A Scalable Frank-Wolfe-Based Algorithm for the Max-Cut SDP

Chi Bach Pham [1] **Wynita Griggs** [1 2] **James Saunderson** [1]

Abstract

We consider the problem of solving large-scale instances of the Max-Cut semidefinite program (SDP), i.e., optimizing a linear function over $n \times n$ positive semidefinite (PSD) matrices with unit diagonal. When the cost matrix is PSD, we show how to exactly reformulate the problem as maximizing a smooth concave function over PSD matrices with unit trace. By applying the Frank-Wolfe method, we obtain a simple algorithm that is compatible with recent sampling-based techniques to solve SDPs using low memory. We demonstrate the practical performance of our method on $10^6 \times 10^6$ instances of the max-cut SDP with costs having up to 5×10^6 non-zero entries. Theoretically, we show that our method solves problems with diagonally dominant costs to relative error ϵ in $O(n\epsilon^{-1})$ calls to a randomized approximate largest eigenvalue subroutine, each of which succeeds with high probability after $O(\log(n)\epsilon^{-1/2})$ matrix-vector multiplications with the cost matrix.

1. Introduction

Semidefinite programs (SDPs) involve optimizing a linear functional over positive semidefinite matrix satisfying linear constraints. SDPs arise naturally in a wide range of machine learning problems, including various clustering formulations, kernel learning (Lanckriet et al., 2004), variational inference (Bach, 2022) and robustness certification for neural networks (Raghunathan et al., 2018). While small and medium-scale SDPs can be solved to high accuracy using interior point methods that exploit second-order information, solving large-scale instances of SDPs remains a challenge. One reason for this is simply because even storing an $n \times n$ dense matrix of decision variables rapidly becomes prohibitive as n grows.

To deal with this basic storage challenge, it is common to consider alternative representations of the positive semidefinite matrix decision variable. Many of these are based on the observation that for $n \times n$ SDPs with $O(d)$ constraints, there is an optimal solution with rank $O(\sqrt{d})$ (Pataki, 1998; Barvinok, 1995). Parameterizing the decision variable as $X = UU^T$ where U is an $n \times r$ matrix (Burer & Monteiro, 2003), allows for a reduction in storage to $O(nr)$. However, the problem is no longer convex in these variables, and spurious second order critical points can occur whenever r is smaller than $\sim \sqrt{2d}$ (Waldspurger & Waters, 2020).

Storage can be further reduced, and convexity preserved, by using sketching-based methods (Yurtsever et al., 2017; 2021). Rather than working with the full decision variable, this approach employs algorithms, based on enhancements of the Frank-Wolfe method, that track a random projection of the decision variable of sufficient dimension to reconstruct a $(1+\zeta)$-optimal best rank r approximation of an ϵ-optimal SDP solution. The working storage cost (i.e., the storage in addition to that required to specify the instance) is $O(d + nr/\zeta)$.

A related approach due to Shinde et al. (2021), seeks algorithms that generate a Gaussian random vector with covariance equal to a (near-optimal, near-feasible) solution of an SDP. Such samples are sufficient to implement many rounding schemes, such as the Goemans-Williamson rounding for Max-Cut (Shinde et al., 2021, Section 4). By applying a sketching method to the samples, a near-optimal best rank r approximation of the SDP solution can also be constructed in low memory (Tropp et al., 2017). The algorithm proposed by Shinde et al. (2021) has working storage $O(n+d)$ and is based on modifying the Frank-Wolfe method to track Gaussian samples with covariance equal to the algorithm's matrix iterates. It is similar in spirit to the basic sketching-based method of Yurtsever et al. (2017).

For general SDPs, both factorization-based methods and the sketching and sampling-based methods typically penalize violation of the linear constraints in the objective, no longer ensuring feasibility. This also tends to cause running times with poor dependence on the desired accuracy ϵ, since the penalty parameter typically depends on ϵ, and in turn affects the smoothness of the objective function.

[1]Department of Electrical and Computer Systems Engineering, Monash University, Australia. [2]Department of Civil Engineering, Monash University, Australia. Correspondence to: James Saunderson <james.saunderson@monash.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

The Max-Cut SDP Seeking to improve upon these low-memory methods, in this paper we focus on the Max-Cut SDP, perhaps the simplest interesting family of SDPs. This problem (see (SDP-P) in Section 3) involves maximizing a linear functional over the set of PSD matrices with diagonal entries equal to one. It arises naturally in SDP relaxations of unconstrained binary quadratic optimization problems, such as the celebrated Goemans-Williamson approximation algorithm for the NP-complete combinatorial optimization problem Max-Cut (Goemans & Williamson, 1995). The Max-Cut SDP also arises, e.g., in convex relaxations of the correlation clustering problem (Charikar & Wirth, 2004), the factor analysis problem in statistics (Della Riccia & Shapiro, 1982), and as a computational primitive in approaches to low-rank matrix optimization problems via max-norm regularization (Srebro & Shraibman, 2005; Jaggi, 2013).

As far as we are aware, the earliest low-memory first-order algorithm for the Max-Cut SDP is due to Klein and Lu (1998), (reducing the memory use of (Klein & Lu, 1996)). For a graph with n vertices and m edges, it achieves relative error ϵ for Laplacian costs in $\tilde{O}(mn/\epsilon^3)$ time using $O(n^{1.5})$ working storage. When specialized to the Max-Cut SDP, the method of (Shinde et al., 2021, Section 4) samples a Gaussian vector with covariance that achieves relative error ϵ for the SDP. The method uses $O(n)$ working storage and $\tilde{O}(n^2/\epsilon^2)$ calls to a largest eigenvalue routine that must be solved to relative error ϵ/n. (Each call costs $\tilde{O}(m(\epsilon/n)^{-\frac{1}{2}})$ if implemented using the Lanczos method—see Section 5.) The time complexity analysis in (Yurtsever et al., 2021) is focused on ϵ-dependence, with $O(\epsilon^{-2})$ calls to a largest eigenvalue routine. However, the method has strong practical performance (see Table 1 in Section 6).

There is a significant literature on theoretical algorithms for SDP in general, and for the Max-Cut SDP in particular. The most prominent methods are based on variations on the matrix multiplicative weights algorithm (Arora et al., 2005; Arora & Kale, 2016). The state-of-the-art for the Max-Cut SDP appears to be (Lee & Padmanabhan, 2020), which achieves $\epsilon \sum_{i,j=1}^{n} |C_{ij}|$ additive error in $\tilde{O}(m\epsilon^{-3.5})$ operations. We are not aware of any practical evaluation of the algorithms proposed in this general line of work.

Among factorization-based methods, one with good practical performance and theoretical guarantees is a block coordinate descent method due to Erdogdu et al. (2022). This applies without assumptions on the cost matrix and achieves ϵ relative error using storage $O(n\epsilon^{-1})$. For diagonally dominant costs, it has theoretical running time $\tilde{O}(nm\epsilon^{-3})$.

1.1. Our Contributions

Our first contribution is to exactly reformulate the Max-Cut SDP as the maximization of a concave function with bounded curvature constant (see Section 2.2) over a compact convex set for which linear optimization can be reduced to a symmetric largest eigenvalue problem. This reformulation is naturally suited to applying the Frank-Wolfe method with the Gaussian-sampling based low-memory modifications of Shinde et al. (2021). Our algorithm is just this lightly modified version of Frank-Wolfe (see Algorithm 2 and Algorithm 3) followed by a non-linear transformation that can be carried out either at the level of the matrix decision variable or the Gaussian samples (see Lemma 4.2).

Our reformulation has two parameters, α and μ, both of which are entry-wise positive vectors. We choose these based on a feasible point z for the dual of the Max-Cut SDP (SDP-D). Carefully doing so ensures that the smooth problem is an exact reformulation, and allows us to reduce the curvature constant of the objective. Our second contribution is to show how to choose these parameters so that the algorithm achieves promising theoretical running time guarantees. If ρ denotes the ratio of the dual cost of z and a lower bound on the primal objective value, the algorithm achieves ϵ relative error in $O(\rho n/\epsilon)$ calls to an approximate largest eigenvalue subroutine (see Theorem 4.10), which needs to be solved to relative error $O(\epsilon/\rho)$. This can be achieved (with high probability) using a randomized (block) Krylov subspace method in $O(\rho^{1/2}\log(n)\epsilon^{-1/2})$ matrix-vector multiplications with the cost matrix. (See Section 5.) The overall running time for a PSD cost matrix with m non-zero entries is $O(mn\log(n)(\epsilon/\rho)^{-3/2})$. For diagonally dominant costs, we can take $\rho = 2$ (see Section 4.3).

We test our method on instances of the Max-Cut SDP with diagonally dominant costs. Our numerical experiments indicate that, at least for these instances, the practical performance of the method is (much) better than the theoretical bounds suggest, with both better dependence on n and ϵ.

Our algorithm is very simple. The novelty comes from the reformulation and understanding its properties. As such, is likely that there is significant scope for further algorithm engineering to improve the method's practical performance. The numerical experiments also suggest that there is scope to improve the theoretical analysis of the algorithm.

2. Preliminaries

2.1. Notation and Terminology

We briefly introduce notation not explicitly defined elsewhere. Let $\mathbb{R}^n$ denote n-dimensional real vectors, $\mathbb{R}^{n\times m}$ denote $n \times m$ real matrices, and $\mathbb{S}^n$ denote $n \times n$ symmetric matrices. If $x, y \in \mathbb{R}^n$ then $\langle x, y\rangle := \sum_{i=1}^{n} x_i y_i$ and $\|x\|_2 := \langle x, x\rangle^{1/2}$. If $X, Y \in \mathbb{R}^{n\times m}$ then $\langle X, Y\rangle := \mathrm{tr}(X^T Y)$ (where X^T denotes the transpose of X) and $\|X\|_F = \langle X, X\rangle^{1/2}$ is the Frobenius norm.

If $\mathcal{A}$ is a linear map between inner product spaces, its adjoint

is $\mathcal{A}^\star$. The map diag : $\mathcal{S}^n \to \mathbb{R}^n$ returns the diagonal entries of an $n \times n$ matrix as a vector. Its adjoint, diag* takes a vector and forms a diagonal matrix with those entries on the diagonal. We use the notation $\mathbf{1}$ for the vector with all entries equal to one, and I for the identity matrix.

For $x \in \mathbb{R}^n$, by $x \geq 0$ (resp., $x > 0$) we mean that x is entry-wise nonnegative (resp., positive). For $x, y \in \mathbb{R}^n$, $x \circ y \in \mathbb{R}^n$ denotes the entry-wise product. For $X \in \mathcal{S}^n$, $X \succeq 0$ indicates that X is positive semidefinite. A symmetric matrix X is *diagonally dominant* if $|X_{ii}| \geq \sum_{j \neq i} |X_{ij}|$ for all i. If $X \in \mathbb{R}^{n \times n}$, the spectral norm, denoted $\|X\|$, is the largest singular value. If $X \succeq 0$ then $\|X\| = \lambda_{\max}(X)$, the largest eigenvalue of X. We also use $\|\cdot\|$ to denote a general norm and $\|x\|_* = \sup_{\|y\| \leq 1} \langle x, y \rangle$ to denote its dual norm.

If $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, ∇f denotes the gradient of f. In the univariate case, f' denotes the derivative of f.

