

Accelerated first-order methods for a class of semidefinite programs

Alex L. Wang ^{*} Fatma Kılınç-Karzan [†]

June 1, 2022

Abstract

This paper introduces a new storage-optimal first-order method (FOM), CertSDP, for solving a special class of semidefinite programs (SDPs) to high accuracy. The class of SDPs that we consider, the *exact QMP-like SDPs*, is characterized by low-rank solutions, *a priori* knowledge of the restriction of the SDP solution to a small subspace, and standard regularity assumptions such as strict complementarity. Crucially, we show how to use a *certificate of strict complementarity* to construct a low-dimensional strongly convex minimax problem whose optimizer coincides with a factorization of the SDP optimizer. From an algorithmic standpoint, we show how to construct the necessary certificate and how to solve the minimax problem efficiently. We accompany our theoretical results with preliminary numerical experiments suggesting that CertSDP significantly outperforms current state-of-the-art methods on large sparse exact QMP-like SDPs.

1 Introduction

Semidefinite programs (SDPs) are among the most powerful tools that optimizers have for tackling both convex *and* nonconvex problems. In the former direction, SDPs are routinely used to model convex optimization problems that arise in a variety of applications such as robust optimization, engineering, and robotics [8, 60]. In the latter direction, many results over the last thirty years have shown that SDPs perform provably well as convex relaxations of certain nonconvex optimization problems; see [8, 15, 26, 51] and references therein. As examples, exciting results in phase retrieval [15] and clustering [1, 39, 52] show that these nonconvex problems have exact SDP relaxations with high probability under certain random models. More abstractly, a line of recent work [4, 6, 7, 11, 13, 14, 30, 33, 36, 58, 63, 64] has investigated general conditions under which *exactness* holds between nonconvex quadratically constrained quadratic programs (QCQPs) or quadratic matrix programs (QMPs) and their standard SDP relaxations.

Despite the expressiveness and strong theoretical guarantees of SDPs, they have seen limited application in practice and have a reputation of being “prohibitively expensive,” especially for large-scale applications. Indeed, standard methods for solving SDPs, such as the interior point methods (IPMs) [2, 46], scale poorly with problem dimension due to both their expensive iterations and also significant memory needs. See [68, Section 8.1] for a more thorough discussion.

In this paper, we show how to derive highly efficient (in iteration complexity, per-iteration-cost, and memory usage) first-order methods (FOMs) for solving general SDPs that admit a desirable *exactness* property. Our developments are inspired by recent results on linearly convergent FOMs for the trust-region subproblem (TRS) and the generalized trust-region subproblem (GTRS) [16, 65]

^{*}Carnegie Mellon University, Pittsburgh, PA, USA, alw1@andrew.cmu.edu

[†]Carnegie Mellon University, Pittsburgh, PA, USA, fkilinc@andrew.cmu.edu

that operate in the original problem space. We briefly discuss these problems now to motivate our assumptions and our problem class. We will discuss this literature in further detail in Section 1.3.

The TRS [41] seeks to minimize a general quadratic objective over the unit ball. The GTRS [40] then replaces the unit ball constraint with a general quadratic equality or inequality constraint:

$$\inf_{x \in \mathbb{R}^{n-1}} \{q_{\text{obj}}(x) : q_1(x) = 0\}$$

(presented as an equality constraint). Here, both q_{obj} and q_1 may be nonconvex, but it is standard to assume that there exists $\hat{\gamma} \in \mathbb{R}$ such that $q_{\text{obj}} + \hat{\gamma}q_1$ is a *strongly convex* quadratic function. Under this assumption, the S-lemma [24] guarantees that the GTRS has an exact SDP relaxation in the following sense: Let M_{obj}, M_1 be symmetric matrices such that $q_{\text{obj}}(x) = \begin{pmatrix} x \\ 1 \end{pmatrix}^\top M_{\text{obj}} \begin{pmatrix} x \\ 1 \end{pmatrix}$ and $q_1(x) = \begin{pmatrix} x \\ 1 \end{pmatrix}^\top M_1 \begin{pmatrix} x \\ 1 \end{pmatrix}$. Then, equality holds between the GTRS, its SDP relaxation, and the dual of the SDP relaxation:

$$\begin{aligned} & \min_{x \in \mathbb{R}^{n-1}} \{q_{\text{obj}}(x) : q_1(x) = 0\} \\ &= \min_{Y \in \mathbb{S}^n} \left\{ \langle M_{\text{obj}}, Y \rangle : \begin{array}{l} \langle M_1, Y \rangle = 0 \\ Y = \begin{pmatrix} * & * \\ * & 1 \end{pmatrix} \succeq 0 \end{array} \right\} \\ &= \sup_{\gamma \in \mathbb{R}, t \in \mathbb{R}} \left\{ t : M_{\text{obj}} + \gamma M_1 - t \begin{pmatrix} 0_{n-1} & \\ & 1 \end{pmatrix} \succeq 0 \right\}. \end{aligned}$$

Here, \mathbb{S}^n is the vector space of $n \times n$ symmetric matrices, the inner product $\langle M, Y \rangle$ is defined as $\langle M, Y \rangle := \text{tr}(M^\top Y)$ and $Y \succeq 0$ indicates that Y is positive semidefinite (PSD).

In particular, the SDP relaxation of the GTRS has an optimal solution Y^* with rank one. Furthermore, we know the value of $(Y^*)_{n,n} = 1$ before we even solve the SDP relaxation. We will think of this as a *a priori* knowledge of the restriction of Y^* to a subspace of dimension $\text{rank}(Y^*)$.

Despite the fact that the SDP relaxation solves the GTRS exactly, the large computational cost of solving SDPs has spurred an extensive line of work developing new algorithms for the GTRS (that avoid explicitly solving large SDPs). Most relatedly, Wang et al. [65] assume that the dual SDP is solvable and that there exists an optimal dual solution (γ^*, t^*) such that $M_{\text{obj}} + \gamma^* M_1 - t^* \begin{pmatrix} 0_{n-1} & \\ & 1 \end{pmatrix}$ has rank $n-1$. This assumption holds generically for GTRS problems conditioned on strict feasibility of the dual SDP and can be phrased as assuming strict complementarity [3] between the dual SDP and the desired rank-one solution Y^* . Wang et al. [65] then showed that it is possible to construct a strongly convex reformulation of the GTRS in the original space using a sufficiently accurate estimate of γ^* .

In our study, we will examine general SDPs satisfying similar structural assumptions and design an efficient *storage-optimal* FOM to solve them. In this respect, our approach also extends a recent line of work [23, 25, 53, 68] towards developing storage-optimal FOMs for SDPs possessing low-rank solutions. We discuss storage optimality in SDP algorithms in Section 1.3.

1.1 Problem setup and assumptions

Consider an SDP in standard form and its dual:

$$\begin{aligned} \inf_{Y \in \mathbb{S}^n} \left\{ \langle M_{\text{obj}}, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle + d_i = 0, \forall i \in [m] \\ Y \succeq 0 \end{array} \right\} & \quad (\text{SDP}) \\ \geq \sup_{\gamma \in \mathbb{R}^m} \left\{ d^\top \gamma : M_{\text{obj}} + \sum_{i=1}^m \gamma_i M_i \succeq 0 \right\}. \end{aligned}$$

For notational convenience, we let $d_{\text{obj}} := 0$ and define $M(\gamma) := M_{\text{obj}} + \sum_{i=1}^m \gamma_i M_i$ and $d(\gamma) := d_{\text{obj}} + \sum_{i=1}^m \gamma_i d_i$.

In this paper, inspired by the structural properties of the GTRS that make it amenable to highly efficient FOMs, we will work under two major assumptions. First, we will assume (Assumption 1) that the primal and dual SDPs are both solvable, strong duality holds, and there exist primal and dual optimal solutions $Y^* \in \mathbb{S}^n$ and $\gamma^* \in \mathbb{R}^m$ such that $\text{rank}(Y^*) = k$ and $\text{rank}(M(\gamma^*)) = n - k$. The assumption that $\text{rank}(Y^*) + \text{rank}(M(\gamma^*)) = n$ is referred to as *strict complementarity* and is known to hold generically conditioned on primal and dual attainability [3].

Second, we will assume (Assumption 2) that the optimal primal solution Y^* is known *a priori* on some k -dimensional subspace W^\perp , on which it is positive definite. This assumption is inspired by QCQP and QMP applications [6, 54, 64]: Recall that the standard SDP relaxation [54] of an equality-constrained QCQP (in the variable $x \in \mathbb{R}^{n-1}$) is given by

$$\begin{aligned} \inf_{x \in \mathbb{R}^{n-1}} \left\{ \begin{pmatrix} x \\ 1 \end{pmatrix}^\top M_{\text{obj}} \begin{pmatrix} x \\ 1 \end{pmatrix} : \begin{pmatrix} x \\ 1 \end{pmatrix}^\top M_i \begin{pmatrix} x \\ 1 \end{pmatrix} = 0, \forall i \in [m] \right\} \\ \geq \inf_{Y \in \mathbb{S}^n} \left\{ \langle M_{\text{obj}}, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle = 0, \forall i \in [m] \\ Y = \begin{pmatrix} * & * \\ * & 1 \end{pmatrix} \succeq 0 \end{array} \right\}. \end{aligned}$$

Thus, the optimal solution (in fact, any feasible solution) to the SDP will have a 1 in the bottom-right corner. Taking W to be the subspace corresponding to the first $(n-1)$ -coordinates of \mathbb{R}^n , we have that the restriction of Y^* to W^\perp is known *a priori* and is positive definite. Similarly, the standard SDP relaxation [6] of an equality-constrained QMP (in the variable $X \in \mathbb{R}^{(n-k) \times k}$) is given by

$$\begin{aligned} \inf_{X \in \mathbb{R}^{(n-k) \times k}} \left\{ \text{tr} \left(\begin{pmatrix} X \\ I_k \end{pmatrix}^\top M_{\text{obj}} \begin{pmatrix} X \\ I_k \end{pmatrix} \right) : \text{tr} \left(\begin{pmatrix} X \\ I_k \end{pmatrix}^\top M_i \begin{pmatrix} X \\ I_k \end{pmatrix} \right) = 0, \forall i \in [m] \right\} \\ \geq \inf_{Y \in \mathbb{S}^n} \left\{ \langle M_{\text{obj}}, Y \rangle : \begin{array}{l} \langle M_i, Y \rangle = 0, \forall i \in [m] \\ Y = \begin{pmatrix} * & * \\ * & I_k \end{pmatrix} \succeq 0 \end{array} \right\}. \end{aligned}$$

Taking W to be the subspace corresponding to the first $(n-k)$ coordinates of \mathbb{R}^n , we have that the restriction of Y^* to W^\perp is known *a priori* and positive definite.

We will refer to SDPs where Assumptions 1 and 2 hold as *rank- k exact QMP-like SDPs* or *k -exact SDPs* for short.

1.2 Overview and outline of the paper

In this paper, we develop a new FOM for rank- k exact QMP-like SDPs. This FOM enjoys low iteration complexity, simple iterative subprocedures, storage optimality, and strong numerical

performance. A summary of our contributions, along with an outline of the remainder of this paper, is as follows. For the sake of presentation, we will assume that W corresponds to the first $n - k$ coordinates of \mathbb{R}^n in the following outline.

- We close this section by discussing thematically related work in storage-optimal or storage-efficient FOMs for solving SDPs and FOMs for solving the GTRS. We then discuss some work on acceleration within FOMs with inexact prox oracles and FOMs for saddle-point problems as these are related to our techniques.
- In Section 2, we show how to reformulate a k -exact SDP as a *strongly convex* quadratic matrix minimax problem (QMMP) using a *certificate of strict complementarity* (see Definition 2). There are two key ideas here: First, in the setting of k -exact SDPs, we may parameterize the rank- k matrices in \mathbb{S}_+^n which agree with the restriction of Y^* to W^\perp as

$$Y(X) := \begin{pmatrix} XX^\top & X(Z^*)^{1/2} \\ (Z^*)^{1/2}X^\top & Z^* \end{pmatrix},$$

where $Z^* \succ 0$ is the known restriction of Y^* to W^\perp and $X \in \mathbb{R}^{(n-k) \times k}$ is unknown. The task of recovering Y^* then reduces to the task of recovering X^* . We replace the variable $Y \in \mathbb{S}_+^n$ with the parameterization $Y(X)$ in the primal SDP to derive a *nonconvex* QMP in the variable X whose optimizer is X^* . This first step can be compared to the Burer–Monteiro reformulation (see Remark 2). The second key idea then shows that this nonconvex QMP can be further reformulated into a strongly convex QMMP (QMMP $_{\mathcal{U}}$) given a certificate of strict complementarity $\mathcal{U} \subseteq \mathbb{R}^m$. Theorem 1 verifies that the minimax problem

$$\min_{X \in \mathbb{R}^{(n-k) \times k}} \max_{\gamma \in \mathcal{U}} \left(\langle M(\gamma), Y(X) \rangle + d(\gamma) \right) \quad (1)$$

has X^* as its unique optimizer and $\text{Opt}_{(\text{SDP})}$ as its optimal value.

- In Section 3, we derive a two-level accelerated FOM for solving strongly convex QMMPs of the form (1). Due to the minimax structure of (1), we focus on Nesterov’s optimal method for strongly convex minimax problems [45, Algorithm 2.3.13]. This algorithm relies on a prox-map (see Definition 3) computation in each iteration, and its analysis assumes that prox-map is given by an explicit expression or can be computed exactly. In our setting, the prox-map will not admit a closed-form expression in general. Instead, we will treat the prox-map as an optimization problem in its own right and solve it via an inner FOM. Therefore, we suggest CautiousAGD (Algorithm 1), a new variant of [45, Algorithm 2.3.13] that handles inexact computations in the prox-map procedure. We extend the original estimating sequences analysis of [45, Algorithm 2.3.13] to prove bounds on the accuracy required in each individual prox-map computation to recover an accelerated linear convergence rate in terms of outer iterations (see Theorem 3). In our case, the prox-map can be computed efficiently using an inner loop via the strongly convex excessive gap technique [45, Chapter 6.2]. In all, CautiousAGD computes an ϵ -optimal solution of a QMMP after $O(\log(\epsilon^{-1}))$ outer iterations and $O(\epsilon^{-1/2})$ total inner iterations.
- In Section 4, we show how to combine any method for producing iterates $\gamma^{(i)} \rightarrow \gamma^*$ with CautiousAGD to construct a certificate of strict complementarity. Combined with Algorithm 1, this completes the description of our new FOM, CertSDP (Algorithm 2), for rank- k exact QMP-like SDPs. Informally, we show that CertSDP returns an ϵ -optimal solution to the underlying SDP after performing a fixed (i.e., independent of ϵ) number of iterations of

$\gamma^{(i)} \rightarrow \gamma^*$ plus either $O(\log(\epsilon^{-1}))$ outer iterations or $O(\epsilon^{-1})$ inner iterations in CautiousAGD. See Theorem 6 for a formal statement.

- In Section 5, we present numerical experiments comparing an implementation of CertSDP with similar algorithms from the literature [23, 47, 56, 68], on random sparse k -exact SDP instances with $n \approx 10^3, 10^4$, and 10^5 . Our code outperforms previous state of the art and is the only algorithm among those we tested that was able to solve our largest instances to high accuracy.

1.3 Related work

Storage-optimal/efficient FOMs. A growing body of literature, itself containing multiple research strands, has explored FOMs for SDPs [5, 9, 20, 23, 25, 34, 37, 38, 47, 53, 56, 66–68]. Below, we recount some recent developments in this direction with a particular view towards storage-efficient or *storage-optimal* FOMs for SDPs admitting low-rank solutions. Storage-optimality alludes to the fact that a rank- k PSD matrix $Y \in \mathbb{S}_+^n$ can be represented as the outer product of an $n \times k$ *factor matrix* with itself, i.e., $Y = XX^\top$ for some $X \in \mathbb{R}^{n \times k}$, so that a primal iterate with rank k can be implicitly stored using only $O(nk)$ memory. Similarly, a dual iterate may be stored using only $O(m)$ memory. Then, a storage-optimal FOM is allowed to use only $O(m + nk)$ storage where k is the rank of the *true* primal SDP solution.

Low-storage and storage-optimal FOMs are particularly attractive for SDPs where $M_{\text{obj}}, M_1, \dots, M_m$ are either structured or sparse, so that it is possible to not only store the instance efficiently, but also to compute matrix-vector products efficiently [23]. The algorithm that we develop in this paper follows this pattern and similarly interacts with $M_{\text{obj}}, M_1, \dots, M_m$ via *only* matrix-vector products.

