss the computation of the projection to its feasible region. In Section 3, we present a surrogate problem for (CRS) and theoretically analyze the complexity of our method when applying the APG to solve the surrogate problem with an approximate minimum eigenvalue computed by the Lanczos method. In Section 4, we compare the numerical performance of our methods with the Krylov subspace method. We conclude our paper in Section 5.

2 Convex reformulation

We first record the optimality condition of (CRS) [16, 8], which is given by the following system of equations in x and λ:

$$Ax + b + \lambda x = 0, \quad A + \lambda I \succeq 0, \quad \text{and} \quad \lambda = \rho\|x\|. \tag{2}$$

This optimality condition will be frequently used in this paper. It is obvious that (CRS) is equivalent to the following problem:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n, y\in\mathbb{R}} & \frac{1}{2}x^TAx + b^Tx + \frac{\rho}{3}y^{\frac{3}{2}} \\ \text{subject to} & \|x\|^2 \le y. \end{array} \tag{RP}$$

Note that the feasible region $\{(x,y) \in \mathbb{R}^n \times \mathbb{R} : \|x\|^2 \le y\}$ of the problem (RP) is convex. Therefore, when $A \succeq 0$, (RP) is a convex optimization problem and can be solved efficiently by various methods, e.g., APG or projected Barzilai-Borwein method (BBM) [2, 20]. Hence, from now on, we assume that the minimum eigenvalue of matrix A, denoted by λ_1, is negative, i.e., $\lambda_1 < 0$. Consider the optimization problem

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n, y\in\mathbb{R}} & f_2(x,y) := \frac{1}{2}x^T(A-\lambda_1 I)x + b^Tx + \frac{\rho}{3}y^{\frac{3}{2}} + \frac{\lambda_1}{2}y \\ \text{subject to} & \|x\|^2 \le y, \end{array} \tag{CP}$$

where I is the $n \times n$ identity matrix. Problem (CP) is a convex problem because f_2 is separable in x and y and is convex in each of these two variables. The following theorem shows that problem (CRS) is equivalent to problem (CP).

Theorem 2.1. *Problem* (CRS) *is equivalent to* (CP) *in the following sense. First, the two problems have the same optimal value. Second, if x^* is an optimal solution to* (CRS)*, then $(x^*, \|x^*\|^2)$ is an optimal solution to* (CP)*. Third, if $(\tilde{x}, \tilde{y})$ is an optimal solution to* (CP)*, then an optimal solution to* (CRS) *is given by*

$$\hat{x} = \begin{cases} \tilde{x} & \text{if } \|\tilde{x}\|^2 = \tilde{y}, \\ \tilde{x} + \zeta v & \text{if } \|\tilde{x}\|^2 < \tilde{y}, \end{cases}$$

where ζ is a root of the quadratic equation $\|\tilde{x} + \zeta v\|^2 = \tilde{y}$ and v is the eigenvector associated with the minimum eigenvalue of A.

Proof. Denote by Val(CRS) and Val(CP) the optimal values of problems (CRS) and (CP), respectively. We first observe that (CP) is a convex problem and satisfies the Slater condition. Assume that x^* is an optimal solution to (CRS). By using the optimality condition (2), we can easily show that the triplet $(x, y, \mu) = (x^*, \|x^*\|^2, \frac{1}{2}(\rho\|x^*\| + \lambda_1))$ satisfies the KKT system of (CP):

$$(A - \lambda_1 I)x + b + 2\mu x = 0 \text{ and } \frac{\rho}{2}y^{\frac{1}{2}} + \frac{\lambda_1}{2} - \mu = 0 \tag{3}$$

This implies that $(x^*, \|x^*\|^2)$ is an optimal solution to (CP) and that Val(CRS) $\ge$ Val(CP). On the other hand, because of the assumption $\lambda_1 < 0$ and the constraint $\|x\|^2 \le y$, we have

that Val(CRS) $\leq$ Val(CP). Therefore, Val(CRS) = Val(CP). This completes the proof of the first and second claims.

To prove the third claim, assume that (CP) has an optimal solution $(\tilde{x}, \tilde{y})$. Suppose μ is a Lagrangian multiplier associated with the constraint in (CP). If $\|\tilde{x}\|^2 = \tilde{y}$, from the KKT system (3), we have that

$$A\tilde{x} - \lambda_1\tilde{x} + b + 2\mu\tilde{x} = 0 \tag{4}$$

and

$$\frac{1}{2}\rho\sqrt{\tilde{y}} + \frac{1}{2}\lambda_1 - \mu = 0. \tag{5}$$

Equation (5) implies $\mu = \rho\sqrt{\tilde{y}}/2 + \lambda_1/2 \geq 0$. This, together with $\|\tilde{x}\|^2 = \tilde{y}$ and $A\tilde{x} - \lambda_1\tilde{x} + b + 2\mu\tilde{x} = 0$, implies that $A\tilde{x} + b + \bar{\lambda}\tilde{x} = 0$ and $A + \bar{\lambda}I \succeq 0$, for $\bar{\lambda} = 2\mu - \lambda_1 = \rho\|\tilde{x}\|$. Hence, due to (2), $\tilde{x}$ is also optimal for (CRS) and the objective values of (CRS) and (CP) are the same due to $\|\tilde{x}\|^2 = \tilde{y}$.

Next, we consider the case of $\|\tilde{x}\|^2 < \tilde{y}$. Let v be an eigenvector of matrix A associated with the minimum eigenvalue λ_1. By complementary slackness, $\mu = 0$. Then, equation (4) implies that $b^T v = 0$. Hence, there exists ζ such that $\|\tilde{x} + \zeta v\| = \sqrt{\tilde{y}}$ and $(\tilde{x} + \zeta v, \tilde{y})$ is still a solution to (CP). Using the same argument for the case of $\|\tilde{x}\|^2 = \tilde{y}$, we can show that $\tilde{x} + \zeta v$ is an optimal solution for (CRS). This completes the proof. □

Optimization problems of the form

$$\min_{x\in\mathbb{R}^n, y\in\mathbb{R}} g(x, y) + h(x, y), \tag{6}$$

where g is a smooth convex function and h is a non-smooth convex function, are called convex composite minimization problems. Letting $S = \{(x, y) : \|x\|^2 \leq y\}$, problem (CP) can be written as a convex composite minimization problem:

$$\min_{x\in\mathbb{R}^n, y\in\mathbb{R}} f_2(x, y) + \iota_S(x, y),$$

where ι_S is the indication function

$$\iota_S(x, y) := \begin{cases} 0 & \text{if } (x, y) \in S, \\ +\infty & \text{otherwise.} \end{cases}$$

General convex composite minimization problems (6) can be solved by many different algorithms such as APG, BBM, proximal quasi-Newton methods [10] and proximal Newton methods [23]. In order to apply these methods, we need to efficiently compute the proximal mapping with respect to the non-smooth function h in (6). In our situation, $h = \iota_S$ and hence the proximal mapping reduces to the orthogonal projection $\Pi_S(x, y)$ onto the closed convex set S, i.e.,

$$\Pi_S(x, y) = \operatorname*{argmin}_{(x',y')\in S} \|(x', y') - (x, y)\|^2.$$

The following theorem shows that such a projection can be done in $O(n)$ time.