2.2. Frank-Wolfe Algorithm

Let f be a concave, continuously differentiable function, and let $\mathcal{S}$ be a compact convex subset of the domain of f. One way to solve the convex optimization problem

$$f^* := \sup_{x \in \mathcal{S}} f(x). \tag{1}$$

is to use the Frank-Wolfe method (Frank & Wolfe, 1956), described in Algorithm 1. It is particularly suitable when we have access to an efficient *linear maximization oracle (LMO)* that performs linear optimization over $\mathcal{S}$.

Algorithm 1 Frank-Wolfe algorithm

input Initial point $x^{(0)} \in \mathcal{S}$, $\epsilon > \delta > 0$
output $x^{(t)} \in \mathcal{S}$ such that $f^* - f(x^{(t)}) \leq \epsilon + \delta$
1: $q^{(0)} := \texttt{LMO}(\nabla f(x^{(0)}), \mathcal{S}, \delta)$
2: $t := 0$
3: **while** $\langle \nabla f(x^{(t)}), q^{(t)} - x^{(t)} \rangle > \epsilon$ **do**
4: $\quad \gamma^{(t)} := \frac{2}{t+2}$
5: $\quad x^{(t+1)} := x^{(t)} + \gamma^{(t)}(q^{(t)} - x^{(t)})$
6: $\quad q^{(t+1)} := \texttt{LMO}(\nabla f(x^{(t+1)}), \mathcal{S}, \delta)$
7: $\quad t := t + 1$
8: **end while**
9: **function** $q = \texttt{LMO}(y, \mathcal{S}, \delta')$
10: $\quad$ Find $q \in \mathcal{S}$ such that $\langle q, y \rangle \geq \sup_{\hat{q} \in \mathcal{S}} \langle \hat{q}, y \rangle - \delta'$
11: **end function**

Stopping criterion If x^* is optimal for (1), and the stopping criterion in line 3 holds, we have that

$$\begin{aligned} f(x^*) - f(x^{(t)}) &\leq \left\langle \nabla f(x^{(t)}), x^* - x^{(t)} \right\rangle \\ &\leq \sup_{\hat{q} \in \mathcal{S}} \left\langle \nabla f(x^{(t)}), \hat{q} - x^{(t)} \right\rangle =: \textsc{gap}(x^{(t)}) \\ &\leq \left\langle \nabla f(x^{(t)}), q^{(t)} - x^{(t)} \right\rangle + \delta \leq \epsilon + \delta. \end{aligned}$$

Curvature constant The convergence analysis of Frank-Wolfe-type algorithms usually relies on the *curvature constant* (see, e.g., (Jaggi, 2013)), defined as

$$\mathcal{M}(f|\mathcal{S}) := \sup_{\substack{x, s \in \mathcal{S}, \\ \gamma \in [0,1]}} \tfrac{2}{\gamma^2} B_f((1-\gamma)x + \gamma s \,||\, x) \tag{2}$$

where, for a concave function f, $B_f(y||x) := [g(x) + \langle \nabla g(x), y - x \rangle] - g(y)$ is the *Bregman divergence*. The curvature constant for f is bounded if ∇f is Lipschitz continuous with respect to some norm.

Lemma 2.1 (Lemma 7, Appendix D, (Jaggi, 2013))**.** *Let $\|\cdot\|$ and $\|\cdot\|_*$ be a dual pair of norms. Let f be a concave and differentiable function and $\mathcal{S}$ a compact subset of the domain of f. Suppose that $\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|$ for all $x, y \in \mathcal{S}$. Then*

$$\mathcal{M}(f|\mathcal{S}) \leq L \operatorname{diam}_{\|\cdot\|}(\mathcal{S})^2$$

where $\operatorname{diam}_{\|\cdot\|}(\mathcal{S}) := \sup_{x,y \in \mathcal{S}} \|x - y\|$.

Convergence analysis The following is a standard convergence result for the Frank-Wolfe algorithm. Since the duality gap $\textsc{gap}(x^{(t)})$ may not be monotonically decreasing, we consider the smallest duality gap up to iteration t.

Lemma 2.2 (Bound 3.1, Theorem 5.1, Theorem 5.2, (Freund & Grigas, 2016))**.** *The iterates $x^{(t)}$, for $t \geq 1$, of Algorithm 1 applied to (1) satisfy*

$$f^* - f(x^{(t)}) \leq \frac{2\mathcal{M}(f|\mathcal{S})}{t+3} + \delta \quad \text{and}$$
$$\min_{\tau \leq t} \textsc{gap}(x^{(\tau)}) \leq \frac{4.5\mathcal{M}(f|\mathcal{S})}{t} + \delta.$$

In particular, if $\delta < \epsilon$, the algorithm terminates after at most $4.5\mathcal{M}(f|\mathcal{S})/(\epsilon - \delta)$ iterations.

3. Formulations of the Max-Cut SDP

3.1. Standard Formulation of the Max-Cut SDP

In this section we state the Max-Cut SDP and record some of its basic properties. Omitted proofs for this section are in Appendix A. The Max-Cut SDP is

$$F_C^* := \sup_{X \in \mathbb{S}^n} \langle C, X \rangle \quad \text{s.t.} \quad \begin{cases} X \succeq 0 \\ \operatorname{diag}(X) = \mathbf{1}. \end{cases} \tag{SDP-P}$$

The corresponding dual SDP is

$$G_C^* := \inf_{z \in \mathbb{R}^n} \langle \mathbf{1}, z \rangle \quad \text{s.t.} \quad \operatorname{diag}^\star(z) \succeq C \tag{SDP-D}$$

Since $X = I$ is strictly feasible for (SDP-P), Slater's condition holds and so this pair of SDPs satisfies strong duality (Vandenberghe & Boyd, 1996).

Lemma 3.1. *If C is symmetric then $F_C^* = G_C^*$.*

Since $X = I$ is primal feasible, we obtain a simple *a priori* lower bound on F_C^*.

Lemma 3.2. *If C is symmetric then $F_C^* \geq \operatorname{tr}(C)$.*

Likewise, the objective value of any dual feasible point for (SDP-D) will also provide us an upper bound on F_C^*. In the special case where C is diagonally dominant with positive diagonal entries, we can use the diagonal to construct such a dual feasible point.

Lemma 3.3. *If C is diagonally dominant and $\operatorname{diag}(C) > 0$ then $z = 2\operatorname{diag}(C)$ is dual feasible and $F_C^* \leq 2\operatorname{tr}(C)$.*

For the smoothing strategy we introduce in Section 4, the following entry-wise bounds on feasible dual variables play an important role.

Lemma 3.4. *If $C \succeq 0$ with $\operatorname{diag}(C) > 0$ and z is feasible for (SDP-D) then $z_i \geq C_{ii} > 0$ for $i = 1, 2, \ldots, n$.*

3.2. Nesterov Formulation of the Max-Cut SDP

If the cost matrix C is positive semidefinite, it has a factorization of $C = A^T A$, where $A \in \mathbb{R}^{m \times n}$. In this case, Nesterov derived an alternative formulation of the Max-Cut SDP (Nesterov, 2011). In this section, we summarize this formulation and its properties. All omitted proofs for this section are in Appendix B. Let

$$\tilde{\mathcal{S}}_C := \{A^T P A \in \mathbb{S}^n : P \succeq 0,\ \operatorname{tr}(P) = 1\} \quad \text{and} \tag{3}$$

$$\mathcal{S}_C := \operatorname{diag}(\tilde{\mathcal{S}}_C) \tag{4}$$

(While the sets $\tilde{\mathcal{S}}_C$ and $\mathcal{S}_C$ do depend on the choice of factorization of $C = A^T A$, this factorization will not play a significant role, so we suppress it from the notation.) The Nesterov reformulation is

$$\sup_{x \in \mathcal{S}_C} \sum_{i=1}^n \sqrt{x_i} = \sup_{W \in \tilde{\mathcal{S}}_C} \sum_{i=1}^n \sqrt{W_{ii}} \tag{Nesterov-P}$$

Note that the second, equivalent, formulation explicitly keeps track of a matrix variable. This matrix is needed to explicitly relate optimal points of (Nesterov-P) to those of (SDP-P) (see Lemma 3.6).

This formulation is potentially amenable to being solved efficiently using the Frank-Wolfe algorithm. This is because linear maximization over the constraint set $\mathcal{S}_C$ can be expressed in terms of solving a largest eigenvalue problem for a symmetric matrix. (The optimal value of the LMO is called the *support function* of $\mathcal{S}_C$, and is denoted $\sigma_{\mathcal{S}_C}(\cdot)$.)

Lemma 3.5. *If $y \in \mathbb{R}^n$ and $C = A^T A$ then*

$$\sigma_{\mathcal{S}_C}(y) := \sup_{q \in \mathcal{S}_C} \langle q, y \rangle = \lambda_{\max}\left(A \operatorname{diag}^\star(y) A^T\right). \tag{5}$$

If $y > 0$ then $\sigma_{\mathcal{S}_C}(y) = \|\operatorname{diag}^\star(y)^{1/2} C \operatorname{diag}^\star(y)^{1/2}\|$.

Recent work on low-memory algorithms for semidefinite programming (Shinde et al., 2021; Yurtsever et al., 2017) also make crucial use of the Frank-Wolfe method over constraint sets with LMOs that reduce to extreme eigenvalue problems. One obstacle to using the Frank-Wolfe algorithm for (Nesterov-P) is that the objective function (7) does not have a bounded curvature constant. In Section 4, we show how to modify (Nesterov-P) to obtain a smooth problem with the same optimal point and optimal value to which the analysis tools introduced in Section 2.2 can be applied.

It is fruitful to view (Nesterov-P) in terms of a smooth saddle point problem. Let $\phi : \mathcal{S}_C \times (0, \infty)^n$ be the concave-convex function defined by

$$\phi(x, y) = \sum_{i=1}^n \frac{1}{4y_i} + x_i y_i. \tag{6}$$

The partial infimum over y is

$$f(x) := \inf_{y > 0} \phi(x, y) = \sum_{i=1}^n \sqrt{x_i}, \tag{7}$$

which holds because $\sqrt{x}$ is the concave conjugate of $-1/(4y)$. The partial supremum over x is

$$g(y) := \sup_{x \in \mathcal{S}_C} \phi(x, y) = \sum_{i=1}^n \frac{1}{4y_i} + \sigma_{\mathcal{S}_C}(y). \tag{8}$$

The problem (Nesterov-P) can then be expressed as

$$f_C^* := \sup_{x \in \mathcal{S}_C} \inf_{y > 0} \phi(x, y) = \sup_{x \in \mathcal{S}_C} f(x)$$

Exchanging the order of the supremum and infimum gives a natural dual problem

$$g_C^* := \inf_{y > 0} \sup_{x \in \mathcal{S}_C} \phi(x, y) = \inf_{y > 0} g(y). \tag{Nesterov-D}$$

Since $\mathcal{S}_C$ is compact, $\phi(x, \cdot)$ has closed convex sublevel sets for any $x \in \mathcal{S}_C$, and $\phi(\cdot, y)$ has closed convex suplevel sets for any $y > 0$, it follows from Sion's minimax theorem (Sion, 1958) that $f_C^* = g_C^*$.

The following result explicitly relates (Nesterov-P) and (SDP-P), by showing how feasible points for one problem can be mapped to feasible points for the other in such a way that the objective value improves. This implies the problems are equivalent (see Corollary 3.7) and also shows how to map near-optimal points of one problem to the other.