One paradigm towards developing storage-optimal FOMs leverages duality to construct surrogate primal SDPs that can be solved with optimal storage. In this paradigm, the variable $Y \in \mathbb{S}_+^n$ is *compressed*, i.e., replaced with $U\bar{Y}U^\top$ for some matrix $U \in \mathbb{R}^{n \times k}$ and $\bar{Y} \in \mathbb{S}_+^k$. Ding et al. [23] give rigorous guarantees for such a method assuming strict complementarity. Specifically, they show that if $U \in \mathbb{R}^{n \times k}$ corresponds to a minimum eigenspace of an approximate dual solution, then the optimal solution \bar{Y} of the compressed SDP (in penalty form) is a good approximation of the true primal solution. Then, combining their bounds with existing FOMs for solving the dual SDP approximately, Ding et al. [23] show that $\|U\bar{Y}U^\top - Y^*\|_F \leq \epsilon$ after $O(\epsilon^{-2})$ -many minimum eigenvector computations. It is unclear how this convergence guarantee changes when only approximate eigenvector computations (which are the only practical option) are allowed. Friedlander and Macêdo [25] explore a similar idea for trace-minimization SDPs (i.e., SDPs where $M_{\text{obj}} = I$) from the viewpoint of *gauge duality*. Specifically, they show that if U corresponds to a minimum eigenspace associated with the true solution to the gauge dual, then the optimal solution of the compressed SDP exactly recovers the true primal SDP solution. Unfortunately, they do not analyze the accuracy of the recovered primal solution when the gauge dual is solved only approximately, which is the case in practice.

A second paradigm towards developing storage efficient/optimal FOMs works simultaneously in both the primal and dual spaces by employing *linear sketches*. Yurtsever et al. [68] apply the Nystrom sketch to the conditional gradient-augmented Lagrangian (CGAL) technique [67] to derive SketchyCGAL. They show that it is possible to reconstruct a $(1 + \zeta)$ -optimal rank- k approximation of an ϵ -optimal solution to the primal SDP¹ by tracking only the dual iterates as well as a $O(nk/\zeta)$ -

¹In [68], a rank- k matrix $\tilde{Y} \in \mathbb{S}_+^n$ is a $(1 + \zeta)$ -optimal rank- k approximation of an ϵ -optimal solution $Y_\epsilon \in \mathbb{S}_+^n$ if

sized sketch of the primal iterates. When the true solution is unique and has rank- k , it is appropriate to take $\zeta = O(1)$ so that the total storage is $O(m + nk)$. Furthermore, Yurtsever et al. [68] bound the required accuracy in the approximate eigenvector computations within SketchyCGAL. In all, they show that it is possible to implement their algorithm in $O(\epsilon^{-2})$ iterations where each iteration involves computing an eigenvector via $\tilde{O}(\epsilon^{-1/2})$ matrix-vector products. In follow-up work, Shinde et al. [53] combine the algorithmic architecture of SketchyCGAL with the additional observation that in specific applications (e.g., max-cut), the goal is simply to *sample* from a Gaussian distribution with variance given by an approximate solution Y_ϵ to the SDP. Under this alternate goal, it is possible to further reduce the storage requirements to $O(n + m)$.

One may compare these storage-optimal FOMs for SDPs with the Burer–Monteiro method [12]. In the Burer–Monteiro method, the convex SDP in the variable $Y \in \mathbb{S}_+^n$ is explicitly replaced with an outer product term involving an $n \times k'$ factor matrix where $k' \geq k$. The resulting *nonconvex* problem is then tackled via local optimization methods. While results [10, 18, 19] have shown that non-global local minima cannot exist when $k' = \Omega(\sqrt{m})$ (so that local optimization methods are certifiably correct), more recent work [61] has shown that such spurious local minima can in fact exist even if $k = 1$ and $k' = \Theta(\sqrt{m})$. In other words, the Burer–Monteiro approach provably cannot achieve storage-optimality.

FOMs for the GTRS. The algorithms developed in the current paper are inspired by recent developments in FOMs for the TRS and the GTRS. There has been extensive work [16, 28, 29, 31, 40, 50, 57, 62, 65] towards developing customized algorithms for the TRS and GTRS that circumvent solving large SDPs; see [65] and references therein for a more thorough account of algorithmic ideas for solving large-scale GTRS instances. We highlight only the two most relevant results from this area.

Carmon and Duchi [16] consider iterative methods that produce Krylov subspace solutions to the TRS, i.e., solutions to the TRS restricted to a Krylov subspace generated by the objective function. They show that these solutions converge to the true TRS solution *linearly* as long as the linear term in q_{obj} is not orthogonal to the minimum eigenspace of the Hessian in q_{obj} . One may interpret this condition as requiring strict complementarity between the SDP relaxation of the TRS and its dual.

More recently, Wang et al. [65] make a connection between the GTRS and optimal FOMs for strongly convex minimax problems [45]. In the language of the current paper, Wang et al. [65] assume strict complementarity between the SDP relaxation of the GTRS and its dual, and show how to construct a strongly convex reformulation of the GTRS using low-accuracy eigenvalue computations. More concretely, they show how to construct $\tilde{\gamma}_-$ and $\tilde{\gamma}_+$ such that the minimax problem

$$\min_{x \in \mathbb{R}^n} \max_{\gamma \in [\tilde{\gamma}_-, \tilde{\gamma}_+]} (q_{\text{obj}}(x) + \gamma \cdot q_1(x))$$

is strongly convex and has as its unique optimizer the optimizer of the underlying GTRS. The resulting strongly convex minimax problem is then solved via [45, Algorithm 2.3.13] to achieve a linear convergence rate. One may compare the strongly convex reformulation of the GTRS in [65] with the more natural Lagrangian reformulation (through S-lemma):

$$\min_{x \in \mathbb{R}^n} \sup_{\gamma \in \Gamma} (q_{\text{obj}}(x) + \gamma \cdot q_1(x))$$

$\|Y_\epsilon - \tilde{Y}\|_* \leq (1 + \zeta) \|Y_\epsilon - [Y_\epsilon]_k\|_*$ where $\|\cdot\|_*$ is the nuclear norm and $[Y_\epsilon]_k$ is the best rank- k approximation of Y_ϵ .

where $\Gamma = \{\gamma \in \mathbb{R}_+ : q_{\text{obj}} + \gamma q_1 \text{ is convex}\}$. Specialized FOMs have also been developed for the GTRS using this Lagrangian reformulation [62]. Unfortunately, since the Lagrangian reformulation may not be strongly convex in general, the resulting algorithms can only achieve sublinear (in terms of ϵ) convergence rates—specifically, rates of the form $O(\epsilon^{-1/2})$ as opposed to rates of the form $O(\log(\epsilon^{-1}))$.

Accelerated FOMs for non-smooth problems via saddle-point problems. One may treat the QMMP reformulation (1) of the SDP as a saddle-point problem in the variables $(X, \gamma) \in \mathbb{R}^{(n-k) \times k} \times \mathbb{R}^m$ as opposed to a non-smooth problem in just $X \in \mathbb{R}^{(n-k) \times k}$. There is a vast body of work developing accelerated FOMs for non-smooth problems that leverages saddle-point structure [32, 42, 45, 49]. Both Nesterov [44] and Nemirovski [42] achieve an accelerated convergence rate of $O(\epsilon^{-1})$ for general convex–concave saddle point problems (see also [59]). This rate can be further improved for the special case of strongly convex–concave saddle-point problems [17, 32, 43]: Nesterov’s excessive gap technique [43, 45] achieves an $O(\epsilon^{-1/2})$ convergence for strongly convex–concave saddle-point problems where the coupling term is linear. This is generalized in [17] to allow nonlinear proximal operators. Hamedani and Aybat [27], Juditsky and Nemirovski [32] generalize this convergence rate to the setting where the gradient of the coupling term is only assumed to be Lipschitz. These rates match the known [48] lower bound of $O(\epsilon^{-1/2})$ for any FOM on the general class of strongly convex-concave saddle-point problems. Note that the assumption that the gradient of the coupling term is Lipschitz does not hold for our setting. Indeed, the saddle point function we are interested in, $\langle M(\gamma), Y(X) \rangle$, is jointly *cubic* in the variables (X, γ) (so that the gradients vary quadratically). Nonetheless, we will show that it is possible to achieve the optimal $O(\epsilon^{-1/2})$ iteration complexity in our setting.

Accelerated FOMs with inexact first-order information. A related line of work [21, 22] has analyzed the convergence rate of (accelerated) FOMs in the presence of inexact first-order information. Devolder et al. [22] analyzes FOMs for smooth convex functions. In [21], the same authors extend these results to FOMs for smooth and strongly convex functions. Our algorithm (Algorithm 1) continues this line of work by considering an inexact *prox-map* for strongly convex *max-type functions*.

1.4 Notation

For a positive integer n , let $[n] := \{1, \dots, n\}$. Let \mathbb{R}_{++} denote the positive real numbers. Let \mathbb{S}^n denote the set of $n \times n$ real symmetric matrices and let I_n denote the $n \times n$ identity matrix. Given $X \in \mathbb{S}^n$, we write $X \succeq 0$ (resp. $X \succ 0$) if X is positive semidefinite (resp. positive definite). Let \mathbb{S}_+^n be the positive semidefinite cone. Given $X \in \mathbb{R}^{n \times m}$ let X^\top , $\ker(X)$, $\text{range}(X)$, $\text{rank}(X)$, $\|X\|_F$ and $\|X\|_2$ denote the transpose, kernel, range, rank, Frobenius norm, and spectral norm of X . Given $X \in \mathbb{R}^{n \times n}$ let $\text{tr}(X)$ denote the trace of X . We endow both \mathbb{S}^n and $\mathbb{R}^{n \times m}$ with the trace inner product $\langle X, Y \rangle := \text{tr}(X^\top Y)$. Given a subspace $W \subseteq \mathbb{R}^n$, let W^\perp denote its orthogonal complement. Abusing notation, we write \mathbb{S}^W to denote the vector space of self-adjoint operators on W and \mathbb{R}^{W, W^\perp} for the vector space of linear maps from W^\perp to W . Given $M \in \mathbb{S}^n$ let $M_W \in \mathbb{S}^W$, $M_{W, W^\perp} \in \mathbb{R}^{W, W^\perp}$, and $M_{W^\perp} \in \mathbb{S}^{W^\perp}$ denote the restrictions of M to the corresponding subspaces. Let $\mathbf{S}^{n-1} \subseteq \mathbb{R}^n$ denote the unit sphere. Given $x \in \mathbb{R}^n$ and $r \geq 0$, let $\mathbb{B}(x, r)$ denote the closed ℓ_2 -ball centered at x with radius r . Given a function in multiple arguments $f(x_1, \dots, x_m)$, we write $\nabla_k f(x_1, \dots, x_m)$ to denote the gradient of f in the k th argument evaluated at x_1, \dots, x_m .

2 Strongly convex reformulations of k -exact SDPs

In this section, we describe how to construct a strongly convex reformulation of a *rank- k exact QMP-like SDP* using a *certificate of strict complementarity* (see Definitions 1 and 2). The following sections will expand on these ideas and show how these properties can be exploited to achieve algorithmic efficiency.

2.1 Definitions and problem setup

We make the following two assumptions on (SDP).

Assumption 1. Assume in (SDP) that the primal and dual problems are both solvable, strong duality holds, and there exist primal and dual optimal solutions $Y^* \in \mathbb{S}^n$ and $\gamma^* \in \mathbb{R}^m$ such that $\text{rank}(Y^*) = k$ and $\text{rank}(M(\gamma^*)) = n - k$. \square

We fix Y^* and γ^* to be solutions to (SDP) satisfying $\text{rank}(Y^*) = k$ and $\text{rank}(M(\gamma^*)) = n - k$.

Assumption 2. Let $W \subseteq \mathbb{R}^n$ be a k -dimensional subspace such that the restriction of Y^* to W^\perp is known and positive definite. \square

Definition 1. We say that an instance of (SDP) is a *rank- k exact QMP-like SDP* or a *k -exact SDP* for short if both Assumptions 1 and 2 hold. \square

Definition 2. We say that a compact subset $\mathcal{U} \subseteq \mathbb{R}^m$ *certifies strict complementarity* if $\gamma^* \in \mathcal{U}$ and, for all $\gamma \in \mathcal{U}$, it holds that $M(\gamma)_W \succ 0$. \square

Remark 1. Suppose we are given a certificate of strict complementarity \mathcal{U} , i.e., $\gamma^* \in \mathcal{U}$ and $M(\gamma)_W \succ 0$ for all $\gamma \in \mathcal{U}$. We immediately deduce that $\text{rank}(M(\gamma^*)) \geq \text{rank}(M(\gamma^*)_W) = n - k$. On the other hand, $\text{rank}(Y^*) \geq \text{rank}(Y^*_{W^\perp}) = k$. This is the sense in which \mathcal{U} *certifies strict complementarity*. \square

2.2 Identifying \mathbb{S}^n with quadratic matrix functions

Suppose (SDP) is a k -exact SDP and that \mathcal{U} certifies strict complementarity. For ease of presentation, we will assume in this subsection that W is the $(n - k)$ -dimensional subspace corresponding to the first $n - k$ coordinates of \mathbb{R}^n . This is without loss of generality and our results extend in the natural way to the setting where W is general.

Our strongly convex reformulation of (SDP) will regard the $M_i \in \mathbb{S}^n$ as inducing *quadratic matrix functions* on the space $\mathbb{R}^{W \times W^\perp} \simeq \mathbb{R}^{(n-k) \times k}$. We begin by writing each M_i , for $i \in \{\text{obj}\} \cup [m]$, as a block matrix

$$M_i = \begin{pmatrix} A_i/2 & \tilde{B}_i/2 \\ \tilde{B}_i^\top/2 & C_i \end{pmatrix},$$

where $A_i \in \mathbb{S}^{n-k}$, $\tilde{B}_i \in \mathbb{R}^{(n-k) \times n}$ and $C_i \in \mathbb{S}^k$.

We will partition Y^* as a block matrix with compatible block structure: Define $Z^* := Y^*_{W^\perp}$ and $X^* := Y^*_{W, W^\perp} (Z^*)^{-1/2}$. Note here that Z^* is known *a priori* due to Assumption 2. Next, by the

assumption that $\text{rank}(Y^*) = k$ (Assumption 1), we have that

$$Y^* = \begin{pmatrix} X^* X^{*\top} & X^* (Z^*)^{1/2} \\ (Z^*)^{1/2} (X^*)^\top & Z^* \end{pmatrix}.$$

Finally, given $X \in \mathbb{R}^{(n-k) \times k}$, define

$$Y(X) := \begin{pmatrix} X X^\top & X (Z^*)^{1/2} \\ (Z^*)^{1/2} X^\top & Z^* \end{pmatrix}$$

and note that $Y(X^*) = Y^*$.

One of our key ideas in building a strongly convex reformulation of (SDP) is that $Y(X)$ is a matrix whose entries are *quadratic* in X . We can thus identify each M_i with a quadratic matrix function. For each $i \in \{\text{obj}\} \cup [m]$, define

$$\begin{aligned} q_i(X) &:= \langle M_i, Y(X) \rangle + d_i = \frac{\text{tr}(X^\top A_i X)}{2} + \langle \tilde{B}_i (Z^*)^{1/2}, X \rangle + \langle C_i, Z^* \rangle + d_i \\ &= \frac{\text{tr}(X^\top A_i X)}{2} + \langle B_i, X \rangle + c_i, \end{aligned}$$

where we let $B_i := \tilde{B}_i (Z^*)^{1/2}$ and $c_i := \langle C_i, Z^* \rangle + d_i$. Finally, given $\gamma \in \mathbb{R}^m$, define $A(\gamma) := A_{\text{obj}} + \sum_{i=1}^m \gamma_i A_i$. We define $B(\gamma)$, $\tilde{B}(\gamma)$, $c(\gamma)$, $d(\gamma)$, and $q(\gamma, X)$ analogously.

Remark 2. One may compare our parameterization of rank- k matrices in \mathbb{S}_+^n with the Burer-Monteiro approach [12]. In the Burer-Monteiro approach, one replaces the matrix variable $Y \in \mathbb{S}_+^n$ with a rank- k matrix variable parameterized by

$$Y = \begin{pmatrix} X \\ X' \end{pmatrix} \begin{pmatrix} X \\ X' \end{pmatrix}^\top$$

where $X \in \mathbb{R}^{(n-k) \times k}$ and $X' \in \mathbb{R}^{k \times k}$. This transformation replaces the $O(n^2)$ -dimensional variable $Y \in \mathbb{S}_+^n$ with the nk -dimensional variable $(X; X') \in \mathbb{R}^{n \times k}$. Unfortunately, this approach also transforms the SDP from a convex problem into a *nonconvex* problem. Our parameterization will allow us to remedy this nonconvexity. As we will see in the next subsection, when Z^* is known (so that we may fix X') *a priori*, we may further reformulate the remaining *nonconvex* problem in $X \in \mathbb{R}^{(n-k) \times k}$ into a *strongly convex* problem. \square

2.3 A strongly convex reformulation of (SDP)

The following theorem states that if \mathcal{U} certifies strict complementarity, then X^* is the unique minimizer of a strongly convex *quadratic matrix minimax problem* (QMMP) that can be constructed from \mathcal{U} .

Theorem 1. *Suppose (SDP) is a rank- k exact QMP-like SDP and that \mathcal{U} certifies strict complementarity. Then, X^* is the unique minimizer of the strongly convex QMMP*

$$\min_{X \in \mathbb{R}^{W \times W^\perp}} \max_{\gamma \in \mathcal{U}} q(\gamma, X). \quad (\text{QMMP}_{\mathcal{U}})$$

Furthermore, $X^* = -A(\gamma^*)^{-1} B(\gamma^*)$ and $\text{Opt}_{(\text{QMMP}_{\mathcal{U}})} = \text{Opt}_{(\text{SDP})}$.