Theorem 2.2. *For any point $(x_0, y_0) \in \mathbb{R}^n \times \mathbb{R}$, the projection $\Pi_S(x_0, y_0)$ is given by*

$$\Pi_S(x_0, y_0) = \begin{cases} (x_0, y_0) & \text{if } \|x_0\|^2 \le y_0, \\ \left(\dfrac{x_0}{1+\mu^*},\, y_0 + \dfrac{\mu^*}{2}\right) & \text{otherwise,} \end{cases} \tag{7}$$

where μ^* is the unique solution in the interval $[\max\{0, -2y_0\}, \infty)$ of the uni-variate cubic equation

$$\frac{1}{2}\mu^3 + (y_0+1)\mu^2 + (2y_0 + \frac{1}{2})\mu - x_0^T x_0 + y_0 = 0. \tag{8}$$

Proof. The case of $x_0^T x_0 \le y_0$ is trivial. So, we consider the case that $x_0^T x_0 > y_0$. The projection is defined as the solution to the (strongly convex) optimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n, y\in\mathbb{R}} \quad & \|(x,y) - (x_0,y_0)\|^2 \\ \text{subject to} \quad & \|x\|^2 \le y. \end{aligned} \tag{9}$$

The KKT optimality condition of problem (9) can be written as

$$2(x - x_0) + 2\mu x = 0, \tag{10}$$
$$2y - 2y_0 - \mu = 0, \tag{11}$$
$$\mu(\|x\|^2 - y) = 0,$$
$$\|x\|^2 \le y,$$
$$\mu \ge 0.$$

We have $x = \frac{x_0}{1+\mu}$ and $y = y_0 + \frac{\mu}{2}$ from (10) and (11), respectively. Suppose that $\mu = 0$. The optimality condition reduces to $x = x_0$ and $y = y_0$, which contradicts to the constraint $\|x\|^2 > y$ of problem (9). Therefore, we have $\mu > 0$ and hence $\|x\|^2 = y$ by complementary slackness. This leads to the uni-variate cubic equation

$$\left(\frac{x_0}{1+\mu}\right)^T \frac{x_0}{1+\mu} = y_0 + \frac{\mu}{2},$$

which is equivalent to (8) and implies, in particular, that $2y_0 + \mu \ge 0$. Define

$$h(\mu) = \frac{1}{2}\mu^3 + (y_0+1)\mu^2 + \left(2y_0 + \frac{1}{2}\right)\mu - x_0^T x_0 + y_0.$$

Since $2y_0 + \mu \ge 0$ and $\mu \ge 0$, the derivative h' satisfies

$$h'(\mu) = \frac{3}{2}\mu^2 + 2(y_0+1)\mu + \left(2y_0 + \frac{1}{2}\right) = \frac{1}{2}\mu^2 + (2y_0+\mu)\mu + (2y_0 + 2\mu) + \frac{1}{2} \ge \frac{1}{2}.$$

Hence $h(\mu)$ is strictly increasing on $[\max\{0,-2y_0\},\infty)$. Observing that $h(0)=y_0-x_0^Tx_0<0$, $h(-2y_0)=-x_0^Tx_0<0$ and $h(+\infty)=+\infty$, there exists exactly one root in the interval $[\max\{0,-2y_0\},\infty)$. Denote the solution of equation $h(\mu)=0$ in this interval by μ^*. Then, we have

$$x=\frac{x_0}{1+\mu^*} \quad \text{and} \quad y=y_0+\frac{\mu^*}{2},$$

which completes the proof. □

In practice, to find a root of the cubic equation (8) in the interval $[\max\{0,-2y_0\},\infty)$, we use a hybrid method obtained by combining the bisection method and the Newton's method. Numerically, our hybrid method is faster and more stable than the function `roots` in MATLAB. The projection can be done in runtime $O(n)$ as formulating the cubic equation cost $O(n)$ and solving the uni-variate cubic equation costs $O(1)$.

3 Complexity to achieve an ϵ-optimal solution of (CRS)

3.1 Another Equivalent Convex Reformulation

To achieve a theoretical complexity for solving convex composite optimization problem (6) with first-order methods such as APG [17], the function g is often required to have a Lipschitz continuous gradient on its domain $\mathrm{dom}(g)$, i.e., there exists a constant $L>0$ such that

$$\|\nabla g(x)-\nabla g(y)\|\le L\|x-y\|,\ \forall x,y\in\mathrm{dom}(g).$$

However, one can easily check that the gradient ∇f_2 of the objective f_2 of (CP) is not Lipschitz continuous at those points (x,y) with $y=0$. To remedy this, instead of (CP), we consider the following problem, which ensures y is bounded below from 0 by imposing an extra constrain $y\ge l$:

$$\begin{array}{ll}\min\limits_{x\in\mathbb{R}^n,y\in\mathbb{R}} & f_2(x,y)\\ \text{subject to} & \|x\|^2\le y,\ y\ge l,\end{array} \tag{BCP}$$

where $l=\lambda_1^2/\rho^2$ is a lower bound for y. Noting that the function $\frac{\rho}{3}y^{\frac{3}{2}}+\frac{\lambda_1}{2}y$ is decreasing when $\sqrt{y}\le-\lambda_1/\rho$, we have that any optimal solution $(\tilde{x},\tilde{y})$ of (CP) must satisfy $\tilde{y}\ge(-\lambda_1/\rho)^2=l$, and hence problem (BCP) has the same objective value and optimal solutions as problem (CP).

Problem (BCP) is again in the form of a convex composite minimization problem (6). Denote by $B=\{(x,y)\in\mathbb{R}^n\times\mathbb{R}:\|x\|^2\le y,y\ge l\}$ the feasible region of problem (BCP). The next theorem shows that the projection onto Π_B the feasible region B is again easily computable.

Theorem 3.1. *For any point $(x_0, y_0) \in \mathbb{R}^n \times \mathbb{R}$, the projection $\Pi_B(x_0, y_0)$ is given by*

$$\Pi_B(x_0, y_0) = \begin{cases} (x_1, y_1) & \text{if } y_1 \geq l, \\ (x_0, l) & \text{if } y_1 < l \text{ and } \|x_0\| < \sqrt{l}, \\ (\sqrt{l}x_0/\|x_0\|, l) & \text{otherwise,} \end{cases}$$

where $(x_1, y_1) = \Pi_S(x_0, y_0)$.

Proof. Let (x_2, y_2) be the projection of (x_0, y_0) onto B. If $y_1 \geq l$, then $(x_1, y_1) = \Pi_S(x_0, y_0)$ is the solution to the problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}} \quad & \|(x, y) - (x_0, y_0)\|^2 \\ \text{subject to} \quad & \|x\|^2 \leq y, \ y \geq l. \end{aligned}$$

Next, we consider the case of $y_1 < l$. In this case, we must have $y_2 = l$ since otherwise (x_2, y_2) is also the projection of (x_0, y_0) onto S, which contradicts with $y_1 < l$. Hence, x_2 is actually the solution to the problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|x - x_0\|^2 \\ \text{subject to} \quad & \|x\|^2 \leq l. \end{aligned}$$

We thus have the following two implications: if $\|x_0\| < \sqrt{l}$, then $x_2 = x_0$; and if $\|x_0\| \geq \sqrt{l}$, then $x_2 = \sqrt{l}x_0/\|x_0\|$. This completes the proof. □

For Theorem 3.1, the projection onto B is as cheap as the projection onto S because the former costs at most two more scalar comparisons, which are negligible, than the latter (note that $\|x_0\|$ is already computed in the computation of the projection onto S).