Lemma 3.6. *Let $C \succeq 0$ with $\operatorname{diag}(C) > 0$.*

- *If X is feasible for (SDP-P) then*

$$\psi_C(X) := (CXC) / \langle C, X \rangle \in \tilde{\mathcal{S}}_C$$

and $f(\operatorname{diag}(\psi_C(X)))^2 \geq \langle C, X \rangle$.

- *Suppose* $W \in \tilde{\mathcal{S}}_C$, *and* $x = \operatorname{diag}(W)$. *For* $i = 1, 2, \ldots, n$ *let*

$$d_i = \begin{cases} x_i^{-1/2} & \text{if } x_i > 0 \\ 0 & \text{otherwise,} \end{cases}$$

and let $\delta_i = 1 - d_i^2 x_i$. *Then*

$$\xi(W) := \operatorname{diag}^\star(d) W \operatorname{diag}^\star(d) + \operatorname{diag}^\star(\delta)$$

is (SDP-P)*-feasible and* $\langle C, \xi(W)\rangle \geq f(\operatorname{diag}(W))^2$.

By considering optimal points for (Nesterov-P) and (SDP-P), we obtain the following.

Corollary 3.7. *If* W^* *is optimal for* (Nesterov-P) *then* $\xi(W^*)$ *is optimal for* (SDP-P). *Conversely if* X^* *is optimal for* (SDP-P) *then* $\psi_C(X^*)$ *is optimal for* (Nesterov-P). *Moreover* $(f_C^*)^2 = F_C^*$.

The following result specifies the relationship between (Nesterov-D) and (SDP-D).

Lemma 3.8. *Let* $C \succeq 0$ *with* $\operatorname{diag}(C) > 0$.

- *If* z *is feasible for* (SDP-D) *then*

$$\psi(z)_i := \langle \mathbf{1}, z\rangle^{1/2} / (2z_i) \quad \text{for } i = 1, 2, \ldots, n$$

satisfies $\psi(z) > 0$ *and* $g(\psi(z))^2 \leq \langle \mathbf{1}, z\rangle$.

- *If* $y > 0$ *then*

$$\xi_C(y)_i := y_i^{-1} \sigma_{\mathcal{S}_C}(y) \quad \text{for } i = 1, 2, \ldots, n \tag{9}$$

is feasible for (SDP-D) *and* $\langle \mathbf{1}, \xi_C(y)\rangle \leq g(y)^2$.

By a similar argument to Lemma 3.6, $(g_C^*)^2 = G_C^*$ and $\psi(z^*)$ and $\xi_C(y^*)$ are corresponding optimal points.

4. Smoothing the Nesterov Formulation

4.1. Smoothing Approach

In this section we describe how we smooth the objective function of (Nesterov-P) so that we can obtain convergence guarantees for the Frank-Wolfe algorithm. All omitted proofs for this section are in Appendix C.

In smoothing the objective we would like to obtain a bounded curvature constant and preserve the optimal value of the original problem. We use a variation on the standard idea of smoothing by regularizing the dual (Nesterov, 2005). To preserve the optimal value we will do this by adding constraints to the dual that are satisfied at optimality.

Initially, let $\alpha > 0$ be any entry-wise positive vector in $\mathbb{R}^n$. Consider adding the (entry-wise) constraint $y \leq \alpha$ to the saddle point problems, i.e., consider

$$\begin{aligned} f_{\alpha,C}^* &:= \sup_{x \in \mathcal{S}_C} \inf_{\alpha \geq y > 0} \phi(x, y) \\ &= \inf_{\alpha \geq y > 0} \sup_{x \in \mathcal{S}_C} \phi(x, y) =: g_{\alpha,C}^*. \end{aligned}$$

By Sion's minimax theorem, the smoothed primal and dual problems have the same optimal value. The new dual is obtained by adding our constraint to (Nesterov-D), i.e.,

$$\inf_{\alpha \geq y > 0} g(y), \tag{Smooth-D}$$

where $g(\cdot)$ is the objective function defined in (8). To write down the smoothed primal problem, we define $h_\beta(x) := \inf_{\beta \geq y > 0} xy + 1/(4y)$, given explicitly by

$$h_\beta(x) = \begin{cases} \sqrt{x} & \text{if } x \geq \frac{1}{4\beta^2} \\ \beta x + \frac{1}{4\beta} & \text{if } 0 \leq x < \frac{1}{4\beta^2}. \end{cases} \tag{10}$$

Note that $\beta x + \frac{1}{4\beta}$ is the linearization of $\sqrt{\cdot}$ at $\frac{1}{4\beta^2}$. Since $\sqrt{\cdot}$ is concave, $h_\beta(x) \geq \sqrt{x}$ for all $x \geq 0$.

The smoothed primal problem is then

$$\sup_{x \in \mathcal{S}_C} f_\alpha(x) = \sup_{W \in \tilde{\mathcal{S}}_C} f_\alpha(\operatorname{diag}(W)), \tag{Smooth-P}$$

$$\text{where} \quad f_\alpha(x) = \inf_{\alpha \geq y > 0} \phi(x, y) = \textstyle\sum_{i=1}^n h_{\alpha_i}(x_i). \tag{11}$$

The following result shows that if we choose α appropriately, the optimal values and points are unchanged by smoothing.

Lemma 4.1. *Let* y^* *be optimal for* (Nesterov-D) *and* W^* *be optimal for* (Nesterov-P). *If* $\alpha \geq y^*$ *then*

$$f_\alpha(\operatorname{diag}(W^*)) = f_{\alpha,C}^* = f_C^* = f(\operatorname{diag}(W^*)).$$

Given a feasible point for (Smooth-P), we can construct a feasible point for the original semidefinite program (SDP-P) with improved objective function value, as long as α is large enough.[1] The construction can also be implemented at the level of Gaussian samples, so that it is also compatible with sample-based memory-efficient variants on the Frank-Wolfe algorithm, as discussed in Section 5.

Lemma 4.2. *Let* C *be positive semidefinite with* $\operatorname{diag}(C) > 0$. *Suppose that* $\alpha_i \geq \frac{(F_C^*)^{1/2}}{C_{ii}}$ *for all* i.

Suppose $W \in \tilde{\mathcal{S}}_C$ *and* $x = \operatorname{diag}(W)$. *For* $i = 1, 2, \ldots, n$ *let* $d_i := \min\{x_i^{-1/2}, 2\alpha_i\}$ *and* $\delta_i = 1 - d_i^2 x_i$. *Then*

$$\zeta_\alpha(W) := \operatorname{diag}^\star(d) W \operatorname{diag}^\star(d) + \operatorname{diag}^\star(\delta)$$

is (SDP-P)*-feasible and* $\langle C, \zeta_\alpha(W)\rangle \geq f_\alpha(\operatorname{diag}(W))^2$.

If $s \sim \mathcal{N}(0, W)$ *and* $s' \sim \mathcal{N}(0, I)$ *are independent then*

$$\operatorname{diag}^\star(d) s + \operatorname{diag}^\star(\delta)^{1/2} s' \sim \mathcal{N}(0, \zeta_\alpha(W))$$

which can be computed based on x *and* s *alone.*

[1] Taking $\alpha_i \to \infty$ for all i recovers one part of Lemma 3.6.

For this approach to be useful, we would like to choose the smoothing parameter to satisfy the requirements both of Lemmas 4.1 and 4.2. The next definition ensures this.

Definition 4.3. Let C be positive semidefinite with positive diagonal entries. A point $\alpha \in \mathbb{R}^n$ is *C-valid* if $\alpha_i \geq (F_C^*)^{1/2}/C_{ii}$ for all $i = 1, 2, \ldots, n$.

Lemma 4.4. *If α is C-valid and y^* is an optimal point for* (Smooth-D) *then $y^* \leq \alpha$.*

In practice we would like to choose an α that is C-valid based on data that are available to us. We do this assuming we have access to some feasible point for (SDP-D), which gives us an upper bound on F_C^*.

Lemma 4.5. *Let C be positive semidefinite with positive diagonal entries. If z is feasible for* (SDP-D) *then $\alpha_i = \frac{\langle \mathbf{1}, z\rangle^{1/2}}{C_{ii}}$ for $i = 1, 2, \ldots, n$ is C-valid.*

4.2. Bounding the Curvature Constant

In this section, we establish bounds on the curvature constant of the smoothed primal objective function f_α. We do this by bounding the diameter and Lipschitz constant of ∇f_α in a norm adapted to the problem. All omitted proofs for this section are in Appendix D.

Given a vector $z > 0$, define $\|x\|_z := \sum_{i=1}^n z_i^{-1}|x_i|$.

Lemma 4.6. *If $z > 0$ is feasible for* (SDP-D) *then* $\operatorname{diam}_{\|\cdot\|_z}(\mathcal{S}_C) \leq 2$.

Next, we bound the Lipschitz constant of ∇f_α.

Lemma 4.7. *Let $\alpha > 0$ and $z > 0$ be entry-wise positive vectors and let $L = \max_i 2\alpha_i^3 z_i^2$. Then ∇f_α is L-Lipschitz with respect to $\|\cdot\|_z$ over $\mathcal{S}_C$.*

Substituting the C-valid choice of α from Lemma 4.5 into these bounds for the Lipschitz constant of ∇f_α and diameter of $\mathcal{S}_C$, directly gives a bound on $\mathcal{M}(f_\alpha|\mathcal{S}_C)$.

Theorem 4.8. *Let C be positive semidefinite with* $\operatorname{diag}(C) > 0$. *Let z satisfy* $\operatorname{diag}^\star(z) \succeq C$, *and let $\alpha_i = \langle \mathbf{1}, z\rangle^{1/2}/C_{ii}$ for $i = 1, 2, \ldots, n$. Then*

$$\mathcal{M}(f_\alpha|\mathcal{S}_C) \leq 8\langle \mathbf{1}, z\rangle^{1/2} \max_i \frac{\langle \mathbf{1}, z\rangle z_i^2}{C_{ii}^3}.$$

Unless the diagonal entries of C are uniformly bounded below, the bound on the curvature constant in Theorem 4.8 could be arbitrarily bad if C has small diagonal entries.

4.3. Improving the Conditioning of diag(C)

Since (SDP-P) has the constraint $\operatorname{diag}(X) = \mathbf{1}$, diagonally shifting the cost matrix from C to $C + \operatorname{diag}^\star(\mu)$ has the effect of just adding the constant value $\langle \mathbf{1}, \mu\rangle$ to the objective function (and the optimal value). It does not change the optimal primal solution. We use this observation to carefully choose a shifting μ that improves the conditioning of the diagonal of the cost matrix while only increasing the optimal value of the problem by a constant factor. All omitted proofs for this section are in Appendix E.

The following result bounds the curvature constant after both choosing the smoothing parameter and shifting the cost based on the dual feasible point z. Crucially, now, the bound is no longer sensitive to small entries on the diagonal of C.

Theorem 4.9. *Let C be positive semidefinite with $diag(C) > 0$. Let z satisfy* $\operatorname{diag}^\star(z) \succeq C$. *Let*

$$\mu_i = z_i + \tfrac{1}{n}\langle \mathbf{1}, z\rangle \quad \text{for } i = 1, 2, \ldots, n$$

and let $C_\mu = C + \operatorname{diag}^\star(\mu)$. Let $z_\mu = z + \mu$ and let $[\alpha_\mu]_i = \langle \mathbf{1}, z_\mu\rangle^{1/2}/[C_\mu]_{ii}$ for $i = 1, 2, \ldots, n$. Then

$$\mathcal{M}(f_{\alpha_\mu}|\mathcal{S}_{C_\mu}) \leq 32 \cdot 3\sqrt{3}n\langle \mathbf{1}, z\rangle^{1/2}.$$

The smoothed problem with the shifted objective can be used to solve the original semidefinite program (SDP-P). We do this by running Algorithm 2 with appropriate parameter choices. We require an *a priori* lower bound $F_{LB} \leq F_C^*$, which we take to be $\operatorname{tr}(C)$ in many cases.