Proof. Without loss of generality, we work in the basis where W is the first $n - k$ coordinates of \mathbb{R}^n . Note that the assumption that \mathcal{U} certifies strict complementarity implies that $A(\gamma^*) = M(\gamma^*)_W \succ 0$. We begin by verifying that $X^* = -A(\gamma^*)^{-1}B(\gamma^*)$. By complementary slackness, we have

$$\begin{aligned} 0 &= \langle M(\gamma^*), Y(X^*) \rangle \\ &= \text{tr} \left(\begin{pmatrix} X^* \\ (Z^*)^{1/2} \end{pmatrix}^\top \begin{pmatrix} A(\gamma^*)/2 & \tilde{B}(\gamma^*)/2 \\ \tilde{B}(\gamma^*)^\top/2 & C(\gamma^*) \end{pmatrix} \begin{pmatrix} X^* \\ (Z^*)^{1/2} \end{pmatrix} \right) \\ &= \text{tr} \left(\frac{(X^* + A(\gamma^*)^{-1}B(\gamma^*))^\top A(\gamma^*) (X^* + A(\gamma^*)^{-1}B(\gamma^*))}{2} \right) \\ &\quad + \left[\langle C(\gamma^*), Z^* \rangle - \text{tr} \left(\frac{B(\gamma^*)^\top A(\gamma^*)^{-1}B(\gamma^*)}{2} \right) \right]. \end{aligned}$$

Here, the second line follows by the definitions of $M(\gamma^*)$ and $Y(X^*)$, and the third line follows from the definition $B(\gamma) := \tilde{B}(\gamma)(Z^*)^{1/2}$. We claim that the square-bracketed term on the final line is zero: By the assumption that $\text{rank}(M(\gamma^*)) = n - k$ and the fact that $A(\gamma^*) \succ 0$, we have that $C(\gamma^*) = \frac{\tilde{B}(\gamma^*)^\top A(\gamma^*)^{-1} \tilde{B}(\gamma^*)}{2}$. Pre- and post-multiplying $C(\gamma^*)$ by $(Z^*)^{1/2}$ and taking the trace of this identity gives

$$\begin{aligned} \langle C(\gamma^*), Z^* \rangle &= \text{tr} \left(\frac{(Z^*)^{1/2} \tilde{B}(\gamma^*)^\top A(\gamma^*)^{-1} \tilde{B}(\gamma^*) (Z^*)^{1/2}}{2} \right) \\ &= \text{tr} \left(\frac{B(\gamma^*)^\top A(\gamma^*)^{-1} B(\gamma^*)}{2} \right). \end{aligned}$$

Thus, we have that

$$0 = \text{tr} \left(\left(X^* + A(\gamma^*)^{-1}B(\gamma^*) \right)^\top A(\gamma^*) \left(X^* + A(\gamma^*)^{-1}B(\gamma^*) \right) \right),$$

so that $X^* = -A(\gamma^*)^{-1}B(\gamma^*)$ by the positive definiteness of $A(\gamma^*)$.

Next, note that by the feasibility of Y^* , we have $q_i(X^*) = \langle M_i, Y^* \rangle + d_i = 0$ for all $i \in [m]$. Similarly, by the optimality of Y^* , we have $q_{\text{obj}}(X^*) = \langle M_{\text{obj}}, Y^* \rangle = \text{Opt}_{(\text{SDP})}$. In particular, $\max_{\gamma \in \mathcal{U}} q(\gamma, X^*) = q(\gamma^*, X^*) = \text{Opt}_{(\text{SDP})}$. On the other hand, for any $X \in \mathbb{R}^{W \times W^\perp}$,

$$\max_{\gamma \in \mathcal{U}} q(\gamma, X) \geq q(\gamma^*, X) = \text{Opt}_{(\text{SDP})} + \text{tr} \left(\frac{(X - X^*)^\top A(\gamma^*) (X - X^*)}{2} \right).$$

As $A(\gamma^*) \succ 0$, we conclude that X^* is the unique minimizer of $(\text{QMMP}_{\mathcal{U}})$ with optimal value $\text{Opt}_{(\text{QMMP}_{\mathcal{U}})} = \text{Opt}_{(\text{SDP})}$.

Finally, strong convexity of $(\text{QMMP}_{\mathcal{U}})$ follows from compactness of \mathcal{U} and the assumption that \mathcal{U} certifies strict complementarity (so that $A(\gamma) = M(\gamma)_W$ is positive definite over \mathcal{U}). \blacksquare

Remark 3. One may compare $(\text{QMMP}_{\mathcal{U}})$ with the more natural Lagrangian formulation of (SDP) , which results in a QMMP in the same space:

$$\min_{X \in \mathbb{R}^{W \times W^\perp}} \sup_{\gamma \in \mathbb{R}^m: A(\gamma) \succeq 0} q(\gamma, X). \quad (2)$$

Indeed, it is possible to show that X^* is also the unique minimizer of (2). Nevertheless, the formulation (2), in contrast to (QMMP $_{\mathcal{U}}$), has two major downsides: First, it may be the case that $\sup_{\gamma \in \mathbb{R}^m: A(\gamma) \succeq 0} q(\gamma, X)$ is a convex function in X that is *not strongly convex*. Second, the domain of the supremum, $\{\gamma \in \mathbb{R}^m : A(\gamma) \succeq 0\}$, is itself a spectrahedron so that *even evaluating* $\sup_{\gamma \in \mathbb{R}^m: A(\gamma) \succeq 0} q(\gamma, X)$ (that is, evaluating *zeroth-order* information in the X variable) requires solving an SDP. In contrast, (QMMP $_{\mathcal{U}}$) is strongly convex by construction. Furthermore, we may pick \mathcal{U} to have efficient projection and linear maximization oracles (e.g., by taking \mathcal{U} to be an ℓ_2 ball). From this viewpoint, (QMMP $_{\mathcal{U}}$) will be much more amenable than (2) to first-order methods. \square

3 Algorithms for strongly convex QMMPs

In this section, we describe and analyze an accelerated first-order method (FOM) for solving strongly convex QMMPs. While we will apply this algorithm to problems arising from the application of Theorem 1, the algorithms from this section can handle *general* strongly convex QMMPs.

We state explicitly the setup and assumptions of this section. Let $q_{\text{obj}}, q_1, \dots, q_m : \mathbb{R}^{(n-k) \times k} \rightarrow \mathbb{R}$ be quadratic matrix functions of the form

$$q_i(X) = \frac{\text{tr}(X^\top A_i X)}{2} + \langle B_i, X \rangle + c_i.$$

Given $\gamma \in \mathbb{R}^m$, let $A(\gamma) := A_{\text{obj}} + \sum_{i=1}^m \gamma_i A_i$. Define $B(\gamma), c(\gamma), q(\gamma, X)$ analogously.

Let $\mathcal{U} \subseteq \mathbb{R}^m$ be a compact convex set with exact projection and linear maximization oracles. Our goal is to find an ϵ -optimal solution to

$$\min_{X \in \mathbb{R}^{(n-k) \times k}} \max_{\gamma \in \mathcal{U}} q(\gamma, X). \quad (\text{QMMP})$$

That is, our goal is to find some $\tilde{X} \in \mathbb{R}^{(n-k) \times k}$ satisfying $\max_{\gamma \in \mathcal{U}} q(\gamma, \tilde{X}) \leq \text{Opt}_{(\text{QMMP})} + \epsilon$. For notational convenience, we will define

$$Q(X) := \max_{\gamma \in \mathcal{U}} q(\gamma, X).$$

While we will treat \mathcal{U} as fixed in this section, in future sections, we will explicitly call attention to the dependence of the function Q on the set \mathcal{U} and write $Q_{\mathcal{U}}$ instead.

We present a FOM for (QMMP) under two assumptions. The first assumption (Assumption 3) requires uniform strong convexity and smoothness of $q(\gamma, X)$ over \mathcal{U} .

Assumption 3. We will assume algorithmic access to parameters $0 < \mu \leq L$ such that $\mu I \preceq A(\gamma) \preceq LI$ for all $\gamma \in \mathcal{U}$. \square

When Assumption 3 holds, we define the *condition number* of (QMMP) as $\kappa := L/\mu$. We will state our second assumption (which bounds the norms of various quantities) when needed in Section 3.3.

Our FOM will closely follow Nesterov's accelerated gradient descent scheme for strongly convex minimax functions [45, Algorithm 2.3.13] (henceforth AGD-MM) with one major difference. In contrast to the presentation in [45] and its application in [62], the necessary *prox-map* (see Definition 3 below) in the QMMP setting cannot be computed explicitly or exactly.

We break our FOM for strongly convex QMMPs into two levels, presented as the first two subsections in this section. In Section 3.1, we give a convergence analysis for a modified version of AGD-MM using an *inexact* prox-map oracle. In particular, we will bound the necessary accuracy of the prox-map to recover accelerated convergence rates. In Section 3.2, we show how to implement the approximate prox-map oracle efficiently for each iteration using the strongly convex excessive gap technique [45, Algorithm 6.2.37]. Finally, in Section 3.3, we state an assumption (Assumption 4) that allows us to bound the iteration cost of the prox-map oracle uniformly across iterations. Taken together with the results from the previous subsections, this will give a rigorous guarantee for the overall FOM.

3.1 An FOM for strongly convex QMMPs using an inexact prox-map oracle

This subsection generalizes AGD-MM by allowing *inexact* prox-map computations. We first recall the definition of the prox-map and the fundamental relation (3) that is used in the convergence rate analysis of AGD-MM. Next, we show how to recover a similar inequality (5) when the prox-map is computed only approximately. Finally, we show how to modify the step-sizes in AGD-MM to prevent error accumulation that may otherwise build up from inexact prox-map computations. These step-sizes allow us to recover the accelerated linear convergence rates of AGD-MM even with inexact prox-map computations.

3.1.1 The prox-map

AGD-MM requires computing the prox-map $X_L(\Xi)$ (defined in Definition 3) *exactly* in every iteration (adapted from [45, Definition 2.3.2]).

Definition 3. Let $\Xi \in \mathbb{R}^{(n-k) \times k}$. Define

$$\begin{aligned} Q(\Xi; X) &:= \max_{\gamma \in \mathcal{U}} (q(\gamma, \Xi) + \langle \nabla_2 q(\gamma, \Xi), X - \Xi \rangle) \\ Q_L(\Xi; X) &:= Q(\Xi; X) + \frac{L}{2} \|X - \Xi\|_F^2 \\ Q_L^*(\Xi) &:= \min_{X \in \mathbb{R}^{(n-k) \times k}} Q_L(\Xi; X) \\ X_L(\Xi) &:= \arg \min_{X \in \mathbb{R}^{(n-k) \times k}} Q_L(\Xi; X) \\ g_L(\Xi) &:= L(\Xi - X_L(\Xi)). \end{aligned}$$

Here, $\nabla_2 q(\gamma, \Xi)$ is the gradient of $q(\gamma, X)$ in X at Ξ and is an affine function of γ (more explicitly, $\nabla_2 q(\gamma, \Xi) = A(\gamma)\Xi + B(\gamma)$). Note that the function $Q(\Xi; X)$ simply replaces the inside function $q(\gamma, X)$ in the definition of $Q(X)$ with its *linearization* around Ξ . The quantities X_L and g_L are the *prox-map* and the *grad-map*. \square

Recall also the main property of the prox-map and grad-map that is used in the analysis of the convergence rate of AGD-MM as given in the following lemma (adapted from [45, Theorem 2.3.2]).

Lemma 1. Let $\Xi \in \mathbb{R}^{(n-k) \times k}$. Then, for all $X \in \mathbb{R}^{(n-k) \times k}$,

$$Q(X) \geq Q(X_L(\Xi)) + \frac{1}{2L} \|g_L(\Xi)\|_F^2 + \langle g_L(\Xi), X - \Xi \rangle + \frac{\mu}{2} \|X - \Xi\|_F^2. \quad (3)$$

3.1.2 An approximate prox-map inequality

In the setting of general QMMPs, it is not possible to compute the prox-map exactly. Instead, we will apply an inner FOM to solve the prox-map $X_L(\Xi)$ to some prescribed accuracy. This necessitates an analysis of (a variant of) AGD-MM that works with inexact prox-map computations. To this end, we show how to recover a version of (3) where $X_L(\Xi)$ is computed only approximately.

Define

$$\tilde{\mu} := \mu/2, \quad \tilde{L} := L - \mu/2, \quad \text{and} \quad \tilde{\kappa} := \tilde{L}/\tilde{\mu}. \quad (4)$$

We will need the following geometric fact.

Lemma 2. *Let $\tilde{X}, X_L \in \mathbb{R}^{(n-k) \times k}$ be such that $\|\tilde{X} - X_L\|_F \leq \delta$. Then, for all $X \in \mathbb{R}^{(n-k) \times k}$,*

$$\frac{L}{2} \|X - X_L\|_F^2 \geq \frac{\tilde{L}}{2} \|X - \tilde{X}\|_F^2 - \frac{L\delta^2}{2} (2\kappa - 1).$$

Proof of Lemma 2. Let $\Delta := \tilde{X} - X_L$. Then,

$$\begin{aligned} \frac{L}{2} \|X - X_L\|_F^2 &= \frac{L}{2} \|X - \tilde{X} + \Delta\|_F^2 \\ &= \frac{\tilde{L}}{2} \|X - \tilde{X}\|_F^2 + \frac{\tilde{\mu}}{2} \|X - \tilde{X}\|_F^2 + L \langle X - \tilde{X}, \Delta \rangle + \frac{L}{2} \|\Delta\|_F^2, \end{aligned}$$

where the second equality follows from expanding the square and the fact that $L = \tilde{L} + \tilde{\mu}$. Moreover,

$$0 \leq \frac{L}{2} \left\| \sqrt{\frac{\tilde{\mu}}{L}} (X - \tilde{X}) + \sqrt{\frac{L}{\tilde{\mu}}} \Delta \right\|_F^2 = \frac{\tilde{\mu}}{2} \|X - \tilde{X}\|_F^2 + L \langle X - \tilde{X}, \Delta \rangle + L\kappa \|\Delta\|_F^2.$$

Combining these two inequalities gives

$$\frac{L}{2} \|X - X_L\|_F^2 \geq \frac{\tilde{L}}{2} \|X - \tilde{X}\|_F^2 - \frac{L\delta^2}{2} (2\kappa - 1). \quad \blacksquare$$

We may now derive a variant of (3) which only uses an approximate prox-map.

Theorem 2. *Let $\Xi \in \mathbb{R}^{(n-k) \times k}$. Suppose \tilde{X} satisfies*

$$Q_L(\Xi; \tilde{X}) \leq Q_L^*(\Xi) + \epsilon.$$

Set $\tilde{g} := \tilde{L}(\Xi - \tilde{X})$. Then, for all $X \in \mathbb{R}^{(n-k) \times k}$,

$$Q(X) \geq Q(\tilde{X}) + \frac{1}{2\tilde{L}} \|\tilde{g}\|_F^2 + \langle \tilde{g}, X - \Xi \rangle + \frac{\tilde{\mu}}{2} \|X - \Xi\|_F^2 - 2\kappa\epsilon. \quad (5)$$

Proof. As $Q_L(\Xi; X)$ is L -strongly convex, from the premise of the lemma we have $\|\tilde{X} - X_L(\Xi)\|_F \leq \sqrt{2\epsilon/L}$.

