Mathematical Programming
https://doi.org/10.1007/s10107-021-01686-3

FULL LENGTH PAPER

Series A

Convergence rate of block-coordinate maximization Burer–Monteiro method for solving large SDPs

Murat A. Erdogdu[1,2] · Asuman Ozdaglar[3] · Pablo A. Parrilo[3] · Nuri Denizcan Vanli[3]

Received: 10 December 2019 / Accepted: 26 June 2021

Abstract

Semidefinite programming (SDP) with diagonal constraints arise in many optimization problems, such as Max-Cut, community detection and group synchronization. Although SDPs can be solved to arbitrary precision in polynomial time, generic convex solvers do not scale well with the dimension of the problem. In order to address this issue, Burer and Monteiro (Math Program 95(2):329–357, 2003) proposed to reduce the dimension of the problem by appealing to a low-rank factorization and solve the subsequent non-convex problem instead. In this paper, we present coordinate ascent based methods to solve this non-convex problem with provable convergence guarantees. More specifically, we prove that the block-coordinate maximization algorithm applied to the non-convex Burer–Monteiro method globally converges to a first-order stationary point with a sublinear rate without any assumptions on the problem. We further show that this algorithm converges linearly around a local maximum provided that the objective function exhibits quadratic decay. We establish that this condition generically holds when the rank of the factorization is sufficiently large. Furthermore,

Part of this work has previously appeared in ICML 2018 Workshop on Modern Trends in Nonconvex Optimization for Machine Learning.

✉ Nuri Denizcan Vanli
denizcan@mit.edu

Murat A. Erdogdu
erdogdu@cs.toronto.edu

Asuman Ozdaglar
asuman@mit.edu

Pablo A. Parrilo
parrilo@mit.edu

1 Department of Computer Science, University of Toronto, Toronto, Canada

2 Department of Statistical Sciences, University of Toronto, Toronto, Canada

3 Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, USA

Published online: 30 July 2021

incorporating Lanczos method to the block-coordinate maximization, we propose an algorithm that is guaranteed to return a solution that provides $1 - \mathcal{O}(1/r)$ approximation to the original SDP without any assumptions, where r is the rank of the factorization. This approximation ratio is known to be optimal (up to constants) under the unique games conjecture, and we can explicitly quantify the number of iterations to obtain such a solution.

Keywords Semidefinite programming · Burer–Monteiro method · Coordinate descent · Non-convex optimization · Large-scale optimization

Mathematics Subject Classification 65K05, 90C22, 90C25, 90C26, 90C27, 90C30, 58C05, 49M37

1 Introduction

A variety of problems in statistical estimation and machine learning require solving a combinatorial optimization problem, which are often intractable [42]. Semidefinite programs (SDP) are commonly used as convex relaxations for these problems, providing efficient algorithms with approximate optimality [34]. A generic SDP in this framework can be written as

$$\begin{aligned} &\text{maximize} && \langle \boldsymbol{A}, \boldsymbol{X} \rangle \\ &\text{subject to} && X_{ii} = 1, \ \text{for } i \in [n], \\ & && \boldsymbol{X} \succeq 0, \end{aligned} \tag{CVX}$$

where $\boldsymbol{A}, \boldsymbol{X} \in \mathrm{Sym}_n$ (real symmetric matrices of size $n \times n$) and $[n] = \{1, 2, ..., n\}$. This problem appears as a convex relaxation to the celebrated Max-Cut problem [22], graphical model inference [19], community detection problems [6], and group synchronization [32].

Although SDPs serve as reliable relaxations to many combinatorial problems, the resulting convex problem is still computationally challenging. Interior point methods can solve SDPs to arbitrary accuracy in polynomial-time, but they do not scale well with the problem dimension n. A popular approach to remedy these limitations is to introduce a low-rank factorization $\boldsymbol{X} = \boldsymbol{\sigma}\boldsymbol{\sigma}^\top$, where $\boldsymbol{\sigma} \in \mathbb{R}^{n \times r}$ with r denoting the rank. This reformulation removes the positive semidefinite cone constraint in (CVX) since $\boldsymbol{X} = \boldsymbol{\sigma}\boldsymbol{\sigma}^\top$ is guaranteed to be a positive semidefinite matrix, and choosing $r \ll n$ provides computational efficiency as well as storage benefits. This method is often referred to as Burer–Monteiro approach [15]. Denoting ith row of $\boldsymbol{\sigma}$ by σ_i, i.e., $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, ..., \sigma_n]^\top$, the resulting non-convex problem can be written as follows

$$\begin{aligned} &\text{maximize} && \langle \boldsymbol{A}, \boldsymbol{\sigma}\boldsymbol{\sigma}^\top \rangle \\ &\text{subject to} && \|\sigma_i\| = 1, \ \text{for } i \in [n]. \end{aligned} \tag{Non-CVX}$$

In the original Burer–Monteiro approach [15], the authors proposed to use an augmented Lagrangian method for a general form SDP. However, it has been recently

observed that feasible methods (such as block-coordinate maximization [25,43], Riemannian gradient [25,32] and Riemannian trust-region methods [1,11,26]) provide empirically faster rates since feasibility can be efficiently guaranteed via projection onto the Cartesian product of spheres. Despite overwhelming empirical evidence [25,32,43], convergence properties of these feasible methods are not well-understood (except the Riemannian trust-region method, for which a sublinear convergence rate is shown in [11] and a local superlinear convergence is shown in [1] with no rate estimate). Among these methods, block-coordinate maximization and Riemannian gradient ascent are simpler to implement and have computational complexity of $\mathcal{O}(nr)$ and $\mathcal{O}(n^2r)$, respectively, whereas Riemannian trust-region requires to solve the trust-region subproblem at each iteration, which is usually solved iteratively using the Lanczos method in a few iterations, whose per iteration requires $\mathcal{O}(n^2r)$ arithmetic operations. Furthermore, block-coordinate maximization does not have any step size or tuning parameters, unlike Riemannian gradient ascent and Riemannian trust-region methods. Empirical studies further motivate the use of block-coordinate maximization by presenting superior performance compared to existing methods on large-scale problems, often with linear convergence [25]. In this paper, we provide the first local and global convergence rate guarantees for the block-coordinate maximization method (applied to Burer–Monteiro approach) in the literature, which are consistent with the empirical performance of the algorithm. Our contributions can be summarized as follows:

- We establish the global sublinear convergence of the block-coordinate maximization algorithm applied to (Non-CVX) without any assumptions on the cost matrix $\boldsymbol{A}$.
- We show that this algorithm enjoys a linear rate around a neighborhood of any local maximum when the objective function satisfies the quadratic decay assumption.
- We establish that the quadratic decay condition that leads to local linear convergence generically holds when the rank of the factorization satisfies $r \geq \sqrt{2n}$.
- Incorporating Lanczos methods into the block-coordinate maximization procedure, we propose an algorithm that returns an approximate second-order stationary point of (Non-CVX). By choosing the rank of the factorization sufficiently large and selecting the parameters of the algorithm according to the cost matrix $\boldsymbol{A}$, we show that the solution returned by this algorithm is not only an approximate local maximum to (Non-CVX), but also provides $1 - \mathcal{O}(1/r)$ approximation to (CVX). We highlight that this approximation ratio is optimal under the unique games conjecture.
- We validate our theoretical results via numerical examples and compare the performance of the block-coordinate maximization algorithm with various manifold optimization methods to demonstrate its performance.

1.1 Related work

There are numerous papers that analyze the landscape of the solution space of (Non-CVX). In particular, it is known that (CVX) admits an optimal solution of rank r such that $r(r+1)/2 \leq n$ [7,35]. Using this observation, it has been shown in

[15,16,26] that when $r \geq \sqrt{2n}$, if $\boldsymbol{\sigma}$ is a rank deficient second-order stationary point of (Non-CVX), then $\boldsymbol{\sigma}$ is a global maximum for (Non-CVX) and $\boldsymbol{X} = \boldsymbol{\sigma}\boldsymbol{\sigma}^\top$ is a global maximum for (CVX). The recent paper [12] showed that when $r \geq \sqrt{2n}$, for almost all $\boldsymbol{A}$, every $\boldsymbol{\sigma}$ that is a first-order stationary point is rank deficient. For arbitrary rank r, it is shown that all local maxima are within a $n\|\boldsymbol{A}\|_2/\sqrt{r}$ gap from the optimum of (CVX) [33], and any ε-approximate concave point is within a $\mathrm{Rg}(\mathrm{Non}-\mathrm{CVX})/(r-1)+n\varepsilon/2$ gap from the optimum of (CVX) [32], where Rg(Non-CVX) is the range of the problem (Non-CVX), i.e., the difference between the maximum and the minimum values of the objective in (Non-CVX).

Javanmard *et al.* [25] showed that when applied to solve (Non-CVX), Riemannian gradient ascent and block-coordinate maximization methods provide excellent numerical results, yet no convergence guarantee is provided. Similar experimental results are also observed in [43] for the block-coordinate maximization algorithm and in [32] for the Riemannian gradient ascent algorithm. Concurrent to this work, in [43], the authors analyzed the convergence of the deterministic block-coordinate maximization algorithm. In particular, they showed that the deterministic block-coordinate maximization algorithm is asymptotically convergent (see [43, Theorem 3.2]) and enjoys a local liner convergence with no explicit rate estimates (see [43, Theorem 3.5]). They also proved that the deterministic block-coordinate maximization approach converges to a local maximum generically under random initialization using the center-stable manifold theorem similar to [30]. These results hold under the assumption that the iterates generated by the algorithm satisfy a certain condition that is seemingly impossible to verify without actually running the algorithm. To alleviate this issue, the authors suggested using a coordinate ascent method with a sufficiently small step size, for which the aforementioned convergence results hold without this precarious assumption. In [9], the authors provided a global sublinear convergence rate for the Riemannian trust-region method for general non-convex problems and these results have been used in [11,32] for the non-convex Burer–Monteiro approach. Augmented Lagrangian methods have been proposed to solve (Non-CVX) as well [15,16], however these methods do not benefit from separability of the manifold constraints, and hence are usually slower [12].

There also exist methods that solve (CVX) by exploiting its special structure [5,21,27,39]. In particular, [27] reduces (CVX) to a sequence of approximate eigenpair computations that is efficiently solved using the power method. In [5,39], matrix multiplicative weights algorithm is used to approximately solve (CVX), and these ideas are extended in [21] using sketching techniques [40]. However, these methods require constructing $\boldsymbol{X} \in \mathrm{Sym}_n$ explicitly, which is prohibitive when n goes beyond a few thousands, whereas the Burer–Monteiro approach we consider in this paper easily scales to very large instances as the low-rank factorization decreases the dimension of the problem from $\mathcal{O}\left(n^2\right)$ to $\mathcal{O}\left(nr\right)$ with $r \ll n$. For time complexity comparison between these methods that are based on Lagrangian relaxation and the Burer–Monteiro approach in this paper, we refer to Corollary 4.

Coordinate descent methods have been successfully applied to non-convex differentiable optimization problems in several papers [31,36,38,41]. In [41], the authors propose a coordinate gradient descent approach that may be viewed as a hybrid of gradient-projection and coordinate descent to minimize the sum of a smooth function

and a convex separable function. They analyze the greedy coordinate selection rule and present local linear convergence, although no rate estimates are provided. [38] considers a similar composite but convex optimization problem and provides explicit rate estimates. These results are then generelized to non-convex problems by [36] and [31]. However, these approaches heavily rely on the Euclidean geometry and cannot handle non-convex constraints, which is the main focus of our paper.

1.2 Notations and preliminaries

Throughout the paper, matrices are denoted with a boldface font, and all vectors are column vectors. The superscripts are used to denote iteration counters, i.e., $\boldsymbol{\sigma}^k$ denotes the value of $\boldsymbol{\sigma}$ at iteration k. For a vector g, $\|g\|$ represents its Euclidean norm. For matrices $\boldsymbol{A}, \boldsymbol{B}$, we write $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \text{trace}(\boldsymbol{A}\boldsymbol{B}^\top)$ for the inner product associated to the Frobenius norm $\|\boldsymbol{A}\|_{\text{F}} = \sqrt{\langle \boldsymbol{A}, \boldsymbol{A} \rangle}$. A_{ij} represents the entry at the ith row and jth column of $\boldsymbol{A}$, A_i represents its ith row as a column vector, and $\|\boldsymbol{A}\|_1 = \max_{1 \le j \le n} \sum_{i=1}^n |A_{ij}|$ represents its 1-norm, and $\|\boldsymbol{A}\|_{1,1} = \sum_{i,j=1}^n |A_{ij}|$ represents its $L_{1,1}$-norm. For a function h, ∇h and grad$h$ represent its Euclidean and Riemannian gradients, respectively. Similarly, $\nabla^2 h$ and Hess$h$ represent its Euclidean and Riemannian Hessians, respectively. We let $\mathbb{S}^{m-1}$ denote the unit sphere in $\mathbb{R}^m$. For a vector y, Diag(y) represents the diagonal matrix whose ith diagonal entry is y_i. Similarly for a matrix $\boldsymbol{A}$, diag$(\boldsymbol{A})$ represents the vector whose ith entry is A_{ii}.

For brevity, we assume without loss of generality that $\boldsymbol{A}$ is a symmetric matrix and $A_{ii} = 0$, for all $i \in [n]$ (the latter assumption is removed in Sect. 4 to keep our presentation consistent with the existing works in the literature). Indeed, if $\boldsymbol{A}$ is not symmetric, then we can replace $\boldsymbol{A}$ by $(\boldsymbol{A} + \boldsymbol{A}^\top)/2$, which is a symmetric matrix, and the objective value (Non-CVX) remains the same for all $\boldsymbol{\sigma} \in \mathbb{R}^{n \times r}$ since $\boldsymbol{\sigma}\boldsymbol{\sigma}^\top$ is symmetric. Similarly, replacing the diagonal entries of $\boldsymbol{A}$ by zeros decreases the objective value by the constant Tr $(\boldsymbol{A})$ for all feasible $\boldsymbol{\sigma}$, since the diagonal entries of $\boldsymbol{\sigma}\boldsymbol{\sigma}^\top$ are equal to 1.

The rest of the paper is organized as follows. In Sect. 2, we present the algorithm and discuss its per iteration cost. In Sect. 3, we prove the global sublinear convergence and local linear convergence of the algorithm with explicit rate estimates. In Sect. 4, we introduce a second-order method based on block-coordinate maximization and Lanczos method that is guaranteed to return solutions with global optimality guarantees. We also provide a global sublinear convergence rate estimate for this algorithm. We perform numerical experiments to validate our theoretical results in Sect. 5 and conclude the paper in Sect. 6.

2 Block-coordinate maximization algorithm

In this section, we discuss the block-coordinate maximization (BCM) algorithm, its update rule and per iteration computational cost. Throughout the paper, we let $f : \mathbb{R}^{n \times r} \to \mathbb{R}$ denote the objective function of (Non-CVX), i.e.,

Algorithm 1 Block-Coordinate Maximization (BCM)

Initialize $\boldsymbol{\sigma}^0 \in \mathbb{R}^{n\times r}$ and calculate $g_i^0 = \sum_{j\neq i} A_{ij}\sigma_j^0$, for all $i \in [n]$.
for $k = 0, 1, 2, \ldots$ **do**
  Choose block $i_k = i$ using one of the coordinate selection rules.
  $\sigma_{i_k}^{k+1} \leftarrow g_{i_k}^k / \|g_{i_k}^k\|$.
  $g_i^{k+1} \leftarrow g_i^k - A_{ii_k}\sigma_{i_k}^k + A_{ii_k}\sigma_{i_k}^{k+1}$, for all $i \neq i_k$.
end for

$$f(\boldsymbol{\sigma}) = \langle \boldsymbol{A}, \boldsymbol{\sigma}\boldsymbol{\sigma}^\top \rangle.$$

Given the current iterate $\boldsymbol{\sigma}^k$, the BCM algorithm chooses a row $i_k \in [n]$ of the matrix $\boldsymbol{\sigma}^k$ and maximizes the following objective

$$f(\boldsymbol{\sigma}^k) = \sum_{i=1}^n \langle \sigma_i^k, g_i^k \rangle, \quad \text{where} \quad g_i^k := \sum_{j\neq i} A_{ij}\,\sigma_j^k,$$

over the block $\sigma_{i_k}^k \in \mathbb{S}^{r-1}$. More formally, we can write the update rule of the algorithm as follows

$$\begin{aligned}
\sigma_{i_k}^{k+1} &= \arg\max_{\|\zeta\|=1} f(\sigma_1^k, \ldots, \sigma_{i_k-1}^k, \zeta, \sigma_{i_k+1}^k, \ldots, \sigma_n^k), \\
&= \arg\max_{\|\zeta\|=1} 2\langle \zeta, g_{i_k}^k \rangle + \sum_{i\neq i_k} \sum_{j\neq i, i_k} A_{ij}\langle \sigma_i^k, \sigma_j^k \rangle, \qquad (1) \\
&= \arg\max_{\|\zeta\|=1} \langle \zeta, g_{i_k}^k \rangle = \frac{g_{i_k}^k}{\|g_{i_k}^k\|}, \qquad (2)
\end{aligned}$$

with the convention that $\sigma_{i_k}^{k+1} = \sigma_{i_k}^k$ when $\|g_{i_k}^k\| = 0$. Blocks $\sigma_{i_k}^k$ that are updated at each iteration can be chosen through any deterministic or randomized rule, and in this paper we focus on three coordinate selection rules:

- Uniform sampling: $i_k = i$ with probability $p_i = 1/n$.
- Importance sampling: $i_k = i$ with probability $p_i = \|g_i^k\| / \sum_{j=1}^n \|g_j^k\|$.
- Greedy coordinate selection: $i_k = \arg\max_{i\in[n]}(\|g_i^k\| - \langle \sigma_i^k, g_i^k \rangle)$.