Theorem 4.10. *Let $C \succeq 0$ with* $\operatorname{diag}(C) > 0$. *Let z satisfy* $\operatorname{diag}^\star(z) \succeq C$. *Define μ, α_μ, C_μ, and z_μ as in Theorem 4.9. Let F_{LB} be any lower bound on F_C^* and let $\rho = \frac{\langle \mathbf{1}, z\rangle}{F_{LB}}$. Let $\epsilon' = \frac{3\epsilon}{8\sqrt{3}}\frac{\langle \mathbf{1}, z\rangle^{1/2}}{\rho}$ and $\delta' = \epsilon'/3$. If*

$$(x, W, s) = \texttt{FW-sampling}(f_{\alpha_\mu}, C_\mu, \epsilon', \delta', x^{(0)})$$

then $F_C^* - \langle C, \zeta_{\alpha_\mu}(W)\rangle \leq \epsilon F_{LB} \leq \epsilon F_C^*$ after at most $O\left(\frac{n\rho}{\epsilon}\right)$ calls to FACTORED-LMO$(\cdot, \mathcal{S}_{C_\mu}, \delta')$.

Shifting the diagonal of C in this z-dependent way also ensures that the optimal value of the LMO is not too large. This is important because the complexity of our LMO will depend on the desired relative error (see Theorem 5.1).

Lemma 4.11. *Let $C \succeq 0$ with* $\operatorname{diag}(C) > 0$. *Let z satisfy* $\operatorname{diag}^\star(z) \succeq C$. *Define μ, α_μ, C_μ, and z_μ as in Theorem 4.9. Then $h_{\mathcal{S}_{C_\mu}}(\nabla f_{\alpha_\mu}(x)) \leq 2\sqrt{3}\langle \mathbf{1}, z\rangle^{1/2}$ for all $x \in \mathcal{S}_{C_\mu}$.*

Remark 4.12. If δ' is chosen as in Theorem 4.10 we can implement FACTORED-LMO$(\cdot, \mathcal{S}_{C_\mu}, \delta')$ by solving the LMO to relative error $\delta_{\text{rel}} = \frac{1}{48}(\epsilon/\rho)$.

If we apply Frank-Wolfe to (Smooth-P) with cost C_μ, we obtain convergence bounds that depend only on $\rho = \frac{\langle \mathbf{1}, z\rangle}{F_{LB}}$.

- If C is diagonally dominant with $\operatorname{diag}(C) > 0$, we can take $z = 2\operatorname{diag}(C)$ and $F_{LB} = \operatorname{tr}(C)$, so $\rho \leq 2$
- If $C \succeq 0$ with $\operatorname{diag}(C) > 0$ and C is sparse with respect to a graph with maximum degree Δ, then (see Lemma E.1) $C \preceq \operatorname{diag}^\star(z)$ where $z = (\Delta + 1)\operatorname{diag}(C)$. Taking $F_{LB} = \operatorname{tr}(C)$ gives $\rho \leq \Delta + 1$.

- For a general $C \in \mathbb{S}^n$ with $\operatorname{diag}(C) > 0$, the matrix $C_d = C + \operatorname{diag}^\star(d)$ where $d_i = \sum_{j \neq i} |C_{ij}|$ is diagonally dominant with $F_{LB} = \operatorname{tr}(C_d) \geq \sum_{i,j} |C_{ij}| =: \|C\|_1$. Applying our method to C_d shows that we can solve general instances of (SDP-P) to additive error $\epsilon \|C\|_1$ (the error model in (Lee & Padmanabhan, 2020)) by reducing to the diagonally dominant case.

5. Implementation

Algorithm 2 gives pseudo-code for Frank-Wolfe specialized to solve problems of the form (Smooth-P). The basic form of the algorithm (skipping lines 6–12) works only with the n-dimensional decision variable $x^{(t)} \in \mathcal{S}_C$, and suffices to (approximately) compute the optimal value. If our aim is to compute a near-optimal point for (SDP-P), it suffices to also instantiate, update (by running lines 6–8), and return the $n \times n$ matrix variable $W^{(t)} \in \tilde{\mathcal{S}}_C$. If the memory required to store the $W^{(t)}$ variable is prohibitive, we can instead instantiate, update (by running lines 9–12), and return the variable $s^{(t)}$, which is updated to maintain the property of being a random vector with distribution $\mathcal{N}(0, W^{(t)})$.

Given $s^{(t)}$ and $x^{(t)}$ (but not $W^{(t)}$), Lemma 4.2 tells us how to sample a Gaussian random variable with zero mean and covariance equal to a near-optimal point for the original Max-Cut SDP, (SDP-P). This sample-based representation of the solution of an SDP was introduced in (Shinde et al., 2021). It is of particular interest for the Max-Cut SDP, since such a Gaussian random vector is all that is required to implement the rounding scheme of (Goemans & Williamson, 1995). For further details see (Shinde et al., 2021).

The other minor difference between Algorithm 2 and the basic version of Frank-Wolfe in Algorithm 1 is the presence of the `FACTORED-LMO` subroutine. This returns q, a valid output of the LMO with respect to $\mathcal{S}_C$, as well as a vector v such that $vv^T \in \tilde{\mathcal{S}}_C$ is a valid output of the LMO with respect to $\tilde{\mathcal{S}}_C$. This takes advantage of the fact that it is always possible for the LMO with respect to $\tilde{\mathcal{S}}_C$ to return a rank one solution, since the extreme rays have rank one. Maintaining this LMO output in factored form (as v rather than vv^T) means that when Algorithm 2 is run without instantiating the matrix variable $W^{(t)}$, the only variables being maintained are n-dimensional vectors.

5.1. LMO Implementation

In this section, we give more details about the LMO in Algorithm 2 (see Algorithm 3). Since $\nabla f_\alpha(x) > 0$ for any $x \in \mathcal{S}_C$, we only need to implement the LMO for entry-wise positive costs. Due to Lemma 3.5, the LMO reduces to approximately finding the largest (magnitude) eigenvalue of a positive semidefinite matrix. The two cases in Algorithm 3 depend on whether we can compute matrix-vector multiplications with A and A^T or only with C. In Lemma F.1 we justify the correctness of choosing v via line 6 of Algorithm 3 in the case where we only have access to C.

Algorithm 2 Frank-Wolfe with sampling $(x, W, s) =$ `FW-sampling`$(f, C, \epsilon, \delta, x^{(0)})$

input $C \succeq 0$, $x^{(0)} \in \mathcal{S}_C$, $\epsilon > \delta > 0$.
output $x^{(t)} \in \mathcal{S}_C$ such that $f^* - f(x^{(t)}) \leq \epsilon + \delta$,
$W^{(t)} \in \tilde{\mathcal{S}}_C$ such that $x^{(t)} = \operatorname{diag}(W^{(t)})$,
Gaussian vector $s^{(t)} \sim \mathcal{N}(0, W^{(t)})$

1: $(q^{(0)}, v^{(0)}) :=$ `FACTORED-LMO`$(\nabla f(x^{(0)}), \mathcal{S}_C, \delta)$
2: $t := 0$
3: **while** $\langle \nabla f(x^{(t)}), q^{(t)} - x^{(t)} \rangle > \epsilon$ **do**
4: $\quad \gamma^{(t)} := 2/(t+2)$
5: $\quad x^{(t+1)} := (1 - \gamma^{(t)}) x^{(t)} + \gamma^{(t)} q^{(t)}$
6: $\quad$ **if** Returning W **then**
7: $\quad\quad W^{(t+1)} := (1 - \gamma^{(t)}) W^{(t)} + \gamma^{(t)} v^{(t)} (v^{(t)})^T$
8: $\quad$ **end if**
9: $\quad$ **if** Returning s **then**
10: $\quad\quad$ sample $\omega \sim \mathcal{N}(0, 1)$
11: $\quad\quad s^{(t+1)} := (1 - \gamma^{(t)})^{1/2} s^{(t)} + (\gamma^{(t)})^{1/2} \omega v^{(t)}$
12: $\quad$ **end if**
13: $\quad (q^{(t+1)}, v^{(t+1)}) :=$ `FACTORED-LMO`$(\nabla f(x^{(t+1)}), \mathcal{S}_C, \delta)$
14: $\quad t := t + 1$
15: **end while**

In either case, the core of the LMO is finding an approximate largest eigenvalue of a positive semidefinite matrix (lines 2 or 5 of Algorithm 3). We could do this using the Lanczos method with random start (see (Yurtsever et al., 2021) for storage-efficient pseudocode), which has the following convergence guarantee (Kuczyński & Woźniakowski, 1992).

Theorem 5.1. *Let $M \succeq 0$, let $\delta_{\text{rel}} \in (0, \frac{1}{8})$ and $p \in (0, \frac{1}{2})$. Then with probability at least p, after $q = O(\delta_{\text{rel}}^{-1/2} \log(n/p^2))$ matrix-vector multiplications with M and $O(qn)$ addition operations, the Lanczos method finds a unit vector u satisfying $u^T M u \geq (1 - \delta_{\text{rel}}) \lambda_{\max}(M)$.*

6. Numerical Experiments

In this section, we summarize our computational results from applying Algorithm 2 with $C = \frac{1}{4} L_G$ where L_G is the Laplacian of an (unweighted) graph. In all cases, C is diagonally dominant and we set $z = 2 \operatorname{diag}(C)$ and $\alpha_i = \frac{(2\operatorname{tr}(C))^{1/2}}{C_{ii}}$ for all i. For the examples we tried, we found that diagonal shifting did not improve the practical performance.

The algorithm was implemented using Julia 1.7.1. All computations were performed on a machine with 4 cores, 8 2.5GHz-CPUs and 16GB of RAM. In all experiments, the starting vector $x^{(0)}$ for Algorithm 2 is given as $x^{(0)} =$

Table 1. Average iteration count and corresponding average relative error comparison between our method and methods of Shinde et al. (2021) and Yurtsever et al. (2021) for Gset graphs. The averages are over the groups of graphs. The smallest average iteration counts for each group are highlighted in bold.