We bound

$$\begin{aligned}
Q(X) &\geq Q(\Xi; X) + \frac{\mu}{2} \|X - \Xi\|_F^2 \\
&= Q_L(\Xi; X) - \frac{\tilde{L}}{2} \|X - \Xi\|_F^2 + \frac{\tilde{\mu}}{2} \|X - \Xi\|_F^2 \\
&\geq Q_L^*(\Xi) + \frac{L}{2} \|X - X_L(\Xi)\|_F^2 - \frac{\tilde{L}}{2} \|X - \Xi\|_F^2 + \frac{\tilde{\mu}}{2} \|X - \Xi\|_F^2 \\
&\geq Q(\tilde{X}) + \frac{\tilde{L}}{2} \|X - \tilde{X}\|_F^2 - \frac{\tilde{L}}{2} \|X - \Xi\|_F^2 + \frac{\tilde{\mu}}{2} \|X - \Xi\|_F^2 - 2\kappa\epsilon \\
&= Q(\tilde{X}) + \frac{\tilde{L}}{2} \left(2 \langle X - \Xi, \Xi - \tilde{X} \rangle + \|\Xi - \tilde{X}\|_F^2 \right) + \frac{\tilde{\mu}}{2} \|X - \Xi\|_F^2 - 2\kappa\epsilon \\
&= Q(\tilde{X}) + \langle \tilde{g}, X - \Xi \rangle + \frac{1}{2\tilde{L}} \|\tilde{g}\|_F^2 + \frac{\tilde{\mu}}{2} \|X - \Xi\|_F^2 - 2\kappa\epsilon.
\end{aligned}$$

Here, the first inequality follows from μ -strong convexity of Q , the first equation follows from the definitions of $Q_L(\Xi; X)$, \tilde{L} and $\tilde{\mu}$, the second inequality follows from optimality of $X_L(\Xi)$, the third inequality follows from Lemma 2 applied with $\delta = \sqrt{2\epsilon/L}$ and the L -smoothness of $q(\gamma, X)$ for each $\gamma \in \mathcal{U}$, and the last two equations follow from expanding the squares and the definition of \tilde{g} . \blacksquare

3.1.3 Estimating sequences

We now modify the estimating sequences analysis of AGD-MM to use (5) instead of (3): Fix $X_0 \in \mathbb{R}^{(n-k) \times k}$ and let $\{\epsilon_t\} \subseteq \mathbb{R}_{++}$ and $\{\Xi_t\} \subseteq \mathbb{R}^{(n-k) \times k}$ to be fixed later. Define

$$\phi_0(X) := Q(X_0) + \frac{\tilde{\mu}}{2} \|X - X_0\|_F^2.$$

For $t \geq 0$, let X_{t+1} be an ϵ -approximate prox-map, i.e., X_{t+1} satisfies

$$Q_L(\Xi_t; X_{t+1}) \leq Q_L^*(\Xi_t) + \epsilon_t,$$

and set $\tilde{g}_t := \tilde{L}(\Xi_t - X_{t+1})$. Let $\alpha := \tilde{\kappa}^{-1/2}$ and recursively define

$$\begin{aligned}
\phi_{t+1}(X) &:= (1 - \alpha)\phi_t(X) \\
&\quad + \alpha \left(Q(X_{t+1}) + \frac{1}{2\tilde{L}} \|\tilde{g}_t\|_F^2 + \langle \tilde{g}_t, X - \Xi_t \rangle + \frac{\tilde{\mu}}{2} \|X - \Xi_t\|_F^2 \right).
\end{aligned}$$

The following lemma shows how $\phi_t(X)$ evolves. Its proof follows verbatim from the standard proof [45, Lemma 2.3.3] and is deferred to Appendix A. Indeed, the standard proof makes no assumption on how Ξ_t and X_t are related.

Lemma 3. *For all $t \geq 0$, $\phi_t(X)$ is a quadratic matrix function in X of the form*

$$\phi_t(X) = \phi_t^* + \frac{\tilde{\mu}}{2} \|X - V_t\|_F^2. \quad (6)$$

The sequences $\{\phi_t^*\}$, $\{V_t\}$ are given by $V_0 = X_0$, $\phi_0^* = Q(X_0)$ and the recurrences

$$\begin{aligned}
V_{t+1} &= (1 - \alpha)V_t + \alpha \left(\Xi_t - \frac{1}{\tilde{\mu}} \tilde{g}_t \right), \text{ and} \\
\phi_{t+1}^* &= (1 - \alpha)\phi_t^* + \alpha \left(Q(X_{t+1}) + \frac{1}{2\tilde{L}} \|\tilde{g}_t\|_F^2 \right) - \frac{\alpha^2}{2\tilde{\mu}} \|\tilde{g}_t\|_F^2 \\
&\quad + \alpha(1 - \alpha) \left(\frac{\tilde{\mu}}{2} \|\Xi_t - V_t\|_F^2 + \langle \tilde{g}_t, V_t - \Xi_t \rangle \right).
\end{aligned}$$

For all $t \geq 0$, we will henceforth set

$$\Xi_t := \frac{X_t + \alpha V_t}{1 + \alpha}.$$

The following lemma shows that Ξ_{t+1} can be written as an *extragradient* step from X_t towards X_{t+1} . Its proof follows verbatim from the standard proof [45, Page 92] and is deferred to Appendix A. Indeed, the standard proof only needs the relation $X_{t+1} = \Xi_t - \tilde{g}_t/\tilde{L}$, which continues to hold in our setting by construction.

Lemma 4. *It holds that $\Xi_0 = X_0$ and $\Xi_{t+1} = X_{t+1} + \frac{1-\alpha}{1+\alpha}(X_{t+1} - X_t)$ for all $t \geq 0$.*

The following two lemmas bound the two types of errors that result from inexact prox-map computations. Define $E_0^{(1)} := 0$, $E_0^{(2)} := 0$, and for all $t \geq 0$ inductively set

$$E_{t+1}^{(1)} := (1 - \alpha)E_t^{(1)} + (1 - \alpha)\epsilon_t \quad \text{and} \quad E_{t+1}^{(2)} := (1 - \alpha)E_t^{(2)} + \alpha\epsilon_t.$$

Let $E_t := E_t^{(1)} + E_t^{(2)}$ be the sum of the two types of errors. Equivalently, let $E_t := 0$ and inductively set $E_{t+1} = (1 - \alpha)E_t + \epsilon_t$ for all $t \geq 0$.

Lemma 5. *It holds that $Q(X_t) \leq \phi_t^* + 2\kappa E_t^{(1)}$ for all $t \geq 0$.*

Proof. It is clear that $Q(X_0) \leq \phi_0^*$. Thus, consider X_{t+1} with $t \geq 0$. By induction and Lemma 3,

$$\begin{aligned} \phi_{t+1}^* &\geq (1 - \alpha)Q(X_t) + \alpha Q(X_{t+1}) + \left(\frac{\alpha}{2\tilde{L}} - \frac{\alpha^2}{2\tilde{\mu}} \right) \|\tilde{g}_t\|_F^2 \\ &\quad + \alpha(1 - \alpha) \langle \tilde{g}_t, V_t - \Xi_t \rangle - (1 - \alpha) \left(2\kappa E_t^{(1)} \right). \end{aligned}$$

As X_{t+1} satisfies $Q_L(\Xi_t; X_{t+1}) \leq Q^*(\Xi_t) + \epsilon_t$, we deduce (see Theorem 2) that

$$Q(X_t) \geq Q(X_{t+1}) + \frac{1}{2\tilde{L}} \|\tilde{g}_t\|_F^2 + \langle \tilde{g}_t, X_t - \Xi_t \rangle + \frac{\tilde{\mu}}{2} \|X_t - \Xi_t\|_F^2 - 2\kappa\epsilon_t.$$

These two inequalities together lead to

$$\begin{aligned} \phi_{t+1}^* &\geq Q(X_{t+1}) - 2\kappa(1 - \alpha)(E_t^{(1)} + \epsilon_t) \\ &\quad + \left(\frac{\alpha}{2\tilde{L}} - \frac{\alpha^2}{2\tilde{\mu}} + \frac{1 - \alpha}{2\tilde{L}} \right) \|\tilde{g}_t\|_F^2 + (1 - \alpha) \langle \tilde{g}_t, \alpha(V_t - \Xi_t) + (X_t - \Xi_t) \rangle. \end{aligned}$$

It is straightforward to show that the two quantities on the final line are identically zero using the relations $\alpha^2 = \tilde{\mu}/\tilde{L}$ and $\Xi_t = \frac{X_t + \alpha V_t}{1 + \alpha}$ (see Lemma 4). \blacksquare

Lemma 6. *For all $t \geq 0$, it holds that*

$$\phi_t(X) \leq (1 - (1 - \alpha)^t)Q(X) + (1 - \alpha)^t \phi_0(X) + 2\kappa E_t^{(2)}, \quad \forall X \in \mathbb{R}^{(n-k) \times k}.$$

Proof. The statement holds for $t = 0$. Thus, consider ϕ_{t+1} for $t \geq 0$. By definition

$$\phi_{t+1}(X) = (1 - \alpha)\phi_t(X) + \alpha \left(Q(X_{t+1}) + \frac{1}{2\tilde{L}} \|\tilde{g}_t\|_F^2 + \langle \tilde{g}_t, X - \Xi_t \rangle + \frac{\tilde{\mu}}{2} \|X - \Xi_t\|_F^2 \right).$$

As X_{t+1} satisfies $Q_L(\Xi_t; X_{t+1}) \leq Q^*(\Xi_t) + \epsilon_t$, we deduce (see Theorem 2) that

$$Q(X) \geq Q(X_{t+1}) + \frac{1}{2\tilde{L}} \|\tilde{g}_t\|_F^2 + \langle \tilde{g}_t, X - \Xi_t \rangle + \frac{\tilde{\mu}}{2} \|X - \Xi_t\|_F^2 - 2\kappa\epsilon_t.$$

Then, these inequalities combined with the inductive hypothesis give

$$\begin{aligned} \phi_{t+1}(X) &\leq (1 - \alpha)\phi_t(X) + \alpha Q(X) + 2\kappa\alpha\epsilon_t \\ &= (1 - (1 - \alpha)^{t+1})Q(X) + (1 - \alpha)(\phi_t(X) - (1 - (1 - \alpha)^t)Q(X)) + 2\kappa\alpha\epsilon_t \\ &\leq (1 - (1 - \alpha)^{t+1})Q(X) + (1 - \alpha)^{t+1}\phi_0(X) + 2\kappa \left((1 - \alpha)E_t^{(2)} + \alpha\epsilon_t \right). \quad \blacksquare \end{aligned}$$

Combining Lemmas 5 and 6, we get a bound on the total error due to inexact prox-maps as a function of the accuracy of each individual prox-map.

Corollary 1. *For all $t \geq 0$, it holds that*

$$Q(X_t) - \text{Opt}_{(\text{QMMP})} \leq (1 - \alpha)^t \left[2 \left(Q(X_0) - \text{Opt}_{(\text{QMMP})} \right) \right] + 2\kappa E_t.$$

Proof. Let $X_{\mathcal{U}}^*$ denote the optimizer of (QMMP) so that $Q(X_{\mathcal{U}}^*) = \text{Opt}_{(\text{QMMP})}$. Then, Lemmas 5 and 6 give

$$\begin{aligned} Q(X_t) - \text{Opt}_{(\text{QMMP})} &\leq \phi_t^* + 2\kappa E_t^{(1)} - Q(X_{\mathcal{U}}^*) \\ &\leq \phi_t(X_{\mathcal{U}}^*) + 2\kappa E_t^{(1)} - Q(X_{\mathcal{U}}^*) \\ &\leq (1 - (1 - \alpha)^t)Q(X_{\mathcal{U}}^*) + (1 - \alpha)^t\phi_0(X_{\mathcal{U}}^*) + 2\kappa E_t - Q(X_{\mathcal{U}}^*) \\ &= (1 - \alpha)^t \left(\phi_0(X_{\mathcal{U}}^*) - \text{Opt}_{(\text{QMMP})} \right) + 2\kappa E_t. \end{aligned}$$

Note also that by the definition of $\phi_0(\cdot)$ and the μ -strong convexity of Q , we have

$$\begin{aligned} \phi_0(X_{\mathcal{U}}^*) - \text{Opt}_{(\text{QMMP})} &= Q(X_0) - \text{Opt}_{(\text{QMMP})} + \frac{\tilde{\mu}}{2} \|X_{\mathcal{U}}^* - X_0\|_F^2 \\ &\leq 2 \left(Q(X_0) - \text{Opt}_{(\text{QMMP})} \right). \end{aligned}$$

Combining the two inequalities completes the proof. \blacksquare

We are now ready to present CautiousAGD (Algorithm 1) and its guarantee.

Algorithm 1 CautiousAGD

Given $q(\gamma, X)$ and \mathcal{U} satisfying Assumption 3; $X_0 \in \mathbb{R}^{(n-k) \times k}$, and a bound $\text{gap}_0 \in \mathbb{R}$ such that $Q(X_0) - \text{Opt}_{(\text{QMMP})} \leq \text{gap}_0$

1. Set $\tilde{\mu}, \tilde{L}, \tilde{\kappa}$ as in (4) and $\alpha := \tilde{\kappa}^{-1/2}$. Set $\Xi_0 := X_0$.
2. For $t \geq 0$
 - (a) Compute an inexact prox-map X_{t+1} satisfying

$$Q_L(\Xi_t; X_{t+1}) \leq Q_L^*(\Xi_t) + \epsilon_t, \quad \text{where} \quad \epsilon_t = \begin{cases} \frac{\text{gap}_0}{\tilde{\kappa}} \left(1 - \frac{\alpha}{2}\right), & \text{if } t = 0, \text{ and} \\ \frac{\text{gap}_0}{\tilde{\kappa}} \left(1 - \frac{\alpha}{2}\right)^t \frac{\alpha}{2}, & \text{else.} \end{cases} \quad (7)$$

- (b) Set $\Xi_{t+1} := X_{t+1} + \frac{1-\alpha}{1+\alpha} (X_{t+1} - X_t)$
-

Theorem 3. Let $q(\gamma, X)$ and \mathcal{U} satisfy Assumption 3. Let gap_0 be a known upper bound on $Q(X_0) - \text{Opt}_{(\text{QMMP})}$ and let X_t denote the iterates produced by Algorithm 1 with starting point X_0 . Then, for all $t \geq 1$, the iterate X_t satisfies

$$Q(X_t) - \text{Opt}_{(\text{QMMP})} \leq \left(1 - \frac{\alpha}{2}\right)^t (4 \cdot \text{gap}_0).$$

In particular, $Q(X_T) - \text{Opt}_{(\text{QMMP})} \leq \epsilon$ after at most

$$T = O\left(\sqrt{\kappa} \log\left(\frac{\text{gap}_0}{\epsilon}\right)\right)$$

iterations. The t -th iteration of Algorithm 1 requires computing a prox-map X_{t+1} satisfying (7).

Proof. We first claim that $E_t = (\text{gap}_0/\kappa)(1 - \alpha/2)^t$ for all $t \geq 1$. Indeed, this claim holds for $t = 1$ as $E_1 = \epsilon_0 = (\text{gap}_0/\kappa)(1 - \alpha/2)$ by construction (see (7)). Then, by induction

$$\begin{aligned} E_{t+1} &= (1 - \alpha)E_t + \epsilon_t \\ &= (1 - \alpha)\frac{\text{gap}_0}{\kappa} \left(1 - \frac{\alpha}{2}\right)^t + \frac{\text{gap}_0}{\kappa} \left(1 - \frac{\alpha}{2}\right)^t \frac{\alpha}{2} = \frac{\text{gap}_0}{\kappa} \left(1 - \frac{\alpha}{2}\right)^{t+1}. \end{aligned}$$

Then, the bound on $Q(X_t) - \text{Opt}_{(\text{QMMP})}$ follows from Corollary 1 and the starting condition $Q(X_0) - \text{Opt}_{(\text{QMMP})} \leq \text{gap}_0$. \blacksquare

Remark 4. We refer to Algorithm 1 as CautiousAGD to allude to the fact that Algorithm 1 is simply AGD-MM with inexact prox-maps and smaller extra-gradient steps. Specifically, AGD-MM sets

$$\Xi_{t+1} = X_{t+1} + \left(\frac{1 - \kappa^{-1/2}}{1 + \kappa^{-1/2}}\right) (X_{t+1} - X_t)$$

whereas CautiousAGD sets

$$\Xi_{t+1} = X_{t+1} + \left(\frac{1 - \tilde{\kappa}^{-1/2}}{1 + \tilde{\kappa}^{-1/2}}\right) (X_{t+1} - X_t).$$

Note that $\kappa \leq \tilde{\kappa} \leq 2\kappa$. \square

3.2 Approximating the prox-map

Recall that the prox-map $X_L(\Xi)$ is the minimizer of $Q_L(\Xi; X)$:

$$\begin{aligned} &\min_{X \in \mathbb{R}^{(n-k) \times k}} Q_L(\Xi; X) \\ &= \min_{X \in \mathbb{R}^{(n-k) \times k}} \max_{\gamma \in \mathcal{U}} \left(\frac{L}{2} \|X - \Xi\|_F^2 + \langle \nabla_2 q(\gamma, \Xi), X - \Xi \rangle + q(\gamma, \Xi) \right). \end{aligned}$$

There are a number of ways to solve for $X_L(\Xi)$. For example, when m is small, one may apply an interior point method to solve for γ in the dual problem:

$$\begin{aligned} &\max_{\gamma \in \mathcal{U}} \left[\min_{X \in \mathbb{R}^{(n-k) \times k}} \left(\frac{L}{2} \|X - \Xi\|_F^2 + \langle \nabla_2 q(\gamma, \Xi), X - \Xi \rangle + q(\gamma, \Xi) \right) \right] \\ &= \max_{\gamma \in \mathcal{U}} \left(-\frac{1}{2L} \|\nabla_2 q(\gamma, \Xi)\|_F^2 + q(\gamma, \Xi) \right). \end{aligned} \tag{8}$$

Note here that strong duality holds as the term inside the parenthesis is linear in γ and convex quadratic in X and \mathcal{U} is a compact convex set so we can apply Sion's Minimax Theorem [55]. An approximate primal solution \tilde{X} can then be reconstructed from an approximate solution $\tilde{\gamma}$ of the dual problem by setting $\tilde{X} = \Xi - \frac{\nabla_2 q(\tilde{\gamma}, \Xi)}{L}$.

