Approximating semidefinite packing programs *

G. Iyengar †, D. Phillips ‡, and C. Stein §

Abstract

In this paper we describe an algorithm to approximately solve *semidefinite packing programs*. This class includes many semidefinite programs that arise in the context of developing approximation algorithms for difficult combinatorial optimization problems. Our algorithm exploits the structural similarity between semidefinite packing programs and linear packing programs.

1 Introduction

Semidefinite programming (SDP) is an optimization problem of the form

$$\begin{array}{ll} \max & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{subject to} & \langle \mathbf{A}_i, \mathbf{X} \rangle \leq b_i, \quad i = 1, \ldots, p, \\ & \mathbf{X} \succeq \mathbf{0}, \end{array} \tag{1}$$

where $b_i \in \mathbb{R}, i = 1, \ldots, p$, $\mathbf{C} \in \mathbb{R}^{n \times n}$ and $\mathbf{A}_i \in \mathbb{R}^{n \times n}, i = 1, \ldots, p$ are all given non-zero scalers and matrices, and $\mathbf{X} \in \mathbb{R}^{n \times n}$ are the decision variables. We use $\langle ., . \rangle$ to denote the *Frobenius inner* product defined as follows:

$$\langle \mathbf{A}, \mathbf{B} \rangle = \mathbf{Tr}(\mathbf{A}^\top \mathbf{B}) = \sum_{j=1}^{n} \sum_{i=1}^{n} a_{ij} b_{ij}. \tag{2}$$

Also, we define

$$r := \text{number of non-zeros in } |\mathbf{C}| + \sum_{i=1}^{p} |\mathbf{A}_i|, \tag{3}$$

where $|\mathbf{C}|$ is the component-wise absolute value of the matrix $\mathbf{C}$. Thus, r is a measure of the sparsity in (1). The constraint $\mathbf{X} \succeq \mathbf{0}$ indicates that the matrix $\mathbf{X}$ is *positive semidefinite*, i.e., $\mathbf{X}$ is symmetric and has nonnegative eigenvalues, or, equivalently, $\mathbf{v}^\top \mathbf{X} \mathbf{v} \geq 0$ for all $\mathbf{v} \in \mathbb{R}^n$. We similarly write $\mathbf{X} \succ \mathbf{0}$ when $\mathbf{X}$ is positive definite, i.e., $\mathbf{X}$ is symmetric and has positive eigenvalues, or, equivalently, $\mathbf{v}^\top \mathbf{X} \mathbf{v} > 0$ for all $\mathbf{v} \in \mathbb{R}^n$. We also write that $\mathbf{A} \succeq \mathbf{B}$ ($\mathbf{A} \succ \mathbf{B}$) when $\mathbf{A} - \mathbf{B} \succeq \mathbf{0}$ ($\mathbf{A} - \mathbf{B} \succ \mathbf{0}$). Note that $\succ$ and $\succeq$ induce a *partial ordering* on symmetric matrices (see, e.g., Horn and Johnson [16]).

For general SDPs, violated inequalities can be determined in polynomial time, so the ellipsoid method of Yudin and Nemirovsky [38, 26], and Shor [32] can approximately solve an SDP in polynomial time. However, the ellipsoid method is only useful as a theoretical tool, since practical implementations have not proven efficient. Fortunately, a computable self-concordant barrier exists for SDP, and so interior point methods can also compute an approximately optimal solution in polynomial time. All known interior point methods share the property that they can solve for a solution within an *additive* constant ϵ of optimality in time proportional to $\log(\frac{1}{\epsilon})$, but they typically do not exploit any sparsity in the matrices $\mathbf{C}$ and $\mathbf{A}_i, i = 1, \ldots, p$. For example,

*A preliminary version of this paper appeared in the proceedings of the Eleventh Conference on Integer Programming and Combinatorial Optimization [17].