Per iteration computational cost of the BCM algorithm with uniform sampling is $\mathcal{O}(nr)$ as after i_k is chosen uniformly at random, $g_{i_k}^k$ can be computed in $2(n-1)r$ floating point operations. On the other hand, the BCM algorithm with importance sampling and greedy coordinate selection requires all $\{\|g_i^k\|\}_{i=1}^n$, which can be naively computed in $\mathcal{O}\left(n^2 r\right)$ floating point operations per iteration. Instead, a smarter implementation is to keep both $\{\sigma_i^k\}_{i=1}^n$ and $\{g_i^k\}_{i=1}^n$'s in the memory (only the current iterates, not all the past ones) and update them as presented in Algorithm 1, which can be done in $2(n-1)r$ floating point operations. Therefore, per iteration computational cost of the BCM method with all three coordinate selection rules is $\mathcal{O}(nr)$ for dense $\boldsymbol{A}$ (i.e., when no structure is available on $\boldsymbol{A}$). However, in many SDP applications (such as

Max-Cut and graphical model inference), $\boldsymbol{A}$ is induced by a graph and letting d denote the maximum degree of the graph that induces $\boldsymbol{A}$, the computational cost of the BCM algorithm becomes $\mathcal{O}(dr)$. In comparison, per iteration computational complexity of the Riemannian gradient ascent algorithm is $\mathcal{O}\left(n^2 r\right)$, whereas the Riemannian trust-region algorithm runs a few iterations of a subroutine (e.g., power method) to solve the trust-region subproblem, whose per iteration cost is typically $\mathcal{O}\left(n^2 r\right)$.

Using any of the coordinate selection rules, the function values of the iterates generated by the BCM algorithm is a non-decreasing sequence by the definition of the algorithm. The increase in the function value per iteration (before reaching to stationarity) can be explicitly computed as we present in the following lemma.

Lemma 1 *Suppose at the kth iteration of the BCM algorithm, i_kth block is chosen (with some coordinate selection rule). Then, the BCM algorithm yields the following ascent on the objective value:*

$$f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) = 2\left(\|g_{i_k}^k\| - \langle \sigma_{i_k}^k, g_{i_k}^k \rangle\right) \geq 0.$$

Proof According to the decomposition in (1), we can compute the objective function as follows:

$$\begin{aligned} f(\boldsymbol{\sigma}^{k+1}) &= 2\langle \sigma_{i_k}^{k+1}, g_{i_k}^{k+1} \rangle + \sum_{i \neq i_k} \sum_{j \neq i, i_k} A_{ij} \langle \sigma_i^{k+1}, \sigma_j^{k+1} \rangle, \\ &= 2\langle \sigma_{i_k}^{k+1}, g_{i_k}^{k} \rangle + \sum_{i \neq i_k} \sum_{j \neq i, i_k} A_{ij} \langle \sigma_i^{k}, \sigma_j^{k} \rangle, \end{aligned} \tag{3}$$

where the latter equality follows since $g_{i_k}^{k+1} = g_{i_k}^k$ and all the terms in the sum are independent of $\sigma_{i_k}^{k+1}$. After adding and subtracting $2\langle \sigma_{i_k}^k, g_{i_k}^k \rangle$ to the right-hand side of (3), we obtain

$$f(\boldsymbol{\sigma}^{k+1}) = f(\boldsymbol{\sigma}^k) + 2\left(\langle \sigma_{i_k}^{k+1}, g_{i_k}^k \rangle - \langle \sigma_{i_k}^k, g_{i_k}^k \rangle\right).$$

By the update rule of the algorithm, we have $\sigma_{i_k}^{k+1} = g_{i_k}^k / \|g_{i_k}^k\|$, and plugging this value in the above equation concludes the proof. □

3 Convergence rate of BCM

In this section, we analyze the convergence rate of the BCM algorithm. As the feasible set of the problem in (Non-CVX) defines a smooth manifold, we will use certain tools from manifold optimization throughout the paper, which are highlighted in the following subsection. We refer to [2, Section 5.4] for a more detailed treatment of this topic.

3.1 Riemannian geometry of the problem

We define the following submanifold of matrices $\mathbb{R}^{n\times r}$ that corresponds to the Riemannian geometry induced by the constraints of the problem (Non-CVX) in the Euclidean space:

$$\mathcal{M}_r := \left\{ \boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)^\top \in \mathbb{R}^{n\times r} : \|\sigma_i\| = 1,\ \forall i \in [n] \right\}.$$

This manifold represents the Cartesian product of n unit spheres in $\mathbb{R}^r$. For any given point $\boldsymbol{\sigma} \in \mathcal{M}_r$, its tangent space can be found by taking the differential of the equality constraints as follows

$$T_{\boldsymbol{\sigma}}\mathcal{M}_r := \left\{ \boldsymbol{u} = (u_1, \ldots, u_n)^\top \in \mathbb{R}^{n\times r} : \langle u_i, \sigma_i \rangle = 0,\ \forall i \in [n] \right\}.$$

The Riemannian gradient of f on this manifold can be computed by the projection of its Euclidean gradient onto the tangent bundle. In particular, let $\mathrm{P}_{\boldsymbol{\sigma}}^{\perp} : \mathbb{R}^{n\times r} \to T_{\boldsymbol{\sigma}}\mathcal{M}_r$ denote the projection operator from the Euclidean space to the tangent space of $\boldsymbol{\sigma}$. When applied to a given matrix $\boldsymbol{w} = (w_1, \ldots, w_n)^\top \in \mathbb{R}^{n\times r}$, this projection operator yields

$$\begin{aligned} \mathrm{P}_{\boldsymbol{\sigma}}^{\perp}(\boldsymbol{w}) &= (w_1 - \langle \sigma_1, w_1 \rangle \sigma_1, \ldots, w_n - \langle \sigma_n, w_n \rangle \sigma_n)^\top, \\ &= \boldsymbol{w} - \mathrm{Diag}(\mathrm{diag}(\boldsymbol{w}\boldsymbol{\sigma}^\top))\, \boldsymbol{\sigma}. \end{aligned}$$

Therefore, the Riemannian gradient of f at $\boldsymbol{\sigma}$ can be computed as follows

$$\mathrm{grad}\, f(\boldsymbol{\sigma}) = \mathrm{P}_{\boldsymbol{\sigma}}^{\perp}(\nabla f(\boldsymbol{\sigma})) = 2\,(\boldsymbol{A} - \boldsymbol{\Lambda})\,\boldsymbol{\sigma},$$

where $\boldsymbol{\Lambda} = \mathrm{Diag}(\mathrm{diag}(\boldsymbol{A}\boldsymbol{\sigma}\boldsymbol{\sigma}^\top))$. Or equivalently, the Riemannian gradient of f at $\boldsymbol{\sigma}$ can be explicitly expressed as follows

$$\mathrm{grad}\, f(\boldsymbol{\sigma}) = 2\,(g_1 - \langle \sigma_1, g_1 \rangle \sigma_1, \ldots, g_n - \langle \sigma_n, g_n \rangle \sigma_n)^\top,$$

and its magnitude is given by

$$\|\mathrm{grad}\, f(\boldsymbol{\sigma})\|_{\mathrm{F}}^2 = 2 \sum_{i=1}^{n} \|g_i - \langle \sigma_i, g_i \rangle \sigma_i\|^2 = 2 \sum_{i=1}^{n} \left(\|g_i\|^2 - \langle \sigma_i, g_i \rangle^2 \right). \tag{4}$$

Using the same approach, we can calculate the Riemannian Hessian of f at $\boldsymbol{\sigma}$ along the direction of a vector $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$ by projecting the directional derivative of the gradient vector field onto the tangent space of $\boldsymbol{\sigma}$ as follows

$$\mathrm{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}] = \mathrm{P}^{\perp}\left(\mathrm{D}\,\mathrm{grad}\, f(\boldsymbol{\sigma})[\boldsymbol{u}]\right),$$

where D grad $f(\boldsymbol{\sigma})[\boldsymbol{u}]$ denotes the directional gradient of grad $f(\boldsymbol{\sigma})$ along the direction $\boldsymbol{u}$. This yields

$$\begin{aligned}\mathrm{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}] &= \mathrm{P}^{\perp}\left(2(\boldsymbol{A}-\boldsymbol{\Lambda})\boldsymbol{u} - 2\mathrm{Diag}(\mathrm{diag}(\boldsymbol{A}\boldsymbol{\sigma}\boldsymbol{u}^{\top}+\boldsymbol{A}\boldsymbol{u}\boldsymbol{\sigma}^{\top}))\boldsymbol{\sigma}\right)\\ &= \mathrm{P}^{\perp}\left(2(\boldsymbol{A}-\boldsymbol{\Lambda})\boldsymbol{u}\right), \end{aligned} \tag{5}$$

and in particular, for any $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$, we have

$$\langle \boldsymbol{u}, \mathrm{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}]\rangle = 2\langle \boldsymbol{u}, (\boldsymbol{A}-\boldsymbol{\Lambda})\boldsymbol{u}\rangle. \tag{6}$$

The geodesics $t \mapsto \boldsymbol{\sigma}(t)$ (i.e., curves of shortest path with zero acceleration) can be expressed as a function of $\boldsymbol{\sigma} = \boldsymbol{\sigma}(0) \in \mathcal{M}_r$ and $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$ as follows

$$\sigma_i(t) = \sigma_i \cos(\|u_i\|t) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|t). \tag{7}$$

This geodesic can be thought as the curve on the manifold that are obtained by moving from $\boldsymbol{\sigma} \in \mathcal{M}_r$ towards the direction pointed by $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$ (Table 1). According to this definition, the exponential map $\mathrm{Exp}_{\boldsymbol{\sigma}} : T_{\boldsymbol{\sigma}}\mathcal{M}_r \to \mathcal{M}_r$ corresponds to evaluating the point at $t = 1$ on the geodesic function, i.e., letting $\boldsymbol{\sigma}' = \mathrm{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})$, where $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$, we have

$$\sigma_i' = \sigma_i \cos(\|u_i\|) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|).$$

According to this geodesic map, we can also define the following geodesic distance between two points $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$ on the manifold:

$$\mathrm{dist}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') = \left(\sum_{i=1}^{n}(\arccos\langle \sigma_i, \sigma_i'\rangle)^2\right)^{1/2}. \tag{8}$$

More specifically, letting $\boldsymbol{\sigma}' = \mathrm{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})$, we obtain

$$\mathrm{dist}(\boldsymbol{\sigma}, \boldsymbol{\sigma}') = \left(\sum_{i=1}^{n}(\arccos\langle \sigma_i, \sigma_i\cos\|u_i\|\rangle)^2\right)^{1/2} = \|\boldsymbol{u}\|_{\mathrm{F}}.$$

Similarly, the distance between a point $\boldsymbol{\sigma}$ and a non-empty, closed and (geodesically) convex set Ω can be found as

$$\mathrm{dist}(\boldsymbol{\sigma}, \Omega) = \min_{\boldsymbol{\sigma}' \in \Omega} \mathrm{dist}(\boldsymbol{\sigma}, \boldsymbol{\sigma}').$$

Table 1 Summary of certain definitions stated in Sect. 3.1

Projection to the tangent space $T_{\boldsymbol{\sigma}}\mathcal{M}_r$ at $\boldsymbol{\sigma}$	$\mathrm{P}_{\boldsymbol{\sigma}}^{\perp}(\boldsymbol{w}) = \boldsymbol{w} - \mathrm{Diag}(\mathrm{diag}(\boldsymbol{w}\boldsymbol{\sigma}^\top))\,\boldsymbol{\sigma}$
Riemannian gradient at $\boldsymbol{\sigma}$	$\mathrm{grad}\, f(\boldsymbol{\sigma}) = 2\,(\boldsymbol{A} - \boldsymbol{\Lambda})\,\boldsymbol{\sigma}$ where $\boldsymbol{\Lambda} = \mathrm{Diag}(\mathrm{diag}(\boldsymbol{A}\boldsymbol{\sigma}\boldsymbol{\sigma}^\top))$
Riemannian Hessian at $\boldsymbol{\sigma}$ along $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$	$\mathrm{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}] = \mathrm{P}^{\perp}\,(2(\boldsymbol{A} - \boldsymbol{\Lambda})\boldsymbol{u})$
Geodesic $t \to \boldsymbol{\sigma}(t)$	$\sigma_i(t) = \sigma_i \cos(\|u_i\|t) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|t)$
Exponential map $\boldsymbol{\sigma}' = \mathrm{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})$	$\sigma_i' = \sigma_i \cos(\|u_i\|) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|)$

3.2 Global rate of convergence

In this section, we show that the BCM algorithm is globally convergent to a first-order stationary point of the problem with a sublinear rate. In particular, in the following theorem, we consider the BCM algorithm with greedy coordinate selection and show that its functional ascent (see Lemma 1) can be related to the norm of the Riemannian gradient of the function evaluated at the current iterate. By doing so, we prove that the BCM algorithm returns a solution with arbitrarily small Riemannian gradient.

Theorem 1 *Let $f^* = \max_{\|\sigma_i\|=1, \forall i\in[n]} f(\boldsymbol{\sigma})$. Then, for any $K \geq 1$, BCM with greedy coordinate selection yields the following guarantee*

$$\min_{k\in[K-1]} \|\mathrm{grad}\, f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 \leq \frac{2n\|\boldsymbol{A}\|_1(f^* - f(\boldsymbol{\sigma}^0))}{K}. \tag{9}$$

Proof From Lemma 1, we have

$$\begin{aligned} f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) &= 2\left(\|g_{i_k}^k\| - \langle \sigma_{i_k}^k, g_{i_k}^k\rangle\right) \\ &= 2\max_{i\in[n]}\left(\|g_i^k\| - \langle \sigma_i^k, g_i^k\rangle\right), \end{aligned}$$

where the latter equality follows by the greedy coordinate selection rule. We can rewrite this equation as follows:

$$\begin{aligned} f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) &= \max_{i\in[n]} \frac{2\|g_i^k\|\left(\|g_i^k\| - \langle \sigma_i^k, g_i^k\rangle\right)}{\|g_i^k\|}, \\ &\geq \max_{i\in[n]} \frac{\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k\rangle^2}{\|g_i^k\|}, \end{aligned}$$

where the inequality follows since $\|g_i^k\| \geq \langle \sigma_i^k, g_i^k\rangle$ for all $\sigma_i^k \in \mathbb{R}^r$. Lower bounding the maximum with the mean of its arguments, we get

$$f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \frac{1}{n}\sum_{i=1}^{n} \frac{\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k\rangle^2}{\|g_i^k\|}. \tag{10}$$

The $\|g_i^k\|$ term in the denominator in (10) can be upper bounded as follows

$$\|g_{i_k}^k\| \leq \sum_{j \neq i_k} |A_{i_k j}| \|\sigma_j^k\| \leq \|\boldsymbol{A}\|_1. \tag{11}$$

Using this bound in (10), we get

$$\begin{aligned} f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) &\geq \frac{1}{n\|A\|_1} \sum_{i=1}^{n} \left(\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k \rangle^2 \right) \\ &= \frac{\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2}{2n\|\boldsymbol{A}\|_1}. \end{aligned} \tag{12}$$

In order to conclude (9), we assume the contrary that $\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 > \epsilon$ for all $k \in [K-1]$. Then, using the boundedness of f, we observe that

$$f^* - f(\boldsymbol{\sigma}^0) \geq f(\boldsymbol{\sigma}^K) - f(\boldsymbol{\sigma}^0) = \sum_{k=0}^{K-1} \left[f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \right].$$

Using the functional ascent bound of BCM in (12), we get

$$f^* - f(\boldsymbol{\sigma}^0) \geq \sum_{k=0}^{K-1} \frac{\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2}{2n\|\boldsymbol{A}\|_1} > \frac{K\epsilon}{2n\|\boldsymbol{A}\|_1},$$

where the latter inequality follows by the assumption. Then, by contradiction, the algorithm returns a solution with $\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 \leq \epsilon$, for some $k \in [K-1]$, provided that

$$K \geq \frac{2n\|\boldsymbol{A}\|_1 (f^* - f(\boldsymbol{\sigma}^0))}{\epsilon}.$$

□

Using a similar approach to Theorem 1, we show in the following corollary that the BCM algorithm with uniform and importance sampling attains a similar sublinear convergence rate in expectation. The proof of this corollary follows similar lines to the proof of Theorem 1, hence is deferred to "Appendix A".

Corollary 1 *Let* $f^* = \max_{\|\sigma_i\|=1, \forall i \in [n]} f(\boldsymbol{\sigma})$. *Then, for any* $K \geq 1$, *randomized BCM yields the following guarantee*

$$\min_{k \in [K-1]} \mathbb{E} \|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 \leq \frac{2L(f^* - f(\boldsymbol{\sigma}^0))}{K}, \tag{13}$$

where

$$L = \begin{cases} n\|\boldsymbol{A}\|_1, & \textit{for uniform sampling,} \\ \|\boldsymbol{A}\|_{1,1}, & \textit{for importance sampling.} \end{cases} \tag{14}$$

We can observe from (9), (13) and (14) that BCM with uniform sampling attains the same sublinear rate as BCM with greedy coordinate selection in expectation as they both require at most $\lceil \big(2n\|\boldsymbol{A}\|_1(f^* - f(\boldsymbol{\sigma}^0))\big)/\epsilon \rceil$ iterations to return a solution $\boldsymbol{\sigma}$ satisfying $\|\text{grad}\, f(\boldsymbol{\sigma})\|_{\text{F}}^2 \leq \epsilon$. On the other hand, we see that BCM with importance sampling enjoys a tighter convergence rate compared to BCM with uniform sampling, as $\|\boldsymbol{A}\|_{1,1} \leq n\|\boldsymbol{A}\|_1$ for all $\boldsymbol{A} \in \mathbb{R}^{n\times n}$.