Graph Group Number	Members of Group	Graph Parameter		Yurtsever et al		Shinde et al.		This paper	
		vxs n	edges m	Avg Iter	Avg Rel Err	Avg Iter	Avg Rel Err	Avg Iter	Avg Rel Err
1	G1 to G5	800	19176	**138.2**	0.008	30469.20	0.027	147.00	0.005
2	G14 to G17	800	≈4675	913.15	0.023	193806.25	0.049	**79.00**	0.017
3	G22 to G26	2000	19990	**126.80**	0.007	154352.60	0.040	155.60	0.004
4	G35 to G37	2000	≈11775	2314	0.020	711903.67	0.047	**99.33**	0.013
5	G43 to G47	1000	9990	**130.20**	0.008	28197.20	0.030	132.60	0.005
6	G48 to G50	3000	6000	**145.67**	0.005	17417.33	0.015	407.67	0.003
7	G51 to G54	1000	≈5915	1149.75	0.022	242651.50	0.038	**92.00**	0.013

Algorithm 3 LMO for $\mathcal{S}_C$ with additive error δ
$(q, v) = \texttt{FACTORED-LMO}(y, \mathcal{S}_C, \delta)$

input vector $y > 0$, accuracy δ, $C \succeq 0$
output v such that $vv^T \in \tilde{\mathcal{S}}_C$ and
$$\langle vv^T, \operatorname{diag}^\star(y)\rangle \geq \sigma_{\tilde{\mathcal{S}}_C}(\operatorname{diag}^\star(y)) - \delta,$$
and $q = v \circ v = \operatorname{diag}(vv^T)$.
if Have access to A such that $C = A^T A$ **then**
  Find u such that $\|u\|_2 = 1$ and
  $u^T(A \operatorname{diag}^\star(y) A^T)u \geq \lambda_{\max}(A \operatorname{diag}^\star(y) A^T) - \delta$.
  $v := A^T u$
else
  Find u such that $\|u\|_2 = 1$ and
  $u^T \operatorname{diag}^\star(y)^{\frac{1}{2}} C \operatorname{diag}^\star(y)^{\frac{1}{2}} u \geq$
  $\lambda_{\max}(\operatorname{diag}^\star(y)^{\frac{1}{2}} C \operatorname{diag}^\star(y)^{\frac{1}{2}}) - \delta$.
  $v := C \operatorname{diag}^\star(y)^{\frac{1}{2}} u / (u^T \operatorname{diag}^\star(y)^{\frac{1}{2}} C \operatorname{diag}^\star(y)^{\frac{1}{2}} u)^{\frac{1}{2}}$
end if
$q := v \circ v$

$\frac{1}{m}\operatorname{diag}(C)$ where m is the number of edges. We stop when

$$\text{RFWgap}(x^{(t)}) := \frac{\langle \nabla f(x^{(t)}), q^{(t)} - x^{(t)} \rangle}{f(x^{(t)})} \leq \epsilon,$$

an easily computed bound on the relative error.

For the LMO subroutine, in practice we use a Julia implementation, `ArnoldiMethod.jl`, of the Krylov-Schur method (Stewart, 2002). The method terminates when $\|Ax - x\lambda\|_2 < \texttt{tol}\,|\lambda|$, where `tol` is the tolerance parameter. We initially set $\texttt{tol} = 1$ and reduce it by a factor of $10^{-0.25}$ every time the RFWgap decreases by a factor of $10^{-0.25}$, while ensuring $\texttt{tol} \leq$ RFWgap.

Gset experiments For our first experiment, we use unweighted graphs from the Gset dataset (gse). To display the results concisely, we grouped the graphs into groups consisting of graphs with similar properties (see Table 1). We tested our method with both the scheduled step size $\gamma^{(t)} = \frac{2}{t+2}$ and a line search. We stopped when RFWgap $< 10^{-2.5}$.

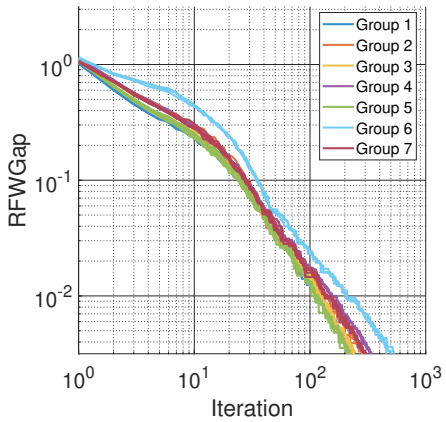

Figure 1. Plot of $\min_{\tau \leq t} \text{RFWgap}(x^{(\tau)})$ vs Frank-Wolfe iterations for the line-search variant of our algorithm applied to Gset graphs. Each line represents a single graph. The lines are colored by group membership.

Figure 1 shows the convergence of RFWgap as a function of the number of Frank-Wolfe iterations for the line-search variant. (Results for the scheduled step size are very similar, and are shown in Figure 3 of Appendix G.)

While our theory predicts a (global) convergence rate of $O(1/t)$, in practice the local convergence rate is significantly better, being possibly more like $O(1/t^2)$.

We compare our results on the Gset graphs with related memory-efficient approaches by Shinde et al. (2021) and Yurtsever et al. (2021). Both are Frank-Wolfe-based methods with LMOs of similar cost to ours, so we compare the number of Frank-Wolfe iterations. Due to differing stopping criteria, for each case we set ϵ so that we reach a relative error at least as good as the best-performing previous method. Because the comparison methods work with (SDP-P) and our experiments work with (Smooth-P) (having optimal value $(F_C^*)^{1/2}$), we scaled the relative error for the comparison methods down by a factor of two. The 'true' optimal

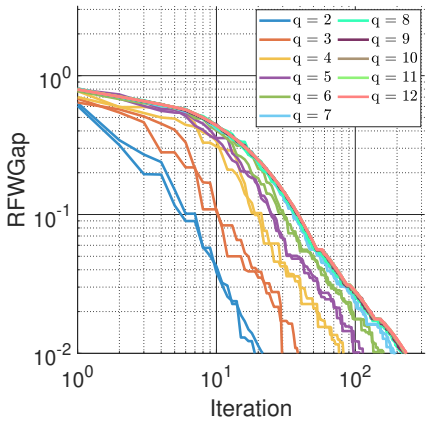

Figure 2. Plot of $\min_{\tau \leq t} \text{RFWgap}(x^{(\tau)})$ vs Frank-Wolfe iterations for our algorithm applied to 3-regular graphs with $10^{q/2}$ vertices. Each line represents a single graph. Colors indicate graphs with the same parameters.

value used for relative error calculations is the square root of the value generated using SDPT3 (Toh et al., 1999).

Table 1 summarizes our results. (See Table 2 in Appendix G for the full results.) Our method significantly outperforms the method proposed by Shinde et al. (2021), and performs better for some graphs, and worse for others, than the method of Yurtsever et al. (2021). The groups of graphs for which our method performs best are those where the method of Yurtsever et al. (2021) performs worst, and conversely. From initial observations, our method performs better than that of Yurtsever et al. (2021) on graphs with skewed degree distributions, while performing worse on instances that are sparse with respect to torus graphs.

Large-scale d-regular random graphs We demonstrate the scalability of our method by applying it to large random d-regular graphs, generated using the method in (Kim & Vu, 2003). For each (n, d) with $d = 3, 5$ and $n = 10^1, 10^{1.5}, 10^2, \ldots, 10^6$ we generated two random graphs. The convergence plots for the 3-regular graphs are shown in Figure 2. The plot for the 5-regular graph is remarkably similar and is included in Figure 4 in Appendix G. While our theoretical convergence rates are $O(n/t)$, Figure 2 shows much more mild dependence on n. The local convergence rate appears close to $O(1/t^2)$, just as in Figure 1.

Acknowledgements

The authors would like to thank Pablo Parrilo for helpful discussions at the early stages of this work. JS is the recipient of an Australian Research Council Discovery Early Career Researcher Award (project number DE210101056) funded by the Australian Government.

References

UF sparse matrix collection: Gset group. https://www.cise.ufl.edu/research/sparse/matrices/Gset/index.html. Accessed: 2023-01-05.

Arora, S. and Kale, S. A combinatorial, primal-dual approach to semidefinite programs. *Journal of the ACM*, 63 (2):1–35, 2016.

Arora, S., Hazan, E., and Kale, S. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pp. 339–348. IEEE, 2005.

Bach, F. Sum-of-squares relaxations for information theory and variational inference. *arXiv preprint arXiv:2206.13285*, 2022.

Barvinok, A. I. Problems of distance geometry and convex properties of quadratic maps. *Discrete & Computational Geometry*, 13:189–202, 1995.

Burer, S. and Monteiro, R. D. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

Charikar, M. and Wirth, A. Maximizing quadratic programs: Extending grothendieck's inequality. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 54–60. IEEE, 2004.

Della Riccia, G. and Shapiro, A. Minimum rank and minimum trace of covariance matrices. *Psychometrika*, 47: 443–448, 1982.

Erdogdu, M. A., Ozdaglar, A., Parrilo, P. A., and Vanli, N. D. Convergence rate of block-coordinate maximization Burer–Monteiro method for solving large SDPs. *Mathematical Programming*, 195(1-2):243–281, 2022.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2): 95–110, 1956.

Freund, R. M. and Grigas, P. New analysis and results for the frank–wolfe method. *Mathematical Programming*, 155(1-2):199–230, 2016.

Goemans, M. X. and Williamson, D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

Jaggi, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pp. 427–435. PMLR, 2013.

Kim, J. and Vu, V. Generating random regular graphs. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pp. 213–222. ACM, 2003.

Klein, P. and 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, 1996.

Klein, P. N. and Lu, H.-I. Space-efficient approximation algorithms for MAXCUT and COLORING semidefinite programs. In *ISAAC*, pp. 387–396. Springer, 1998.

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

Lanckriet, G. R., Cristianini, N., Bartlett, P., Ghaoui, L. E., and Jordan, M. I. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5(Jan):27–72, 2004.

Lee, Y. T. and Padmanabhan, S. An $\widetilde{\mathcal{O}}(m/\varepsilon^{3.5})$-cost algorithm for semidefinite programs with diagonal constraints. In *Conference on Learning Theory*, pp. 3069–3119. PMLR, 2020.

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

Nesterov, Y. Barrier subgradient method. *Mathematical Programming*, 127(1):31–56, 2011.

Pataki, G. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23(2):339–358, 1998.

Raghunathan, A., Steinhardt, J., and Liang, P. S. Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in Neural Information Processing Systems*, 31, 2018.

Shinde, N., Narayanan, V., and Saunderson, J. Memory-efficient structured convex optimization via extreme point sampling. *SIAM Journal on Mathematics of Data Science*, 3(3):787–814, 2021.

Sion, M. On general minimax theorems. *Pacific J. Math.*, 8: 171–176, 1958.

Srebro, N. and Shraibman, A. Rank, trace-norm and max-norm. In *Learning Theory: 18th Annual Conference on Learning Theory, COLT 2005, Bertinoro, Italy, June 27-30, 2005. Proceedings 18*, pp. 545–560. Springer, 2005.

Stewart, G. W. A Krylov-Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2002.

Toh, K. C., Todd, M. J., and Tütüncü, R. H. SDPT3 - A Matlab software package for semidefinite programming, Version 1.3. *Optimization methods & Software*, 11(1-4): 545–581, 1999.

Tropp, J. A., Yurtsever, A., Udell, M., and Cevher, V. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. *Advances in Neural Information Processing Systems*, 30, 2017.

Vandenberghe, L. and Boyd, S. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

Waldspurger, I. and Waters, A. Rank optimality for the burer–monteiro factorization. *SIAM Journal on Optimization*, 30(3):2577–2602, 2020.

Yurtsever, A., Udell, M., Tropp, J., and Cevher, V. Sketchy decisions: Convex low-rank matrix optimization with optimal storage. In *Artificial Intelligence and Statistics*, pp. 1188–1196. PMLR, 2017.

Yurtsever, A., Tropp, J. A., Fercoq, O., Udell, M., and Cevher, V. Scalable semidefinite programming. *SIAM Journal on Mathematics of Data Science*, 3(1):171–200, 2021.