†The Department of Industrial Engineering & Operations Research, Columbia University, garud@ieor.columbia.edu

‡Mathematics Department, The College of William & Mary, phillips@math.wm.edu

§The Department of Industrial Engineering & Operations Research, Columbia University, cliff@ieor.columbia.edu

on the general SDP described in (1), the complexity of the barrier method is $O(\log(\epsilon^{-1})n^{1/2}(pn^3+p^2n^2+p^3)$ (see Nesterov and Nemirovsky [28]). Practical implementations of interior point algorithms exist, and SDP has been used in a variety of fields to help model and solve different problems (see, e.g., Vandenberghe and Boyd [35]). We are interested in solving SDPs that arise in the context of approximation algorithms for hard combinatorial optimization problems.

Using SDP in combinatorial optimization is an idea that started in the early 1970s. Donath and Hoffman [8] and Cullum, Donath, and Wolfe [7] examined the connection between eigenvalue optimization and some hard graph-partitioning problems. Lovász [22] used SDP to compute the Shannon-capacity of a graph, which, with the work of Grötschel, Lovász, and Schrijver [14], led to the first polynomial-time algorithm for finding the largest stable set in a perfect graph. Beginning with the work of Goemans and Williamson [10], SDP has been used as a tool for approximating NP-hard optimization problems. In this case, an SDP relaxation of the original problem is formulated and solved, and then a rounding step is used to output a feasible and approximately optimal solution to the original problem. Karger, Motwani, and Sudan [18] use an SDP relaxation and rounding strategy to approximate graph coloring, Skutella [33] uses an SDP to approximately solve a scheduling problem, and recently, Arora, Rao and V. Vazirani [2] use SDP to approximate graph partitioning problems. In all of these problems, using the SDP relaxation yields better approximation bounds than using a linear programming relaxation or combinatorial techniques. However, solving the SDP relaxations is the key bottleneck in the computational complexity of these algorithms.

Since the SDPs used in algorithms for combinatorial optimization possess considerable structure, specialized algorithms to solve specific SDPs have been developed. The dual interior point method of Benson et al [3] was designed to take advantage of sparsity and improved the running time to $O(\log(\frac{1}{\epsilon}n^{1/2}(p^3+np^2+n^2p+n^3)))$ (see also Burer and Monteiro [6]). However, for MAXCUT, both the dual and the standard barrier interior point method has a worst case running time of $O(\log(\epsilon^{-1})n^{3.5})$ when used to solve the MAXCUT SDP of a graph with n nodes and m edges [28]. Note that the number of constraints in the MAXCUT SDP is n, i.e., $p = n$, and that the barrier interior point method does not take advantage of the sparsity of the graph, i.e., how large the number of edges, m, is relative to n.

In order to improve the worst-case complexity, Klein and Lu [19] developed algorithms to solve SDPs for MAXCUT and coloring that have a better running time dependence on n and m. Their methods build on approximation algorithms for *vector packing and covering programs* (see Plotkin et al [30] and Grigoriadis and Khachiyan [13]) to solve the MAXCUT [10] and graph coloring SDPs [18]. The Klein-Lu algorithm can be described as a Lagrangian penalty method where, at each iteration, an improving direction is found by computing the optimal solution to the eigenvalue problem, $\min\{\mathbf{C}^\top\mathbf{X} : \mathbf{X} \succeq \mathbf{0} : \mathbf{diag}(\mathbf{X}) \leq \mathbf{1}\}$. The Klein-Lu algorithm computes a solution to the MAXCUT SDP to within a $(1-\epsilon)$ multiplicative factor of the true optimum in $O(\epsilon^{-2}\log(\epsilon^{-1})nm)$ time. Compared to the interior point method, the Klein-Lu algorithm significantly improves running time dependence on n and m at a cost of increased dependence on the approximation constant, ϵ. In particular, the Klein-Lu algorithm is able to employ linear algebraic methods that take advantage of sparsity in the underlying graph. Using their algorithm for the MAXCUT SDP as a subroutine, Klein and Lu also describe an algorithm to solve the SDP relaxation for the graph coloring problem. Recently, Arora and Kale [1] have described algorithms that approximately solve the SDP relaxations of the Sparsest Cut, Balanced Seperator and Minimum UnCut problems to within a $(1+\epsilon)$ multiplicative factor of optimality. The Arora-Kale algorithms use a similar Lagrangian penalty method as Klein and Lu and have a similar running time of $O(\epsilon^{-2}p(n,m))$ where $p(n,m)$ is a problem dependent polynomial in n and m. We emphasize that both of these algorithms attain much of their improvement by shifting the running time dependence from requiring fully dense matrices, n^2, to the number of edges, m (i.e., sparsity).