Sticking with FOMs, one may apply the strongly convex excessive gap technique [45, Chapter 6.2] to compute the prox-map $X_L(\Xi)$ as well. We will rewrite $Q_L(\Xi; X)$ in a form that is more natural for applying the excessive gap technique [45, Algorithm 6.2.37]. Note that $\nabla_2 q(\gamma, \Xi) = A(\gamma)\Xi + B(\gamma)$. Thus, defining the matrix $G_{\text{obj}} := A_{\text{obj}}\Xi + B_{\text{obj}}$ and the linear operator $\mathcal{G} : \gamma \mapsto \sum_{i=1}^m \gamma_i (A_i \Xi + B_i)$, we have $\nabla_2 q(\gamma, \Xi) = A(\gamma)\Xi + B(\gamma) = G_{\text{obj}} + \mathcal{G}\gamma$. Hence, we arrive at

$$Q_L(\Xi; X) = \frac{L}{2} \|X - \Xi\|_F^2 + \langle G_{\text{obj}}, X - \Xi \rangle + \max_{\gamma \in \mathcal{U}} \{ \langle \mathcal{G}\gamma, X \rangle + (q(\gamma, \Xi) - \langle \mathcal{G}\gamma, \Xi \rangle) \}. \quad (9)$$

The inner saddle-point function is strongly convex in X and linear in γ so that we may approximate the prox-map by approximately solving a strongly convex-concave saddle point problem. Thus, applying [45, Theorem 6.2.4] to $Q_L(\Xi; X)$ in the form (9) gives the following result.

Theorem 4. *Initialize [45, Theorem 6.2.4] with initial iterate $\gamma_0 \in \mathcal{U}$. Let $(\tilde{\gamma}, \tilde{X})$ denote the output of [45, Algorithm 6.2.37] after*

$$O\left(\frac{\max_{\gamma \in \mathbf{S}^{m-1}} \|\mathcal{G}\gamma\|_F \cdot \max_{\gamma \in \mathcal{U}} \|\gamma - \gamma_0\|_2}{\sqrt{L\epsilon}}\right)$$

iterations. Here, each iteration may require two exact projections onto \mathcal{U} . Then,

$$Q_L(\Xi; \tilde{X}) - Q_L^*(\Xi) \leq Q_L(\Xi; \tilde{X}) - \left(q(\tilde{\gamma}_k, \Xi) - \frac{\|\nabla_2 q(\tilde{\gamma}_k, \Xi)\|_F^2}{2L} \right) \leq \epsilon. \quad (10)$$

Remark 5. For simplicity, in our numerical implementation of CertSDP, we opt to run the accelerated gradient descent method for simple sets [45, Algorithm 2.2.63] on the dual problem (8). \square

3.3 Putting the pieces together

We conclude this section by showing how to combine Theorems 3 and 4 to get a guarantee on the total iteration count (including iterations *within* the inexact prox-map calls). To this end, we will need an additional assumption on the norms of various quantities.

Assumption 4. Suppose Algorithm 1 starts at $X_0 = 0_{(n-k) \times k}$. Suppose $R > 0$ satisfies

$$\frac{\|\nabla_2 q(\tilde{\gamma}, X_0)\|_F}{L} = \frac{\|B(\tilde{\gamma})\|_F}{L} \leq R$$

where $\tilde{\gamma} \in \arg \max_{\gamma \in \mathcal{U}} q(\gamma, X_0)$. Let D denote the diameter of \mathcal{U} ; this is the natural scale parameter for the dual iterates. We will see soon that R is a natural scale parameter for the primal iterates $X_t, \Xi_t \in \mathbb{R}^{(n-k) \times k}$. Suppose $H \geq 1$ bounds

$$\frac{D \|\sum_{i=1}^m \gamma_i A_i\|_2}{\mu}, \quad \text{and} \quad \frac{D \|\sum_{i=1}^m \gamma_i B_i\|_F}{\mu \kappa R}$$

for all $\gamma \in \mathbf{S}^{m-1}$. We will assume algorithmic access to D , H , and R . \square

Lemma 7. Under Assumption 4, it holds that $Q(X_0) - \text{Opt}_{(\text{QMMP})} \leq \frac{\mu\kappa^2 R^2}{2}$. In particular, we may take $\text{gap}_0 = \frac{\mu\kappa^2 R^2}{2}$ in Algorithm 1. Then, for every $t \geq 0$, the iterate Ξ_t computed by Algorithm 1 satisfies

$$\|\Xi_t\|_F \leq 10\kappa R.$$

Proof. Let $\tilde{\gamma} \in \arg \max_{\gamma \in \mathcal{U}} q(\gamma, X_0)$. By μ -strong convexity of $Q(X)$, we have that

$$\begin{aligned} Q(X) &\geq q(\tilde{\gamma}, X) \\ &\geq q(\tilde{\gamma}, X_0) + \langle \nabla_2 q(\tilde{\gamma}, X_0), X - X_0 \rangle + \frac{\mu}{2} \|X - X_0\|_F^2 \\ &= Q(X_0) - \frac{1}{2\mu} \|\nabla_2 q(\tilde{\gamma}, X_0)\|_F^2 + \frac{\mu}{2} \left\| X - X_0 + \frac{\nabla_2 q(\tilde{\gamma}, X_0)}{\mu} \right\|_F^2. \end{aligned}$$

In particular, taking $X = \arg \min_{X \in \mathbb{R}^{(n-k) \times k}} Q(X)$ gives

$$Q(X_0) - \text{Opt}_{(\text{QMMP})} \leq \frac{\|\nabla_2 q(\tilde{\gamma}, X_0)\|_F^2}{2\mu} \leq \frac{\mu\kappa^2 R^2}{2},$$

where the last inequality follows from Assumption 4. This proves the first claim. Next, by Theorem 3, we have that for all $t \geq 0$, that $Q(X_t) - Q(X_0) \leq Q(X_t) - \text{Opt}_{(\text{QMMP})} \leq 2\mu\kappa^2 R^2$ and hence

$$\frac{\mu}{2} \left\| X_t - X_0 + \frac{\nabla_2 q(\tilde{\gamma}, X_0)}{\mu} \right\|_F^2 \leq Q(X_t) - Q(X_0) + \frac{\|\nabla_2 q(\tilde{\gamma}, X_0)\|_F^2}{2\mu} \leq \frac{5\mu\kappa^2 R^2}{2}.$$

Using the assumption $X_0 = 0_{(n-k) \times k}$ in Assumption 4 and applying triangle inequality together with the bound $\|\nabla_2 q(\tilde{\gamma}, X_0)\|_F^2 \leq \mu^2 \kappa^2 R^2$ derived from Assumption 4, we deduce that for all $t \geq 0$,

$$\|X_t\|_F \leq (1 + \sqrt{5}) \kappa R.$$

Then, as $\Xi_{t+1} = X_{t+1} + \frac{1-\alpha}{1+\alpha} (X_{t+1} - X_t)$, we have

$$\|\Xi_{t+1}\|_F \leq 3(1 + \sqrt{5}) \kappa R \leq 10\kappa R. \quad \blacksquare$$

With this bound on $\|\Xi_t\|_F$ we are now able to bound the operator norm $\max_{\gamma \in \mathbf{S}^{m-1}} \|\mathcal{G}\gamma\|_F$ in Theorem 4.

Lemma 8. Suppose Assumption 4 holds and we set $\text{gap}_0 = \frac{\mu\kappa^2 R^2}{2}$ in Algorithm 1. Then, for every iterate $t \geq 0$, we have

$$\max_{\gamma \in \mathbf{S}^{m-1}} \|\mathcal{G}\gamma\|_F \leq 11 \frac{\mu\kappa HR}{D}.$$

Proof. Recall that by definition, the linear operator \mathcal{G} maps γ to $\sum_{i=1}^m \gamma_i (A_i \Xi_t + B_i)$. Thus, for any $\gamma \in \mathbf{S}^{m-1}$,

$$\begin{aligned} \|\mathcal{G}\gamma\|_F &= \left\| \sum_{i=1}^m \gamma_i (A_i \Xi_t + B_i) \right\|_F \\ &\leq \left\| \sum_{i=1}^m \gamma_i A_i \right\|_2 \|\Xi_t\|_F + \left\| \sum_{i=1}^m \gamma_i B_i \right\|_F \\ &\leq 11 \frac{\mu\kappa HR}{D}. \quad \blacksquare \end{aligned}$$

The following theorem gives the iteration complexity of Algorithm 1 instantiated with the excessive gap technique to compute the prox-map. It follows as a corollary to Theorems 3 and 4 and Lemma 8.

Theorem 5. *Let $q(\gamma, X)$ and \mathcal{U} satisfy Assumptions 3 and 4. Suppose gap_0 is set to $\frac{\mu\kappa^2 R^2}{2}$ in Algorithm 1. Let X_t denote the iterates produced by Algorithm 1 with starting point $X_0 = 0_{(n-k) \times k}$. Then, for all $t \geq 1$, the iterate X_t satisfies*

$$Q(X_t) - \text{Opt}_{(\text{QMMP})} \leq \left(1 - \frac{\alpha}{2}\right)^t (2\mu\kappa^2 R^2).$$

In particular, $Q(X_T) - \text{Opt}_{(\text{QMMP})} \leq \epsilon$ after at most

$$T = O\left(\sqrt{\kappa} \log\left(\frac{\mu\kappa^2 R^2}{\epsilon}\right)\right)$$

outer iterations of Algorithm 1. The iterate X_T is computed after a total (including iterations within the inexact prox-map computations) of $O\left(\frac{\kappa^{5/4} H R \sqrt{L}}{\sqrt{\epsilon}}\right)$ iterations.

Proof. We will take T to be the first positive integer such that

$$\left(1 - \frac{\alpha}{2}\right)^T (2\mu\kappa^2 R^2) \leq \epsilon.$$

Clearly, $T = O(\sqrt{\kappa} \log(\kappa L R^2 / \epsilon))$. Next, if $T > 1$, then by the maximality of T we have

$$\left(1 - \frac{\alpha}{2}\right)^T \geq \left(1 - \frac{\alpha}{2}\right) \left(\frac{\epsilon}{2\mu\kappa^2 R^2}\right) \geq \frac{\epsilon}{4\mu\kappa^2 R^2}.$$

From (7) and $\text{gap}_0 = \frac{\mu\kappa^2 R^2}{2}$, we deduce that $\epsilon_t \geq \frac{\mu\kappa R^2}{2} (1 - \frac{\alpha}{2})^t \frac{\alpha}{2}$. By Lemma 8 and Assumption 4, we may bound

$$\max_{\gamma \in \mathbf{S}^{m-1}} \|\mathcal{G}\gamma\|_F \cdot \max_{\gamma \in \mathcal{U}} \|\gamma - \gamma_0\|_2 \leq 11\mu\kappa H R.$$

Thus, X_t can be computed in

$$O\left(\frac{\mu\kappa H R}{\sqrt{L\epsilon_t}}\right) = O\left(\kappa^{1/4} H (1 - \alpha/2)^{-t/2}\right)$$

iterations. Summing over the first T outer iterations and observing our lower bound on $(1 - \frac{\alpha}{2})^T$, we have that

$$\sum_{t=0}^T \left(1 - \frac{\alpha}{2}\right)^{-t/2} \leq O\left(\frac{(1 - \frac{\alpha}{2})^{-T/2}}{\alpha}\right) = O\left(\frac{\kappa R \sqrt{L}}{\sqrt{\epsilon}}\right). \quad \blacksquare$$

4 Solving k -exact SDPs via strongly convex QMMP algorithms

In this section, we show how to combine Theorems 1 and 5 to develop first-order methods for approximately solving rank- k exact QMP-like SDPs. We will use the following notion of an approximate solution to (SDP).

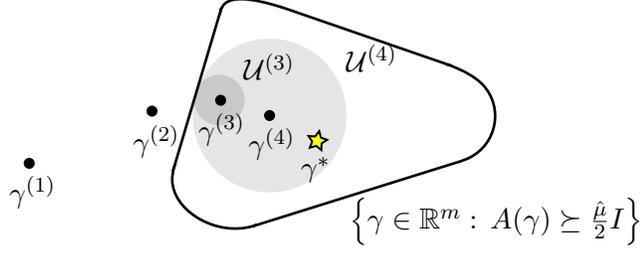


Figure 1: CertSDP (Algorithm 2) produces a series of iterates $\gamma^{(i)} \rightarrow \gamma^*$. For each $\gamma^{(i)}$, CertSDP constructs a ball $\mathcal{U}^{(i)}$ around $\gamma^{(i)}$. Intuitively, we want to pick $\mathcal{U}^{(i)}$ to be the largest ball around $\gamma^{(i)}$ for which we can solve the associated QMMP efficiently, in hopes of enclosing γ^* . We will thus choose $\mathcal{U}^{(i)}$ to satisfy certain regularity estimates (see (11) and Lemma 11). At the minimum, we will ensure $A(\gamma) \succeq \hat{\mu}/2$ for all $\gamma \in \mathcal{U}^{(i)}$.

Definition 4. We will say that $\tilde{Y} \in \mathbb{S}^n$ is ϵ -optimal and δ -feasible for (SDP) if

$$\langle M_{\text{obj}}, \tilde{Y} \rangle \leq \text{Opt}_{\text{(SDP)}} + \epsilon, \quad \left(\sum_{i=1}^m (\langle M_i, \tilde{Y} \rangle + d_i)^2 \right)^{1/2} \leq \delta,$$

and $\tilde{Y} \succeq 0$. □

The final piece towards this goal is developing algorithms for constructing a certificate of strict complementarity \mathcal{U} .

By Definition 2, the properties we need to ensure for \mathcal{U} are that $\gamma^* \in \mathcal{U}$ and $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{U}$. We will construct \mathcal{U} by taking it to be an ℓ_2 -ball centered at a sufficiently accurate estimate $\tilde{\gamma}$ of γ^* .

We begin by verifying that $A(\gamma^*) \succ 0$.

Lemma 9. Suppose $M^*, Y^* \in \mathbb{S}_+^n$ have rank $n - k$ and k respectively and that $\langle M^*, Y^* \rangle = 0$. Let W be an $(n - k)$ -dimensional subspace. Then, $M_W^* \succ 0$ if and only if $Y_{W^\perp}^* \succ 0$.

Proof. It suffices to prove the forward direction as we may interchange the roles of Y^* and M^* .

We prove the contrapositive. Suppose $Y_{W^\perp}^* \not\succeq 0$ so that $\ker(Y_{W^\perp}^*)$ is nontrivial. As $Y^* \succeq 0$, we have that in fact $\ker(Y^*) \cap W^\perp$ is nontrivial. Then, $\text{range}(Y^*)$ is a k -dimensional subspace contained in $(\ker(Y^*) \cap W^\perp)^\perp$. Similarly, W is an $(n - k)$ -dimensional subspace contained in $(\ker(Y^*) \cap W^\perp)^\perp$. Then, as $(\ker(Y^*) \cap W^\perp)^\perp$ has dimension at most $n - 1$, we deduce that $\text{range}(Y^*) \cap W$ is nontrivial and $\langle Y^*, M^* \rangle > 0$, a contradiction. ■

Clearly then, for all $\tilde{\gamma}$ close enough to γ^* , we have that $A(\tilde{\gamma}) \succ 0$ and there exists some $r > 0$ such that $\tilde{\mathcal{U}} := \mathbb{B}(\tilde{\gamma}, r)$ satisfies $A(\gamma) \succ 0$ for all $\gamma \in \tilde{\mathcal{U}}$. We consider one setting for r below. It remains to ask, does the condition that $\gamma^* \in \tilde{\mathcal{U}}$ hold? Below, we show that this condition indeed holds when $\tilde{\gamma}$ is a sufficiently accurate estimate of γ^* and that we can effectively check this condition using CautiousAGD.

Assumption 5. Suppose we have algorithmic access to

- parameters $0 < \hat{\mu} \leq \hat{L}$ such that $\hat{\mu}I \preceq A(\gamma^*) \preceq \hat{L}I$,
- parameters $\hat{R}_p, \hat{R}_d > 0$ such that $\|X^*\|_F \leq \hat{R}_p$ and $\|\gamma^*\|_2 \leq \hat{R}_d$, and

- a parameter $\hat{\rho} > 0$ upper bounding

$$\frac{\hat{\mu}}{\hat{R}_d}, \quad \left\| \sum_{i=1}^m \gamma_i A_i \right\|_2, \quad \text{and} \quad \frac{\|\sum_{i=1}^m \gamma_i B_i\|_F}{\hat{R}_p} \quad \forall \gamma \in \mathbf{S}^{m-1}.$$

For notational simplicity, we will additionally assume $\hat{R}_d \geq 1$. This is not strictly necessary and simply allows us to write $O(\hat{R}_d)$ in place of $O(1 + \hat{R}_d)$. \square

Note from the identity $X^* = -A(\gamma^*)^{-1}B(\gamma^*)$ that $\|B(\gamma^*)\|_F \leq \hat{L}\hat{R}_p$. Now, suppose $\gamma^{(1)}, \gamma^{(2)}, \dots$ is a sequence converging to γ^* (such a sequence can be constructed via subgradient methods [35]; see also [23, Section 6.2.2]). Given $\gamma^{(i)}$, define

$$r^{(i)} := \min \left(\frac{\hat{\mu}}{2\hat{\rho}}, 2\hat{R}_d - \|\gamma^{(i)}\|_2, \frac{\lambda_{\min}(A(\gamma^{(i)})) - \hat{\mu}/2}{\hat{\rho}}, \frac{2\hat{L} - \lambda_{\max}(A(\gamma^{(i)}))}{\hat{\rho}}, \frac{2\hat{L}\hat{R}_p - \|B(\gamma^{(i)})\|_F}{\hat{\rho}\hat{R}_p} \right). \quad (11)$$

If $r^{(i)}$ is positive, define $\mathcal{U}^{(i)} := \mathbb{B}(\gamma^{(i)}, r^{(i)})$.