3.2 A Surrogate Problem

When the dimension n is high, the exact computation of the minimum eigenvalue is prohibitively expensive, if not impossible. For computational efficiency, an approximate eigenvalue is preferred when only an approximate solution of (CRS) is needed, which is often the case in practice. When an approximate minimum eigenvalue $\theta \approx \lambda_1$ is used in the problem (BCP), the objective $\frac{1}{2}x^T(A - \theta I)x + b^T x + \frac{\rho}{3}y^{\frac{3}{2}} + \frac{\theta}{2}y$ could be non-convex. Therefore, we need to slightly modify the problem (BCP). Let the approximate minimum eigenvalue θ satisfies $\lambda_1 \leq \theta \leq \lambda_1 + \epsilon$ and define $\eta := -\theta + \epsilon + \lambda_1 \geq 0$. Noting that $-\theta + \epsilon = -\lambda_1 + \eta$ (we will frequently use this equality in the following analysis), we obtain the following problem as a surrogate problem to (CRS):

$$\begin{aligned} \min_{x \in \mathbb{R}^n, y \in \mathbb{R}} \quad & f_3(x, y) := \frac{1}{2}x^T \left(A + (-\theta + \epsilon)I\right) x + b^T x + \frac{\rho}{3}y^{\frac{3}{2}} - \frac{-\theta + \epsilon}{2}y \\ \text{subject to} \quad & \|x\|^2 \leq y, \ y \geq \hat{l}, \end{aligned} \tag{SP}$$

where $\hat{l} = (-\theta + \epsilon)^2/\rho^2$. When $y \leq \hat{l}$, $\frac{\rho}{3}y^{\frac{3}{2}} - \frac{-\theta+\epsilon}{2}y$ is decreasing, and hence $\hat{l}$ is a lower bound for y. Define $\hat{B} := \{(x, y) : \|x\|^2 \leq y,\ y \geq \hat{l}\}$ which, by Theorem 3.1, admits an easily computable projection. Our theoretical convergence rate of solving problem (CRS) is based the surrogate problem (SP). In particular, we shall specialize the backtracking line search version of APG [3] to (SP) (see Algorithm 1) and analyze its convergence rate for this specific problem. The only difference of Algorithm 1 with the original APG is that we reset the final solution returned by APG (in Lines 8–12 of Algorithm 1) to achieve an equal or smaller objective value (see the proof in Theorem 3.4). From now on, let us denote (x^η, y^η) an optimal solution to (SP).

Remark: If we directly use the approximate minimum eigenvalue θ to replace the exact minimum eigenvalue λ_1 in (BCP), we get the following problem:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n, y\in\mathbb{R}} & \frac{1}{2}x^T(A-\theta I)x + b^T x + \frac{\rho}{3}y^{\frac{3}{2}} + \frac{\theta}{2}y \\ \text{subject to} & \|x\|^2 \leq y,\ y \geq l, \end{array} \tag{AP}$$

In Appendix A, we show that solving (AP) yields an approximate optimal solution to (CRS) if ϵ is sufficiently small, i.e., the eigenvalue computation is sufficiently accurate. We also show in Appendix A that either all the stationary points, which are approximate optimal solutions of (AP), share the same objective value, or there is a unique stationary point that is the optimal solution of (AP) if $-\theta > \bar{\lambda}$, where $\bar{\lambda}$ is some constant such that $\bar{\lambda} < -\lambda_1$. Note that when $\epsilon \leq -\lambda_1 - \bar{\lambda}$, we always have that $\theta < \lambda_1 + \epsilon < -\bar{\lambda}$ and hence that $-\theta > \bar{\lambda}$. However, the constant $\bar{\lambda}$ is unknown a priori and hence our formulation (AP) may have a non-optimal stationary point if we choose a θ that is not close enough to λ_1. This is why we focus on (SP) in this paper. Nevertheless, we will compare the empirical performance between (SP) and (AP) in the numerical section.

3.2.1 Approximate Computation of Eigen-pairs

To obtain an approximate eigen-pair, we recall the Lanczos method for approximately finding the minimum eigenvalue and its associated eigenvector [11]. The Lanczos method achieves a fast complexity bound for eigenvalue computation [15] and is an important component for proving complexity bounds for non-convex unconstrained optimization in the literature [1, 7, 19]. The specific result on the Lanczos method we need is the following lemma.

Lemma 3.2 ([15] and Lemma 9 in [19])**.** *Let H be a symmetric matrix satisfying $\|H\|_2 \leq U_H$ for some $U_H > 0$, where $\|\cdot\|_2$ denotes the operator 2-norm of a matrix. Suppose that the Lanczos procedure is applied to find the largest eigenvalue of $U_H I - H$ starting at a random vector distributed uniformly over the unit sphere. Then, for any $\epsilon > 0$ and $\delta \in (0, 1)$,*

Algorithm 1 APG for (SP)

Input: f_3, ∇f_3, $L_0 > 0, \xi > 1$, $\epsilon > 0$, $\theta < 0$ and some initial point α_0

Output: an approximate optimal solution (x_k, y_k)

choose $\beta_1 = \alpha_0$ and $t_1 = 1$

for $k = 1, 2, ..., k_{\max}$ **do**

find the smallest non-negative integer i_k such that $\bar{L} = \xi^{i_k} L_{k-1}$ and

$$f_3(\alpha_k) \geq f_3(\beta_k) + \nabla f_3(\beta_k)^T(\alpha_k - \beta_k) + \frac{\bar{L}}{2}\|\alpha_k - \beta_k\|^2,$$

where $\alpha_k = \Pi_{\hat{B}}(\beta_k - \frac{1}{\bar{L}}\nabla f_3(\beta_k))$

set $L_k = \xi^{i_k} L_{k-1}$

compute $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$

compute $\beta_{k+1} = \alpha_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\alpha_k - \alpha_{k-1})$

end for

if $\alpha_k(n+1) > \|\alpha_k(1:n)\|^2$ and $\sqrt{\alpha_k(n+1)} > (-\theta + \epsilon)/\rho$ **then**

set $x_k = \alpha_k(1:n)$ and $y_k = \max\{\|\alpha_k(1:n)\|^2, (-\theta+\epsilon)^2/\rho^2\}$

else

set $(x_k^T, y_k)^T = \alpha_k$

end if

there is a probability at least $1 - \delta$ *that the procedure outputs a unit vector* v *such that* $v^T H v \leq \lambda_1(H) + \epsilon$ *in at most* $\min\left\{n, \frac{\log(n/\delta^2)}{2\sqrt{2}}\sqrt{\frac{U_H}{\epsilon}}\right\}$ *iterations.*

3.2.2 Convergence Rate of APG for (SP)

We first collect some basic properties of APG.