3.3 Local rate of convergence

Although the BCM algorithm enjoys the sublinear convergence rates presented in Sect. 3.2, it is numerically observed that the rate of convergence is linear when $\boldsymbol{\sigma}^k$ is close to a local maximum [25,43]. In this section, we investigate this behavior and prove that indeed BCM attains a linear convergence rate around a local maximum under the quadratic decay condition on the objective function, which is classically defined as follows [4,8]: Consider the unconstrained maximization problem: $\max_x \varphi(x)$, and let $\Omega_{\bar{x}}$ denote the set of local maximizers with objective value $\varphi(\bar{x})$. Then, the quadratic decay condition is said to be satisfied at $\bar{x}$ for φ, if there exists constants $\mu, \delta > 0$ such that $\varphi(x) \leq \varphi(\bar{x}) - \mu\, \text{dist}^2(x, \Omega_{\bar{x}})$, for all x such that $\|x - \bar{x}\| \leq \delta$, where dist measures the distance between point x and set $\Omega_{\bar{x}}$.

For the constrained optimization problem that we are considering in (Non-CVX), this definition needs to be slightly reworked. In particular, let $\boldsymbol{\sigma}$ be a local maximum of (Non-CVX) and consider the Taylor expansion of $\text{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})$ around $\boldsymbol{\sigma}$:

$$f(\text{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})) = f(\boldsymbol{\sigma}) + \frac{1}{2}\langle \boldsymbol{u}, \text{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}]\rangle + \mathcal{O}\left(\|\boldsymbol{u}\|_{\text{F}}^3\right),$$

where the first-order term is zero as $\boldsymbol{\sigma}$ is a local maximum. Then, for a sufficiently small neighborhood of $\boldsymbol{\sigma}$, the quadratic decay condition is satisfied if and only if there exists a constant $\mu > 0$ such that $\langle \boldsymbol{u}, \text{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}]\rangle \leq -\mu\, \text{dist}^2(\text{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u}), \Omega_{\boldsymbol{\sigma}})$, for all $\text{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})$ sufficiently close to $\boldsymbol{\sigma}$, where $\Omega_{\boldsymbol{\sigma}}$ is the set on which f has constant value $f(\boldsymbol{\sigma})$. Assume for the sake of simplicity that $\boldsymbol{\sigma}$ is a strict local maximum, i.e., $\Omega_{\boldsymbol{\sigma}} = \{\boldsymbol{\sigma}\}$. Then, the distance between $\text{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})$ and $\boldsymbol{\sigma}$ can be found as the norm of the tangent vector that connects these two points via the geodesic curve, i.e., $\text{dist}(\text{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u}), \boldsymbol{\sigma}) = \|\boldsymbol{u}\|_{\text{F}}$. Therefore, the quadratic decay condition is satisfied if and only if there exists a constant $\mu > 0$ such that $\langle \boldsymbol{u}, \text{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}]\rangle \leq -\mu\|\boldsymbol{u}\|_{\text{F}}^2$ for all $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$, where we note that the condition that $\text{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})$ is sufficiently close to $\boldsymbol{\sigma}$ is dropped considering the limit as $\boldsymbol{u} \to \boldsymbol{0}$.

Unfortunately, no local maximum is a strict local maximum for the problem (Non-CVX). To observe this, let $\text{O}(r) = \{\boldsymbol{Q} \in \mathbb{R}^{r\times r} : \boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}\}$ denote the orthogonal group in dimension r. Then, it can be observed that $f(\boldsymbol{\sigma}\,\boldsymbol{Q}) =$

$\langle \boldsymbol{A}, \boldsymbol{\sigma}\, \boldsymbol{Q}\boldsymbol{Q}^\top \boldsymbol{\sigma}^\top \rangle = \langle \boldsymbol{A}, \boldsymbol{\sigma}\boldsymbol{\sigma}^\top \rangle = f(\boldsymbol{\sigma})$, for any $\boldsymbol{Q} \in \mathrm{O}(r)$. Therefore, in order to measure the distance between $\mathrm{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})$ and $\Omega_{\boldsymbol{\sigma}}$, we define the following equivalence relation $\sim$:

$$\boldsymbol{\sigma} \sim \boldsymbol{\sigma}' \iff \exists \boldsymbol{Q} \in O(r) : \boldsymbol{\sigma} = \boldsymbol{\sigma}' \boldsymbol{Q}. \tag{15}$$

This equivalence relation induces a quotient space denoted by $\mathcal{M}_r / \sim$ and we let $[\boldsymbol{\sigma}]$ denote the equivalence class of a given matrix $\boldsymbol{\sigma} \in \mathcal{M}_r$. According to this definition, f has constant value of $f(\boldsymbol{\sigma})$ on the set $[\boldsymbol{\sigma}]$, i.e., $\Omega_{\boldsymbol{\sigma}} = [\boldsymbol{\sigma}]$. We let $\mathcal{V}_{\boldsymbol{\sigma}} \subset T_{\boldsymbol{\sigma}}\mathcal{M}_r$ denote the tangent space to the equivalence class $[\boldsymbol{\sigma}]$, which can be found as $\mathcal{V}_{\boldsymbol{\sigma}} = \{\boldsymbol{\sigma}\boldsymbol{B} : \boldsymbol{B} \in \mathbb{R}^{r\times r} \text{ and } \boldsymbol{B}^\top = -\boldsymbol{B}\}$.[1] Therefore, $\mathrm{dist}(\mathrm{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u}), [\boldsymbol{\sigma}]) = \|\boldsymbol{u}\|_{\mathrm{F}}$ if the closest point to $\mathrm{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u})$ in $[\boldsymbol{\sigma}]$ is $\boldsymbol{\sigma}$, or equivalently $\mathrm{dist}(\mathrm{Exp}_{\boldsymbol{\sigma}}(\boldsymbol{u}), [\boldsymbol{\sigma}]) = \|\boldsymbol{u}\|_{\mathrm{F}}$ if $u \in T_{\boldsymbol{\sigma}}\mathcal{M}_r \setminus \mathcal{V}_{\boldsymbol{\sigma}}$. Consequently, we say that quadratic decay is satisfied at $\boldsymbol{\sigma}$ for f if $\mathrm{Hess}\, f(\boldsymbol{\sigma})$ is negative definite on the orthogonal complement of $\mathcal{V}_{\boldsymbol{\sigma}}$ in $T_{\boldsymbol{\sigma}}\mathcal{M}_r$. The formal statement of this definition is as follows.

Definition 1 (*Quadratic decay*) Let $\boldsymbol{\sigma}$ be a local maximum of (Non-CVX). Quadratic decay condition is said to be satisfied at $\boldsymbol{\sigma}$ for f if there exists a constant $\mu > 0$ such that

$$\langle \boldsymbol{u}, \mathrm{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}] \rangle \leq -\mu \|\boldsymbol{u}\|_{\mathrm{F}}^2, \quad \text{for all } \boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r \setminus \mathcal{V}_{\boldsymbol{\sigma}}, \tag{16}$$

where $\mathcal{V}_{\boldsymbol{\sigma}}$ is the tangent space to the equivalence class $[\boldsymbol{\sigma}]$.

In the following theorem, we present the linear convergence rate of the BCM algorithm under the quadratic decay condition. We defer the validity of this condition to Sect. 3.4 where we show that quadratic decay generically (over the set of matrices A) holds for f when r is sufficiently large.

Theorem 2 *Let $\bar{\boldsymbol{\sigma}}$ be a limit point of the BCM algorithm and assume that $\bar{\boldsymbol{\sigma}}$ is a local maximum that satisfies the quadratic decay condition. If $\boldsymbol{\sigma}^0$ is sufficiently close to the equivalent class $[\bar{\boldsymbol{\sigma}}]$, then the iterates generated by the BCM algorithm with greedy coordinate selection enjoy the following linear convergence rate*

$$f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^{k+1}) \leq \left(1 - \frac{\mu}{4n^2\|\boldsymbol{A}\|_1}\right)\left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k)\right). \tag{17}$$

Proof We first discuss the outline of the proof for clarity. By (12), we have the following functional ascent bound on the iterates of the algorithm

$$f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \frac{\|\mathrm{grad}\, f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2}{2n\|\boldsymbol{A}\|_1}. \tag{18}$$

In order to prove linear convergence, our aim is to show that $\|\mathrm{grad}\, f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 \geq c(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k))$ for some positive constant c such that $c < 2n\|\boldsymbol{A}\|_1$, in a neighborhood around the limit points of the iterates generated by the algorithm. To prove this,

[1] Note that the dimension of $\mathcal{V}_{\boldsymbol{\sigma}}$ depends on the rank of $\boldsymbol{\sigma}$, and hence the quotient space is not a manifold.

we consider the Taylor approximation of $\|\mathrm{grad}\, f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2$ and $f(\boldsymbol{\sigma}^k)$ around $\boldsymbol{\sigma} \in [\bar{\boldsymbol{\sigma}}]$, where $\boldsymbol{\sigma}$ is the closest point to $\boldsymbol{\sigma}^k$ in the set $\bar{\boldsymbol{\sigma}}$. In the remainder of this proof, we show that the desired inequality holds by relating the most significant terms in these Taylor expansions. We defer bounding the higher-order terms to "Appendix B" in order not to distract the reader from the content of the paper.

Let $\bar{\boldsymbol{\sigma}}$ be the limit point of a subsequence $\{\boldsymbol{\sigma}^{k_\ell}\}_{k_\ell \geq 0}$ that contains $\boldsymbol{\sigma}^k$. Then, we consider the solution $\boldsymbol{\sigma} \in [\bar{\boldsymbol{\sigma}}]$ such that $\boldsymbol{\sigma}$ is the projection of $\boldsymbol{\sigma}^k$ onto $[\bar{\boldsymbol{\sigma}}]$, i.e., $\mathrm{dist}(\boldsymbol{\sigma}, \boldsymbol{\sigma}^k) \leq \mathrm{dist}(\boldsymbol{\sigma}', \boldsymbol{\sigma}^k)$ for all $\boldsymbol{\sigma}' \in [\bar{\boldsymbol{\sigma}}]$. Then, by construction there exists $\bar{\boldsymbol{u}} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r \setminus \mathcal{V}_{\boldsymbol{\sigma}}$ such that $\mathrm{Exp}_{\boldsymbol{\sigma}}(\bar{\boldsymbol{u}}) = \boldsymbol{\sigma}^k$. For ease of presentation, we let $\boldsymbol{u} = \bar{\boldsymbol{u}}/\|\bar{\boldsymbol{u}}\|_{\mathrm{F}}$ denote the normalized tangent vector and consider the following geodesic to describe $\boldsymbol{\sigma}^k$:

$$\sigma_i^k = \sigma_i \cos(\|u_i\| t) + \frac{u_i}{\|u_i\|} \sin(\|u_i\| t), \tag{19}$$

where it can be observed that $t = \|\bar{\boldsymbol{u}}\|_{\mathrm{F}}$ recovers the original exponential map $\boldsymbol{\sigma}^k = \mathrm{Exp}_{\boldsymbol{\sigma}}(\bar{\boldsymbol{u}})$. The second order Taylor approximation to (19) yields (note that $t = \|\bar{\boldsymbol{u}}\|_{\mathrm{F}} < 1$, when $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}^k$ are sufficiently close):

$$\sigma_i^k = \sigma_i + t u_i - \frac{t^2}{2}\|u_i\|^2 \sigma_i + \mathcal{O}\left(t^3\right),$$

and using this approximation, we obtain

$$g_i^k = g_i + t v_i - \frac{t^2}{2}\tilde{g}_i + \mathcal{O}\left(t^3\right),$$

where

$$v_i^k = \sum_{j \neq i} A_{ij} u_j \quad \text{and} \quad \tilde{g}_i = \sum_{j \neq i} A_{ij} \|u_j\|^2 \sigma_j.$$

This yields the following Taylor approximation to $\|\mathrm{grad}\, f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2$:

$$\begin{aligned}
\|\mathrm{grad}\, f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 &= 2\sum_{i=1}^{n}\left(\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k\rangle^2\right) \\
&= 2\sum_{i=1}^{n}\Bigg(\|g_i + t v_i - \frac{t^2}{2}\tilde{g}_i\|^2 \\
&\qquad -\langle \sigma_i + t u_i - \frac{t^2}{2}\|u_i\|^2\sigma_i,\ g_i + t v_i - \frac{t^2}{2}\tilde{g}_i\rangle^2\Bigg) + \mathcal{O}\left(t^3\right), \\
&= 2\sum_{i=1}^{n}\Big\{\|g_i\|^2 + 2t\langle g_i, v_i\rangle - t^2\langle g_i, \tilde{g}_i\rangle + t^2\|v_i\|^2
\end{aligned}$$

$$
\begin{aligned}
&- \Big(\langle \sigma_i, g_i \rangle + t \langle \sigma_i, v_i \rangle - \frac{t^2}{2} \langle \sigma_i, \tilde{g}_i \rangle \\
&+ t \langle u_i, g_i \rangle + t^2 \langle u_i, v_i \rangle - \frac{t^2}{2} \|u_i\|^2 \langle \sigma_i, g_i \rangle \Big)^2 \Big\} + \mathcal{O}\left(t^3\right).
\end{aligned}
$$

Observe that as $\boldsymbol{\sigma}$ is a local maximum, we have $\sigma_i = g_i / \|g_i\|$ for all $i \in [n]$. This follows since the first-order stationarity condition implies $\sigma_i = \pm g_i / \|g_i\|$ for all $i \in [n]$; and having $\sigma_i = -g_i / \|g_i\|$ for some $i \in [n]$ conflicts with the assumption that $\boldsymbol{\sigma}$ is a local maximum as replacing σ_i with any other feasible point on the sphere increases the objective function. We also have that $\langle \sigma_i, u_i \rangle = 0$ for all $i \in [n]$, as $\boldsymbol{u} \in T_{\boldsymbol{\sigma}} \mathcal{M}_r$. Using these facts in the above equality, we get

$$
\begin{aligned}
\|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 &= 2 \sum_{i=1}^{n} \Big[\|g_i\|^2 + 2t \|g_i\| \langle \sigma_i, v_i \rangle - t^2 \|g_i\| \langle \sigma_i, \tilde{g}_i \rangle + t^2 \|v_i\|^2 \\
&\quad - \Big(\|g_i\| + t \langle \sigma_i, v_i \rangle - \frac{t^2}{2} \langle \sigma_i, \tilde{g}_i \rangle \\
&\quad + t^2 \langle u_i, v_i \rangle - \frac{t^2}{2} \|u_i\|^2 \|g_i\| \Big)^2 \Big] + \mathcal{O}\left(t^3\right), \\
&= 2 \sum_{i=1}^{n} \Big[\|g_i\|^2 + 2t \|g_i\| \langle \sigma_i, v_i \rangle - t^2 \|g_i\| \langle \sigma_i, \tilde{g}_i \rangle + t^2 \|v_i\|^2 \\
&\quad - \Big(\|g_i\|^2 + 2t \|g_i\| \langle \sigma_i, v_i \rangle - t^2 \|g_i\| \langle \sigma_i, \tilde{g}_i \rangle + 2t^2 \|g_i\| \langle u_i, v_i \rangle \\
&\quad - t^2 \|u_i\|^2 \|g_i\|^2 + t^2 \langle \sigma_i, v_i \rangle^2 \Big) \Big] + \mathcal{O}\left(t^3\right), \\
&= 2t^2 \sum_{i=1}^{n} \Big(\|v_i\|^2 - \langle \sigma_i, v_i \rangle^2 \\
&\quad - 2 \|g_i\| \langle u_i, v_i \rangle + \|u_i\|^2 \|g_i\|^2 \Big) + \mathcal{O}\left(t^3\right). \qquad (20)
\end{aligned}
$$

Since $\langle \sigma_i, u_i \rangle = 0$ for all $i \in [n]$, we have by the Pythagorean theorem that

$$
\|v_i\|^2 - \langle \sigma_i, v_i \rangle^2 - \langle \frac{u_i}{\|u_i\|}, v_i \rangle^2 \geq 0.
$$

Using this inequality in (20), we get

$$
\begin{aligned}
\|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 &\geq 2t^2 \sum_{i=1}^{n} \Big(\langle \frac{u_i}{\|u_i\|}, v_i \rangle^2 - 2 \|g_i\| \langle u_i, v_i \rangle + \|u_i\|^2 \|g_i\|^2 \Big) + \mathcal{O}\left(t^3\right), \\
&= 2t^2 \sum_{i=1}^{n} \Big(\|u_i\| \|g_i\| - \langle \frac{u_i}{\|u_i\|}, v_i \rangle \Big)^2 + \mathcal{O}\left(t^3\right). \qquad (21)
\end{aligned}
$$