A. Additional proofs for Section 3.1

Proof of Lemma 3.3. If $z = 2\text{diag}(C)$ then $[\text{diag}^\star(z) - C]_{ii} = C_{ii}$ for all i, and $[\text{diag}^\star(z) - C]_{ij} = -C_{ij}$ for all $i \neq j$. Therefore if C is diagonally dominant with positive diagonal entries, then so is $\text{diag}^\star(z) - C$. By the Gershgorin circle theorem, diagonally dominant matrices with positive diagonal entries are positive semidefinite. Therefore z is dual feasible and $F_C^* \leq \langle \mathbf{1}, z\rangle = 2\,\text{tr}(C)$. □

Proof of Lemma 3.4. This follows from the fact that positive semidefinite matrices have nonnegative diagonal entries. Then $\text{diag}^\star(z) - C \succeq 0$ implies that $z_{ii} \geq C_{ii}$ for all i. □

B. Additional proofs for Section 3.2

Proof of Lemma 3.5. If $y \in \mathbb{R}^n$ then

$$\sup_{q \in \mathcal{S}_C} \langle q, y\rangle = \sup_{\substack{P \succeq 0,\\ \text{tr}(P)=1}} \left\langle \text{diag}(A^T P A), y\right\rangle = \sup_{\substack{P \succeq 0,\\ \text{tr}(P)=1}} \left\langle P, A\,\text{diag}^\star(y) A^T\right\rangle = \lambda_{\max}\left(A\,\text{diag}^\star(y) A^T\right).$$

If $y > 0$ then we can factor $A\,\text{diag}^\star(y)A^T = (A\,\text{diag}^\star(y)^{1/2})(A\,\text{diag}^\star(y)^{1/2})^T$. Then

$$\begin{aligned}\lambda_{\max}\left(A\,\text{diag}^\star(y)A^T\right) = \left\|A\,\text{diag}^\star(y)^{1/2}\right\|^2 &= \lambda_{\max}\left(\text{diag}^\star(y)^{1/2} A^T A\,\text{diag}^\star(y)^{1/2}\right)\\ &= \lambda_{\max}\left(\text{diag}^\star(y)^{1/2} C \text{diag}^\star(y)^{1/2}\right) = \left\|\text{diag}^\star(y)^{1/2} C \text{diag}^\star(y)^{1/2}\right\|.\end{aligned}$$

□

Proof of Lemma 3.6. Let $\psi(X) = \frac{AXA^T}{\text{tr}(AXA^T)}$. Clearly $\psi(X) \succeq 0$ and $\text{tr}(\psi(X)) = 1$ so $\psi(X)$ is feasible for (Nesterov-P). Let $x_1, x_2, \ldots, x_n$ be unit vectors such that $X_{ij} = \langle x_i, x_j\rangle$, and let R be the matrix with columns given by the x_i. Let $Y(X) = RA^T/\|RA^T\|_F$ and note that $\psi(X) = Y(X)^T Y(X)$. Now,

$$\begin{aligned}f(\psi(X)) &= \sum_{i=1}^n (A_i^T \psi(X) A_i)^{1/2}\\ &= \sum_{i=1}^n \|Y(X)A_i\|_2\\ &\geq \sum_{i=1}^n \langle Y(X)A_i, x_i\rangle \quad \text{(since } \|x_i\| = 1 \text{ for all } i)\\ &= \sum_{i=1}^n \left\langle RA^T, x_i A_i^T\right\rangle / \|RA^T\|_F\\ &= \left\langle RA^T, RA^T\right\rangle / \|RA^T\|_F\\ &= \langle C, X\rangle^{1/2}.\end{aligned}$$

The fact that $\xi(W) = \text{diag}^\star(d) W \text{diag}^\star(d) + \text{diag}^\star(\delta)$ is feasible for (SDP-P) and satisfies $\langle C, \xi(W)\rangle^{1/2} \geq f(W)$ is the limiting case of Lemma 4.2 as $\alpha_i \to \infty$ for all i, so we omit the proof. □

Proof of Corollary 3.7. Given W^* such that $f(\text{diag}(W^*)) = f^*(C)$ we have

$$F^*(C) \geq \langle C, \xi(W^*)\rangle \geq f(\text{diag}(W^*))^2 = f^*(C)^2.$$

Given X^* such that $\langle C, X^*\rangle = F^*(C)$ we have

$$f^*(C)^2 \geq f(\text{diag}(\psi_C(X^*)))^2 \geq \langle C, X^*\rangle = F^*(C).$$

Combining these gives $F^*(C) = f^*(C)^2$. Moreover $\langle C, \xi(W^*)\rangle \geq \langle C, X^*\rangle$ and $f(\text{diag}(\psi_C(X^*))) \geq f(\text{diag}(W^*))$, establishing optimality of $\xi(W^*)$ and $\psi_C(X^*)$, respectively. □

Proof of Lemma 3.8. Suppose that $\operatorname{diag}^\star(z) \succeq C$ and $\operatorname{diag}(C) > 0$. Then $z_{ii} > 0$ for all i and so $\psi(z)_i > 0$. Therefore $\psi(z)$ is feasible for (Nesterov-D). Since $X^T X \preceq I$ is equivalent to $\|XX^T\| \leq 1$ for any real matrix X, it follows that

$$\operatorname{diag}^\star(z) \succeq C = A^T A \quad \iff \quad I \succeq \operatorname{diag}^\star(z)^{-1/2} A^T A \operatorname{diag}^\star(z)^{-1/2} \quad \iff \quad \sigma_{\mathcal{S}}(z^{-1}) = \left\| A \operatorname{diag}^\star(z)^{-1} A^T \right\| \leq 1.$$

Now,

$$g(\psi(z)) = \sum_{i=1}^n \frac{1}{4\psi(z)_i} + \sigma_{\mathcal{S}}(\psi(z)) = \frac{1}{2\langle \mathbf{1}, z\rangle^{1/2}} \sum_{i=1}^n z_i + \frac{\langle \mathbf{1}, z\rangle^{1/2}}{2} \sigma_{\mathcal{S}}(z^{-1}) \leq \langle \mathbf{1}, z\rangle^{1/2}.$$

On the other hand, suppose that $y > 0$. Then

$$\operatorname{diag}^\star(\xi(y)) \succeq C = A^T A \quad \iff \quad \left\| \sum_{i=1}^n A_i A_i^T y_i \right\| I \succeq \operatorname{diag}^\star(y)^{1/2} A^T A \operatorname{diag}^\star(y)^{1/2}$$

which holds because $X^T X \preceq \|XX^T\| I$ for any real matrix X. Therefore $\xi(y)$ is feasible for (SDP-D). Now,

$$g(y) = \sum_{i=1}^n \frac{1}{4y_i} + \sigma_{\mathcal{S}}(y) = \frac{1}{4\sigma_{\mathcal{S}}(y)} \sum_{i=1}^n \xi(y)_i + \sigma_{\mathcal{S}}(y) \geq \min_{\alpha > 0} \frac{1}{\alpha} \frac{\langle \mathbf{1}, \xi(y)\rangle}{4} + \alpha = \langle \mathbf{1}, \xi(y)\rangle^{1/2},$$

where we have used the fact that if $\beta > 0$ then $\min_{\alpha>0} \beta/\alpha + \alpha = 2\sqrt{\beta}$. □

C. Additional proofs for Section 4.1

Proof of Lemma 4.1. We first show that y^* is optimal for (Smooth-D). Since y^* is feasible for (Smooth-D) by assumption, we have that $g^*_{\alpha,C} \leq g(y^*) = g^*_C$. Since (Smooth-D) is obtained from (Nesterov-D) by adding constraints, we have that $g^*_C \leq g^*_{\alpha,C}$. Therefore $g^*_C = g^*_{\alpha,C}$ and y^* is optimal for both problems.

Since $\sqrt{x} \leq h_\beta(x)$ for all $x \geq 0$, we have that

$$f^*_C = f(\operatorname{diag}(W^*)) \leq f_\alpha(\operatorname{diag}(W^*)) \leq f^*_{\alpha,C} = g^*_{\alpha,C} = g^*_C = f^*_C.$$

We conclude that all these quantities are equal. □

Proof of Lemma 4.4. If z^* is optimal for (SDP-D) then, by Lemma 3.8, we have that

$$y^*_i = \frac{\langle \mathbf{1}, z^*\rangle^{1/2}}{z^*_i} = \frac{(F^*_C)^{1/2}}{z^*_i} \quad \text{for } i = 1, 2, \ldots, n.$$

The result then follows from the bound $z^*_i \geq C_{ii}$ from Lemma 3.4. □

C.1. Proof of Lemma 4.2

Before giving the proof, we establish some preliminary results. Let $H_\alpha : \mathbb{R}^n \to \mathbb{R}$ be defined by

$$H_\alpha(x) = \begin{cases} \|x\|_2 & \text{if } \|x\|_2 > \frac{1}{2\alpha} \\ \alpha\|x\|_2^2 + \frac{1}{4\alpha} & \text{if } \|x\|_2 \leq \frac{1}{2\alpha}. \end{cases} \tag{12}$$

This function is of interest because it has the property that

$$H_\alpha(x) = h_\alpha(\|x\|^2)$$

where h_α is the smoothing of the square root function defined in (10).

Lemma C.1. *The convex conjugate of H_α is the extended real-valued function given by*

$$H^*_\alpha(w) = \begin{cases} \frac{\|w\|_2^2 - 1}{4\alpha} & \text{if } \|w\|_2 \leq 1 \\ \infty & \text{otherwise.} \end{cases} \tag{13}$$

Proof. We need to compute

$$H_\alpha^*(w) = \sup_x \langle w, x\rangle - H_\alpha(x).$$

If $\|w\|_2 > 1$ then consider $x = \tau w$ for some $\tau > 1/(2\alpha)$. Then $\|x\|_2 > 1/(2\alpha)$ and so

$$\langle x, w\rangle - H_\alpha(x) = \tau\|w\|_2^2 - \tau\|w\|_2 = \tau\|w\|_2(\|w\|_2 - 1) \to \infty \quad \text{as } \tau \to \infty.$$

This shows that $H_\alpha^*(w) = \infty$ if $\|w\|_2 > 1$.

Assume $\|w\|_2 \le 1$. The first-order optimality conditions give

$$w = \nabla H_\alpha(x) = \begin{cases} \frac{x}{\|x\|_2} & \text{if } \|x\|_2 > \frac{1}{2\alpha} \\ 2\alpha x & \text{if } \|x\|_2 \le \frac{1}{2\alpha}. \end{cases}$$

If $\|w\|_2 \le 1$ then $x = w/2\alpha$ satisfies the first-order optimality conditions. Substituting back into the objective gives

$$H_\alpha^*(w) = \|w\|^2/(2\alpha) - \alpha\|w\|_2^2/(4\alpha^2) - \frac{1}{4\alpha} = \frac{\|w\|_2^2 - 1}{4\alpha}.$$

□

Proof of Lemma 4.2. Suppose that W is feasible for (Smooth-P). Since $W \succeq 0$ it has a representation as $W = R^T R$. Let $r_1, \ldots, r_n$ denote the columns of R. Wince $W = A^T P A$ for some $P \succeq 0$ with $\text{tr}(P) = 1$, it follows that $r_i = P^{1/2} A_i$ where $\|P^{1/2}\|_F = \text{tr}(P)^{1/2} = 1$. Then we have

$$f_\alpha(\text{diag}(W)) = \sum_{i=1}^n h_{\alpha_i}(W_{ii}) = \sum_{i=1}^n H_{\alpha_i}(r_i)$$

where H_α is defined in (12).