Lemma 3.3 ([17, 3])**.** *Consider a function* $F(x) = g(x) + h(x)$*, where* g *is continuously differentiable, convex function with the gradient* ∇g *being* L*-Lipschitz continuous on its domain* $\mathrm{dom}(g)$ *and* h *is a proper, closed, and convex function that can possibly be non-smooth. Let* $\{x_k\}_{k=1}^{\infty}$ *be the sequence generated by APG. Then, we have*

$$F(x_k) - F^* \leq \frac{2\xi L\|x^* - x_0\|^2}{(k+1)^2},$$

where x^* *is an optimal solution and* F^* *is the optimal value of* $F(x)$*. Equivalently, in order to guarantee* $F(x_k) - F^* \leq \epsilon$*, we need at most* $k = \sqrt{2\xi L}\|x^* - x_0\|\epsilon^{-1/2} - 1$ *iterations.*

Restricting the objective function f_3 in (SP) to the set $\hat{B}$, the gradient ∇f_3 is then

γ-Lipschitz continuous, where

$$\gamma = \max\left\{\|A + (-\lambda_1 + \eta)I\|_2, \frac{\rho}{4\sqrt{\hat{l}}}\right\}. \tag{12}$$

Applying Lemma 3.3 to problem (SP) with $g = f_3$ and $h = \iota_{\hat{B}}$, we obtain that

$$f(x_k, y_k) - f^* \leq \epsilon$$

after at most $k = \sqrt{2\xi L}\sqrt{\|x^\eta - x_0\|^2 + \|y^\eta - y_0\|^2}\epsilon^{-1/2} - 1$ iterations.

The next theorem shows that with high probability, our algorithm returns an ϵ-approximate optimal solution to problem (CRS) using at most $O(\epsilon^{-1/2}\log(n/\delta))$ operations (including those in the approximate eigenpiar computation and the APG).

Theorem 3.4. *Let X^* be the optimal solution set, (x^η, y^η) be any optimal solution to problem* (SP)*, $R = \inf_{(x,y)\in X^*}\|(x,y) - (x_0,y_0)\|$ the initial distance to the optimal solution, (x^*, y^*) an optimal solution to problem* (BCP) *with $\|x^*\|^2 = y^*$ (which always exists) and (x_k, y_k) the solution returned by Algorithm 1, where $k \geq \sqrt{2\xi\gamma}R\epsilon^{-1/2} - 1$ and γ is as defined in* (12)*. Define*

$$\tilde{x} = \begin{cases} x_k & \text{if } \|x_k\|^2 = y_k, \\ x_k + tv & \text{otherwise,} \end{cases}$$

where v is an approximate eigenvector that satisfies $v^TAv \leq \lambda_1 + \epsilon$ and $\|v\| = 1$, and t is chosen such that $t(v^TAx_k + b^Tv + (-\lambda_1+\eta)x_k^Tv) \leq 0$ and $\|x_k + tv\|^2 = y_k$ (which also always exists). Then, we have

$$f_1(\tilde{x}) - f_1(x^*) \leq \epsilon + (-\lambda_1 + \eta)^2\epsilon/\rho^2 = O(\epsilon),$$

where f_1 is the objective function in (CRS)*. Furthermore, when the approximate eigen-pair is computed by the Lanczos method, the output is correct with probability at least $1 - \delta$ and the total number of matrix-vector products is at most*

$$\sqrt{2\xi\gamma}R\epsilon^{-1/2} - 1 + \frac{\log(n/\delta^2)}{2\sqrt{2}}\sqrt{\frac{\|A\|_2}{\epsilon}} = O(\epsilon^{-1/2}\log(n/\delta)).$$

Proof. Recall that f_2 and f_3 are the objective functions of (BCP) and (SP), respectively. For any optimal solution x^* of (CRS), $(x^*, \|x^*\|^2)$ is an optimal solution of (BCP). Therefore, an optimal solution (x^*, y^*) satisfying $\|x^*\|^2 = y^*$ always exists. Let $E_k = f_3(x_k, y_k) - f_3(x^\eta, y^\eta)$. From Lemma 3.3, we obtain that $f_3(\alpha_k) - f_3(x^\eta, y^\eta) < \epsilon$. If $\alpha_k(n+1) > \|\alpha_k(1:n)\|^2$ and $\sqrt{\alpha_k(n+1)} > (-\lambda_1 + \eta)/\rho$, we then go to Line 8 and Algorithm 1 outputs (x_k, y_k) instead of α_k. The y-part of the objective function f_3, i.e.,

$$\frac{\rho}{3}y^{\frac{3}{2}} - \frac{-\theta + \epsilon}{2}y,$$

is increasing when $\sqrt{y} \geq (-\theta+\epsilon)/\rho$, and hence Line 8 outputs a solution whose objective value is at most $f_3(\alpha_k)$. Hence $E_k \leq \epsilon$ for all $k \geq \sqrt{2\xi\gamma}R\epsilon^{-1/2} - 1$. Using this, we have

$$\begin{aligned} & f_3(x_k, y_k) - f_2(x^*, y^*) \\ =& f_3(x_k, y_k) - f_3(x^\eta, y^\eta) + f_3(x^\eta, y^\eta) - f_3(x^*, y^*) + f_3(x^*, y^*) - f_2(x^*, y^*) \\ \leq& E_k + 0 + \frac{\eta}{2}(\|x^*\|^2 - y^*) \\ =& E_k, \end{aligned} \tag{13}$$

where the inequality follows from the fact $f_3(x^\eta, y^\eta) - f_3(x^*, y^*) \leq 0$ because (x^η, y^η) is an optimal solution to (SP) and the last equality from the fact that $\|x^*\|^2 = y^*$.

If $\|x_k\|^2 = y_k$, we have that $\tilde{x} = x_k$ and hence that $f_3(x_k, y_k) = f_1(\tilde{x})$. Substituting $f_3(x_k, y_k) = f_1(\tilde{x})$ to (13) and noting that $f_1(x^*) = f_2(x^*, y^*)$, we have that $f_1(\tilde{x}) - f_1(x^*) \leq \epsilon$. If $\|x_k\|^2 < y_k$, we have $\tilde{x} = x_k + tv$ with $\|\tilde{x}\|^2 = y_k$ and hence

$$\begin{aligned} & f_1(\tilde{x}) - f_3(x_k, y_k) \\ =& \frac{1}{2}(x_k + tv)^T A(x_k + tv) + b^T(x_k + tv) + \frac{\rho}{3}\|(x_k + tv)\|^3 \\ & - \left(\frac{1}{2}x_k^T A x_k + b^T x_k + \frac{\rho}{3}y_k^{3/2} + \frac{-\lambda_1 + \eta}{2}(\|x_k\|^2 - y_k)\right) \\ =& tv^T A x_k + \frac{t^2}{2}v^T A v + tb^T v - \frac{-\lambda_1 + \eta}{2}(\|x_k\|^2 - \|x_k + tv\|^2) \\ =& t(v^T A x_k + b^T v) + \frac{t^2}{2}(\lambda_1 + \epsilon - \eta) - \frac{-\lambda_1 + \eta}{2}(-2tx_k^T v - t^2) \\ =& t(v^T A x_k + b^T v + (-\lambda_1 + \eta)x_k^T v) + \epsilon t^2/2 \\ \leq& \epsilon t^2/2, \end{aligned} \tag{14}$$

where the third equality follows from $v^T A v = \theta = \lambda_1 + \epsilon - \eta$ and the inequality from $t(v^T A x_k + b^T v + (-\lambda_1 + \eta)x_k^T v) \leq 0$. Note that a constant t satisfying such an inequality always exists. Indeed, since $\|x_k\|^2 < y_k$, the equation $\|x_k + tv\|^2 = y_k$ (in t) have two roots of opposite signs. Hence, we can always choose a t such that $t(v^T A x_k + b^T v + (-\lambda_1 + \eta)x_k^T v) \leq 0$. Using the inequalities (13), (14) and the fact that $f_1(x^*) = f_2(x^*, y^*)$, we get $f_1(\tilde{x}) - f_1(x^*) \leq \epsilon + \epsilon t^2/2$. Also, $\|x_k + tv\|^2 = y_k$ implies that $t \leq \|x_k\| + \sqrt{y_k} \leq 2\left(\frac{-\lambda_1 + \eta}{\rho}\right)$. Thus, we have

$$f_1(\tilde{x}) - f_1(x^*) \leq \epsilon + 2\epsilon\left(\frac{-\lambda_1 + \eta}{\rho}\right)^2 \leq \epsilon + 2\left(\frac{-\lambda_1 + \epsilon}{\rho}\right)^2 \epsilon,$$

where the last inequality follows from $0 \leq \eta := -\theta + \lambda_1 + \epsilon \leq \epsilon$.