In order to lower bound (21) by $c(f(\boldsymbol{\sigma}) - f(\boldsymbol{\sigma}^k))$, we consider the second order Taylor approximation of $f(\boldsymbol{\sigma}^k)$, which can be written as follows

$$\begin{aligned}
f(\boldsymbol{\sigma}^k) &= \sum_{i=1}^n \langle \sigma_i^k, g_i^k \rangle, \\
&= \sum_{i=1}^n \langle \sigma_i + t u_i - \frac{t^2}{2}\|u_i\|^2 \sigma_i,\ g_i + t v_i - \frac{t^2}{2}\tilde{g}_i \rangle + \mathcal{O}\left(t^3\right), \\
&= \sum_{i=1}^n \Big(\langle \sigma_i, g_i \rangle + t\langle \sigma_i, v_i \rangle - \frac{t^2}{2}\langle \sigma_i, \tilde{g}_i \rangle + t\langle u_i, g_i \rangle + t^2 \langle u_i, v_i \rangle \\
&\quad - \frac{t^2}{2}\|u_i\|^2 \langle \sigma_i, g_i \rangle \Big) + \mathcal{O}\left(t^3\right).
\end{aligned}$$

Similar to the previous derivations, using the fact that $\sigma_i = g_i / \|g_i\|$ and $\langle \sigma_i, u_i \rangle = 0$ for all $i \in [n]$, we obtain

$$\begin{aligned}
f(\boldsymbol{\sigma}^k) &= f(\boldsymbol{\sigma}) + \sum_{i=1}^n \left(t\langle \sigma_i, v_i \rangle - \frac{t^2}{2}\langle \sigma_i, \tilde{g}_i \rangle + t^2 \langle u_i, v_i \rangle - \frac{t^2}{2}\|u_i\|^2 \langle \sigma_i, g_i \rangle \right) + \mathcal{O}\left(t^3\right), \\
&= f(\boldsymbol{\sigma}) + \sum_{i=1}^n \Bigg(t \sum_{j \neq i} A_{ij} \langle \sigma_i, u_j \rangle - \frac{t^2}{2} \sum_{j \neq i} A_{ij} \|u_j\|^2 \langle \sigma_i, \sigma_j \rangle \\
&\quad + t^2 \langle u_i, v_i \rangle - \frac{t^2}{2}\|u_i\|^2 \langle \sigma_i, g_i \rangle \Bigg) + \mathcal{O}\left(t^3\right), \\
&= f(\boldsymbol{\sigma}) + t \sum_{j=1}^n \sum_{i \neq j} A_{ji} \langle \sigma_i, u_j \rangle - \frac{t^2}{2} \sum_{j=1}^n \sum_{i \neq j} A_{ji} \|u_j\|^2 \langle \sigma_i, \sigma_j \rangle \\
&\quad + t^2 \sum_{i=1}^n \left(\langle u_i, v_i \rangle - \frac{1}{2}\|u_i\|^2 \langle \sigma_i, g_i \rangle \right) + \mathcal{O}\left(t^3\right),
\end{aligned}$$

where the last line follows since $\boldsymbol{A}$ is symmetric. Using the definition $g_j = \sum_{i \neq j} A_{ji} \sigma_i$ and $\sigma_i = g_i / \|g_i\|$ in the above inequality yields

$$\begin{aligned}
f(\boldsymbol{\sigma}^k) &= f(\boldsymbol{\sigma}) + t \sum_{j=1}^n \langle g_j, u_j \rangle - \frac{t^2}{2} \sum_{j=1}^n \|u_j\|^2 \langle g_j, \sigma_j \rangle \\
&\quad + t^2 \sum_{i=1}^n \left(\langle u_i, v_i \rangle - \frac{1}{2}\|u_i\|^2 \langle \sigma_i, g_i \rangle \right) + \mathcal{O}\left(t^3\right), \\
&= f(\boldsymbol{\sigma}) + t^2 \sum_{i=1}^n \left(\langle u_i, v_i \rangle - \|u_i\|^2 \|g_i\| \right) + \mathcal{O}\left(t^3\right). \qquad (22)
\end{aligned}$$

Reorganizing terms, we get

$$f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k) = f(\boldsymbol{\sigma}) - f(\boldsymbol{\sigma}^k) = t^2 \sum_{i=1}^{n} \left(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \right) + \mathcal{O}\left(t^3\right). \tag{23}$$

Turning back our attention to (21), we can lower bound the right-hand side as follows

$$\begin{aligned} \|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 &\geq 2t^2 \sum_{i=1}^{n} \frac{1}{\|u_i\|^2} \left(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \right)^2 + \mathcal{O}\left(t^3\right), \\ &\geq 2t^2 \sum_{i=1}^{n} \left(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \right)^2 + \mathcal{O}\left(t^3\right), \\ &\geq \frac{2t^2}{n} \left(\sum_{i=1}^{n} \left(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \right) \right)^2 + \mathcal{O}\left(t^3\right), \end{aligned}$$

where the second inequality follows since $\|u_i\|^2 \leq \|\boldsymbol{u}\|_{\mathrm{F}}^2 = 1$ and the last inequality follows since $\left(\sum_{i=1}^{n} a_i\right)^2 \leq n \sum_{i=1}^{n} a_i^2$, for all $a_i \in \mathbb{R}$, $i \in [n]$. Using the second order approximation derived in (23) in the above inequality, we obtain

$$\begin{aligned} \|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 &\geq \frac{f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k)}{n} \sum_{i=1}^{n} 2 \left(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \right) + \mathcal{O}\left(t^3\right), \\ &= \frac{2\langle \boldsymbol{u}, (\boldsymbol{\Lambda} - \boldsymbol{A})\boldsymbol{u} \rangle}{n} \left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k) \right) + \mathcal{O}\left(t^3\right), \end{aligned}$$

where $\boldsymbol{\Lambda} = \operatorname{Diag}(\|g_1\|, \ldots, \|g_n\|)$. Since we have $2\langle \boldsymbol{u}, (\boldsymbol{A} - \boldsymbol{\Lambda})\boldsymbol{u} \rangle \leq -\mu \|\boldsymbol{u}\|_{\mathrm{F}}^2$ by the quadratic decay condition, we conclude that

$$\|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 \geq \frac{\mu}{n} \left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k) \right) + \mathcal{O}\left(t^3\right). \tag{24}$$

This implies that whenever σ^k is sufficiently close to σ, i.e., whenever t is sufficiently small (cf. (19)), the remainder in the Taylor approximation, i.e., the $\mathcal{O}\left(t^3\right)$ terms, will be dominated by $\frac{\mu}{n}\left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k)\right)$. In particular, if $\boldsymbol{\sigma}^0$ is sufficiently close to $\bar{\boldsymbol{\sigma}}$ to satisfy $\mathcal{O}\left(t^3\right) \geq -\frac{\mu}{2n}\left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k)\right)$ in the above inequality (see Appendix B for a proof of this), we then have

$$\|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 \geq \frac{\mu}{2n} \left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k) \right). \tag{25}$$

Combining this inequality with (18), we get

$$f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \frac{\mu}{4n^2 \|\boldsymbol{A}\|_1} \left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k) \right). \tag{26}$$

Rearranging terms in the above inequality concludes the proof. □

The linear convergence rate of the BCM algorithm with greedy coordinate selection in Theorem 2 can be extended for importance sampling and uniform sampling as we highlight in the following (its proof follows similar lines to the proofs of Theorem 2 and Corollary 1 and hence is omitted).

Corollary 2 *Let the conditions in Theorem 2 hold. Then, the iterates generated by the BCM algorithm enjoys the local linear convergence rate*

$$f(\bar{\boldsymbol{\sigma}}) - \mathbb{E} f(\boldsymbol{\sigma}^k) \leq (1-\rho)^k \left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^0) \right), \tag{27}$$

where $\rho = \frac{\mu}{4n\|\boldsymbol{A}\|_{1,1}}$ *for importance sampling and* $\rho = \frac{\mu}{4n^2\|\boldsymbol{A}\|_1}$ *for uniform sampling.*

3.4 Quadratic decay condition holds generically

In this section, we consider the quadratic decay condition, which is a condition on (Non-CVX), and relate it to a condition on the original problem in (CVX). In particular, we characterize sufficient conditions on (CVX) for quadratic decay to hold. We first provide some background on semidefinite programming (see for example [3], for a more detailed treatment of this topic). Consider the SDP in (CVX):

$$\begin{aligned} \text{maximize} \quad & \langle \boldsymbol{A}, \boldsymbol{X} \rangle \\ \text{subject to} \quad & X_{ii} = 1, \ \text{for } i \in [n], \\ & \boldsymbol{X} \succeq 0, \end{aligned}$$

and its dual:

$$\begin{aligned} \text{minimize} \quad & \langle 1, y \rangle \\ \text{subject to} \quad & \boldsymbol{Z} = \text{Diag}(y) - \boldsymbol{A}, \\ & \boldsymbol{Z} \succeq 0, \end{aligned}$$

where 1 is the vector of ones of appropriate size. Let $\boldsymbol{X}^*$ and $(y^*, \boldsymbol{Z}^*)$ denote the primal and dual optimal solutions, respectively, and let r^* denote the rank of $\boldsymbol{X}^*$. Then, there exists a $\boldsymbol{Q} \in O(n)$ such that

$$\begin{aligned} \boldsymbol{X}^* &= \boldsymbol{Q}\, \text{Diag}(\lambda_1, \ldots, \lambda_{r^*}, 0, \ldots, 0)\, \boldsymbol{Q}^\top, \\ \boldsymbol{Z}^* &= \boldsymbol{Q}\, \text{Diag}(0, \ldots, 0, \omega_{r^*+1}, \ldots, \omega_n)\, \boldsymbol{Q}^\top. \end{aligned}$$

We say that *strict complementarity* holds if $\lambda_i > 0$ for $i = 1, \ldots, r^*$ and $\omega_j > 0$ for $j = r^* + 1, \ldots, n$. Furthermore, let $\boldsymbol{Q}_1 \in \mathbb{R}^{n \times r^*}$ and $\boldsymbol{Q}_2 \in \mathbb{R}^{n \times (n-r^*)}$ respectively denote the first r^* columns and the last $n - r^*$ columns of $\boldsymbol{Q}$ and let q_i denote the ith row of $\boldsymbol{Q}_1$, i.e., $\boldsymbol{Q}_1 = [q_1, q_2, \ldots, q_n]^\top$. Then, $(y^*, \boldsymbol{Z}^*)$ is *dual nondegenerate* if and only if $\{q_1 q_1^\top, \ldots, q_n q_n^\top\}$ spans Sym_{r^*}, i.e., the set of real symmetric $r^* \times r^*$ matrices

[3, Theorem 3]. Strict complementarity and dual nondegeneracy are known to hold generically (over the set of possible cost matrices $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, i.e., they fail to hold only on a subset of measure zero of $\mathbb{R}^{n\times n}$) as proven in [3, Lemma 2]. Using these definitions, we show in the next theorem that strict complementarity and dual nondegeneracy are sufficient for quadratic decay to hold at the maximizer of (Non-CVX).

Theorem 3 *Suppose that* $\boldsymbol{X}^* = \boldsymbol{\sigma}\boldsymbol{\sigma}^\top$ *and* $(y^*, \boldsymbol{Z}^*) = (\mathrm{diag}(\boldsymbol{\Lambda}), \boldsymbol{\Lambda} - \boldsymbol{A})$ *are respectively primal and dual optimal solutions satisfying strict complementarity and dual nondegeneracy, where* $\boldsymbol{\Lambda} = \mathrm{Diag}(\|g_1\|, \ldots, \|g_n\|)$. *If* $r \geq \mathrm{rank}(\boldsymbol{X}^*)$, *then quadratic decay is satisfied for* f *at all* $\bar{\boldsymbol{\sigma}}$ *such that* $\bar{\boldsymbol{\sigma}}\bar{\boldsymbol{\sigma}}^\top = \boldsymbol{X}^*$.

Proof Suppose $\mathrm{rank}(\boldsymbol{X}^*) = r^* \leq r$, then by strict complementarity, we have $\mathrm{rank}(\boldsymbol{Z}^*) = n - r^*$ and kernel of $\boldsymbol{Z}^*$ is equal to the column space of $\boldsymbol{X}^*$, i.e., $\ker(\boldsymbol{Z}^*) = \mathrm{col}(\boldsymbol{X}^*)$. Since $\boldsymbol{X}^* = \boldsymbol{\sigma}\boldsymbol{\sigma}^\top$ and $\boldsymbol{Z}^* = \boldsymbol{\Lambda} - \boldsymbol{A}$, we equivalently have $\ker(\boldsymbol{\Lambda} - \boldsymbol{A}) = \mathrm{col}(\boldsymbol{\sigma})$. As $\boldsymbol{Z}^*$ is feasible for the dual, then $\boldsymbol{Z}^* = \boldsymbol{\Lambda} - \boldsymbol{A} \succeq 0$, and consequently $\langle \boldsymbol{u}, (\boldsymbol{\Lambda} - \boldsymbol{A})\boldsymbol{u}\rangle \geq 0$, for all $\boldsymbol{u} \in \mathbb{R}^{n\times r}$.

Now consider the quadratic form $h(\boldsymbol{u}) := \langle \boldsymbol{u}, (\boldsymbol{\Lambda} - \boldsymbol{A})\boldsymbol{u}\rangle$ over $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$. First, we show that $h(\boldsymbol{u}) = 0$ if and only if $\boldsymbol{u} \in \mathcal{V}_{\boldsymbol{\sigma}}$. The *if* direction of the proof is straightforward, i.e., $(\boldsymbol{\Lambda} - \boldsymbol{A})\boldsymbol{\sigma} = 0$ and $\boldsymbol{u} = \boldsymbol{\sigma}\boldsymbol{B}$ for some skew-symmetric matrix $\boldsymbol{B}$ directly imply $h(\boldsymbol{u}) = 0$ for all $\boldsymbol{u} \in \mathcal{V}_{\boldsymbol{\sigma}}$. To show the *only if* direction, let $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$ such that $h(\boldsymbol{u}) = 0$, or equivalently $\mathrm{tr}((\boldsymbol{\Lambda} - \boldsymbol{A})\boldsymbol{u}\boldsymbol{u}^\top) = 0$. As both $\boldsymbol{\Lambda} - \boldsymbol{A}$ and $\boldsymbol{u}\boldsymbol{u}^\top$ are positive semidefinite matrices, this implies $(\boldsymbol{\Lambda} - \boldsymbol{A})\boldsymbol{u} = 0$. Therefore, columns of $\boldsymbol{u}$ are in $\ker(\boldsymbol{\Lambda} - \boldsymbol{A}) = \mathrm{col}(\boldsymbol{\sigma})$, which implies there exists $\boldsymbol{B} \in \mathbb{R}^{r\times r}$ such that $\boldsymbol{u} = \boldsymbol{\sigma}\boldsymbol{B}$ (note that it is not possible to make this claim without strict complementarity). As $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r$, then $\langle \sigma_i, u_i\rangle = \langle \sigma_i, \boldsymbol{B}^\top\sigma_i\rangle = \langle \sigma_i\sigma_i^\top, \boldsymbol{B}\rangle = 0$, for all $i \in [n]$. Without loss of generality, assume that the last $r - r^*$ columns of $\boldsymbol{\sigma}$ are equal to zero. Then, by dual nondegeneracy of the SDP, the principal submatrices of dimension $r^* \times r^*$ of $\{\sigma_i\sigma_i^\top\}_{i=1}^n$ spans $\mathcal{S}^{r^*}$. Consider the decomposition

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} \\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22} \end{bmatrix},$$

where $\boldsymbol{B}_{11} \in \mathbb{R}^{r^*\times r^*}$ and $\boldsymbol{B}_{22} \in \mathbb{R}^{(r-r^*)\times(r-r^*)}$. Then, the dual nondegeneracy implies that $\boldsymbol{B}_{11}$ is a skew-symmetric matrix, i.e., $\boldsymbol{B}_{11}^\top = -\boldsymbol{B}_{11}$. Furthermore, as the last $r - r^*$ columns of $\boldsymbol{\sigma}$ are equal to zero, then $\boldsymbol{u} = \boldsymbol{\sigma}\boldsymbol{B}$ does not depend on $\boldsymbol{B}_{21}$ and $\boldsymbol{B}_{22}$. Therefore, we can pick $\boldsymbol{B}_{21} = -\boldsymbol{B}_{12}^\top$ and $\boldsymbol{B}_{22} = 0$ such that $\boldsymbol{B}$ is a skew-symmetric matrix and observe that $\boldsymbol{u} \in \mathcal{V}_{\boldsymbol{\sigma}}$. The same argument can be extended for all $\bar{\boldsymbol{\sigma}}$ such that $\bar{\boldsymbol{\sigma}}\bar{\boldsymbol{\sigma}}^\top = \boldsymbol{X}^*$ using parallel transport.

To conclude the proof, we let $\{\boldsymbol{u}^\ell\}_{\ell=1}^{n(r-1)}$ be an orthogonal basis to $T_{\boldsymbol{\sigma}}\mathcal{M}_r$ such that $\{\boldsymbol{u}^\ell\}_{\ell=1}^{s}$ is a basis for $\mathcal{V}_{\boldsymbol{\sigma}}$. Let $\boldsymbol{M} \in \mathbb{R}^{n(r-1)\times n(r-1)}$ such that $M_{ij} = \langle \boldsymbol{u}^i, (\boldsymbol{\Lambda} - \boldsymbol{A})\boldsymbol{u}^j\rangle$. Consider the function $\bar{h} : \mathbb{R}^{n(r-1)} \to \mathbb{R}^{n(r-1)}$ such that $\bar{h}(\boldsymbol{v}) = \boldsymbol{v}^\top\boldsymbol{M}\boldsymbol{v}$ and observe that $\bar{h}(\mathrm{vec}(\boldsymbol{u})) = h(\boldsymbol{u})$. Let $\boldsymbol{L} = [\mathrm{vec}(\boldsymbol{u}^1), \ldots, \mathrm{vec}(\boldsymbol{u}^s)]^\top \in \mathbb{R}^{s\times n(r-1)}$, then $\boldsymbol{v}^\top\boldsymbol{M}\boldsymbol{v} > 0$ for all $\boldsymbol{v}$ such that $\boldsymbol{L}\boldsymbol{v} = 0$ and $\boldsymbol{v} \neq \boldsymbol{0}$. Then, by Finsler's Lemma, $\boldsymbol{L}_\perp^\top\boldsymbol{M}\boldsymbol{L}_\perp \succ 0$, where $\boldsymbol{L}_\perp$ is any basis of the right null-space of $\boldsymbol{L}$. Equivalently, there exists $\mu > 0$ such that $h(\boldsymbol{u}) \geq \mu\|\boldsymbol{u}\|_{\mathrm{F}}^2$ for all $\boldsymbol{u} \in T_{\boldsymbol{\sigma}}\mathcal{M}_r \setminus \mathcal{V}_{\boldsymbol{\sigma}}$. □

Remark 1 Finsler's Lemma [13, Lemma C.11.2] also yields that $\mu = \lambda_{\min}(\boldsymbol{L}_\perp^\top\boldsymbol{M}\boldsymbol{L}_\perp)$.