We present three lemmas below. The first lemma states that $r^{(i)}$ is positive and $\gamma^* \in \mathcal{U}^{(i)}$ for all $\gamma^{(i)}$ sufficiently close to γ^* . The second lemma establishes parameters for which the regularity conditions of Assumption 4 hold for $q(\gamma, X)$ along with $\mathcal{U}^{(i)}$. Finally, the third lemma shows that for each $\mathcal{U}^{(i)}$, an approximate solution of the corresponding strongly convex QMMP (which can be computed using Algorithm 1) can be used to either produce an approximate optimizer of the underlying SDP or declare $\gamma^* \notin \mathcal{U}^{(i)}$.

Lemma 10. *Suppose Assumption 5 holds. Then, $r^{(i)}$ is positive and $\gamma^* \in \mathcal{U}^{(i)}$ if*

$$\|\gamma^{(i)} - \gamma^*\|_2 \leq \frac{\hat{\mu}}{4\hat{\rho}}.$$

Proof. Let $r := \|\gamma^{(i)} - \gamma^*\|_2$. Using Assumption 5, we may bound the individual terms within the definition of $r^{(i)}$ as

$$\begin{aligned} 2\hat{R}_d - \|\gamma^{(i)}\|_2 &\geq \hat{R}_d - r \geq \frac{\hat{\mu}}{\hat{\rho}} - r, \\ \frac{\lambda_{\min}(A(\gamma^{(i)})) - \hat{\mu}/2}{\hat{\rho}} &\geq \frac{\hat{\mu}/2 - \hat{\rho}r}{\hat{\rho}} = \frac{\hat{\mu}}{2\hat{\rho}} - r, \\ \frac{2\hat{L} - \lambda_{\max}(A(\gamma^{(i)}))}{\hat{\rho}} &\geq \frac{\hat{L} - \hat{\rho}r}{\hat{\rho}} = \frac{\hat{L}}{\hat{\rho}} - r, \text{ and} \\ \frac{2\hat{L}\hat{R}_p - \|B(\gamma^{(i)})\|_F}{\hat{\rho}\hat{R}_p} &\geq \frac{\hat{L} - \hat{\rho}r}{\hat{\rho}} = \frac{\hat{L}}{\hat{\rho}} - r. \end{aligned}$$

Thus, $r^{(i)} \geq \min\left(\frac{\hat{\mu}}{2\hat{\rho}}, \frac{\hat{\mu}}{2\hat{\rho}} - r\right) = \frac{\hat{\mu}}{2\hat{\rho}} - r$. Then, when $r \leq \frac{\hat{\mu}}{4\hat{\rho}}$, we have $r^{(i)} > 0$ and furthermore, $r^{(i)} \geq r = \|\gamma^{(i)} - \gamma^*\|_2$. \blacksquare

Lemma 11. Suppose Assumption 5 holds and $r^{(i)}$ is positive. Then, $q(\gamma, X)$ and $\mathcal{U}^{(i)}$ satisfy Assumption 4 with $\mu = \frac{\hat{\mu}}{2}$, $L = 2\hat{L}$, $R = \hat{R}_p$, $D = 2r^{(i)}$, and $H = 2$.

Proof. Begin by noting that for all $\gamma \in \mathcal{U}^{(i)}$,

$$\frac{\hat{\mu}}{2}I \preceq A(\gamma^{(i)}) - r^{(i)}\hat{\rho}I \preceq A(\gamma) \preceq A(\gamma^{(i)}) + r^{(i)}\hat{\rho}I \preceq 2\hat{L}I.$$

Let $\tilde{\gamma} \in \arg \max_{\gamma \in \mathcal{U}^{(i)}} q(\gamma, 0_{(n-k) \times k})$. Then,

$$\|B(\tilde{\gamma})\|_F \leq \|B(\gamma^{(i)})\|_F + \hat{\rho}r^{(i)}\hat{R}_p \leq 2\hat{L}\hat{R}_p = LR.$$

Next, for $\gamma \in \mathbf{S}^{m-1}$

$$\begin{aligned} \frac{D \|\sum_{i=1}^m \gamma_i A_i\|_2}{\mu} &\leq \frac{4r^{(i)}\hat{\rho}}{\hat{\mu}} \leq 2 \\ \frac{D \|\sum_{i=1}^m \gamma_i B_i\|_F}{LR} &\leq \frac{r^{(i)}\hat{\rho}}{\hat{L}} \leq 1/2. \end{aligned} \quad \blacksquare$$

Lemma 12. Suppose Assumption 5 holds, $r^{(i)}$ is positive, and $0 < \epsilon \leq 9\hat{\rho}\hat{R}_d\hat{R}_p^2$. Set $\delta := \frac{\hat{\mu}\epsilon^2}{(9\hat{\rho}\hat{R}_d\hat{R}_p)^2}$ and $\eta := \frac{4\epsilon}{9\hat{R}_d}$. Suppose $\tilde{X} \in \mathbb{R}^{(n-k) \times k}$ satisfies

$$Q_{\mathcal{U}^{(i)}}(\tilde{X}) \leq \min_{X \in \mathbb{R}^{(n-k) \times k}} Q_{\mathcal{U}^{(i)}}(X) + \delta.$$

Then,

- If $\gamma^* \in \mathcal{U}^{(i)}$, then $Y(\tilde{X})$ is η -feasible.
- If $Y(\tilde{X})$ is η -feasible, then $Y(\tilde{X})$ is ϵ -optimal and ϵ -feasible.

Proof. Suppose $\gamma^* \in \mathcal{U}^{(i)}$ and define $\Delta := \tilde{X} - X^*$. By strong convexity and Theorem 1, we have that $\frac{\hat{\mu}}{2} \|\Delta\|_F^2 \leq \delta$. Next, recalling that $q_i(X^*) = 0$ for all $i \in [m]$, we deduce

$$\begin{aligned} \left(\sum_{i=1}^m (\langle M_i, Y(\tilde{X}) \rangle + d_i)^2 \right)^{1/2} &= \left(\sum_{i=1}^m q_i(\tilde{X})^2 \right)^{1/2} \\ &= \max_{\|\gamma\|_2=1} \sum_{i=1}^m \gamma_i \left(\frac{\text{tr}(\Delta^\top A_i \Delta)}{2} + \langle A_i X^* + B_i, \Delta \rangle \right) \\ &\leq \hat{\rho} \left(\frac{\delta}{\hat{\mu}} \right) + \sqrt{8}\hat{\rho}\hat{R}_p \sqrt{\frac{\delta}{\hat{\mu}}} \\ &\leq \frac{4\hat{\rho}\hat{R}_p}{\sqrt{\hat{\mu}}} \sqrt{\delta} = \eta. \end{aligned}$$

Here, the first inequality follows from $\|\Delta\|_F^2 \leq \frac{2\delta}{\hat{\mu}}$ and Assumption 5, and the last inequality follows as $\delta = \frac{\hat{\mu}\epsilon^2}{(9\hat{\rho}\hat{R}_d\hat{R}_p)^2} \leq \hat{\mu}\hat{R}_p^2$ since $0 < \epsilon \leq 9\hat{\rho}\hat{R}_d\hat{R}_p^2$.

Now, suppose $Y(\tilde{X})$ is η -feasible. Note that $\eta \leq \epsilon$ (as $\hat{R}_d \geq 1$) and thus $Y(\tilde{X})$ is immediately ϵ -feasible. Let $\tilde{\gamma} \in \arg \max_{\gamma \in \mathcal{U}^{(i)}} q(\gamma, \tilde{X})$ so that $Q_{\mathcal{U}^{(i)}}(\tilde{X}) = q(\tilde{\gamma}, \tilde{X})$. Then,

$$\begin{aligned} \langle M_{\text{obj}}, Y(\tilde{X}) \rangle &= q_{\text{obj}}(\tilde{X}) = q(\tilde{\gamma}, \tilde{X}) - \sum_{i=1}^m \tilde{\gamma}_i q_i(\tilde{X}) \\ &\leq Q_{\mathcal{U}^{(i)}}(\tilde{X}) + \|\tilde{\gamma}\|_2 \eta \\ &\leq \left(\min_{X \in \mathbb{R}^{(n-k) \times k}} Q_{\mathcal{U}^{(i)}}(X) + \delta \right) + 2\hat{R}_d \eta \\ &\leq \text{Opt}_{(\text{SDP})} + (\delta + 2\hat{R}_d \eta), \end{aligned}$$

where the first inequality follows from the η -feasibility of $Y(\tilde{X})$, the second inequality from the premise of the lemma on \tilde{X} and the fact that $\|\tilde{\gamma}\|_2 \leq \|\tilde{\gamma} - \gamma^{(i)}\|_2 + \|\gamma^{(i)}\|_2 \leq 2\hat{R}_d$ (this holds because $\tilde{\gamma} \in \mathcal{U}^{(i)}$, $\mathcal{U}^{(i)}$ is the ℓ_2 -ball of radius $r^{(i)}$ centered at $\gamma^{(i)}$, and by definition of $r^{(i)}$ we have $r^{(i)} \leq 2\hat{R}_d - \|\gamma^{(i)}\|_2$). We may then use the definitions of δ and η to bound

$$\delta + 2\hat{R}_d \eta = \frac{\hat{\mu} \epsilon^2}{(9\hat{\rho} \hat{R}_d \hat{R}_p)^2} + \frac{8\epsilon}{9} \leq \frac{\hat{\mu} \epsilon}{9\hat{\rho} \hat{R}_d} + \frac{8\epsilon}{9} \leq \epsilon.$$

Here, the first inequality follows from the upper bound on ϵ and the second inequality follows from $\frac{\hat{\mu}}{\hat{R}_d} \leq \hat{\rho}$ (Assumption 5). This then shows that $Y(\tilde{X})$ is ϵ -optimal. \blacksquare

We are now ready to present our full algorithm for computing approximate solutions to (SDP). CertSDP (Algorithm 2) assumes access to a sequence $\gamma^{(i)} \rightarrow \gamma^*$ and applies a guess-and-double scheme to guess when $\|\gamma^{(i)} - \gamma^*\|_2$ is sufficiently small. It then applies Algorithm 1 to either compute an ϵ -optimal ϵ -feasible solution $Y(\tilde{X})$ or to declare that $\gamma^* \notin \mathcal{U}^{(i)}$.

Algorithm 2 CertSDP

Given a rank- k exact QMP-like SDP satisfying Assumption 5, a sequence $\gamma^{(1)}, \gamma^{(2)}, \dots \rightarrow \gamma^*$, and $0 < \epsilon \leq 9\hat{\rho} \hat{R}_d \hat{R}_p^2$

1. Set δ and η as in Lemma 12
2. For each $i = 2^0, 2^1, 2^2, \dots$
 - If $r^{(i)} > 0$
 - (a) Let $\mathcal{U}^{(i)} := \mathbb{B}(\gamma^{(i)}, r^{(i)})$ and compute \tilde{X} satisfying

$$Q_{\mathcal{U}^{(i)}}(\tilde{X}) \leq \min_{X \in \mathbb{R}^{(n-k) \times k}} Q_{\mathcal{U}^{(i)}}(X) + \delta$$

using Algorithm 1

- (b) If $Y(\tilde{X})$ is η -feasible, output $Y(\tilde{X})$
-

The next theorem gives rigorous guarantees on CertSDP and follows from Lemmas 10 to 12 and Theorem 5.

Theorem 6. *Suppose (SDP) is a rank- k exact QMP-like SDP satisfying Assumption 5, $\gamma^{(1)}, \gamma^{(2)}, \dots \rightarrow \gamma^*$ and $0 < \epsilon \leq 9\hat{\rho} \hat{R}_d \hat{R}_p^2$. Let T be such that $\|\gamma^{(t)} - \gamma^*\|_2 \leq \frac{\hat{\mu}}{4\hat{\rho}}$ for all $t \geq T$. Then, CertSDP (Algorithm 2) accesses at most $2T$ iterates of the sequence $\gamma^{(i)}$ and outputs an ϵ -optimal and ϵ -feasible*

solution in

$$O\left(\sqrt{\hat{\kappa}} \log\left(\frac{\hat{\kappa}\hat{\rho}\hat{R}_p\hat{R}_d}{\epsilon}\right) \cdot \log(T)\right) \text{ prox-map calls, and}$$

$$O\left(\frac{\hat{\kappa}^{7/4}\hat{\rho}\hat{R}_p^2\hat{R}_d}{\epsilon} \cdot \log(T)\right) \text{ iterations within all prox-map calls.}$$

5 Numerical experiments

In this section, we investigate the numerical performance of our new FOM, CertSDP, on rank- k exact QMP-like SDPs that are both large and sparse. Specifically, we consider random instances of distance-minimization QMPs and their primal and dual SDP relaxations of the form

$$\begin{aligned} & \inf_{X \in \mathbb{R}^{(n-k) \times k}} \left\{ \frac{\|X\|_F^2}{2} : \operatorname{tr}\left(\frac{X^\top A_i X}{2}\right) + \langle B_i, X \rangle + c_i = 0, \forall i \in [m] \right\} \\ & \geq \inf_{Y \in \mathbb{S}^n} \left\{ \left\langle \begin{pmatrix} I_{n-k}/2 & \\ & 0_k \end{pmatrix}, Y \right\rangle : \left\langle \begin{pmatrix} A_i/2 & B_i/2 \\ B_i^\top/2 & \frac{c_i}{k} I_k \end{pmatrix}, Y \right\rangle = 0, \forall i \in [m] \right. \\ & \quad \left. Y = \begin{pmatrix} * & * \\ * & I_k \end{pmatrix} \succeq 0 \right\} \\ & \geq \sup_{\gamma \in \mathbb{R}^m, T \in \mathbb{S}^k} \left\{ \operatorname{tr}(T) : \begin{pmatrix} A(\gamma)/2 & B(\gamma)/2 \\ B(\gamma)^\top/2 & \frac{c(\gamma)}{k} I_k - T \end{pmatrix} \succeq 0 \right\}. \end{aligned} \quad (12)$$

In our instance generation procedure, we ensure that equality holds throughout this chain of inequalities.

We will compare the performance of CertSDP on instances of (12) to that of several first-order methods from the literature: the complementary slackness SDP algorithm (CSSDP) [23], ProxSDP [56], and the splitting cone solver (SCS) [47]. We discuss these algorithms and relevant implementation details in Section 5.1 and the instance generation procedure in Section 5.2 before presenting the numerical results in Section 5.3.

All algorithms and experiments are implemented in Julia and run on a machine with an AMD Opteron 4184 processor with 12 CPUs and 70GB of RAM. Our code is publicly available at:

<https://github.com/alexlihengwang/CertSDP>

We additionally implement a variant of SketchyCGAL [68] adapted to our setting (see Section 5.1). We choose to leave this algorithm off of our large-format experiments as its performance was very similar to that of CSSDP in preliminary experimentation (see Appendix B).

5.1 Implementation details

CertSDP. We implemented CertSDP (Algorithm 2) as presented in this paper except a few modifications. In addition to simplifying the overall algorithm, these modifications enable CertSDP to be run *without* knowledge of the parameters $\hat{\mu}$ and \hat{L} . While the convergence guarantees of Theorem 6 may no longer hold, we find empirically that CertSDP continues to perform very effectively with these modifications.

- We instantiate CertSDP with Accelegrad [35] as the iterative method for producing iterates $\gamma^{(i)}$. As in [23], we apply Accelegrad to the penalized dual problem

$$\max_{\gamma \in \mathbb{R}^m, T \in \mathbb{S}^k} \text{tr}(T) + \text{penalty} \cdot \min \left(0, \lambda_{\min} \begin{pmatrix} A(\gamma)/2 & B(\gamma)/2 \\ B(\gamma)^\top/2 & \frac{c(\gamma)}{k} I_k - T \end{pmatrix} \right)$$

for some large value for the penalty parameter. It can be shown that the optimal value and optimizers of this penalized dual problem coincide with that of the dual SDP whenever the penalty parameter is larger than $\text{tr}(Y^*)$; see [23]. In our experiments, we set the penalty parameter to be $20 \cdot \text{tr}(Y^*)$.

- In practice, it is extremely cheap to solve $(\text{QMMP}_{\mathcal{U}})$ even to high accuracy. Thus, we replace the guess-and-double scheme in Algorithm 2 with a linear schedule, i.e., we solve $(\text{QMMP}_{\mathcal{U}})$ once every ≈ 250 iterations. Additionally, we replace the excessive gap technique used in Theorem 5 with accelerated gradient descent (see Remark 5).
- We set

$$r^{(i)} = \frac{1}{\hat{\rho}} \cdot \frac{\lambda_{\max}(A(\gamma^{(i)}))\lambda_{\min}(A(\gamma^{(i)}))}{\lambda_{\max}(A(\gamma^{(i)})) + \lambda_{\min}(A(\gamma^{(i)}))}$$

if $A(\gamma^{(i)}) \succ 0$, and $r^{(i)} = 0$ else. Equivalently, $\mathcal{U}^{(i)} := \mathbb{B}(\gamma^{(i)}, r^{(i)})$ is the largest ball centered at $\gamma^{(i)}$ for which the condition number of $A(\gamma)$ for any $\gamma \in \mathbb{B}(\gamma^{(i)}, r^{(i)})$ is guaranteed to be at most twice the condition number of $A(\gamma^{(i)})$. Note that it still holds that $A(\gamma) \succ 0$ for all $\gamma \in \mathcal{U}^{(i)}$ (as long as $r^{(i)}$ is positive) and that $\gamma^* \in \mathcal{U}^{(i)}$ for all $\gamma^{(i)}$ close enough to γ^* .