From Lemma 3.2, with probability at least $1 - \delta$, such θ and v can be computed in at most $\frac{\log(n/\delta^2)}{2\sqrt{2}}\sqrt{\frac{\|A\|_2}{\epsilon}}$ iterations. And Lemma 3.3 shows that the number of operations required by Algorithm 1 is at most $\sqrt{2\xi\gamma}R\epsilon^{-1/2} - 1$. This completes the proof. □

4 Numerical experiments

In this section, we compare the numerical performance between our methods and the Krylov subspace method [8] on randomly generated instances. The problem instances are generated in the same manner as in [6], except that we replace both the original diagonal matrix A and vector b by Q^TAQ and Q^Tb, respectively to make the problem more computationally involved and less trivial. The matrix Q is a random block diagonal matrix (with n/K blocks) and each block is generated by the MATLAB command `orth(rand(K))` with K being a positive integer. Note that the random matrices generated in this manner are of full rank almost surely. As pointed out in [6], by construction, the optimal values are -1 for all cases. Problems with different dimensions n and different sparsity levels are tested. The sparsity of matrix A is then K/n, i.e., a proportion K/n of the total entries are nonzero. For fixed K and n, problems with different condition numbers κ and eigen-gaps γ (to be defined later) in the easy and hard cases were also tested, which are believed to strongly affect the hardness of problem (CRS) and the Krylov subspace method [6]. In the easy case, we tested problems with the condition number $\kappa = \frac{\lambda_n+\lambda^*}{\lambda_1+\lambda^*}$, which is an indicator for the hardness of the problem [6]. In the hard case, we tested problems with different eigen-gap $\gamma = \lambda_2 - \lambda_1$, where λ_2 is the second smallest eigenvalue of matrix A. All experiments were run on a Windows workshop with 16 Intel Xeon W-2145 cores (3.70GHz) and 64GB of RAM.

The approximate eigenvalue in formulating the surrogate problem was computed by the MATLAB function `eigs`. We found empirically that setting the tolerance (an input argument of the MATLAB function `eigs`) to be $5/\kappa$ in the easy case and 10^{-6} in the hard case yields a reasonable trade-off between accuracy and efficiency. Both (SP) and (AP) were tested. Besides APG, we have also applied BBM to solve the problems (SP) and (AP). For APG, we used a restarting strategy, which is a common method for speeding up the algorithm [18, 13]. For BBM, we used a simple line search rule to guarantee the decrease of the objective function values. As we know the optimal value is -1, we terminate our algorithm and the Krylov subspace method[2] if the objective value is less than `-1+1e-6`.

Tables 1–3 show the performance comparison of our methods and the Krylov subspace method. In the tables, `fval-opt` denotes the objective value accuracy, which is the objective value returned by the algorithm minus the optimal value; `iter` denotes the iteration number of each algorithm; `time` denotes the total time of each algorithm; $\texttt{time}_{\texttt{eig}}$ denotes the time cost for approximately computing the minimum eigenvalue, which is 0 for the Krylov subspace method. From Tables 1–2 , we see that in the easy case, our methods achieved the prescribed accuracy when $\kappa < 10^4$ and were a bit slower than the Krylov subspace method.

[2]The authors are indebted to Coralia Cartis for her kind sharing of the MATLAB codes for the Krylov subspace method.

From Tables 3 , we see that in the hard case, our methods performed much better than the Krylov subspace method in terms of both the solution quality and the CPU time. As our methods always outperform the Krylov subspace method in the hard case, we do not report more results for the hard case in this paper. In fact, the Krylov subspace method fails to find an approximate solution, while our methods always find a good approximate solution with an accuracy 10^{-6}. We also notice that, in both the easy and hard cases, APG are slightly better than BBM, especially for instances with a large condition number.

5 Conclusion

In this paper, we developed a novel approach for solving the problem (CRS). We first equivalently reformulate the problem (CRS) to a convex constrained optimization problem, where the feasible region admits an easy projection and the objective function is formed by using the minimum eigenvalue of the Hessian matrix. To circumvent the expensive cost due to the the exact computation of the minimum eigenvalue, we then constructed a surrogate problem which is again a convex constrained optimization problem with a feasible region that admits an easy projection and can be solved a variety of methods such as APG and BBM. Furthermore, we proved that an ϵ-approximate solution to (CRS) can be obtained in at most $\tilde{O}(\epsilon^{-1/2})$ matrix-vector multiplications if we use the Lanczos method for approximate eigenvalue computation and APG to approximately solve the surrogate problem. Numerical results showed that our methods are comparable to the Krylov subspace method in the easy case and significantly outperform the Krylov subspace method in the hard case. As future work, we would like to explore the empirical performance of our methods as a subproblem solver for the cubic regularization method. It is also interesting to see whether one can, based on our equivalent reformulation (CP) or the surrogate problem (SP), prove a complexity bound $\tilde{O}(\epsilon^{7/4})$ for finding a local minimum of smooth non-convex optimization problems using the cubic regularization method.

A Analysis for problem (AP)

The purpose of this appendix is to show that when the approximate minimum eigenvalue θ of A is close enough to the exact minimum eigenvalue λ_1 ($\lambda_1 \leq \theta < -\bar{\lambda}$ for some $\bar{\lambda}$ defined in the following paragraphs), problem (AP) can be used to construct an approximate solution to (CRS). Define $\hat{B} := \{(x, y) : \|x\|^2 \leq y,\ y \geq \hat{l}\}$. We claim that the problem (AP)

$K = 10,\ n = 2000$

Methods	$\kappa = 10$				$\kappa = 10^2$				$\kappa = 10^3$				$\kappa = 10^4$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
BB	5.5e-06	13.2	2.85e-03	1.95e-03	8.3e-06	52.4	3.58e-03	1.76e-03	6.9e-06	120.4	1.11e-02	5.79e-03	8.3e-06	338.8	1.71e-02	7.61e-03
BB+e	4.3e-06	15.0	2.70e-03	1.95e-03	6.3e-06	42.2	3.15e-03	1.76e-03	6.8e-06	123.8	1.08e-02	5.79e-03	6.4e-04	361.6	1.73e-02	7.61e-03
APG	4.1e-06	15.2	2.85e-03	1.95e-03	8.6e-06	55.4	3.77e-03	1.76e-03	8.6e-06	99.8	1.01e-02	5.79e-03	2.1e-03	276.4	1.60e-02	7.61e-03
APG+e	6.4e-06	15.2	2.71e-03	1.95e-03	8.0e-06	54.0	3.69e-03	1.76e-03	9.0e-06	108.8	9.56e-03	5.79e-03	3.2e-03	326.0	1.75e-02	7.61e-03
Krylov	6.8e-06	9.0	1.69e-03	0	7.7e-06	26.6	3.12e-03	0	9.5e-06	61.8	7.31e-03	0	8.9e-06	90.8	9.58e-03	0