This theorem states that quadratic decay holds for all global maxima of (Non-CVX) provided that the rank of the factorization is large enough so that the global maximum values of (CVX) and (Non-CVX) are equal to one another. For this case, the set of all global maxima is an equivalence class corresponding to a solution since strict complementarity and dual nondegeneracy imply that the primal solution of (CVX) is unique. On top of this, when $r \geq \sqrt{2n}$, it is known that (see [11, Theorem 2]) any local maximum is global generically (i.e., for almost all cost matrices $\boldsymbol{A}$). As strict complementarity and dual nondegeneracy also hold generically for (CVX), then consequently, when $r \geq \sqrt{2n}$, quadratic decay holds for all local maxima generically as we highlight in the following corollary.

Corollary 3 *If $r \geq \sqrt{2n}$, then quadratic decay holds for all local maxima generically.*

4 Approximately achieving the maximum value of (CVX)

Our results in Sect. 3 show that the BCM algorithm converges with a sublinear rate to a first-order stationary solution and with a linear rate to a local maximum when initialized sufficiently close to it. In this section, we incorporate a second-order oracle to BCM in order to obtain an algorithm, which we refer to as BCM2, that returns an approximate second-order stationary point. More specifically, at the current iteration of the algorithm, if the norm of the gradient is large, we take a BCM step. Otherwise, we run a subroutine (e.g., Lanczos method) to find the leading eigenvector of the Hessian. The main motivation for designing such an algorithm is that the approximate second-order stationary solutions provide $\mathcal{O}(1/r)$ approximation to (CVX). In particular, call $\boldsymbol{\sigma}$ an *ε-approximate concave point* if $\langle \boldsymbol{u}, \mathrm{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}] \rangle \leq \varepsilon \langle \boldsymbol{u}, \boldsymbol{u} \rangle$, for all $\boldsymbol{u} \in T_{\boldsymbol{\sigma}} \mathcal{M}_r$. Then, the following theorem provides an approximation ratio between the approximate concave points of (Non-CVX) and the maximum value of (CVX).

Theorem 4 [32, Theorem 1] *Let $\boldsymbol{\sigma} \in \mathcal{M}_r$ be an ε-approximate concave point. Then, for any positive semidefinite $\boldsymbol{A}$, the following approximation ratio holds:*

$$f(\boldsymbol{\sigma}) \geq \left(1 - \frac{1}{r-1}\right) \mathrm{SDP}(\boldsymbol{A}) - \frac{n}{2}\varepsilon, \tag{28}$$

where $\mathrm{SDP}(\boldsymbol{A})$ is the maximum value of (CVX).

This approximation ratio follows due to a generalization of the randomized rounding approach (most famously presented by [22]) applied to an ε-approximate concave point. In fact, it can be shown that it is not possible to find a better approximation ratio (in terms of the dependence on the rank of the factorization r) for all problems $\boldsymbol{A}$. This result is highlighted in the following theorem.

Theorem 5 [14, Theorems 1 & 3] *Let $\mathrm{SDP}(\boldsymbol{A})$ be the maximum value of (CVX) and $\mathrm{SDP}_r(\boldsymbol{A})$ be the maximum value of (Non-CVX). Then, for all positive semidefinite matrices $\boldsymbol{A}$, the following approximation ratio holds:*

$$1 \geq \frac{\mathrm{SDP}_r(\boldsymbol{A})}{\mathrm{SDP}(\boldsymbol{A})} \geq \gamma(r) = \frac{2}{r}\left(\frac{\Gamma((r+1)/2)}{\Gamma(r/2)}\right)^2 = 1 - \Theta(1/r), \tag{29}$$

Algorithm 2 BCM2

Initialize $\boldsymbol{\sigma}^0 \in \mathbb{R}^{n\times r}$ and calculate $g_i^0 = \sum_{j\neq i} A_{ij}\sigma_j^0$, for all $i \in [n]$.
for $k = 0, 1, 2, \ldots$ **do**
  Compute $\|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 = 2\sum_{i=1}^n (\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k\rangle^2)$.
  if $\|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 > \varepsilon^3/(1350\|\boldsymbol{A}\|_1)$ **then**
    $i_k \leftarrow \arg\max_{i\in[n]}(\|g_i^k\| - \langle \sigma_i^k, g_i^k\rangle)$
    $\sigma_{i_k}^{k+1} \leftarrow g_{i_k}^k/\|g_{i_k}^k\|$.
    $g_i^{k+1} \leftarrow g_i^k - A_{ii_k}\sigma_{i_k}^k + A_{ii_k}\sigma_{i_k}^{k+1}$, for all $i \neq i_k$.
  else
    Find a direction $\boldsymbol{u}^k \in T_{\boldsymbol{\sigma}^k}\mathcal{M}_r$ such that $\langle \boldsymbol{u}^k, \operatorname{Hess} f(\boldsymbol{\sigma}^k)[\boldsymbol{u}^k]\rangle \geq \lambda_{\max}(\operatorname{Hess} f(\boldsymbol{\sigma}^k))/2$, $\langle \boldsymbol{u}^k, \operatorname{grad} f(\boldsymbol{\sigma}^k)\rangle \geq 0$, and $\|\boldsymbol{u}^k\|_{\mathrm{F}} = 1$.
    $\sigma_i^{k+1} \leftarrow \sigma_i^k \cos(\|u_i^k\|t) + \frac{u_i^k}{\|u_i^k\|}\sin(\|u_i^k\|t)$, for all $i \in [n]$, where $t = \varepsilon/(15\|\boldsymbol{A}\|_1)$.
    $g_i^{k+1} \leftarrow \sum_{j\neq i} A_{ij}\sigma_j^{k+1}$, for all $i \in [n]$.
  end if
end for

where $\Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx$ is the Gamma function. Furthermore, under the unique games conjecture, there is no polynomial-time algorithm that approximates $\mathrm{SDP}_r(\boldsymbol{A})$ *with an approximation ratio greater than $\gamma(r) + \varepsilon$ for any $\varepsilon > 0$.*

These results provide motivation to design algorithms with second-order guarantees to solve (Non-CVX) and for this reason, we propose the BCM2 algorithm (see Algorithm 2), which can be described as follows: When the Frobenius norm of the Riemannian gradient is at least as large as $\|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 > \epsilon^3/(1350\|\boldsymbol{A}\|_1)$, we use BCM to update the current solution. Otherwise, we assume that there is a second-order oracle that returns an update direction $\boldsymbol{u}^k \in T_{\boldsymbol{\sigma}^k}\mathcal{M}_r$ such that $\langle \boldsymbol{u}^k, \operatorname{Hess} f(\boldsymbol{\sigma}^k)[\boldsymbol{u}^k]\rangle \geq \lambda_{\max}(\operatorname{Hess} f(\boldsymbol{\sigma}^k))/2$, $\langle \boldsymbol{u}^k, \operatorname{grad} f(\boldsymbol{\sigma}^k)\rangle \geq 0$, and $\|\boldsymbol{u}^k\|_{\mathrm{F}} = 1$. Notice that finding a tangent vector $\boldsymbol{u}^k$ that satisfy $\langle \boldsymbol{u}^k, \operatorname{Hess} f(\boldsymbol{\sigma}^k)[\boldsymbol{u}^k]\rangle \geq \lambda_{\max}(\operatorname{Hess} f(\boldsymbol{\sigma}^k))/2$ and $\|\boldsymbol{u}^k\|_{\mathrm{F}} = 1$ is an eigenpair problem and can be solved efficiently using the Lanczos method. The condition $\langle \boldsymbol{u}^k, \operatorname{grad} f(\boldsymbol{\sigma}^k)\rangle \geq 0$, on the other hand, can always be satisfied by switching the sign of $\boldsymbol{u}^k$. It is a straightforward exercise to explicitly construct such a vector and it can be found in [9, Lemma 11]. Once the update direction $\boldsymbol{u}^k \in T_{\boldsymbol{\sigma}^k}\mathcal{M}_r$ is obtained, we take a step towards this direction using the geodesics on the manifold. When the step size is carefully chosen, it can be shown that the objective value of the iterates generated by this procedure is a monotonically increasing sequence until the approximate second-order stationary condition is satisfied. This property is presented in the following lemma.

Lemma 2 *Let $\boldsymbol{u}^k \in T_{\boldsymbol{\sigma}^k}\mathcal{M}_r$ such that $\|\boldsymbol{u}^k\|_{\mathrm{F}} = 1$, $\langle \boldsymbol{u}^k, \operatorname{grad} f(\boldsymbol{\sigma}^k)\rangle \geq 0$, and $\langle \boldsymbol{u}^k, \operatorname{Hess} f(\boldsymbol{\sigma}^k)[\boldsymbol{u}^k]\rangle \geq \varepsilon/2$. Consider the update rule given by the exponential map $\boldsymbol{\sigma}^{k+1} = \operatorname{Exp}_{\boldsymbol{\sigma}^k}(t\boldsymbol{u}^k)$, i.e.,*

$$\sigma_i^{k+1} = \sigma_i^k \cos(\|u_i^k\|t) + \frac{u_i^k}{\|u_i^k\|}\sin(\|u_i^k\|t), \quad \text{for all } i \in [n], \tag{30}$$

where $t = \frac{\varepsilon}{15\|A\|_1}$ is the step size. These iterates satisfy the following ascent in the function value:

$$f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \frac{\varepsilon^3}{2700\|\boldsymbol{A}\|_1^2}.$$

Proof The Taylor expansion of $\boldsymbol{\sigma}^{k+1}$ around $\boldsymbol{\sigma}^k$ is given by

$$\begin{aligned}\sigma_i^{k+1} &= \sigma_i^k \sum_{\ell=0}^{\infty} \frac{(-1)^\ell}{(2\ell)!}(\|u_i^k\|t)^{2\ell} + u_i^k \sum_{\ell=0}^{\infty} \frac{(-1)^\ell}{(2\ell+1)!}(\|u_i^k\|t)^{2\ell+1}, \\ &= \sigma_i^k + t u_i^k - \frac{t^2}{2}\|u_i^k\|^2\sigma_i^k - \frac{t^3}{6}\|u_i^k\|^2 u_i^k + \ldots,\end{aligned}$$

and using this, we can compute the Taylor expansion of $f(\boldsymbol{\sigma}^{k+1})$ as follows

$$\begin{aligned}f(\boldsymbol{\sigma}^{k+1}) &= \sum_{i=1}^{n}\sum_{j\neq i} A_{ij}\langle \sigma_i^{k+1}, \sigma_j^{k+1}\rangle, \\ &= \sum_{i=1}^{n}\sum_{j\neq i} A_{ij}\Bigg[\langle \sigma_i^k, \sigma_j^k\rangle + t\left(\langle \sigma_i^k, u_j^k\rangle + \langle u_i^k, \sigma_j^k\rangle\right) \\ &\quad + \frac{t^2}{2}\left(-\|u_j^k\|^2\langle \sigma_i^k, \sigma_j^k\rangle + 2\langle u_i^k, u_j^k\rangle - \|u_i^k\|^2\langle \sigma_i^k, \sigma_j^k\rangle\right)\Bigg] - t^3\beta,\end{aligned}$$

where β represents the third and higher-order terms. Using the definitions of $f(\boldsymbol{\sigma}^k)$ and its derivatives, the above equality can be written as follows

$$f(\boldsymbol{\sigma}^{k+1}) = f(\boldsymbol{\sigma}^k) + t\langle \boldsymbol{u}^k, \operatorname{grad} f(\boldsymbol{\sigma}^k)\rangle + \frac{t^2}{2}\langle \boldsymbol{u}^k, \operatorname{Hess} f(\boldsymbol{\sigma}^k)[\boldsymbol{u}^k]\rangle - t^3\beta. \quad (31)$$

Here, our aim is to upper bound the magnitude of the remainder term corresponding to the third and higher-order terms. To this end, we upper bound the higher-order terms using the Cauchy-Schwarz inequality for each term individually. This yields

$$|\beta| \leq \sum_{i=1}^{n}\sum_{j\neq i} |A_{ij}| \left(\sum_{\ell=3}^{\infty} \frac{t^{\ell-3}}{\ell!}(\|u_i^k\| + \|u_j^k\|)^\ell\right).$$

As $t < 1$ and $\boldsymbol{A}$ is a symmetric matrix, we can upper bound the right hand-side of the above inequality as follows

$$|\beta| \leq \|\boldsymbol{A}\|_1 \sum_{i=1}^{n} \left(\sum_{\ell=3}^{\infty} \frac{2^\ell}{\ell!}\|u_i^k\|^\ell\right).$$

Since $\|u_i\| \leq 1$ for all $i \in [n]$, we consequently have

$$|\beta| \leq \|\boldsymbol{A}\|_1 \left(\sum_{i=1}^{n} \|u_i^k\|^2 \right) \left(\sum_{\ell=3}^{\infty} \frac{2^\ell}{\ell!} \right) = \|\boldsymbol{A}\|_1 \sum_{\ell=3}^{\infty} \frac{2^\ell}{\ell!}.$$

where the latter equality follows since $\|\boldsymbol{u}^k\|_{\mathrm{F}} = 1$. Using $\sum_{\ell=3}^{\infty} \frac{2^\ell}{\ell!} = e^2 - 5 \leq 5/2$ above and plugging this bound back in (31), we obtain

$$f(\boldsymbol{\sigma}^{k+1}) \geq f(\boldsymbol{\sigma}^k) + t\langle \boldsymbol{u}^k, \operatorname{grad} f(\boldsymbol{\sigma}^k) \rangle + \frac{t^2}{2} \langle \boldsymbol{u}^k, \operatorname{Hess} f(\boldsymbol{\sigma}^k)[\boldsymbol{u}^k] \rangle - \frac{5\|\boldsymbol{A}\|_1}{2} t^3. \tag{32}$$

Since we are given that $\langle \boldsymbol{u}^k, \operatorname{grad} f(\boldsymbol{\sigma}^k) \rangle \geq 0$ and $\langle \boldsymbol{u}^k, \operatorname{Hess} f(\boldsymbol{\sigma}^k)[\boldsymbol{u}^k] \rangle \geq \varepsilon/2$, (32) yields

$$f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \frac{\varepsilon}{4} t^2 - \frac{5\|\boldsymbol{A}\|_1}{2} t^3.$$

Choosing $t = \frac{\varepsilon}{15\|\boldsymbol{A}\|_1}$ maximizes the right-hand side of the above inequality and concludes the proof. □

Using this ascent lemma, we next analyze the global convergence of Algorithm 2 in Theorem 6, where we assume that we have access to a subroutine that solves the eigenpair problem to the desired accuracy. We then implement the subroutine using the Lanczos algorithm (presented in Algorithm 3) and present its convergence in Theorem 7. In particular, we have the following theorem for the former case.

Theorem 6 *Consider Algorithm 2, where BCM is used at iteration k if* $\|\operatorname{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 \geq \varepsilon^3/(1350\|\boldsymbol{A}\|_1)$ *and a second-order step (see lines 9-11 of Algorithm 2) is taken otherwise. Let* K_{BCM} *denote the number of BCM epochs made and let* K_{H} *denote the number of second-order oracle iterations made such that* $K = nK_{\mathrm{BCM}} + K_{\mathrm{H}}$. *Then, as soon as*

$$K_{\mathrm{BCM}} + K_{\mathrm{H}} = \left\lceil \frac{675n\|\boldsymbol{A}\|_1^2}{\varepsilon^2} \right\rceil, \tag{33}$$

Algorithm 2 is guaranteed to return a solution $\boldsymbol{\sigma}^K$ *that satisfies*

$$f(\boldsymbol{\sigma}^K) \geq \left(1 - \frac{1}{r-1}\right) \mathrm{SDP}(\boldsymbol{A}) - \frac{n}{2}\varepsilon, \tag{34}$$

where $\mathrm{SDP}(\boldsymbol{A})$ *is the maximum value of* (CVX).