- In CautiousAGD (Algorithm 1), we terminate early if $\max_{i \in [m]} |q_i(X_t)|$ does not decrease to zero geometrically. Indeed, this can only happen if $\gamma^* \notin \mathcal{U}^{(i)}$.
- Theorem 4 gives an *a priori* guarantee on the number of inner iterations required for solving each prox-map. Instead of using this number of iterations, in our code, we will monitor the saddle point gap, i.e., the second term in (10), and break as soon as the saddle point gap is small enough.
- We warm-start the iterate X in CautiousAGD using the last iterate of the previous run of CautiousAGD and warm-start γ in the prox-map computation using the last iterate of the previous run of the prox-map computation.
- Unless the time limit is met first, the overall algorithm is terminated once CautiousAGD produces a (10^{-13}) -optimal solution of $(\text{QMMP}_{\mathcal{U}})$ that satisfies $\max_{i \in [m]} |q_i(X_t)| \leq 10^{-13}$.

CSSDP. The complementary slackness SDP algorithm (CSSDP) [23] similarly constructs a sequence of iterates $\gamma^{(i)} \rightarrow \gamma^*$ and occasionally solves a compressed k -dimensional SDP [23, MinFeasSDP] in the vector space corresponding to the k -many minimum eigenvalues of the slack matrix $M(\gamma^{(i)})$. As in our implementation of CertSDP, we instantiate CSSDP with Accelegrad [35] as the iterative method for producing iterates $\gamma^{(i)}$ and solve the compressed SDP once every ≈ 250 iterations. The compressed SDPs are solved using SCS solver with all error parameters set to 10^{-13} . Since CSSDP needs to solve the compressed SDP frequently, we make sure to instantiate the optimization problem just once in order to amortize the cost of allocating the $k \times k$ symmetric matrix variable.

SketchyCGAL. Yurtsever et al. [68] observe that one may track any *linear image* of the primal matrix iterates (as opposed to the matrix iterate itself) in the CGAL [67] algorithm. Combining this observation with the Nyström sketch gives SketchyCGAL. In our code, we implement a variant of this idea, where we replace the Nyström sketch with the linear map sending a matrix in \mathbb{S}^n to its top-right $(n - k) \times k$ submatrix. We omit this algorithm in our large-format experiments as its performance was very similar to that of CSSDP in preliminary experiments (see Appendix B).

ProxSDP and SCS. ProxSDP [56] and the splitting cone solver (SCS) [47] are first-order methods that can be used to tackle large-scale SDPs. ProxSDP combines the primal-dual hybrid gradient method with an approximate projection operation that allows it to replace a full eigendecomposition with a partial one whenever the rank of the true SDP solution is small. SCS employs a first-order method to tackle the homogeneous self-dual embedding but does not explicitly take advantage of possible low rank solutions.

In our experiments, we pass the SDP relaxations of our QMPs to the corresponding Julia interfaces `ProxSDP.jl` and `SCS.jl` with all error parameters set to 10^{-13} . In contrast to CertSDP and CSSDP, which achieve storage optimality, ProxSDP and SCS both store matrix iterates and thus require substantially more memory.

5.2 Random instance generation

We generate random sparse instances of distance-minimization QMPs (12) as follows: Let $(n, k, m, \mu^*, \mathbf{nnz})$ be input parameters. Here, (n, k, m) control the size of (12), μ^* is the desired value of $\lambda_{\min}(A(\gamma^*))$ and \mathbf{nnz} approximately controls the number of nonzero entries in each A_1, \dots, A_m .

- Let $A_1, \dots, A_m \in \mathbb{S}^{n-k}$ be sparse symmetric matrices each with $\approx \mathbf{nnz}$ nonzero entries that are i.i.d. normal. We scale A_1, \dots, A_m such that $\|A_i\|_2 = 1$ for all $i \in [m]$.
- Let $B_1, \dots, B_m \in \mathbb{R}^{(n-k) \times k}$ be matrices where all entries are i.i.d. normal. We scale B_1, \dots, B_m such that $\|B_i\|_F = 1$ for all $i \in [m]$.
- Pick a direction $\hat{\gamma}$ uniformly from the surface of the sphere \mathbf{S}^{m-1} , then set $\gamma^* := r\hat{\gamma}$ where $r > 0$ solves $\lambda_{\min}(A(\gamma^*)) = 1 + r\lambda_{\min}(\sum_{i=1}^m \hat{\gamma}_i A_i) = \mu^*$. Let $X^* := -A(\gamma^*)^{-1}B(\gamma^*)$.
- Finally, for each $i \in [m]$, set c_i such that $\text{tr}\left(\frac{X^{*\top} A_i X^*}{2}\right) + \langle B_i, X^* \rangle + c_i = 0$.

Exactness is guaranteed to hold throughout (12) as (γ^*, T^*) , where

$$T^* := \frac{c(\gamma^*)}{k} I_k - \frac{B(\gamma^*)^\top A(\gamma^*)^{-1} B(\gamma^*)}{2},$$

achieves the value $\frac{\|X^*\|_F^2}{2}$ in the third line of (12) (see Appendix A).

5.3 Numerical results

To investigate the scalability of CertSDP in terms of n , we fix $k = 10$, $m = 10$, $\mu^* = 0.1$ and $\mathbf{nnz} = n$. Note that in this regime, the A_i matrices are each individually very sparse with approximately one nonzero entry per row or column. We then vary n such that the height of the matrix variable $X \in \mathbb{R}^{(n-k) \times k}$, i.e., $n - k$, takes the values 10^3 , 10^4 , 10^5 . For each value of $n - k$, we generate 10 random instances of (12) according to Section 5.2 and measure the time, error, and memory consumption of the tested algorithms.

Remark 6. We measure the memory consumption of each algorithm by monitoring the virtual memory size (*vsz*) of the process throughout the run of the algorithm and report the difference between the maximum value and the starting value. This is the same measurement that is performed in [68]. We caution that this number should only be treated as a *very rough* estimate of the storage requirements. Indeed, virtual memory need not be allocated at all for small enough programs (so that some algorithms register as using no memory at all for small enough values of $n - k$) and furthermore, when it is allocated, it is not always fully used. Experimentally, we found that on our machine, storage of up to ≈ 1.0 MB was often measured as not using any memory at all. We report such measurements as 0.0 MB in our tables (Tables 1 to 3) and as 1.0 MB in our log-scale plot Figure 2. \square

We ran each algorithm with time limits of 3×10^3 , 10^4 , and 5×10^4 seconds for $n - k = 10^3$, 10^4 , 10^5 respectively. SCS is not tested for $n - k = 10^4$ as it was unable to complete a single iteration within the time limits and utilized over 70GB of memory. Similarly, ProxSDP and SCS were not tested for $n - k = 10^5$ as both came to complete failures due to excessive memory allocation.

Detailed numerical results are reported in Tables 1 to 3 for $n - k = 10^3$, 10^4 , and 10^5 respectively. The average memory usage of the four algorithms is plotted in Figure 2. We compare the convergence behavior of CertSDP with that of CSSDP on a single instance of each size in Figure 3. The plots on the left of Figure 3 show the primal squared distance $\|X - X^*\|_F^2$ and the dual suboptimality

$$\text{Opt}_{(12)} - \left(\text{tr}(T) + \text{penalty} \cdot \min \left(0, \lambda_{\min} \begin{pmatrix} A(\gamma)/2 & B(\gamma)/2 \\ B(\gamma)^\top/2 & \frac{c(\gamma)}{k} I_k - T \end{pmatrix} \right) \right)$$

for the iterates produced by CertSDP and CSSDP as a function of time. The plots on the right of Figure 3 show the primal squared distance for the iterates produced by CertSDP within the final call to CautiousAGD.

Algorithm	time (s)	std.	$\ X - X^*\ _F^2$	std.	memory (MB)	std.
CertSDP	1.3×10^3	7.6×10^2	1.9×10^{-22}	4.2×10^{-23}	0.0	0.0
CSSDP	3.0×10^3	5.8×10^{-1}	7.3×10^{-2}	3.4×10^{-2}	0.0	0.0
ProxSDP	2.1×10^2	1.1×10^1	1.2×10^{-19}	3.2×10^{-19}	4.8×10^1	1.9×10^1
SCS	3.1×10^3	2.5×10^1	5.1×10^{-5}	9.5×10^{-5}	5.3×10^2	4.3×10^1

Table 1: Experimental results for $(n - k) = 10^3$ (10 instances) with time limit 3×10^3 seconds.

Algorithm	time (s)	std.	$\ X - X^*\ _F^2$	std.	memory (MB)	std.
CertSDP	4.5×10^3	7.0×10^2	1.9×10^{-22}	5.2×10^{-23}	8.5	1.2×10^1
CSSDP	1.0×10^4	6.6×10^{-1}	2.7	9.4×10^{-1}	6.2	1.5×10^1
ProxSDP	1.2×10^4	1.1×10^2	2.9	9.9×10^{-1}	1.9×10^4	1.2×10^2

Table 2: Experimental results for $(n - k) = 10^4$ (10 instances) with time limit 10^4 seconds. SCS was unable to complete a single iteration within the time limit and utilized over 70GB of memory.

We make a few observations:

- For $n - k = 10^3$ (see Table 1), both CertSDP and ProxSDP were able to achieve high accuracy within the time limit, while CSSDP and SCS could not. ProxSDP was faster than CertSDP while CertSDP used significantly less memory.

Algorithm	time (s)	std.	$\ X - X^*\ _F^2$	std.	memory (MB)	std.
CertSDP	5.0×10^4	6.2×10^2	2.5×10^{-2}	6.5×10^{-2}	2.3×10^2	2.0×10^2
CSSDP [†]	5.0×10^4	4.7	2.8	5.1×10^{-1}	2.0×10^2	2.5×10^2

Table 3: Experimental results for $(n - k) = 10^5$ (10 instances) with time limit 5×10^4 seconds. SCS and ProxSDP are not tested as they both come to complete failure due to memory allocation. [†]CSSDP failed due to numerical issues within the eigenvalue subroutine on three instances.

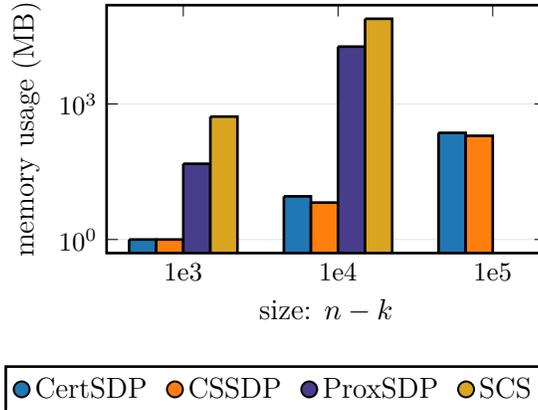


Figure 2: Memory usage of different algorithms as a function of the size $n - k$. In this chart, we plot 0.0 MB at 1.0 MB (see Remark 6 for a discussion on measuring memory usage).

- For $n - k = 10^4$ (see Table 2), CertSDP was the only algorithm that was able to achieve high accuracy within the time limit. The measured memory usage of CertSDP and CSSDP both had high variance, however it is clear that these algorithms use much less memory than ProxSDP and SCS. As previously mentioned, SCS used over 70GB of memory at this size.
- For $n - k = 10^5$ (see Table 3), CertSDP and CSSDP were the only algorithms that could be run without memory allocation errors. While neither algorithm was able to achieve the desired accuracy within the time limit, CertSDP (average primal squared distance of 2.5×10^{-2}) significantly outperformed CSSDP (average primal squared distance of 2.8).
- The dual suboptimality for CertSDP and CSSDP behave identically. This is expected as we employ Accelegrad to generate both sequences.
- The primal squared distance and the dual suboptimality for CSSDP track quite closely. This is expected from [23, Theorem 4.1, Table 3], which bounds the primal squared distance by a constant factor of the dual suboptimality for CSSDP.
- The convergence behavior of CautiousAGD depends on whether $\mathcal{U}^{(i)}$ in CertSDP is a certificate of strict complementarity.

When $\mathcal{U}^{(i)}$ is *not* a certificate of strict complementarity, CautiousAGD behaves as in the bottom-right plot of Figure 3: It briefly converges linearly before plateauing. This makes sense as the iterates in CautiousAGD should converge linearly to $\arg \min_X Q_{\mathcal{U}^{(i)}}(X) \neq X^*$.

When $\mathcal{U}^{(i)}$ is a certificate of strict complementarity, the iterates of CautiousAGD converge

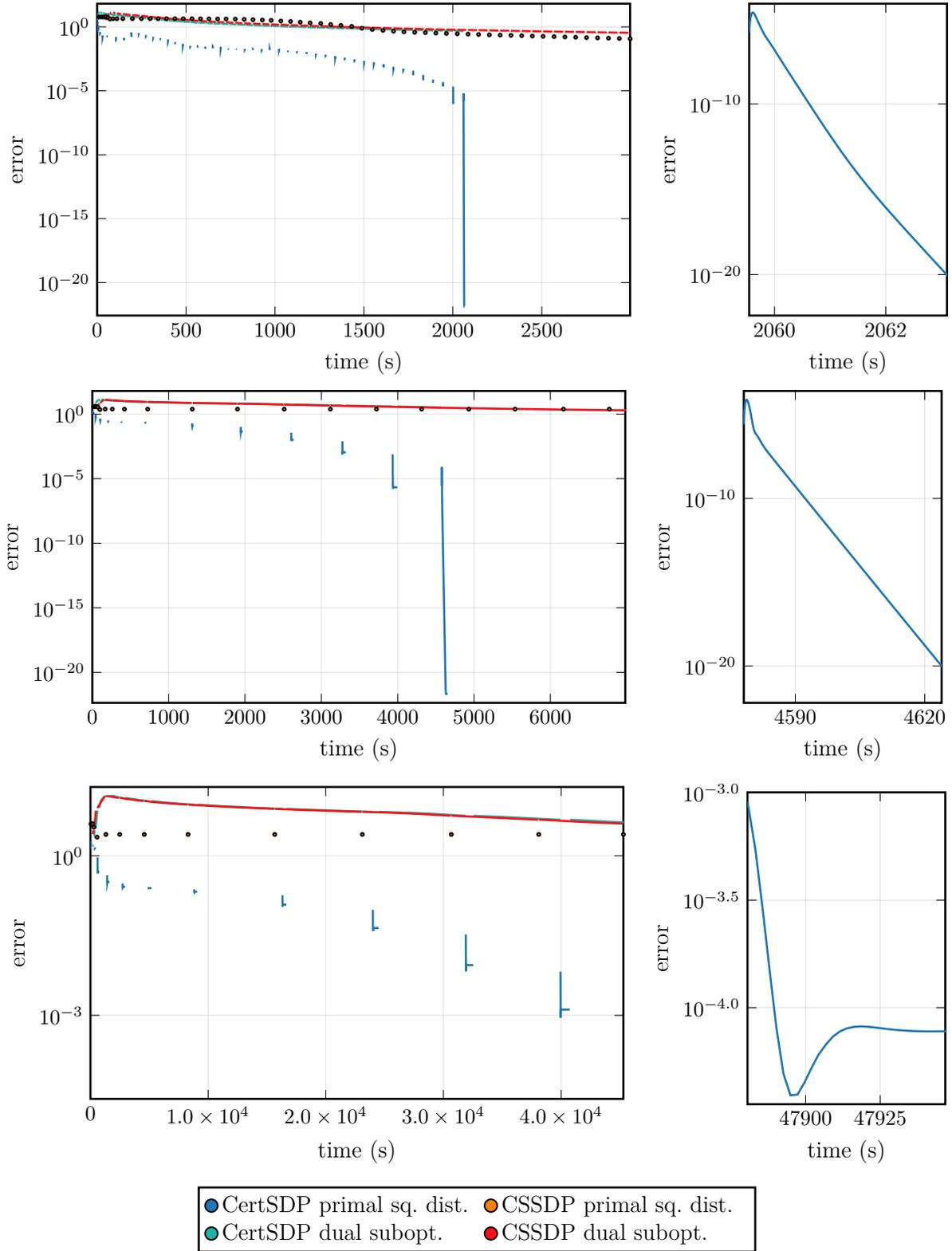


Figure 3: Comparison of convergence behavior between CertSDP (Algorithm 2) and CSSDP. The first, second, and third rows show experiments with $n - k = 10^3$, 10^4 , and 10^5 respectively. The right subplots give zoomed-in views of the primal squared distance in CertSDP on the final call to Algorithm 1.

linearly to X^* (see the top-right and middle-right plots of Figure 3).