$K = 10,\ n = 10000$

Methods	$\kappa = 10$				$\kappa = 10^2$				$\kappa = 10^3$				$\kappa = 10^4$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
BB	3.2e-06	11.0	1.31e-02	8.11e-03	7.4e-06	55.0	2.57e-02	8.22e-03	8.6e-06	168.0	1.11e-01	6.77e-02	8.5e-06	348.0	1.59e-01	7.49e-02
BB+e	8.4e-06	11.0	1.25e-02	8.11e-03	9.9e-06	57.0	2.28e-02	8.22e-03	7.5e-06	152.0	1.04e-01	6.77e-02	2.8e-06	415.0	1.69e-01	7.49e-02
APG	2.2e-06	29.0	2.02e-02	8.11e-03	1.0e-05	68.0	2.87e-02	8.22e-03	9.6e-06	120.0	1.01e-01	6.77e-02	9.9e-06	283.0	1.49e-01	7.49e-02
APG+e	6.9e-06	12.0	1.35e-02	8.11e-03	9.5e-06	70.0	2.78e-02	8.22e-03	9.4e-06	135.0	1.06e-01	6.77e-02	1.0e-05	324.0	1.59e-01	7.49e-02
Krylov	6.1e-06	9.0	4.05e-03	0	8.2e-06	27.0	9.30e-03	0	9.2e-06	68.0	2.17e-02	0	1.0e-05	146.0	4.55e-02	0

$K = 100,\ n = 100$

Methods	$\kappa = 10$				$\kappa = 10^2$				$\kappa = 10^3$				$\kappa = 10^4$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
BB	5.4e-06	13.6	1.07e-03	7.94e-04	6.6e-06	55.2	1.60e-03	8.09e-04	1.2e-05	104.1	2.35e-03	1.11e-03	5.5e-04	276.6	4.23e-03	1.39e-03
BB+e	4.6e-06	15.2	1.06e-03	7.94e-04	6.9e-06	56.3	1.53e-03	8.09e-04	7.8e-06	101.8	2.20e-03	1.11e-03	8.8e-04	286.4	4.21e-03	1.39e-03
APG	5.4e-06	17.2	1.13e-03	7.94e-04	8.1e-06	60.2	1.61e-03	8.09e-04	8.7e-06	96.5	2.21e-03	1.11e-03	1.4e-03	266.8	4.05e-03	1.39e-03
APG+e	5.8e-06	18.0	1.11e-03	7.94e-04	7.9e-06	60.1	1.60e-03	8.09e-04	8.4e-06	90.3	2.12e-03	1.11e-03	2.8e-03	295.4	4.38e-03	1.39e-03
Krylov	5.9e-06	9.0	7.99e-04	0	7.2e-06	21.9	1.80e-03	0	7.0e-06	33.1	2.32e-03	0	6.9e-06	34.4	2.15e-03	0

$K = 100,\ n = 1000$

Methods	$\kappa = 10$				$\kappa = 10^2$				$\kappa = 10^3$				$\kappa = 10^4$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
BB	5.2e-06	13.3	4.97e-03	3.20e-03	6.9e-06	69.8	9.76e-03	3.61e-03	8.8e-06	150.7	2.09e-02	8.10e-03	1.3e-05	246.1	3.94e-02	1.71e-02
BB+e	4.5e-06	15.1	4.92e-03	3.20e-03	6.7e-06	70.5	9.76e-03	3.61e-03	8.0e-06	142.6	2.00e-02	8.10e-03	1.1e-03	337.9	4.43e-02	1.71e-02
APG	5.0e-06	17.3	5.29e-03	3.20e-03	8.6e-06	86.3	1.13e-02	3.61e-03	9.2e-06	127.6	1.92e-02	8.10e-03	2.4e-03	288.6	4.09e-02	1.71e-02
APG+e	7.8e-06	16.7	5.21e-03	3.20e-03	8.6e-06	76.7	1.05e-02	3.61e-03	9.5e-06	120.8	1.85e-02	8.10e-03	3.2e-03	285.5	4.06e-02	1.71e-02
Krylov	6.8e-06	9.0	2.52e-03	0	7.9e-06	26.1	6.69e-03	0	9.0e-06	60.6	1.50e-02	0	9.1e-06	87.0	2.08e-02	0

Table 1: Comparison between Krylov subspace methods and our methods for solving (CRS) for different dimensions and sparsity levels in the **easy** case. Time unit: second.

$K = 100,\ n = 5000$

Methods	$\kappa = 10$				$\kappa = 10^2$				$\kappa = 10^3$				$\kappa = 10^4$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
BB	4.9e-06	12.6	2.74e-02	1.56e-02	7.1e-06	57.6	5.39e-02	1.75e-02	6.7e-06	143.2	1.26e-01	5.29e-02	2.6e-04	382.8	3.06e-01	1.21e-01
BB+e	5.0e-06	12.7	2.68e-02	1.56e-02	8.0e-06	58.2	4.95e-02	1.75e-02	5.7e-06	159.6	1.32e-01	5.29e-02	2.6e-04	390.9	3.07e-01	1.21e-01
APG	6.3e-06	14.1	2.77e-02	1.56e-02	8.3e-06	69.0	5.64e-02	1.75e-02	9.7e-06	115.0	1.13e-01	5.29e-02	3.9e-04	344.1	2.90e-01	1.21e-01
APG+e	4.0e-06	18.5	3.04e-02	1.56e-02	8.2e-06	57.8	5.07e-02	1.75e-02	9.4e-06	121.4	1.16e-01	5.29e-02	6.4e-04	305.8	2.71e-01	1.21e-01
Krylov	6.4e-06	9.0	8.01e-03	0	8.4e-06	26.5	2.37e-02	0	9.4e-06	68.6	6.02e-02	0	9.5e-06	134.9	1.16e-01	0

$K = 100,\ n = 10000$

Methods	$\kappa = 10$				$\kappa = 10^2$				$\kappa = 10^3$				$\kappa = 10^4$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
BB	6.4e-06	13.1	4.91e-02	2.70e-02	7.2e-06	45.0	8.61e-02	2.83e-02	8.1e-06	144.1	2.68e-01	1.02e-01	1.0e-05	383.1	7.73e-01	3.38e-01
BB+e	4.5e-06	13.0	4.86e-02	2.70e-02	8.6e-06	47.8	8.87e-02	2.83e-02	8.3e-06	153.0	2.78e-01	1.02e-01	1.3e-04	361.7	7.46e-01	3.38e-01
APG	5.1e-06	17.1	5.50e-02	2.70e-02	7.7e-06	46.1	8.92e-02	2.83e-02	8.9e-06	130.9	2.57e-01	1.02e-01	9.0e-04	381.9	7.80e-01	3.38e-01
APG+e	5.6e-06	18.2	5.68e-02	2.70e-02	7.0e-06	47.3	9.08e-02	2.83e-02	9.5e-06	123.9	2.50e-01	1.02e-01	9.4e-06	380.9	7.76e-01	3.38e-01
Krylov	6.5e-06	9.0	1.76e-02	0	8.2e-06	26.1	5.12e-02	0	9.5e-06	69.9	1.37e-01	0	9.7e-06	147.3	2.91e-01	0