Proof As we have proven previously in (12), each iteration of BCM yields the following functional ascent

$$f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^{k}) \geq \frac{\|\mathrm{grad}\, f(\boldsymbol{\sigma}^{k})\|_{\mathrm{F}}^2}{2n\|\boldsymbol{A}\|_1} \geq \frac{\varepsilon^3}{2700n\|\boldsymbol{A}\|_1^2}, \tag{35}$$

where the latter inequality holds since BCM is applied at iteration k of Algorithm 2 if $\|\mathrm{grad}\, f(\boldsymbol{\sigma}^{k})\|_{\mathrm{F}}^2 \geq \frac{\varepsilon^3}{1350\|\boldsymbol{A}\|_1}$. Similarly, by Lemma 2, each iteration of the second-order oracle yields the following functional ascent

$$f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^{k}) \geq \frac{\varepsilon^3}{2700\|\boldsymbol{A}\|_1^2}. \tag{36}$$

Hence, an epoch (n iterations) of BCM yields the same amount of function value improvement as an iteration of the second-order oracle. Let

$$f^* = \left(1 - \frac{1}{r-1}\right) \mathrm{SDP}(\boldsymbol{A})$$

denote the desired approximation ratio and consider the approximation gap of the solution $\boldsymbol{\sigma}$ with respect to f^* that is given by

$$h(\boldsymbol{\sigma}) = f^* - f(\boldsymbol{\sigma}). \tag{37}$$

The aim of the algorithm is to find a solution $\boldsymbol{\sigma}$ that satisfy $h(\boldsymbol{\sigma}) \leq \epsilon$ for some $\epsilon > 0$. Consider that the BCM2 algorithm runs K_{BCM} epochs of BCM and K_{H} iterations of the second-order oracle such that a total of $K = nK_{\mathrm{BCM}} + K_{\mathrm{H}}$ iterations are made. Let $\mathcal{G} = \{0 \leq k \leq K-1 : \|\mathrm{grad}\, f(\boldsymbol{\sigma}^{k})\|_{\mathrm{F}}^2 \geq \frac{\varepsilon^3}{1350\|A\|_1}\}$ be the set of iterations at which BCM step is taken and let $\mathcal{H} = \{0 \leq k \leq K-1\} \setminus \mathcal{G}$ be the set of iterations at which a second-order oracle step is taken. Then, the approximation gap decreases at each iteration by the following amount:

$$h(\boldsymbol{\sigma}^{k}) - h(\boldsymbol{\sigma}^{k+1}) \geq \frac{\varepsilon^3}{2700\|\boldsymbol{A}\|_1^2}\delta_k, \tag{38}$$

where, for notational simplicity, we introduced

$$\delta_k = \begin{cases} \frac{1}{n}, & \text{if } k \in \mathcal{G}, \\ 1, & \text{if } k \in \mathcal{H}. \end{cases} \tag{39}$$

By Theorem 4, we are given that any ε-approximate concave point $\boldsymbol{\sigma}$ satisfies

$$h(\boldsymbol{\sigma}) \leq \frac{n}{2}\varepsilon. \tag{40}$$

Hence, the right-hand side of (38) can be lower bounded as follows

$$h(\boldsymbol{\sigma}^k) - h(\boldsymbol{\sigma}^{k+1}) \geq \frac{2\delta_k}{675n^3\|\boldsymbol{A}\|_1^2}\, h^3(\boldsymbol{\sigma}^k). \tag{41}$$

Considering the reciprocal of the approximation gap, we observe that

$$\begin{aligned}\frac{1}{h^2(\boldsymbol{\sigma}^{k+1})} - \frac{1}{h^2(\boldsymbol{\sigma}^k)} &= \frac{\left(h(\boldsymbol{\sigma}^k) - h(\boldsymbol{\sigma}^{k+1})\right)\left(h(\boldsymbol{\sigma}^k) + h(\boldsymbol{\sigma}^{k+1})\right)}{h^2(\boldsymbol{\sigma}^{k+1})h^2(\boldsymbol{\sigma}^k)}, \\ &\geq \frac{2\delta_k}{675n^3\|\boldsymbol{A}\|_1^2}\frac{h(\boldsymbol{\sigma}^k)\left(h(\boldsymbol{\sigma}^k) + h(\boldsymbol{\sigma}^{k+1})\right)}{h^2(\boldsymbol{\sigma}^{k+1})},\end{aligned} \tag{42}$$

where the inequality follows by (41). As the right-hand side of (41) is lower bounded by zero, we have $h(\boldsymbol{\sigma}^k) \geq h(\boldsymbol{\sigma}^{k+1})$. Thus, we can lower bound the right-hand side of (42) as follows

$$\frac{1}{h^2(\boldsymbol{\sigma}^{k+1})} - \frac{1}{h^2(\boldsymbol{\sigma}^k)} \geq \frac{4\delta_k}{675n^3\|\boldsymbol{A}\|_1^2}. \tag{43}$$

Summing (43) over $k = 0, 1, \ldots, K-1$, we get

$$\frac{1}{h^2(\boldsymbol{\sigma}^K)} - \frac{1}{h^2(\boldsymbol{\sigma}^0)} \geq \sum_{k=0}^{K-1} \frac{4\delta_k}{675n^3\|\boldsymbol{A}\|_1^2} = \frac{4}{675n^3\|\boldsymbol{A}\|_1^2}(K_{\mathrm{BCM}} + K_{\mathrm{H}}).$$

Given that $\boldsymbol{\sigma}^0$ is not an ε-approximate concave point (or else, there is nothing to prove), we have

$$\frac{1}{h^2(\boldsymbol{\sigma}^K)} \geq \frac{4}{675n^3\|\boldsymbol{A}\|_1^2}(K_{\mathrm{BCM}} + K_{\mathrm{H}}). \tag{44}$$

Since by (40), we know that $\frac{1}{h(\boldsymbol{\sigma})} \geq \frac{2}{n\varepsilon}$ for any ε-approximate concave point, then as soon as

$$K_{\mathrm{BCM}} + K_{\mathrm{H}} \geq \frac{675n\|\boldsymbol{A}\|_1^2}{\varepsilon^2} \tag{45}$$

iterations made, BCM2 is guaranteed to return an ε-approximate concave point, i.e., there exists a solution $\boldsymbol{\sigma}^k$ for some $1 < k < K$ such that $h(\boldsymbol{\sigma}^k) \leq \frac{n}{2}\varepsilon$. Since $\{h(\boldsymbol{\sigma}^k)\}_{k\geq 0}$ is a nonincreasing sequence (as we have already shown in (41)), then the final iterate of the algorithm $\boldsymbol{\sigma}^K$ is guaranteed to satisfy $h(\boldsymbol{\sigma}^K) \leq \frac{n}{2}\varepsilon$, i.e., $\boldsymbol{\sigma}^K$ is an ε-approximate concave point. □

In Theorem 6, $K_{\mathrm{BCM}} + K_{\mathrm{H}}$ represents the total number of epochs to guarantee (34), whereas the iteration counter of the algorithm is given in terms of $K = nK_{\mathrm{BCM}} + K_{\mathrm{H}}$. This is due to the fact that, at each iteration of the BCM algorithm, a single row

of $\boldsymbol{\sigma}$ is updated and consequently n iterations of the BCM algorithm add up to an epoch. On the other hand, at each iteration of the second-order step, all entries of $\boldsymbol{\sigma}$ are updated, and hence each second-order iteration is an epoch. In terms of the computational cost, an iteration of BCM requires $\mathcal{O}(nr)$ operations and consequently an epoch of BCM requires $\mathcal{O}(n^2 r)$ operations, whereas the second-order direction of update is typically found approximately via a few iterations of the power method or the Lanczos method (see Theorem 7 for a more rigorous treatment of this statement), which require $\mathcal{O}(n^2 r)$ operations. Therefore, an epoch of Algorithm 2 typically has a computational complexity of $\mathcal{O}(n^2 r)$. Furthermore, by Theorem 6, we observe that in at most $\mathcal{O}(n\|\boldsymbol{A}\|_1^2/\varepsilon^2)$ epochs, Algorithm 2 returns a solution that achieves the optimal approximation ratio up to an accuracy of $\mathcal{O}(n\varepsilon)$. In particular, picking $\varepsilon = 2\,\mathrm{SDP}(\boldsymbol{A})/(n(r-1))$, we obtain the following corollary.

Corollary 4 *Consider the setup of Theorem 6 and set* $\varepsilon = 2\,\mathrm{SDP}(\boldsymbol{A})/(n(r-1))$. *Then, as soon as*

$$K = \left\lceil \frac{675n^3(r-1)^2\|\boldsymbol{A}\|_1^2}{4(\mathrm{SDP}(\boldsymbol{A}))^2} \right\rceil, \tag{46}$$

Algorithm 2 is guaranteed to return a solution $\boldsymbol{\sigma}^K$ *that satisfies*

$$f(\boldsymbol{\sigma}^K) \geq \left(1 - \frac{2}{r-1}\right)\mathrm{SDP}(\boldsymbol{A}).$$

Remark 2 In order to understand the total running time of BCM2, consider the following example. Let $\boldsymbol{A}$ be the adjacency matrix of a random Erdos-Rényi graph on n nodes and $\lfloor cn \rfloor$ edges. The size of the maximum cut in this graph normalized by the number of nodes can be bounded between $[c/2 + 0.4\sqrt{c}, c/2 + 0.6\sqrt{c}]$ with high probability as n increases, for all sufficiently large c [20]. Since the maximum value of (CVX) is within 0.878 of the maximum cut [22], we can then conclude that $\mathrm{SDP}(\boldsymbol{A})/n = \mathcal{O}(c)$ with high probability. We can also observe that for this graph, the degree of a node approximately follows a Poisson distribution with mean $2c$, which can be approximated by a normal distribution with mean $2c$ and variance $\sqrt{2c}$, for large c [20]. Then, we have $\|\boldsymbol{A}\|_1 = \mathcal{O}(c \log n)$ with high probability. Therefore, for this problem, Corollary 4 states that in $\widetilde{\mathcal{O}}(nr^2)$ iterations (where tilde is used to hide the logarithmic dependences), Algorithm 2 returns a $\mathcal{O}(1/r)$-optimal solution with high probability. Per iteration computational cost of the algorithm is $\mathcal{O}(nrc)$, which results in a total running time of $\widetilde{\mathcal{O}}(n^2 r^3 c)$. In comparison, Klein-Lu method (see [27, Lemma 4]) requires $\widetilde{\mathcal{O}}(n^2 r^3 c)$ running time and the matrix multiplicative weights method (see [5, Theorem 3]) requires $\widetilde{\mathcal{O}}(n^2 r^{3.5}/c)$ running time to return a $1/r$-optimal solution.

Remark 3 It has been shown in [37, Theorem 3.1] and [17, Theorem 3.5] that an exactly feasible approximately second-order stationary point to (Non-CVX) is also approximately optimal for (CVX). Our BCM2 algorithm returns such a solution and in light of these results, we can conclude that it finds a high-quality solution to (CVX)

with high probability whenever $r \geq \sqrt{2n}$. See Fig. 4 for an empirical validation of this result.

In the description of Algorithm 2 (see line 9), we assumed that we have access to a vector in the tangent space of the current iterate, which satisfies certain second-order conditions. In Algorithm 3, we describe an efficient subroutine to find this desired tangent vector based on the Lanczos method. In particular, the Lanczos method returns a tridiagonal real symmetric matrix whose diagonal entries are $\{\alpha_\ell\}_{\ell \geq 1}$ and off-diagonal entries are $\{\beta_\ell\}_{\ell \geq 2}$, where ℓ denotes the iteration counter in Algorithm 3. The entire spectrum of such a symmetric tridiagonal matrix can be efficiently computed in almost linear time in the dimension of the matrix [18]. Consequently, letting y denote the leading eigenvector of this tridiagonal matrix, we can construct the desired tangent vector in Algorithm 2 as $\boldsymbol{u}^k = \sum_{\ell \geq 1} y_\ell \boldsymbol{u}_\ell$. It is well-known that after $n(r-1)$ iterations, the Lanczos method constructs the leading eigenvector exactly (since order-$n(r-1)$ Krylov subspace spans the entire tangent space). Furthermore, it is also possible to analyze the performance of the Lanczos method with early termination [28]. Building on these ideas, we characterize the quality of the solution returned by Algorithms 2+3 in the following theorem, whose proof can be found in "Appendix C".

Theorem 7 *Suppose in Algorithm 3, we initialize $\boldsymbol{u}_1$ uniformly at random over $T_{\boldsymbol{\sigma}} \mathcal{M}_r$. Let*

$$\ell^* = \left\lceil \left(\frac{1}{2} + 2\sqrt{\frac{\|\boldsymbol{A}\|_1}{\varepsilon}} \right) \log \left(\frac{\left\lceil \frac{675 n \|\boldsymbol{A}\|_1^2}{\varepsilon^2} \right\rceil 1.648 \sqrt{n(r-1)}}{\delta} \right) \right\rceil,$$

and consider that Algorithm 3 is run for $\min(\ell^*, n(r-1))$ iterations at each call from Algorithm 2. Then, after K iterations (defined as in (46)), Algorithm 2 returns a solution σ^K that satisfies

$$f(\boldsymbol{\sigma}^K) \geq \left(1 - \frac{1}{r-1}\right) \mathrm{SDP}(\boldsymbol{A}) - \frac{n}{2}\varepsilon,$$

with probability at least $1 - \delta$.

Algorithm 3 Lanczos Method

1: Given $\boldsymbol{\sigma}$, define $H[\boldsymbol{u}] = \mathrm{Hess} f(\boldsymbol{\sigma})[\boldsymbol{u}] + 4\|A\|_1 \boldsymbol{u}$. Initialize $\boldsymbol{u}_1 \in T_{\boldsymbol{\sigma}} \mathcal{M}_r$ such that $\|\boldsymbol{u}_1\|_{\mathrm{F}} = 1$. Let $\alpha_1 = \langle \boldsymbol{u}_1, H[\boldsymbol{u}_1] \rangle$ and $\boldsymbol{r}_1 = H[\boldsymbol{u}_1] - \alpha_1 \boldsymbol{u}_1$.
2: **for** $\ell \geq 2$ **do**
3: $\quad \beta_\ell = \|\boldsymbol{r}_{\ell-1}\|_{\mathrm{F}}$
4: $\quad \boldsymbol{u}_\ell = \boldsymbol{r}_{\ell-1}/\beta_\ell$ (If $\beta_\ell = 0$, pick $\boldsymbol{u}_\ell \perp \mathrm{span}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\ell-1})$ arbitrarily)
5: $\quad \alpha_\ell = \langle \boldsymbol{u}_\ell, H[\boldsymbol{u}_\ell] \rangle$
6: $\quad \boldsymbol{r}_\ell = H[\boldsymbol{u}_\ell] - \alpha_\ell \boldsymbol{u}_\ell - \beta_\ell \boldsymbol{u}_{\ell-1}$
7: **end for**

5 Numerical experiments

In this section, we evaluate the empirical performance of the BCM algorithm. In what follows, n is the dimension of the cost matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, and r refers to the rank of factorization. All algorithms are implemented on Matlab and the experiments are run on a computer with 2.9 GHz processor and 16 GB memory. RGD and RTR algorithms are implemented using the Manopt package [10] with the default options and the algorithms are terminated when the maximum allowed time is reached.

In all experiments, the cost matrix is generated as $\boldsymbol{A} = (\boldsymbol{G} + \boldsymbol{G}^\top)/n$, where $G_{ij} \sim N(0, 1)$ for all $i \neq j$, and $G_{ii} = 0$ for all $i \in [n]$. All experiments are based on 50 Monte Carlo runs over the initialization. For each run, the initial iterate $\boldsymbol{\sigma}^0 \in \mathbb{R}^{n\times r}$ is the same for all algorithms and each row of $\boldsymbol{\sigma}^0$ is generated uniformly at random on $\mathbb{S}^{r-1}$.

First, we compare various coordinate selection schemes for BCM (see Algorithm 1). We compare cyclic order $i = (1, 2, \ldots, n)$, uniform random selection, random permutation order (i follows a cyclic order of a uniformly random permutation of $[n]$), greedy coordinate selection, and selection by importance sampling. Figure 1 summarizes the results of our experiments on $n \in \{200, 1000\}$ with $r = \lceil\sqrt{2n}\rceil$. We observe that greedy coordinate selection achieves higher function value after running each algorithm the same number of iterations; yet, due to its high per-iteration cost, cyclic, uniformly random, and random permutation selection rules perform better in terms of overall runtime complexity. Furthermore, randomized rules that do not cycle through all coordinates achieve lower function values after running each algorithm the same number of iterations. This phenomenon is observed for a number of numerical examples in different papers and unfortunately we do not have a good theoretical understanding about this behavior except for a few preliminary results [23,24,29]. It would be an interesting future direction to theoretically understand the slower convergence of randomized coordinate selection rules in practice.

Next, we evaluate the performance of these algorithms for $n \in \{200, 1000, 5000\}$ with $r = 2$ and $r = \sqrt{2n}$. In Fig. 2, empirical results illustrate the fast convergence of BCM and BCM2 (see Algorithm 2) compared to RGD and RTR, for both $r = \lceil\sqrt{2n}\rceil$ and $r = 2$. The numerical results indicate our algorithms return a high-quality solution much faster than RGD and RTR regardless of the rank of the factorization is larger or smaller than the Barvinok-Pataki bound.

We next compare the final performance of different methods after convergence. That is, we run all algorithms sufficiently enough until their function value stabilize, and compare the final value obtained. In Fig. 3, we clearly observe that the final function values obtained through BCM2 (Algorithm 2 with Lanczos method) and RTR are larger than those obtained by other algorithms. We also observe that even when the problem size is large (e.g., $n = 20,000$), BCM returns a desirable solution within $\sim$20 seconds, whereas it takes approximately a minute for RTR to return such a solution.