Acknowledgments

This research is supported in part by ONR grant N00014-19-1-2321.

References

- [1] E. Abbe, A. S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *IEEE Trans. Inform. Theory*, 62(1):471–487, 2015.
- [2] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51, 1995.
- [3] F. Alizadeh, J. A. Haeberly, and M. L. Overton. Complementarity and nondegeneracy in semidefinite programming. *Math. Program.*, 77:111–128, 1997.
- [4] C.J. Argue, F. Kılınç-Karzan, and A. L. Wang. Necessary and sufficient conditions for rank-one generated cones. *arXiv preprint*, 2007.07433, 2020.
- [5] M. Baes, M. Burgisser, and A. Nemirovski. A randomized mirror-prox method for solving structured large-scale matrix saddle-point problems. *SIAM J. Optim.*, 23(2):934–962, 2013.
- [6] A. Beck. Quadratic matrix programming. *SIAM J. Optim.*, 17(4):1224–1238, 2007.
- [7] A. Beck, Y. Drori, and M. Teboulle. A new semidefinite programming relaxation scheme for a class of quadratic matrix problems. *Oper. Res. Lett.*, 40(4):298–302, 2012.
- [8] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*, volume 2 of *MPS-SIAM Ser. Optim.* SIAM, 2001.
- [9] A. Ben-Tal and A. Nemirovski. Solving large scale polynomial convex problems on ℓ_1 /nuclear norm balls by randomized first-order algorithms. *CoRR*, 2012.
- [10] N. Boumal, V. Voroninski, and A. Bandeira. The non-convex Burer–Monteiro approach works on smooth semidefinite programs. volume 29, 2016.
- [11] S. Burer and F. Kılınç-Karzan. How to convexify the intersection of a second order cone and a nonconvex quadratic. *Math. Program.*, 162:393–429, 2017.
- [12] S. Burer and R. D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95:329–357, 2003.
- [13] S. Burer and B. Yang. The trust region subproblem with non-intersecting linear constraints. *Math. Program.*, 149:253–264, 2014.
- [14] S. Burer and Y. Ye. Exact semidefinite formulations for a class of (random and non-random) nonconvex quadratic programs. *Math. Program.*, 181:1–17, 2019.
- [15] E. J. Candès, Y. C. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. *SIAM Rev.*, 57(2):225–251, 2015.
- [16] Y. Carmon and J. C. Duchi. Analysis of Krylov subspace solutions of regularized nonconvex quadratic problems. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10728–10738, 2018.
- [17] A. Chambolle and T. Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Math. Program.*, 159:253–287, 2016.
- [18] D. Cifuentes. On the Burer–Monteiro method for general semidefinite programs. *Opt. Lett.*, 15(6):2299–2309, 2021.

- [19] D. Cifuentes and A. Moitra. Polynomial time guarantees for the Burer-Monteiro method. *arXiv preprint*, 1912.01745, 2019.
- [20] A. d’Aspremont and N. El Karoui. A stochastic smoothing algorithm for semidefinite programming. *SIAM J. Optim.*, 24(3):1138–1177, 2014.
- [21] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods with inexact oracle: the strongly convex case. Technical Report 2013016, 2013.
- [22] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Math. Program.*, 146(1):37–75, 2014.
- [23] L. Ding, A. Yurtsever, V. Cevher, J. A. Tropp, and M. Udell. An optimal-storage approach to semidefinite programming using approximate complementarity. *SIAM J. Optim.*, 31(4):2695–2725, 2021.
- [24] A. L. Fradkov and V. A. Yakubovich. The S-procedure and duality relations in nonconvex problems of quadratic programming. *Vestnik Leningrad Univ. Math.*, 6:101–109, 1979.
- [25] M. P. Friedlander and I. Macêdo. Low-rank spectral optimization via gauge duality. *SIAM Journal on Scientific Computing*, 38(3):A1616–A1638, 2016.
- [26] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.
- [27] E. Y. Hamedani and N. C. Aybat. A primal-dual algorithm with line search for general convex-concave saddle point problems. *SIAM J. Optim.*, 31(2):1299–1329, 2021.
- [28] E. Hazan and T. Koren. A linear-time algorithm for trust region problems. *Math. Program.*, 158:363–381, 2016.
- [29] N. Ho-Nguyen and F. Kılınç-Karzan. A second-order cone based approach for solving the Trust Region Subproblem and its variants. *SIAM J. Optim.*, 27(3):1485–1512, 2017.
- [30] V. Jeyakumar and G. Y. Li. Trust-region problems with linear inequality constraints: Exact SDP relaxation, global optimality and robust optimization. *Math. Program.*, 147:171–206, 2014.
- [31] R. Jiang and D. Li. Novel reformulations and efficient algorithms for the Generalized Trust Region Subproblem. *SIAM J. Optim.*, 29(2):1603–1633, 2019.
- [32] A. Juditsky and A. Nemirovski. First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure. *Optimization for Machine Learning*, 30(9):149–183, 2011.
- [33] F. Kılınç-Karzan and A. L. Wang. Exactness in SDP relaxations of QCQPs: Theory and applications. Tut. in Oper. Res. INFORMS, 2021.
- [34] G. Lan, Z. Lu, and R. D.C. Monteiro. Primal-dual first-order methods with $o(1/\epsilon)$ iteration-complexity for cone programming. *Math. Program.*, 126:1–29, 2011.
- [35] K. Y. Levy, A. Yurtsever, and V. Cevher. Online adaptive methods, universality and acceleration. In *Advances in Neural Information Processing Systems*, 2018.

- [36] M. Locatelli. KKT-based primal-dual exactness conditions for the Shor relaxation. *arXiv preprint*, 2011.05033, 2020.
- [37] Z. Lu, A. Nemirovski, and R. D.C. Monteiro. Large-scale semidefinite programming via a saddle point mirror-prox algorithm. *Math. Program.*, 109:211–237, 2007.
- [38] A. Majumdar, G. Hall, and A. A. Ahmadi. Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:331–360, 2020.
- [39] D. G. Mixon, S. Villar, and R. Ward. Clustering subgaussian mixtures by semidefinite programming. *arXiv preprint*, 1602.06612, 2016.
- [40] J. J. Moré. Generalizations of the Trust Region Problem. *Optimization methods and Software*, 2(3-4):189–209, 1993.
- [41] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4:553–572, 1983.
- [42] A. Nemirovski. Prox-method with rate of convergence $o(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.*, 15(1):229–251, 2004.
- [43] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optim.*, 16(1):235–249, 2005.
- [44] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103:127–152, 2005.
- [45] Y. Nesterov. *Lectures on convex optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer, 2018.
- [46] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.
- [47] B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.
- [48] Y. Ouyang and Y. Xu. Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems. *Math. Program.*, 185:1–35, 2021.
- [49] B. Palaniappan and F. Bach. Stochastic variance reduction methods for saddle-point problems. volume 29, 2016.
- [50] T. K. Pong and H. Wolkowicz. The generalized trust region subproblem. *Computational Optimization and Applications*, 58(2):273–322, 2014.
- [51] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254, 2008.

- [52] N. Rujeerapaiboon, K. Schindler, D. Kuhn, and W. Wiesemann. Size matters: Cardinality-constrained clustering and outlier detection via conic optimization. *SIAM J. Optim.*, 29(2): 1211–1239, 2019.
- [53] N. Shinde, V. Narayanan, and J. Saunderson. Memory-efficient structured convex optimization via extreme point sampling. *SIAM Journal on Mathematics of Data Science*, 3(3):787–814, 2021.
- [54] N. Z. Shor. Dual quadratic estimates in polynomial and boolean programming. *Ann. Oper. Res.*, 25:163–168, 1990.
- [55] M. Sion. On general minimax theorems. *Pacific J. Math.*, 8(1):171–176, 1958.
- [56] M. Souto, J. D. Garcia, and Á. Veiga. Exploiting low-rank structure in semidefinite programming by approximate operator splitting. *Optimization*, pages 1–28, 2020.
- [57] R. J. Stern and H. Wolkowicz. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optim.*, 5(2):286–313, 1995.
- [58] J. F. Sturm and S. Zhang. On cones of nonnegative quadratic functions. *Math. Oper. Res.*, 28(2):246–267, 2003.
- [59] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization, 2008.
- [60] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.
- [61] I. Waldspurger and A. Waters. Rank optimality for the Burer–Monteiro factorization. *SIAM J. Optim.*, 30(3):2577–2602, 2020.
- [62] A. L. Wang and F. Kılınç-Karzan. The generalized trust region subproblem: solution complexity and convex hull results. *Math. Program.*, 2020. doi: 10.1007/s10107-020-01560-8. Forthcoming.
- [63] A. L. Wang and F. Kılınç-Karzan. A geometric view of SDP exactness in QCQPs and its applications. *arXiv preprint*, 2011.07155, 2020.
- [64] A. L. Wang and F. Kılınç-Karzan. On the tightness of SDP relaxations of QCQPs. *Math. Program.*, 2021. doi: 10.1007/s10107-020-01589-9. Forthcoming.
- [65] A. L. Wang, Y. Lu, and F. Kılınç-Karzan. Implicit regularity and linear convergence rates for the generalized trust-region subproblem. *arXiv preprint*, 2112.13821, 2021.
- [66] H. Yang, L. Liang, L. Carlone, and K. Toh. An inexact projected gradient method with rounding and lifting by nonlinear programming for solving rank-one semidefinite relaxation of polynomial optimization. *arXiv preprint*, 2105.14033, 2021.
- [67] A. Yurtsever, O. Fercoq, and V. Cevher. A conditional-gradient-based augmented Lagrangian framework. In *International Conference on Machine Learning*, pages 7272–7281, 2019.
- [68] A. Yurtsever, J. A. Tropp, O. Fercoq, M. Udell, and V. Cevher. Scalable semidefinite programming. *SIAM Journal on Mathematics of Data Science*, 3(1):171–200, 2021.

A Deferred proofs

The following proof is adapted from [45].

Proof of Lemma 3. It is evident that $\phi_t(X)$ are quadratic matrix functions of the form (6) with $V_0 = X_0$ and $\phi_0^* = Q(X_0)$. The remainder of the proof verifies the recurrences on V_{t+1} and ϕ_{t+1}^* . We suppose that the stated form holds for some t , and we will show that it will hold for $t + 1$ as well. We compute

$$\frac{1}{\tilde{\mu}} \nabla \phi_{t+1}(X) = (1 - \alpha)(X - V_t) + \alpha \left(X - \left(\Xi_t - \frac{1}{\tilde{\mu}} \tilde{g}_t \right) \right).$$

We deduce that $V_{t+1} = (1 - \alpha)V_t + \alpha \left(\Xi_t - \frac{1}{\tilde{\mu}} \tilde{g}_t \right)$. Noting that $\phi_{t+1}^* = \phi_{t+1}(V_{t+1})$, and applying the recursive definition of $\phi_{t+1}(X)$ gives us

$$\begin{aligned} \phi_{t+1}^* &= (1 - \alpha) \left(\phi_t^* + \frac{\tilde{\mu}}{2} \|V_{t+1} - V_t\|_F^2 \right) \\ &\quad + \alpha \left(Q(X_{t+1}) + \frac{1}{2\tilde{L}} \|\tilde{g}_t\|_F^2 + \langle \tilde{g}_t, V_{t+1} - \Xi_t \rangle + \frac{\tilde{\mu}}{2} \|V_{t+1} - \Xi_t\|_F^2 \right) \\ &= (1 - \alpha) \phi_t^* + \alpha \left(Q(X_{t+1}) + \frac{1}{2\tilde{L}} \|\tilde{g}_t\|_F^2 \right) \\ &\quad + (1 - \alpha) \frac{\tilde{\mu}}{2} \|V_{t+1} - V_t\|_F^2 + \frac{\alpha \tilde{\mu}}{2} \left\| V_{t+1} - \left(\Xi_t - \frac{1}{\tilde{\mu}} \tilde{g}_t \right) \right\|_F^2 - \frac{\alpha}{2\tilde{\mu}} \|\tilde{g}_t\|_F^2 \\ &= (1 - \alpha) \phi_t^* + \alpha \left(Q(X_{t+1}) + \frac{1}{2\tilde{L}} \|\tilde{g}_t\|_F^2 \right) \\ &\quad + \frac{\tilde{\mu}(1 - \alpha)\alpha^2}{2} \left\| V_t - \left(\Xi_t - \frac{1}{\tilde{\mu}} \tilde{g}_t \right) \right\|_F^2 + \frac{\tilde{\mu}\alpha(1 - \alpha)^2}{2} \left\| V_t - \left(\Xi_t - \frac{1}{\tilde{\mu}} \tilde{g}_t \right) \right\|_F^2 - \frac{\alpha}{2\tilde{\mu}} \|\tilde{g}_t\|_F^2 \\ &= (1 - \alpha) \phi_t^* + \alpha \left(Q(X_{t+1}) + \frac{1}{2\tilde{L}} \|\tilde{g}_t\|_F^2 \right) \\ &\quad + \alpha(1 - \alpha) \left(\frac{\tilde{\mu}}{2} \|\Xi_t - V_t\|_F^2 + \langle \tilde{g}_t, V_t - \Xi_t \rangle \right) - \frac{\alpha^2}{2\tilde{\mu}} \|\tilde{g}_t\|_F^2, \end{aligned}$$

where the third equation follows from substituting the expression for V_{t+1} , and the last one from regrouping the terms. \blacksquare

The following proof is adapted from [45, Page 92].

Proof of Lemma 4. Note that

$$\begin{aligned} \Xi_t &= \frac{X_t + \alpha V_t}{1 + \alpha} \\ X_{t+1} &= \Xi_t - \frac{\tilde{g}_t}{\tilde{L}} \\ V_{t+1} &= (1 - \alpha)V_t + \alpha \left(\Xi_t - \frac{1}{\tilde{\mu}} \tilde{g}_t \right). \end{aligned}$$

Therefore,

$$\begin{aligned}
V_{t+1} &= (1 - \alpha) \frac{(1 + \alpha)\Xi_t - X_t}{\alpha} + \alpha \left(\Xi_t - \frac{1}{\tilde{\mu}} \tilde{g}_t \right) \\
&= X_t + \frac{1}{\alpha} \left(\Xi_t - X_t - \frac{1}{\tilde{L}} \tilde{g}_t \right) \\
&= X_t + \frac{1}{\alpha} (X_{t+1} - X_t).
\end{aligned}$$

Then,

$$\begin{aligned}
\Xi_{t+1} &= X_{t+1} + \frac{\alpha}{1 + \alpha} (V_{t+1} - X_{t+1}) \\
&= X_{t+1} + \frac{1 - \alpha}{1 + \alpha} (X_{t+1} - X_t). \quad \blacksquare
\end{aligned}$$

Lemma 13. *Consider an instance of (12) generated by the random procedure in Section 5.2. Then equality holds throughout (12).*

Proof. It suffices to show that γ^* and T^* are feasible and achieve value $\|X^*\|_F^2$ in the dual SDP (i.e., the third line of (12)).

Note that by Schur Complement Theorem,

$$\begin{pmatrix} A(\gamma^*)/2 & B(\gamma^*)/2 \\ B(\gamma^*)^\top/2 & \frac{c(\gamma^*)}{k} I_k - T^* \end{pmatrix} \sim \begin{pmatrix} I_{n-k} & \\ & \frac{c(\gamma^*)}{k} I_k - T^* - \frac{B(\gamma^*)^\top A(\gamma^*)^{-1} B(\gamma^*)}{2} \end{pmatrix} = \begin{pmatrix} I_{n-k} & \\ & 0_k \end{pmatrix}.$$

Here, \sim indicates matrix similarity. Thus γ^* and T^* are feasible in the dual SDP.

Next,

$$\begin{aligned}
\text{tr}(T^*) &= \text{tr} \left(\frac{c(\gamma^*)}{k} I_k - \frac{B(\gamma^*)^\top A(\gamma^*)^{-1} B(\gamma^*)}{2} \right) \\
&= \frac{\text{tr}((X^*)^\top A(\gamma^*) X^*)}{2} + \langle B(\gamma^*), X^* \rangle + c(\gamma^*) \\
&= \frac{\|X^*\|_F^2}{2} + \sum_{i=1}^m \gamma_i^* \left(\text{tr} \left(\frac{(X^*)^\top A_i X^*}{2} \right) + \langle B_i, X^* \rangle + c_i \right) = \frac{\|X^*\|_F^2}{2}. \quad \blacksquare
\end{aligned}$$

B Additional numerical results

Table 4 displays numerical results for a variant of SketchyCGAL (see Section 5.1 for implementation details) on one random instance of (12) for each of $n - k = 10^3$, 10^4 , and 10^5 .

$n - k$	time (s)	$\ X - X^*\ _F^2$	memory (MB)
10^3	3.0×10^3	2.1	2.2×10^1
10^4	1.0×10^4	2.5	6.0×10^1
10^5	5.6×10^4	3.0	2.9×10^2

Table 4: Preliminary experimental results for $(n - k) = 10^3, 10^4, 10^5$ (1 instance) with time limit $3 \times 10^3, 10^4, 5 \times 10^4$ seconds for a variant of SketchyCGAL.