For $i = 1, 2, \ldots, n$ let

$$w_i(r_i) = \arg\max_w \langle r_i, w\rangle - H_\alpha^*(w) = \begin{cases} 2\alpha r_i & \text{if } \|r_i\|_2 \le 1/(2\alpha) \\ r_i/\|r_i\|_2 & \text{if } \|r_i\|_2 > 1/(2\alpha). \end{cases}$$

Then

$$H_{\alpha_i}(r_i) = \langle w_i(r_i), r_i\rangle - H_{\alpha_i}^*(w_i(r_i)).$$

Therefore

$$\begin{aligned}
f_\alpha(\text{diag}(W)) &= \sum_{i=1}^n \langle w_i(r_i), r_i\rangle - H_{\alpha_i}^*(w_i(r_i)) \\
&= \sum_{i=1}^n \left\langle w_i(r_i), P^{1/2} A_i \right\rangle - H_{\alpha_i}^*(w_i(r_i)) \\
&\le \sup_{\|Y\|_F = 1} \left\langle \sum_{i=1}^n w_i(r_i) A_i^T, Y \right\rangle - H_{\alpha_i}^*(w_i(r_i)) \quad (\text{since } \|P^{1/2}\|_F = 1) \\
&= \left\| \sum_{i=1}^n w_i(r_i) A_i^T \right\|_F + \sum_{i=1}^n \frac{1 - \|w_i(r_i)\|_2^2}{4\alpha_i} \quad (\text{since } \sup_{\|Y\|_F=1} \langle X, Y\rangle = \|X\|_F) \\
&= \left[\text{tr}\left(\sum_{i,j=1}^n w_i(r_i) A_i^T A_j w_j(r_i)^T \right) \right]^{1/2} + \sum_{i=1}^n \frac{1 - \|w_i(r_i)\|_2^2}{4\alpha_i} \\
&= \langle C, M\rangle^{1/2} + \sum_{i=1}^n \frac{1 - \|w_i(r_i)\|_2^2}{4\alpha_i}
\end{aligned}$$

where M is the positive semidefinite matrix with $M_{ij} = \langle w_i(r_i), w_j(r_j)\rangle$. Note that $M = \text{diag}^\star(d)A^T PA\text{diag}^\star(d) = \zeta_\alpha(W) - \text{diag}^\star(\delta)$. Therefore

$$f_\alpha(\text{diag}(W)) \leq \langle C, \zeta_\alpha(W) - \text{diag}^\star(\delta)\rangle^{1/2} + \sum_{i=1}^n \frac{\delta_i}{4\alpha_i}. \tag{14}$$

Since $\sqrt{x}$ is concave, $\sqrt{x} \leq \sqrt{a} + \frac{x-a}{2\sqrt{a}}$ for all $x, a > 0$. Applying this with $a = \langle C, \zeta_\alpha(W)\rangle$ and $x = \langle C, \zeta_\alpha(W)\rangle - \langle C, \text{diag}^\star(\delta)\rangle$ gives

$$(\langle C, \zeta_\alpha(W)\rangle - \langle C, \text{diag}^\star(\delta)\rangle)^{1/2} + \sum_{i=1}^n \frac{\delta_i}{4\alpha_i} \leq \langle C, \zeta_\alpha(W)\rangle^{1/2} - \frac{1}{2\langle C, \zeta_\alpha(W)\rangle^{1/2}} \sum_{i=1}^n C_{ii}\delta_i + \sum_{i=1}^n \frac{\delta_i}{4\alpha_i}. \tag{15}$$

Now, by assumption, $\alpha_i \geq \frac{(F_C^*)^{1/2}}{C_{ii}}$ for $i = 1, 2, \ldots, n$. Furthermore, since $\zeta_\alpha(W)$ is positive semidefinite with unit diagonal, it is feasible for (SDP-P) and so $\langle C, \zeta_\alpha(W)\rangle \leq F_C^*$. Substituting these two inequalities in (15) gives

$$\langle C, \zeta_\alpha(W)\rangle^{1/2} - \frac{1}{2\langle C, \zeta_\alpha(W)\rangle^{1/2}} \sum_{i=1}^n C_{ii}\delta_i + \sum_{i=1}^n \frac{\delta_i}{4\alpha_i} \leq \langle C, \zeta_\alpha(W)\rangle^{1/2} - \frac{1}{4(F_C^*)^{1/2}} \sum_{i=1}^n C_{ii}\delta_i \leq \langle C, \zeta_\alpha(W)\rangle^{1/2}. \tag{16}$$

Combining (14), (15), and (16) completes the proof. □

D. Additional proofs for Section 4.2

Proof of Lemma 4.6. First, note that

$$\text{diam}_{\|\cdot\|_z}(\mathcal{S}) = \sup_{x_1, x_2 \in \mathcal{S}} \|x_1 - x_2\|_z \leq 2 \sup_{x \in \mathcal{S}} \|x\|_z.$$

So it is enough to show that $\sup_{x\in\mathcal{S}} \|x\|_z \leq 1$. By the definition of $\mathcal{S}$, if $x \in \mathcal{S}$ then there exists $P \succeq 0$ with $\text{tr}(P) = 1$ such that $x = \text{diag}(A^T PA)$. Now,

$$\begin{aligned}
\|x\|_z &= \sum_{i=1}^n z_i^{-1} |A_i^T PA_i| \\
&= \left\langle P, \sum_{i=1}^n z_i^{-1} A_i A_i^T \right\rangle \\
&\leq \text{tr}(P)\lambda_{\max}(A\text{diag}^\star(z)^{-1}A^T) \\
&= \lambda_{\max}(A\text{diag}^\star(z)^{-1}A^T).
\end{aligned}$$

Now, since z is feasible for (SDP-D), we have that $\text{diag}^\star(z) \succeq C = A^T A$. It follows that $I \succeq \text{diag}^\star(z)^{-1/2}A^T A\text{diag}^\star(z)^{-1/2}$ which is equivalent to $\lambda_{\max}(\text{diag}^\star(z)^{-1/2}A^T A\text{diag}^\star(z)^{-1/2}) \leq 1$. Since $\lambda_{\max}(X^T X) = \lambda_{\max}(XX^T)$, it follows that $\lambda_{\max}(A\text{diag}^\star(z)^{-1}A^T) \leq 1$, as required. □

Proof of Lemma 4.7. First we note that $h_\alpha(x)$ (defined in (10)) has first derivative

$$h'_\alpha(x) = \begin{cases} \alpha & \text{if } 0 < x \leq 1/(4\alpha^2) \\ (1/2)x^{-1/2} & \text{if } x \geq 1/(4\alpha^2) \end{cases}$$

and second derivative (defined except at $x = 1/(4\alpha^2)$)

$$h''_\alpha(x) = \begin{cases} 0 & \text{if } 0 < x < 1/(4\alpha^2) \\ -(1/4)x^{-3/2} & \text{if } x > 1/(4\alpha^2). \end{cases}$$

It follows that $\sup_{x>0} |h''_\alpha(x)| = (1/4)(4\alpha^2)^{3/2} = 2\alpha^3$. Therefore, by the mean value theorem, we have that

$$|h'_\alpha(x_1) - h'_\alpha(x_2)| \leq 2\alpha^3 |x_1 - x_2|$$

for all $x_1, x_2 > 0$.

The dual norm of $\|\cdot\|_z$ is

$$\|x\|_{z,*} = \max_i z_i |x_i|.$$

We now consider the Lipschitz constant of ∇f_α. First, note that

$$\frac{\partial f_\alpha}{\partial x_i} = h'_{\alpha_i}(x_i).$$

Let $x, w \in \mathcal{S} \subseteq \mathbb{R}^n_+$, be chosen arbitrarily. Then

$$\begin{aligned}
\|\nabla f_\alpha(x) - \nabla f_\alpha(w)\|_{z,*} &= \max_i z_i |h'_{\alpha_i}(x_i) - h'_{\alpha_i}(w_i)| \\
&\leq \max_i z_i 2\alpha_i^3 |x_i - w_i| \\
&\leq \left(\max_i 2\alpha_i^3 z_i^2\right) \sum_{j=1}^n z_j^{-1} |x_j - w_j| \\
&= L\|x - w\|_z.
\end{aligned}$$

□

E. Additional proofs for Section 4.3

Proof of Theorem 4.9. We substitute z_μ and C_μ into Theorem 4.8. It is straightforward to check that $\langle \mathbf{1}, z_\mu \rangle = 3 \langle \mathbf{1}, z \rangle$. Furthermore, we have that

$$\begin{aligned}
\frac{\langle \mathbf{1}, z_\mu \rangle}{[C_\mu]_{ii}} &= 3 \frac{\langle \mathbf{1}, z \rangle}{C_{ii} + z_i + \frac{1}{n} \langle \mathbf{1}, z \rangle} \leq 3n \quad \text{and} \\
\frac{[z_\mu]_{ii}}{[C_\mu]_{ii}} &= \frac{2z_i + \frac{1}{n} \langle \mathbf{1}, z \rangle}{C_{ii} + z_i + \frac{1}{n} \langle \mathbf{1}, z \rangle} \leq \frac{2z_i + \frac{2}{n} \langle \mathbf{1}, z \rangle}{z_i + \frac{1}{n} \langle \mathbf{1}, z \rangle} = 2.
\end{aligned}$$

□

Proof of Theorem 4.10. From our choice of $\epsilon' = \frac{3\epsilon}{8\sqrt{3}} \frac{F_{LB}}{\langle \mathbf{1}, z \rangle^{1/2}}$ and $\delta' = \epsilon'/3$ it follows from the properties of the stopping criterion discussed in Section 2.2 that

$$f^*_{\alpha_\mu, C_\mu} - f_{\alpha_\mu}(\text{diag}(W)) \leq \frac{\epsilon}{2\sqrt{3}} \frac{F_{LB}}{\langle \mathbf{1}, z \rangle^{1/2}} \leq \frac{\epsilon}{2\sqrt{3}} \frac{F^*_C}{\langle \mathbf{1}, z \rangle^{1/2}}.$$

Since α_μ is C_μ-valid, we have that $f^*_{\alpha_\mu, C_\mu} = f^*_{C_\mu} = (F^*_{C_\mu})^{1/2}$.

Multiplying both sides of the inequality by

$$(F^*_{C_\mu})^{1/2} + f_{\alpha_\mu}(\text{diag}(W)) \leq 2(F^*_{C_\mu})^{1/2} \leq 2 \langle \mathbf{1}, z_\mu \rangle^{1/2} = 2\sqrt{3} \langle \mathbf{1}, z \rangle^{1/2}$$

gives

$$F^*_{C_\mu} - (f_{\alpha_\mu}(\text{diag}(W)))^2 \leq \epsilon F^*_C.$$

Since α_μ is C_μ-valid we have that

$$\left\langle C_\mu, \zeta_{\alpha_\mu}(W) \right\rangle \geq (f_{\alpha_\mu}(\text{diag}(W)))^2.$$

Let X^* be optimal for (SDP-P). Then

$$\langle C, X^* \rangle - \left\langle C, \zeta_{\alpha_\mu}(W) \right\rangle = \langle C_\mu, X^* \rangle - \left\langle C_\mu, \zeta_{\alpha_\mu}(W) \right\rangle \leq F^*_{C_\mu} - (f_{\alpha_\mu}(\text{diag}(W)))^2 \leq \epsilon\, F^*_C,$$

as required.