Finally, we consider a random SDP with a planted solution. In particular, we consider a matrix $\boldsymbol{X} \succeq 0$ such that $\text{rank}(\boldsymbol{X}) = r$ and $\boldsymbol{X} \in \mathbb{S}^n$ where $n = \frac{r(r+1)}{2}$. We then generate a MaxCut SDP for which $\boldsymbol{X}$ is an optimal solution, i.e., we find a cost matrix $\boldsymbol{A}$ in the normal cone of $\boldsymbol{X}$ (this requires solving an auxiliary SDP). For each $r \in \{4, 7, 10\}$, we generate 100 random MaxCut SDPs as described above. We perform

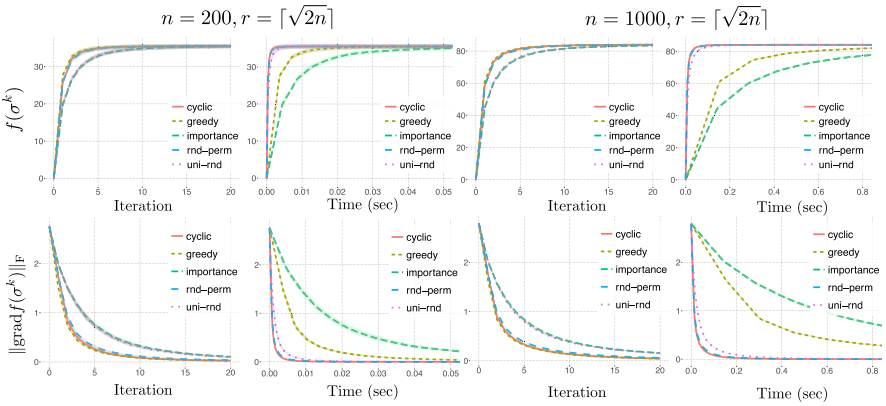

Fig. 1 Comparisons of different randomization schemes for $n \in \{200, 1000\}$ with $r = \lceil\sqrt{2n}\rceil$

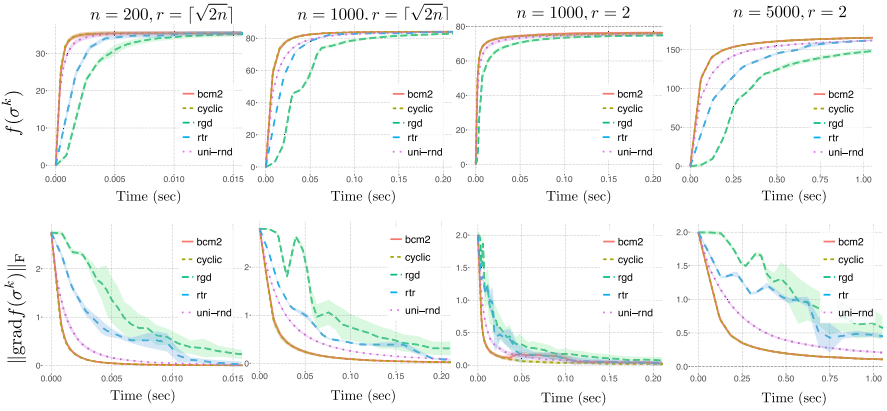

Fig. 2 Performance of BCM and BCM2 (Algorithms 1 and 2) compared to other methods. Here, RTR and RGD refer to Riemannian Trust Region and Riemannian Gradient Descent, respectively

a Burer–Monteiro factorization to these SDPs for a range of ranks in $[r-4, r+4]$. We solve the resulting non-convex problem using our BCM2 algorithm. Figure 4 shows the percentage of experiments solved correctly for each value of r. We consider a trial correct if the solution returned by BCM2 is sufficiently close to the maximizer of the SDP. Figure 4 shows that there is a sharp phase transition at the Barvinok-Pataki bound. Above this bound, the solutions returned by our BCM2 algorithm is approximately optimal to (CVX) with high probability.

6 Conclusion

In this paper, we studied the Burer–Monteiro approach to solve large-scale SDPs. We considered to solve this non-convex problem using the block-coordinate maximization

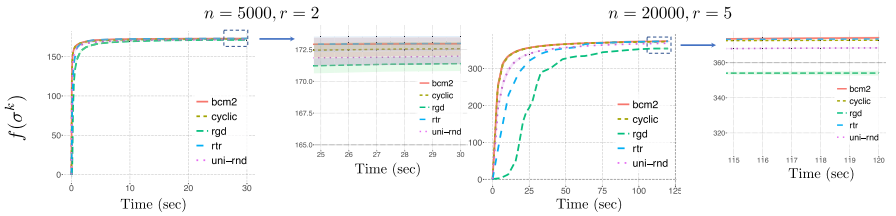

Fig. 3 Comparing the final performance of different methods after (near) convergence

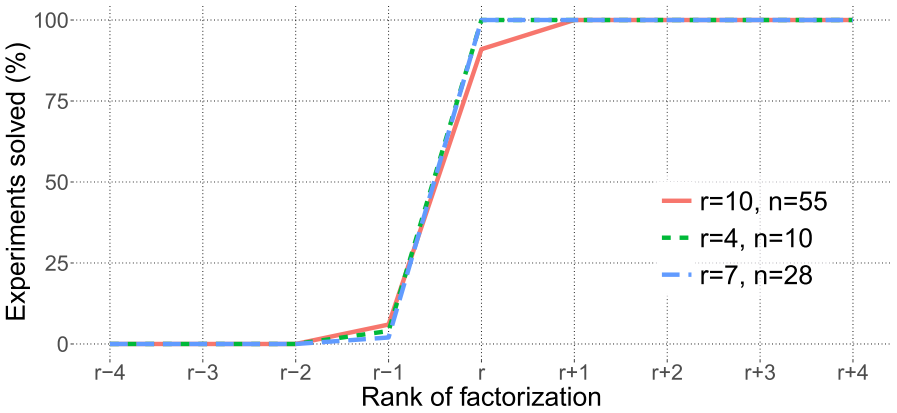

Fig. 4 Phase transition in recovering the optimal solution of (CVX) by an approximately second-order stationary solution of (Non-CVX)

algorithm that is extremely simple to implement. We proved that for various coordinate selection rules, BCM attains a global sublinear convergence rate of $\mathcal{O}(1/\epsilon)$ to guarantee $\mathbb{E}\|\text{grad}\, f(\boldsymbol{\sigma}^k)\|_{\text{F}}^2 \leq \epsilon$. We also showed the linear convergence of BCM around a local maximum that satisfy the quadratic decay condition. We proved that the quadratic decay condition generically holds for all local maxima provided that $r \geq \sqrt{2n}$. These are the first precise rate estimates for the non-convex Burer–Monteiro approach in the literature to the best of our knowledge. We then introduced a new algorithm called BCM2 based on BCM and Lanczos methods. We showed that BCM2 is guaranteed to return a solution that provides $1 - \mathcal{O}(1/r)$ approximation to the SDP without any assumptions on the cost matrix $\boldsymbol{A}$, where the r-dependence of this approximation is optimal under the unique games conjecture. We also presented numerical results that verify our theoretical findings and show that BCM is faster than the state-of-the-art methods. Even though in this paper, we only considered SDPs with diagonal constraints, it would be of interest to study the block-coordinate maximization approach in more generic problems.

A Proof of Corollary 1

Similar to the proof of Theorem 1, from Proposition 1, we have

$$\begin{aligned} f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) &= 2\left(\|g_{i_k}^k\| - \langle \sigma_{i_k}^k, g_{i_k}^k \rangle\right), \\ &= \frac{2\|g_{i_k}^k\|\left(\|g_{i_k}^k\| - \langle \sigma_{i_k}^k, g_{i_k}^k \rangle\right)}{\|g_{i_k}^k\|}, \\ &\geq \frac{\|g_{i_k}^k\|^2 - \langle \sigma_{i_k}^k, g_{i_k}^k \rangle^2}{\|g_{i_k}^k\|}, \end{aligned} \tag{47}$$

where the inequality follows since $\|g_{i_k}^k\| \geq \langle \sigma_{i_k}^k, g_{i_k}^k \rangle$, for all $\sigma_{i_k}^k \in \mathbb{R}^{n\times r}$. Letting $\mathbb{E}_k$ denote the expectation over i_k given σ^k, we have

$$\mathbb{E}_k f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \sum_{i=1}^{n} p_i \frac{\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k \rangle^2}{\|g_i^k\|}.$$

In particular, when $p_i = \frac{1}{n}$, for all $i \in [n]$ (i.e., for uniform sampling case), we have

$$\mathbb{E}_k f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \frac{1}{n\|\boldsymbol{A}\|_1} \sum_{i=1}^{n} \left(\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k \rangle^2\right),$$

since $\|g_i^k\| \leq \|\boldsymbol{A}\|_1$, for all $i \in [n]$ by (11). Therefore, we have

$$\mathbb{E}_k f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \frac{\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2}{2n\|\boldsymbol{A}\|_1}. \tag{48}$$

On the other hand, when $p_i = \frac{\|g_i^k\|}{\sum_{j=1}^n \|g_j^k\|}$ (i.e., for importance sampling case), we have

$$\mathbb{E}_k f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \frac{\sum_{i=1}^n \|g_i^k\|^2 - \langle \sigma_i^k, g_i^k \rangle^2}{\sum_{j=1}^n \|g_j^k\|} = \frac{\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2}{2\sum_{j=1}^n \|g_j^k\|}.$$

Letting $\|\boldsymbol{A}\|_{1,1} = \sum_{i,j=1}^n |\boldsymbol{A}_{ij}|$ denote the $L_{1,1}$ norm of matrix $\boldsymbol{A}$, we observe that $\sum_{j=1}^n \|g_j^k\| \leq \|\boldsymbol{A}\|_{1,1}$, which in the above inequality yields

$$\mathbb{E}_k f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k) \geq \frac{\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2}{2\|\boldsymbol{A}\|_{1,1}}. \tag{49}$$

In order to prove (13), which corresponds to uniform sampling case, we assume the contrary that $\mathbb{E}\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 > \epsilon$, for all $k \in [K-1]$. Then, using the boundedness

of f, we get

$$f^* - f(\boldsymbol{\sigma}^0) \geq \mathbb{E} f(\boldsymbol{\sigma}^K) - f(\boldsymbol{\sigma}^0) = \sum_{k=0}^{K-1} \mathbb{E}\Big[f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k)\Big]$$
$$= \sum_{k=0}^{K-1} \mathbb{E}\Big[\mathbb{E}_k f(\boldsymbol{\sigma}^{k+1}) - f(\boldsymbol{\sigma}^k)\Big].$$

Using the expected functional ascent of BCM in (48) above, we get

$$f^* - f(\boldsymbol{\sigma}^0) \geq \sum_{k=0}^{K-1} \frac{\mathbb{E}\|\mathrm{grad}\, f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2}{2n\|\boldsymbol{A}\|_1} > \frac{K\epsilon}{2n\|\boldsymbol{A}\|_1}, \tag{50}$$

where the last inequality follows by the assumption. Then, by contradiction, the algorithm returns a solution with $\mathbb{E}\|\mathrm{grad}\, f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 \leq \epsilon$, for some $k \in [K-1]$, provided that

$$K \geq \frac{2n\|\boldsymbol{A}\|_1 (f^* - f(\boldsymbol{\sigma}^0))}{\epsilon}.$$

The proof of (14), which corresponds to importance sampling case, can be obtained by using (49) (instead of (48)) in (50), and hence is omitted.

B Rest of the Proof of Theorem 2

In order to quantify how close $\boldsymbol{\sigma}^0$ and $\boldsymbol{\sigma}$ should be so that this convergence rate holds, we need to derive explicit bounds on the higher order terms in (21) and (23), which we do in the following. The Taylor expansion of $\boldsymbol{\sigma}^k$ around $\boldsymbol{\sigma}$ yields

$$\sigma_i^k = \sigma_i \cos(\|u_i\| t) + \frac{u_i}{\|u_i\|} \sin(\|u_i\| t),$$
$$= \sigma_i \left[\sum_{\ell=0}^{\infty} \frac{(-1)^\ell}{(2\ell)!} (\|u_i\| t)^{2\ell}\right] + \frac{u_i}{\|u_i\|} \left[\sum_{\ell=0}^{\infty} \frac{(-1)^\ell}{(2\ell+1)!} (\|u_i\| t)^{2\ell+1}\right].$$

Using this expansion, we can compute $f(\boldsymbol{\sigma}^k) = \sum_{i,j=1}^n A_{ij} \langle \sigma_i^k, \sigma_j^k \rangle$. The first three terms in the expansion are already given in (22) as follows

$$f(\boldsymbol{\sigma}^k) = f(\boldsymbol{\sigma}) + t^2 \sum_{i=1}^n \Big(\langle u_i, v_i \rangle - \|u_i\|^2 \|g_i\|\Big) + \beta_f, \tag{51}$$

where β_f represents the higher order terms. In order to find an upper bound on $|\beta_f|$, we use the Cauchy-Schwarz inequality in the higher order terms in the expansion of

$f(\boldsymbol{\sigma}^k)$, which yields the following bound

$$|\beta_f| \leq \sum_{i,j=1}^{n} |A_{ij}| \left(\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} (\|u_i\| + \|u_j\|)^\ell \right).$$

As $\|\boldsymbol{u}\|_{\mathrm{F}} = 1$, we have $\|u_i\| \leq 1$ for all $i \in [n]$, which implies

$$|\beta_f| \leq \sum_{i,j=1}^{n} |A_{ij}| \left(\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} 2^\ell \right),$$

where we note that t denotes the geodesic distance between $\boldsymbol{\sigma}^k$ and $[\bar{\boldsymbol{\sigma}}]$ as highlighted in (19). Assuming that $t \leq 1$, we obtain the following upper bound

$$|\beta_f| \leq t^3 n \|\boldsymbol{A}\|_1 \left(\sum_{\ell=3}^{\infty} \frac{2^\ell}{\ell!} \right).$$

Using the inequality $\sum_{\ell=3}^{\infty} \frac{2^\ell}{\ell!} = e^2 - 5 \leq 5/2$ above, we get

$$|\beta_f| \leq \frac{5n\|\boldsymbol{A}\|_1 t^3}{2}.$$

Plugging this value back in (51), we obtain

$$f(\boldsymbol{\sigma}^k) \leq f(\boldsymbol{\sigma}) + t^2 \sum_{i=1}^{n} \Big(\langle u_i, v_i \rangle - \|u_i\|^2 \|g_i\| \Big) + \frac{5n\|\boldsymbol{A}\|_1 t^3}{2}. \tag{52}$$

Considering the same expansion for $\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 = 2\sum_{i=1}^{n} (\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k \rangle^2)$, we get the following (see (21)):

$$\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 = 2t^2 \sum_{i=1}^{n} \left(\|u_i\| \|g_i\| - \langle \frac{u_i}{\|u_i\|}, v_i \rangle \right)^2 + \beta_g, \tag{53}$$

where β_g represents the higher order terms. Upper bounding each higher order terms using the Cauchy-Schwarz inequality as follows, we obtain

$$\begin{aligned} |\beta_g| \leq 2 \sum_{i=1}^{n} \Bigg[& \sum_{j,m=1}^{n} |A_{ij}||A_{im}| \left(\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} (\|u_j\| + \|u_m\|)^\ell \right) \\ & + \sum_{j,m=1}^{n} |A_{ij}||A_{im}| \left(\sum_{\substack{\ell,s=0 \\ \ell+s\geq 3}}^{\infty} \frac{t^{\ell+s}}{\ell! s!} (\|u_i\| + \|u_j\|)^{\ell+s} \right) \Bigg]. \end{aligned}$$

Using the fact that $\|u_i\| \leq 1$ for all $i \in [n]$, we get the following upper bound

$$|\beta_g| \leq 2 \sum_{i=1}^{n} \left[\sum_{j,m=1}^{n} |A_{ij}||A_{im}| \left(\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} 2^\ell \right) + \sum_{j,m=1}^{n} |A_{ij}||A_{im}| \left(\sum_{\substack{\ell,s=0 \\ \ell+s\geq 3}}^{\infty} \frac{t^{\ell+s}}{\ell! s!} 2^{\ell+s} \right) \right].$$

Using the upper bound $\sum_{j,m=1}^{n} |A_{ij}||A_{im}| \leq \|\boldsymbol{A}\|_1^2$ above, we obtain

$$|\beta_g| \leq 2\|\boldsymbol{A}\|_1^2 \sum_{i=1}^{n} \left[\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} 2^\ell + \sum_{\substack{\ell,s=0 \\ \ell+s\geq 3}}^{\infty} \frac{t^{\ell+s}}{\ell! s!} 2^{\ell+s} \right].$$

Introducing a change of variables in the last sum, we get

$$\begin{aligned} |\beta_g| &\leq 2\|\boldsymbol{A}\|_1^2 \sum_{i=1}^{n} \left[\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} 2^\ell + \sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} 2^\ell \left(\sum_{s=0}^{\ell} \frac{\ell!}{s!(\ell-s)!} \right) \right], \\ &= 2\|\boldsymbol{A}\|_1^2 \sum_{i=1}^{n} \left[\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} \left(2^\ell + 4^\ell \right) \right]. \end{aligned}$$

Assuming that $t \leq 1$, we obtain the following upper bound

$$|\beta_g| \leq 2\|\boldsymbol{A}\|_1^2 t^3 \sum_{i=1}^{n} \left[\sum_{\ell=3}^{\infty} \frac{1}{\ell!} \left(2^\ell + 4^\ell \right) \right].$$

Using the inequality $\sum_{\ell=3}^{\infty} \frac{2^\ell + 4^\ell}{\ell!} = e^2 + e^4 - 18 \leq 44$ above, we get

$$|\beta_g| \leq 88n\|\boldsymbol{A}\|_1^2 t^3.$$

Plugging this value back in (53), we obtain

$$\|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 \geq 2t^2 \sum_{i=1}^{n} \left(\|u_i\| \|g_i\| - \langle \frac{u_i}{\|u_i\|}, v_i \rangle \right)^2 - 88n\|\boldsymbol{A}\|_1^2 t^3. \tag{54}$$

Using the same bounding technique as in (24), we get

$$\begin{aligned} \|\mathrm{grad} f(\boldsymbol{\sigma}^k)\|_{\mathrm{F}}^2 &\geq \frac{\mu}{n} \left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k) - \frac{5n\|\boldsymbol{A}\|_1 t^3}{2} \right) - 88n\|\boldsymbol{A}\|_1^2 t^3, \\ &= \frac{\mu}{n} \left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k) \right) - t^3 \|\boldsymbol{A}\|_1 \left(3\mu + 88n\|\boldsymbol{A}\|_1 \right). \end{aligned}$$

Therefore, in order for (25) to hold, we need

$$t^3\|\boldsymbol{A}\|_1\,(3\mu + 88n\|\boldsymbol{A}\|_1) \leq \frac{\mu}{2n}\left(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k)\right),$$

which can be equivalently rewritten as follows

$$t^3 \leq \frac{\mu(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k))}{2n\|\boldsymbol{A}\|_1\,(3\mu + 88n\|\boldsymbol{A}\|_1)}.$$

As $f(\boldsymbol{\sigma}^k)$ is a monotonically non-decreasing sequence, then as soon as $\boldsymbol{\sigma}^0$ is sufficiently close to $[\bar{\boldsymbol{\sigma}}]$ in the sense that

$$\mathrm{dist}(\boldsymbol{\sigma}^0, [\bar{\boldsymbol{\sigma}}]) \leq \left(\frac{\mu(f(\bar{\boldsymbol{\sigma}}) - f(\boldsymbol{\sigma}^k))}{2n\|\boldsymbol{A}\|_1\,(3\mu + 88n\|\boldsymbol{A}\|_1)}\right)^{1/3},$$

then the linear convergence rate presented in (26) holds.