$K = 1000,\ n = 1000$

Methods	$\kappa = 10$				$\kappa = 10^2$				$\kappa = 10^3$				$\kappa = 10^4$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
BB	6.0e-06	12.3	4.33e-02	2.38e-02	6.3e-06	59.9	9.52e-02	2.54e-02	7.8e-06	132.9	1.93e-01	5.60e-02	7.0e-04	332.9	4.57e-01	1.16e-01
BB+e	7.0e-06	13.9	4.46e-02	2.38e-02	5.8e-06	55.3	8.77e-02	2.54e-02	7.9e-06	127.0	1.87e-01	5.60e-02	9.8e-04	283.1	4.05e-01	1.16e-01
APG	5.0e-06	15.8	4.76e-02	2.38e-02	7.5e-06	61.1	9.46e-02	2.54e-02	8.9e-06	102.7	1.65e-01	5.60e-02	1.2e-03	312.1	4.37e-01	1.16e-01
APG+e	5.6e-06	16.4	4.79e-02	2.38e-02	7.7e-06	57.1	9.07e-02	2.54e-02	9.5e-06	109.3	1.71e-01	5.60e-02	1.8e-03	358.6	4.83e-01	1.16e-01
Krylov	6.4e-06	9.0	1.69e-02	0	8.0e-06	26.5	5.11e-02	0	9.0e-06	59.9	1.13e-01	0	8.8e-06	81.4	1.58e-01	0

$K = 1000,\ n = 5000$

Methods	$\kappa = 10$				$\kappa = 10^2$				$\kappa = 10^3$				$\kappa = 10^4$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
BB	4.9e-06	13.6	2.86e-01	1.61e-01	6.0e-06	54.9	5.46e-01	1.66e-01	8.1e-06	144.1	1.45e+00	5.33e-01	9.7e-06	367.2	3.42e+00	1.15e+00
BB+e	5.8e-06	13.7	2.83e-01	1.61e-01	7.6e-06	51.3	5.19e-01	1.66e-01	7.3e-06	163.1	1.55e+00	5.33e-01	7.8e-04	394.9	3.58e+00	1.15e+00
APG	5.7e-06	15.6	2.99e-01	1.61e-01	8.8e-06	54.5	5.44e-01	1.66e-01	9.2e-06	120.8	1.31e+00	5.33e-01	9.9e-06	333.7	3.22e+00	1.15e+00
APG+e	5.4e-06	17.4	3.11e-01	1.61e-01	8.1e-06	55.8	5.49e-01	1.66e-01	9.4e-06	128.0	1.35e+00	5.33e-01	5.3e-04	328.1	3.18e+00	1.15e+00
Krylov	6.6e-06	9.0	1.02e-01	0	8.3e-06	26.1	2.98e-01	0	9.4e-06	69.2	7.92e-01	0	9.6e-06	131.7	1.52e+00	0

$K = 1000,\ n = 10000$

Methods	$\kappa = 10$				$\kappa = 10^2$				$\kappa = 10^3$				$\kappa = 10^4$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
BB	4.9e-06	12.7	5.21e-01	2.91e-01	7.5e-06	51.4	1.02e+00	3.01e-01	7.7e-06	140.1	2.68e+00	8.24e-01	1.5e-05	419.7	8.25e+00	2.95e+00
BB+e	5.2e-06	13.5	5.27e-01	2.91e-01	8.3e-06	46.1	9.51e-01	3.01e-01	8.9e-06	131.8	2.56e+00	8.24e-01	1.5e-05	428.3	8.33e+00	2.95e+00
APG	6.1e-06	14.1	5.50e-01	2.91e-01	7.5e-06	51.6	1.04e+00	3.01e-01	9.3e-06	107.8	2.26e+00	8.24e-01	1.4e-05	364.3	7.58e+00	2.95e+00
APG+e	5.5e-06	15.3	5.65e-01	2.91e-01	8.6e-06	50.1	1.01e+00	3.01e-01	9.3e-06	105.3	2.23e+00	8.24e-01	2.5e-04	364.5	7.58e+00	2.95e+00
Krylov	6.6e-06	9.0	2.02e-01	0	8.1e-06	26.5	6.12e-01	0	9.3e-06	65.4	1.53e+00	0	9.6e-06	149.5	3.47e+00	0

Table 2: Comparison between Krylov subspace methods and our methods for solving (CRS) for different dimensions and sparsity levels in the **easy** case. Time unit: second.

$K = 1000,\ n = 10000$																
Methods	$\gamma = 10^{-1}$				$\gamma = 10^{-2}$				$\gamma = 10^{-3}$				$\gamma = 10^{-4}$			
	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}	fval-opt	iter	time	time_{eig}
`BB`	8.2e-06	8.8	6.49e-01	4.47e-01	7.7e-06	27.1	1.52e+00	1.09e+00	8.6e-06	53.7	3.61e+00	2.83e+00	8.8e-06	63.4	8.07e+00	7.18e+00
`BB+e`	8.2e-06	8.8	6.41e-01	4.47e-01	7.6e-06	27.4	1.52e+00	1.09e+00	8.0e-06	54.7	3.59e+00	2.83e+00	9.6e-06	62.2	8.04e+00	7.18e+00
`APG`	5.6e-06	7.6	6.43e-01	4.47e-01	7.5e-06	16.0	1.39e+00	1.09e+00	9.2e-06	36.3	3.39e+00	2.83e+00	9.9e-06	41.1	7.79e+00	7.18e+00
`APG+e`	5.6e-06	7.6	6.41e-01	4.47e-01	7.5e-06	16.0	1.39e+00	1.09e+00	9.7e-06	35.8	3.38e+00	2.83e+00	1.0e-05	39.1	7.78e+00	7.18e+00
`Krylov`	8.8e-01	455.9	1.07e+01	0	1.9e+01	175.1	4.12e+00	0	3.7e-03	442.5	1.04e+01	0	1.1e-03	500.0	1.17e+01	0

Table 3: Comparison between Krylov subspace methods and our methods for solving (CRS) for different dimensions and sparsity levels in the **hard** case. Time unit: second.

simplifies to

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n, y\in\mathbb{R}} & \frac{1}{2}x^T(A-\theta I)x + b^T x + \frac{\rho}{3}y^{\frac{3}{2}} + \frac{\theta}{2}y \\ \text{subject to} & \|x\|^2 \le y, \end{array} \tag{AP$_0$}$$

i.e., the constraint $y \ge l$ is redundant. This is because when $y \le l$ in (AP$_0$), $\frac{\rho}{3}y^{\frac{3}{2}} + \frac{\theta}{2}y$ is decreasing in y, and thus the optimal solution of (AP$_0$) must satisfy $y \ge l$. We consider the following two cases.