To bound the number of iterations, we use Lemma 2.2 and Theorem 4.9. Indeed, we know that the stopping criterion holds after at most

$$T = \frac{\frac{27}{2}\mathcal{M}(f_{\alpha_\mu}|\mathcal{S}_{C_\mu})}{\epsilon' - \delta'}$$

iterations. Substituting in the bound on the curvature constant from Theorem 4.9 and using $\epsilon' - \delta' = \frac{\epsilon}{4\sqrt{3}}\frac{F_{LB}}{\langle \mathbf{1}, z\rangle^{1/2}}$, gives

$$T \leq \frac{32 \cdot 3\sqrt{3} \cdot 4\sqrt{3}}{\epsilon} \frac{\langle \mathbf{1}, z\rangle}{F_{LB}}$$

iterations. □

Proof of Lemma 4.11. Let $A_\mu^T A_\mu = C_\mu$ and note that

$$\begin{aligned} h_{\mathcal{S}_{C_\mu}}(y) &= \left\| A_\mu \text{diag}^\star(y) A_\mu^T \right\| \\ &= \left\| \text{diag}^\star(y)^{1/2} C_\mu \text{diag}^\star(y)^{1/2} \right\|. \end{aligned}$$

For simplicitly of notation let $d_\mu = \text{diag}(C_\mu)$, and note that $d_\mu \geq \mu \geq z$.

Now, since $0 < \nabla f_{\alpha_\mu}(x) \leq \alpha_\mu$ it follows that

$$\begin{aligned} h_{\mathcal{S}_{C_\mu}}(\nabla f_{\alpha_\mu}(x)) &\leq h_{\mathcal{S}_{C_\mu}}(\alpha_\mu) \\ &= \langle \mathbf{1}, z_\mu \rangle \left\| \text{diag}^\star(d_\mu)^{-1/2} C_\mu \text{diag}^\star(d_\mu)^{-1/2} \right\| \\ &= \sqrt{3} \langle \mathbf{1}, z \rangle \left\| \text{diag}^\star(d_\mu)^{-1/2} C_\mu \text{diag}^\star(d_\mu)^{-1/2} \right\| \end{aligned}$$

where the last equality holds because $\langle \mathbf{1}, z_\mu \rangle = 3 \langle \mathbf{1}, z \rangle$.

Now

$$C_\mu = C + \text{diag}^\star(\mu) \preceq \text{diag}^\star(z + \mu) \preceq 2\text{diag}^\star(d_\mu)$$

Therefore

$$\text{diag}^\star(d_\mu)^{-1/2} C_\mu \text{diag}^\star(d_\mu)^{-1/2} \preceq 2I,$$

completing the proof. □

Lemma E.1. *Suppose that C is positive semidefinite with* $\text{diag}(C) > 0$. *Assume, in addition, that C is sparse with respect to a graph $G = (V, E)$ with maximum degree Δ, i.e., $C_{ij} = 0$ if $(i, j) \notin E$. If $z = (\Delta + 1)\text{diag}(C)$ then $C \preceq \text{diag}^\star(z)$.*

Proof. Let $d = \text{diag}(C)$ and consider $\tilde{C} = \text{diag}^\star(d)^{-1/2} C \text{diag}^\star(d)^{-1/2}$. The entries of $\tilde{C}$ are

$$\tilde{C}_{ij} = \begin{cases} 1 & \text{if } i = j \\ \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}} & \text{if } i \neq j. \end{cases}$$

Since $C \succeq 0$, we have that every 2×2 principal minor is nonnegative, and so that $C_{ii}C_{jj} \geq C_{ij}^2$ for all $i \neq j$. Now consider the matrix

$$(\Delta + 1)I - \tilde{C} = \text{diag}^\star(d)^{-1/2}(\text{diag}^\star(z) - C)\text{diag}^\star(d)^{-1/2}.$$

Let Δ_j denote the degree of the jth vertex of G. Then, for any j, we have

$$\sum_{i \neq j} |\tilde{C}_{ij}| \leq \Delta_j \leq \Delta = [(\Delta + 1)I - \tilde{C}]_{jj}.$$

This shows that $(\Delta + 1)I - \tilde{C}$ is diagonally dominant and so that $C \preceq \text{diag}^\star(z)$. □

F. Correctness of LMO implementation without factorizing C

Lemma F.1. *Let $A \in \mathbb{R}^{n\times m}$ be such that $C = A^T A$. Let $y > 0$ be a positive vector. Suppose that u is a unit vector such that*

$$u^T \text{diag}^\star(y)^{1/2} C \text{diag}^\star(y)^{1/2} u \geq \lambda_{\max}(\text{diag}^\star(y)^{1/2} C \text{diag}^\star(y)^{1/2}) - \delta$$

and

$$v = \frac{C\text{diag}^\star(y)^{1/2} u}{(u^T \text{diag}^\star(y)^{1/2} C \text{diag}^\star(y)^{1/2} u)^{1/2}}.$$

Then $vv^T \in \tilde{\mathcal{S}}_C$ and

$$v^T \text{diag}^\star(y) v \geq u^T (\text{diag}^\star(y)^{1/2} C \text{diag}^\star(y)^{1/2} u.$$

Proof. First we check that $vv^T \in \tilde{\mathcal{S}}_C$. We can write $vv^T = A^T P A$ where P is the unit trace positive semidefinite matrix given by

$$P = \frac{A\text{diag}^\star(y)^{1/2} uu^T \text{diag}^\star(y)^{1/2} A^T}{\text{tr}(A\text{diag}^\star(y)^{1/2} uu^T \text{diag}^\star(y)^{1/2} A^T)}.$$

To see that $vv^T = A^T P A$ we observe that $A^T A = C$ and that

$$\text{tr}(A\text{diag}^\star(y)^{1/2} uu^T \text{diag}^\star(y)^{1/2} A^T) = u^T \text{diag}^\star(y)^{1/2} C \text{diag}^\star(y)^{1/2} u.$$

Now, let $M = \text{diag}^\star(y)^{1/2} C \text{diag}^\star(y)^{1/2}$. Our aim is to show that $v^T \text{diag}^\star(y) v \geq u^T M u$. This can be reformulated as

$$v^T \text{diag}^\star(y) v = \frac{u^T M^2 u}{u^T M u} \geq u^T M u \quad \iff \quad (u^T M^2 u)^{1/2} \geq u^T M u.$$

Let $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$ denote the eigenvalues of M with corresponding eigenvectors $w_1, \ldots, w_n$. Since the w_i form an orthonormal basis for $\mathbb{R}^n$, we have that

$$\sum_{i=1}^n (u^T w_i)^2 = \left\langle uu^T, \sum_{i=1}^n w_i w_i^T \right\rangle = \left\langle uu^T, I \right\rangle = 1.$$

So the $(u^T w_i)^2$ are non-negative and sum to one. By the concavity of the square root,

$$(u^T M^2 u)^{1/2} = \left[\sum_{i=1}^n \lambda_i^2 (u^T w_i)^2 \right]^{1/2} \geq \sum_{i=1}^n (u^T w_i)^2 (\lambda_i^2)^{1/2} = u^T M u,$$

completing the proof. □

G. Additional experimental results

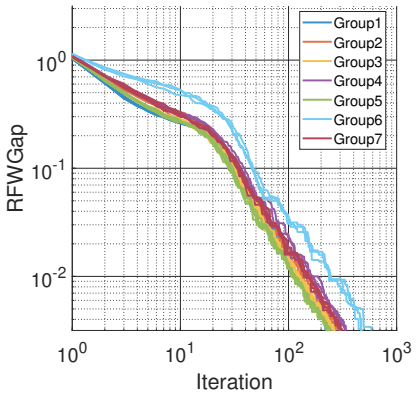

Figure 3. Plot of $\min_{\tau \leq t} \text{RFWgap}(x^{(\tau)})$ vs Frank-Wolfe iterations for our algorithm applied to Gset graphs with scheduled step size. Each line represents a single graph. The lines are colored by group membership.

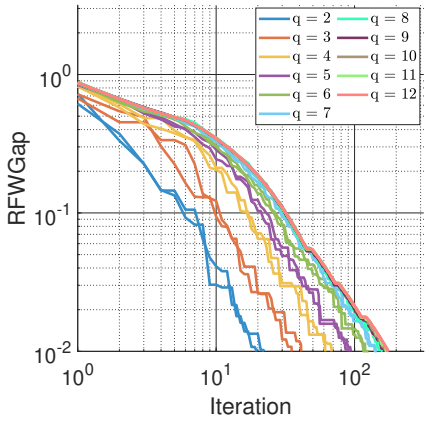

Figure 4. Plot of $\min_{\tau \leq t} \text{RFWgap}(x^{(\tau)})$ vs Frank-Wolfe iterations for our algorithm applied to 5-regular graphs with $10^{q/2}$ vertices. Each line represents a single graph. Colors indicate graphs with the same parameters.

Table 2. Iteration count and relative error comparison between our method and the methods of Shinde et al. (2021) and Yurtsever et al. (2021) for Gset graphs. The iteration count for each graph is highlighted in bold.

Graph	Graph parameter		Yurtsever et al.		Shinde et al.		This paper	
	n	m	Iter	Rel Err	Iter	Rel Err	Iter	Rel Err
G1	800	19176	**149**	0.007	25285	0.025	156	0.005
G2	800	19176	139	0.007	25027	0.024	**122**	0.007
G3	800	19176	**123**	0.008	53597	0.039	149	0.005
G4	800	19176	**141**	0.008	23282	0.022	146	0.005
G5	800	19176	**139**	0.008	25155	0.024	162	0.004
G14	800	4694	897	0.024	186125	0.049	**73**	0.017
G15	800	4661	910	0.024	213059	0.052	**69**	0.021
G16	800	4672	933	0.019	180269	0.047	**91**	0.013
G17	800	4667	915	0.024	195772	0.047	**83**	0.015
G22	2000	19990	**118**	0.008	359679	0.045	172	0.004
G23	2000	19990	**119**	0.007	127076	0.040	149	0.005
G24	2000	19990	**123**	0.007	114487	0.040	155	0.004
G25	2000	19990	**127**	0.007	75340	0.037	156	0.004
G26	2000	19990	**147**	0.008	95181	0.039	146	0.005
G35	2000	11778	2245	0.019	600032	0.048	**119**	0.011
G36	2000	11766	1919	0.018	730590	0.046	**93**	0.014
G37	2000	11785	2778	0.024	805089	0.047	**86**	0.016
G43	1000	9990	**122**	0.008	27416	0.029	133	0.005
G44	1000	9990	**127**	0.007	25333	0.029	143	0.004
G45	1000	9990	**116**	0.008	28319	0.030	118	0.006
G46	1000	9990	164	0.008	25574	0.027	**137**	0.005
G47	1000	9990	**122**	0.008	34344	0.033	132	0.005
G48	3000	6000	**153**	0.005	14145	0.013	381	0.004
G49	3000	6000	**148**	0.005	18803	0.016	387	0.004
G50	3000	6000	**136**	0.004	19304	0.017	455	0.003
G51	1000	5909	1102	0.023	327568	0.037	**89**	0.014
G52	1000	5916	979	0.018	139494	0.037	**103**	0.011
G53	1000	5914	1201	0.025	232629	0.038	**84**	0.015
G54	1000	5916	1317	0.024	270915	0.038	**92**	0.013