C Proof of Theorem 7

Before presenting the proof of Theorem 7, we first introduce the following theorem that characterizes the convergence rate of the Lanczos method with random initialization.

Theorem 8 [28, Theorem 4.2] *Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ be a positive semidefinite matrix, $b \in \mathbb{R}^n$ be an arbitrary vector and $\lambda_L^\ell(\boldsymbol{A}, b)$ denote the output of the Lanczos algorithm after ℓ iterations when applied to find the leading eigenvalue of $\boldsymbol{A}$ (denoted by $\lambda_1(\boldsymbol{A})$) with initialization b. In particular,*

$$\lambda_L^\ell(\boldsymbol{A}, b) = \max\left\{\frac{\langle x, \boldsymbol{A}x\rangle}{\langle x, x\rangle} : 0 \neq x \in \mathrm{span}(b, \ldots, \boldsymbol{A}^{\ell-1}b)\right\}.$$

Assume that b is uniformly distributed over the set $\{b \in \mathbb{R}^n : \|b\| = 1\}$ and let $\epsilon \in [0, 1)$. Then, the probability that the Lanczos algorithm does not return an ϵ-approximation to the leading eigenvalue of $\boldsymbol{A}$ exponentially decreases as follows

$$\mathbb{P}\left(\lambda_L^\ell(\boldsymbol{A}, b) < (1-\epsilon)\lambda_1(\boldsymbol{A})\right)\begin{cases}\leq 1.648\sqrt{n}e^{-\sqrt{\epsilon}(2\ell-1)}, & \text{if } 0 < \ell < n(r-1),\\ = 0, & \text{if } \ell \geq n(r-1).\end{cases}$$

Using this result, Theorem 7 is proven as follows. Since the tangent space $T_{\boldsymbol{\sigma}}\mathcal{M}_r$ has dimension $n(r-1)$, then we can define a symmetric matrix (where we drop the notational dependency on $\boldsymbol{\sigma}$ for simplicity) $\boldsymbol{H} \in \mathbb{R}^{n(r-1)\times n(r-1)}$ that represents the linear operator $\mathrm{Hess}\, f(\boldsymbol{\sigma})$ in the basis $\{\boldsymbol{u}^1, \ldots, \boldsymbol{u}^{n(r-1)}\}$ such that $\mathrm{span}(\boldsymbol{u}^1, \ldots, \boldsymbol{u}^{n(r-1)}) = T_{\boldsymbol{\sigma}}\mathcal{M}_r$. In particular, letting $H_{ij} = \langle \boldsymbol{u}^i, \mathrm{Hess}\, f(\boldsymbol{\sigma})[\boldsymbol{u}^j]\rangle$ yields the desired matrix $\boldsymbol{H}$ and the Lanczos algorithm is run to find the leading eigenvalue of this matrix. Here, it is important to note that $\boldsymbol{H}$ is not a psd matrix, so it is required to shift $\boldsymbol{H}$ with a

large enough multiple of the identity matrix so that the resulting matrix is guaranteed to be positive semidefinite. In particular, by inspecting the definition of $\mathrm{Hess}\, f(\boldsymbol{\sigma})$ in (5), it is easy to observe that $\|\mathrm{Hess}\, f(\boldsymbol{\sigma})\|_{\mathrm{op}} \leq 4\|\boldsymbol{A}\|_1$. Therefore, it is sufficient to run the Lanczos algorithm to find the leading eigenvalue of $\widetilde{\boldsymbol{H}} = \boldsymbol{H} + 4\|\boldsymbol{A}\|_1 \boldsymbol{I}$, where $\boldsymbol{I}$ denotes the appropriate sized identity matrix. On the other hand, we initialize the Lanczos algorithm with a random vector $\boldsymbol{u}$ of unit norm (i.e., $\|\boldsymbol{u}\|_{\mathrm{F}} = 1$) in the tangent space $T_{\boldsymbol{\sigma}}\mathcal{M}_r$. Notice that $\boldsymbol{u}$ can equivalently be represented as a vector $b \in \mathbb{R}^{n(r-1)}$ in the basis $\{\boldsymbol{u}^1, \ldots, \boldsymbol{u}^{n(r-1)}\}$ as $\boldsymbol{u} = \sum_{i=1}^{n(r-1)} b_i \boldsymbol{u}^i$ such that $\|b\| = 1$. Then, by Theorem 8, we have

$$\mathbb{P}\left(\lambda_L^{\ell}(\widetilde{\boldsymbol{H}}, b) < (1-\epsilon)\lambda_1(\widetilde{\boldsymbol{H}})\right) \leq 1.648\sqrt{n(r-1)}e^{-\sqrt{\epsilon}(2\ell-1)}.$$

Letting $\lambda_1(\boldsymbol{H})$ denote the leading eigenvalue of $\boldsymbol{H}$, we run the Lanczos algorithm to obtain a vector b^* such that $\|b^*\| = 1$ and $\langle b^*, \boldsymbol{H}b^*\rangle \geq \lambda_1(\boldsymbol{H})/2$. Thus, we want $\mathbb{P}\left(\lambda_L^{\ell}(\widetilde{\boldsymbol{H}}, b) < 4\|\boldsymbol{A}\|_1 + \lambda_1(\boldsymbol{H})/2\right)$ to be small. Setting $\epsilon^* = \frac{\lambda_1(\boldsymbol{H})}{16\|\boldsymbol{A}\|_1}$, we can observe that

$$\begin{aligned}
\left(1-\epsilon^*\right)\lambda_1(\widetilde{\boldsymbol{H}}) &= \left(1 - \frac{\lambda_1(\boldsymbol{H})}{16\|\boldsymbol{A}\|_1}\right)\left(4\|\boldsymbol{A}\|_1 + \lambda_1(\boldsymbol{H})\right), \\
&= 4\|\boldsymbol{A}\|_1 + \frac{3\lambda_1(\boldsymbol{H})}{4} - \frac{(\lambda_1(\boldsymbol{H}))^2}{16\|\boldsymbol{A}\|_1}, \\
&\geq 4\|\boldsymbol{A}\|_1 + \frac{\lambda_1(\boldsymbol{H})}{2},
\end{aligned}$$

where the inequality follows since $\lambda_1(\boldsymbol{H}) \leq 4\|\boldsymbol{A}\|_1$. Consequently, we have

$$\begin{aligned}
&\mathbb{P}\left(\lambda_L^{\ell}(\widetilde{\boldsymbol{H}}, b) < 4\|\boldsymbol{A}\|_1 + \lambda_1(\boldsymbol{H})/2\right) \\
&\quad\leq \mathbb{P}\left(\lambda_L^{\ell}(\widetilde{\boldsymbol{H}}, b) < (1-\epsilon^*)\lambda_1(\widetilde{\boldsymbol{H}})\right) \leq 1.648\sqrt{n(r-1)}e^{-\sqrt{\epsilon^*}(2\ell-1)}.
\end{aligned}$$

By Theorem 6, we know that the Lanczos method is called at most $\lceil 675n\|\boldsymbol{A}\|_1^2/\varepsilon^2 \rceil$ times to search for an ε-approximate concave point and for any non-desired solution we have $\lambda_1(\boldsymbol{H}) \geq \varepsilon$ by the definition of ε-approximate concave point. Then, by using a union bound over all calls to the Lanczos method, we conclude that when the Lanczos method is run for ℓ iterations, we have the following guarantee

$$\begin{aligned}
&\mathbb{P}\,(\text{Algorithm 2+3 fails to return an}\varepsilon\text{-approximate concave point}) \\
&\quad\leq \left\lceil \frac{675n\|\boldsymbol{A}\|_1^2}{\varepsilon^2} \right\rceil 1.648\sqrt{n(r-1)}e^{-\sqrt{\frac{\varepsilon}{16\|\boldsymbol{A}\|_1}}(2\ell-1)}.
\end{aligned}$$

In order to set this probability to some $\delta \in (0, 1)$, we let

$$\ell^* = \left\lceil \left(\frac{1}{2} + 2\sqrt{\frac{\|\boldsymbol{A}\|_1}{\varepsilon}} \right) \log \left(\frac{\left\lceil \frac{675n\|A\|_1^2}{\varepsilon^2} \right\rceil 1.648\sqrt{n(r-1)}}{\delta} \right) \right\rceil$$
$$= \widetilde{\mathcal{O}} \left(\sqrt{\frac{\|\boldsymbol{A}\|_1}{\varepsilon}} \log \left(\frac{n\sqrt{n(r-1)}}{\delta} \right) \right),$$

where tilde is used to hide poly-logarithmic factors in $\|\boldsymbol{A}\|_1/\varepsilon$. Since the Lanczos algorithm is guaranteed to return the leading eigenvalue with probability 1 in at most $n(r-1)$ iterations, then running each Lanczos subroutine for $\min(\ell^*, n(r-1))$ iterations, it is guaranteed that Algorithm 2+3 returns an ε-approximate concave point with probability at least $1 - \delta$.

References

1. Absil, P.-A., Baker, C.G., Gallivan, K.A.: Trust-region methods on Riemannian manifolds. Found. Comput. Math. **7**(3), 303–330 (2007)
2. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2007)
3. Alizadeh, F., Haeberly, J.-P.A., Overton, M.L.: Complementarity and nondegeneracy in semidefinite programming. Math. Program. **77**(1), 111–128 (1997)
4. Anitescu, M.: Degenerate nonlinear programming with a quadratic growth condition. SIAM J. Optim. **10**(4), 1116–1135 (2000)
5. Arora, S., Hazan, E., Kale, S.: Fast algorithms for approximate semidefiniite programming using the multiplicative weights update method. In: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS'05, pp. 339–348 (2005)
6. Bandeira, A.S., Boumal, N., Voroninski, V.: On the low-rank approach for semidefinite programs arising in synchronization and community detection. arXiv:1602.04426 (2016)
7. Barvinok, A.I.: Problems of distance geometry and convex properties of quadratic maps. Discrete Comput. Geom. **13**(2), 189–202 (1995)
8. Bonnans, J.F., Ioffe, A.: Second-order sufficiency and quadratic growth for nonisolated minima. Math. Oper. Res. **20**(4), 801–817 (1995)
9. Boumal, N., Absil, P.-A., Cartis, C.: Global rates of convergence for nonconvex optimization on manifolds. arXiv preprint arXiv:1605.08101 (2016)
10. Boumal, N., Mishra, B., Absil, P.-A., Sepulchre, R.: Manopt, a Matlab toolbox for optimization on manifolds. J. Mach. Learn. Res. **15**, 1455–1459 (2014)
11. Boumal, N., Voroninski, V., Bandeira, A.S.: The non-convex Burer–Monteiro approach works on smooth semidefinite programs. In: Advances in Neural Information Processing Systems, pp. 2757–2765 (2016)
12. Boumal, N., Voroninski, V., Bandeira, A.S.: Deterministic guarantees for Burer–Monteiro factorizations of smooth semidefinite programs. arXiv preprint arXiv:1804.02008 (2018)
13. Briat, C.: Linear Parameter-Varying and Time-Delay Systems. Springer (2014)
14. Briët, J., de Oliveira Filho, F.M., Vallentin, F.: The positive semidefinite grothendieck problem with rank constraint. In: Automata, Languages and Programming, pp. 31–42 (2010)
15. Burer, S., Monteiro, R.D.C.: A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Math. Program. **95**(2), 329–357 (2003)
16. Burer, S., Monteiro, R.D.C.: Local minima and convergence in low-rank semidefinite programming. Math. Program. **103**(3), 427–444 (2005)

17. Cifuentes, D., Moitra, A.: Polynomial time guarantees for the Burer–Monteiro method. arXiv preprint arXiv:1912.01745 (2019)
18. Coakley, E.S., Rokhlin, V.: A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices. Appl. Comput. Harmon. Anal. **34**(3), 379–414 (2013)
19. Erdogdu, M.A., Deshpande, Y., Montanari, A.: Inference in graphical models via semidefinite programming hierarchies. In: Advances in Neural Information Processing Systems, pp. 416–424 (2017)
20. Gamarnik, D., Li, Q.: On the max-cut of sparse random graphs. arXiv preprint arXiv:1411.1698 (2014)
21. Garber, D., Hazan, E.: Approximating semidefinite programs in sublinear time. In: Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11, pp. 1080–1088 (2011)
22. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. ACM **42**(6), 1115–1145 (1995)
23. Gurbuzbalaban, M., Ozdaglar, A., Parrilo, P.A., Vanli, N.D.: When cyclic coordinate descent outperforms randomized coordinate descent. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, and (eds.) Advances in Neural Information Processing Systems, volume 30, pp. 6999–7007. Curran Associates, Inc. (2017)
24. Gurbuzbalaban, M., Ozdaglar, A., Vanli, N.D., Wright, S.J.: Randomness and permutations in coordinate descent methods. Math. Program. **181**, 03 (2018)
25. Javanmard, A., Montanari, A., Ricci-Tersenghi, F.: Phase transitions in semidefinite relaxations. Proc. Natl. Acad. Sci. **113**(16), E2218–E2223 (2016)
26. Journee, M., Bach, F., Absil, P.-A., Sepulchre, R.: Low-rank optimization on the cone of positive semidefinite matrices. SIAM J. Optim. **20**(5), 2327–2351 (2010)
27. Klein, P., Lu, H.-I.: Efficient approximation algorithms for semidefinite programs arising from MAX CUT and COLORING. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC'96, pp. 338–347. ACM, New York, NY, USA (1996)
28. Kuczyński, J., Woźniakowski, H.: Estimating the largest eigenvalues by the power and Lanczos algorithms with a random start. SIAM J. Matrix Anal. Appl. **13**(4), 1094–1122 (1992)
29. Lee, C.-P., Wright, S.J.: Random permutations fix a worst case for cyclic coordinate descent. IMA J. Numer. Anal. **39**, 07 (2016)
30. Lee, J.D., Simchowitz, M., Jordan, M.I., Recht, B.: Gradient descent only converges to minimizers. In: 29th Annual Conference on Learning Theory, vol. 49, pp. 1246–1257. PMLR (2016)
31. Lu, Z., Xiao, L.: Randomized block coordinate non-monotone gradient method for a class of nonlinear programming. Technical Report MSR-TR-2013-66 (2013)
32. Mei, S., Misiakiewicz, T., Montanari, A., Oliveira, R.I.: Solving SDPs for synchronization and MaxCut problems via the Grothendieck inequality. arXiv preprint arXiv:1703.08729 (2017)
33. Montanari, A.: A Grothendieck-type inequality for local maxima. arXiv preprint arXiv:1603.04064 (2016)
34. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. Math. Program. **96**(2), 293–320 (2003)
35. Pataki, G.: On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. Math. Oper. Res. **23**(2), 339–358 (1998)
36. Patrascu, A., Necoara, I.: Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. J. Glob. Optim. **61**, 05 (2013)
37. Pumir, T., Jelassi, S., Boumal, N.: Smoothed analysis of the low-rank approach for smooth semidefinite programs. In: Advances in Neural Information Processing Systems, pp. 2281–2290 (2018)
38. Richtárik, P., Takáč, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math. Program. **144**, 07 (2011)
39. Steurer, D.: Fast SDP algorithms for constraint satisfaction problems. In: Proceedings of the Twenty-First Annual ACM–SIAM Symposium on Discrete Algorithms, pp. 684–697 (2010)
40. Tropp, J.A., Yurtsever, A., Udell, M., Cevher, V.: Practical sketching algorithms for low-rank matrix approximation. SIAM J. Matrix Anal. Appl. **38**(4), 1454–1485 (2017)
41. Tseng, P., Yun, S.: A coordinate gradient descent method for nonsmooth separable minimization. Math. Program. **117**, 387–423 (2009)
42. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Rev. **38**(1), 49–95 (1996)
43. Wang, P.-W., Chang, W.-C., Kolter, J.Z.: The mixing method: coordinate descent for low-rank semidefinite programming. arXiv preprint arXiv:1706.00476 (2017)