The hard case: Recall the optimality condition (2) for (CRS). Note that $\|(A+\lambda I)^\dagger b\|^2 - \lambda^2/\rho^2 \le 0$ and $\|(A+\lambda I)^\dagger b\|^2 - \lambda^2/\rho^2$ is a decreasing function in λ, where $(\cdot)^\dagger$ denotes the *Moore–Penrose pseudoinverse*, when $\lambda \ge -\lambda_1$ because we are in the hard case [8]. First consider the case that $\|(A-\lambda_1 I)^\dagger b\|^2 - \lambda_1^2/\rho^2 < 0$. Let $\bar{\lambda}$ be the largest $\lambda \in [0, -\lambda_1)$ such that $\|(A+\lambda I)^\dagger b\|^2 - \lambda^2/\rho^2 = 0$, if it exists. If such $\bar{\lambda}$ does not exist, we set $\bar{\lambda} = 0$. Using the fact $\|(A+\lambda I)^\dagger b\|^2 - \lambda^2/\rho^2 < 0$ for $\lambda \ge -\lambda_1$, we have

$$\|(A+\lambda I)^\dagger b\|^2 - \frac{\lambda^2}{\rho^2} < 0, \ \forall \lambda > \bar{\lambda}. \tag{15}$$

Suppose that $\lambda_1 \le \theta < -\bar{\lambda}$ and that (x^θ, y^θ) is an optimal solution of (AP$_0$). Let μ be the Lagrange multiplier corresponding to the constraint $\left\|x^\theta\right\|^2 \le y^\theta$. Then, the KKT condition of (AP$_0$) implies $Ax^\theta - \theta x^\theta + b + 2\mu x^\theta = 0$ and $\rho\sqrt{y^\theta}/2 + \theta/2 - \mu = 0$. Due to $-\theta + 2\mu > \bar{\lambda}$, we have $\|x^\theta\|^2 < y^\theta, \forall \mu \ge 0$ from (15). Using the complementary slackness $\mu(\|x^\theta\|^2 - y^\theta) = 0$, we have that $\mu = 0$. Thus, every possible stationary point of (AP$_0$) can be written as $(x^\theta, y^\theta) = ((A-\theta I)^\dagger b + tv, \ (-\theta/\rho)^2)$, where t is a scalar satisfying $\|x^\theta + tv\| \le \sqrt{y^\theta}$. This in turn yields an approximate optimal solution $x^\theta + tv$ to (CRS), where one should note that different t yields the same objective value. Next, we consider the remaining case that $\|(A+\lambda_1 I)^\dagger b\|^2 - \lambda_1^2/\rho^2 = 0$. This case is similar to the easy case, where we can recover an optimal solution if $\theta \in [\lambda_1, -\bar{\lambda})$, where $\bar{\lambda}$ is the largest $\lambda \in [0, -\lambda_1)$ such that $\|(A+\lambda I)^\dagger b\|^2 - \lambda^2/\rho^2 = 0$. (If such $\bar{\lambda}$ does not exist, we take $\bar{\lambda} = 0$.) The analysis is similar to the easy case below and hence omitted here.

The easy case: Recall that in the easy case we have a unique optimal solution x^* satisfying $\rho\|x^*\| > -\lambda_1$ and $x^* = (A + \rho\|x^*\| I)^{-1} b$; see, e.g., Theorem 3.1 in [8]. Let $\bar{\lambda}$ be the largest $\lambda \in [0, -\lambda_1)$ such that $h(\lambda) := \|(A+\lambda I)^\dagger b\|^2 - \lambda^2/\rho^2 = 0$, if it exists. If such $\bar{\lambda}$ does not exist, we take $\bar{\lambda} = 0$. Then for all $\lambda \in (\bar{\lambda}, -\lambda_1)$, $h(\lambda) > 0$ as $\lim_{\lambda \to -\lambda_1} h(\lambda) = +\infty$. This, together with the definition of $\bar{\lambda}$ and the fact that $h(\lambda)$ is a decreasing function on $(-\lambda_1, +\infty)$, implies that there is only one point, denoting $\tilde{\lambda}$ in $(\bar{\lambda}, +\infty)$ satisfying $\|(A+\lambda I)^\dagger b\|^2 - \lambda^2/\rho^2 = 0$. Let $\tilde{x} = (A + \tilde{\lambda} I)^\dagger b$. The optimality condition (2) implies that $\tilde{x}$ is the unique optimal solution of (CRS). We again suppose that $\lambda_1 \le \theta < -\bar{\lambda}$. If $\theta = \lambda_1$, (AP$_0$) reduces to the exact reformulation (CP). Next, we consider the case that $\lambda_1 < \theta < -\bar{\lambda}$. Assuming that $\mu = 0$, the inequality $\lambda_1 < \theta < -\bar{\lambda}$ implies that $\|x^\theta\| = \|(A-\theta I)^\dagger b\| > \sqrt{y^\theta} = -\theta/\rho$, which

violates the constraint $\|x\|^2 \leq y$. Hence, we must have $\mu > 0$, and thus we always have $\|x^\theta\| = \sqrt{y^\theta}$, i.e., $\|(A + (-\theta + \mu)I)^\dagger b\| = (-\theta + \mu)/\rho$. This implies $-\theta + \mu = \lambda^*$. That is, we recover the optimal solution if $\theta < -\bar{\lambda}$.

References

[1] N. Agarwal, Z. Allen-Zhu, B. Bullins, E. Hazan, and T. Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199. ACM, 2017.

[2] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] E. Birgin and J. Martínez. A Newton-like method with mixed factorizations and cubic regularization for unconstrained minimization. *Computational Optimization and Applications*, 73(3):707–753, 2019.

[5] Y. Carmon and J. Duchi. Gradient descent finds the cubic-regularized nonconvex Newton step. *SIAM Journal on Optimization*, 29(3):2146–2178, 2019.

[6] Y. Carmon and J. C. Duchi. Analysis of Krylov subspace solutions of regularized nonconvex quadratic problems. In *Advances in Neural Information Processing Systems*, pages 10705–10715, 2018.

[7] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

[8] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011.

[9] O. E. Flippo and B. Jansen. Duality and sensitivity in nonconvex quadratic optimization over an ellipsoid. *European Journal of Operational Research*, 94(1):167–178, 1996.

[10] H. Ghanbari and K. Scheinberg. Proximal quasi-Newton methods for regularized convex optimization with linear and accelerated sublinear convergence rates. *Computational Optimization and Applications*, 69(3):597–627, 2018.

[11] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.

[12] N. Ho-Nguyen and F. Kilinc-Karzan. A second-order cone based approach for solving the trust-region subproblem and its variants. *SIAM Journal on Optimization*, 27(3):1485–1512, 2017.

[13] N. Ito, A. Takeda, and K.-C. Toh. A unified formulation and fast accelerated proximal gradient method for classification. *The Journal of Machine Learning Research*, 18(1):510–558, 2017.

[14] R. Jiang and D. Li. Novel reformulations and efficient algorithms for the generalized trust region subproblem. *SIAM Journal on Optimization*, 29(2):1603–1633, 2019.

[15] J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992.

[16] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[17] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[18] B. O'Donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015.

[19] C. W. Royer and S. J. Wright. Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1448–1477, 2018.

[20] J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010.

[21] J. Wang and Y. Xia. A linear-time algorithm for the trust region subproblem based on hidden convexity. *Optimization Letters*, 11(8):1639–1646, 2017.

[22] M.-C. Yue, Z. Zhou, and A. Man-Cho So. On the quadratic convergence of the cubic regularization method under a local error bound condition. *SIAM Journal on Optimization*, 29(1):904–932, 2019.

[23] M.-C. Yue, Z. Zhou, and A. M.-C. So. A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo–Tseng error bound property. *Mathematical Programming*, 174(1-2):327–358, 2019.