The dependence on ϵ of both the Klein-Lu and the Arora-Kale algorithms is $O(\epsilon^{-2})$. In fact, their algorithms are lower bounded by $\Omega(\epsilon^{-2})$ via a simple extension (see Phillips [29]) of the Klein-Young lower bound [20]. The Klein-Young lower bound applies to algorithms which require an oracle that returns extreme point solutions to the problem $\min\{\mathbf{c}^\top\mathbf{x} : \mathbf{x} \in \mathcal{P}\}$ where $\mathcal{P}$ is a fixed polyhedron and $\mathbf{c}$ changes in each iteration. The extension in [29] shows how this result also applies to algorithms that replace $\mathcal{P}$ with a fixed spectrahedra such as the feasible region of the eigenvalue problem used in the Klein-Lu algorithm.

The difference between the worst case running times of interior point methods and the Lagrangian

approaches of Klein and Lu [19] and Arora and Kale [1] is the tradeoff between accuracy and dependence on n, p and r (see (3)). In particular, the Lagrangian approaches generally have a better dependence on n by leveraging sparsity, as can be seen in Figure 1 for MAXCUT and coloring. However, the Lagrangian approaches have a dependence on ϵ of $O(\epsilon^{-2})$ as opposed to the $O(\log(\epsilon^{-1}))$ of interior point. As shown in Figure 1, our algorithm, **SDP-Pack**, fills the accuracy-to-sparsity tradeoff "gap" in the sense that it leverages sparsity to have a better running time dependence on n, p and r than interior point, but has a dependence on ϵ of $O(\epsilon^{-1})$. However, the running time dependence on n, p and r of our method is not as good as Klein-Lu, and the dependence on ϵ is not as good as interior point. **SDP-Pack** is also a Lagrangian penalty approach, but the Klein-Young lower bound does *not* apply to it.

We avoid the conditions of the Klein-Young lower bound by extending methods developed for *subgradient algorithms*, which are methods in which a non-differentiable, convex function is minimized using a gradient-like method (such as steepest descent). Since non-differentiable functions have no gradients, a *subgradient* of the function is used instead, generally with the property that the subgradient represents an "improving" direction for the function at any particular point. Lagrangian penalty approaches such as Klein-Yu [19] and Arora-Kale [1] are examples of subgradient algorithms for SDP (see Bienstock [4]). More general subgradient algorithms can compute a solution to non-differentiable convex minimization problems with an objective value within an additive ϵ of the optimal solution in $O(\epsilon^{-2})$ time, although the running times of these general methods are typically not polynomial in the input length of the problem. Nemirovski and Yudin [26] proved a $\Omega(\epsilon^{-2})$ lower bound on the running time of subgradient methods that assume the only about information known about the objective function is function evaluation and subgradients at given points. Note that the Klein-Young result (and its extension to SDP) is concerned with *relative* approximation, i.e., computing solutions within a multiplicative factor ϵ of optimality, for the specific problems of conic optimization, whereas the lower bound of Nemirovski and Yudin is on *absolute* approximation, i.e., computing solutions within an additive factor ϵ of optimality, for the more general problem of non-differentiable, convex optimization. Both lower bounds are established by constructing relatively simple problems.

Our results

A breakthrough by Nesterov [27] and, independently, by Nemirovski [25], resulted in subgradient algorithms to approximately solve *saddle point* problems of the form:

$$\min_{\mathbf{x}} \max_{\mathbf{v}} \{f(\mathbf{x}) - \mathbf{v}^\top \mathbf{L}\mathbf{x} + d(\mathbf{v}) : \mathbf{x} \in \mathcal{X}, \mathbf{v} \in \mathcal{V}\}, \tag{4}$$

where f and $-d$ are strongly convex and differentiable, $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{V} \subseteq \mathbb{R}^m$ and $\mathbf{L} \in \mathbb{R}^{m \times n}$. The algorithms of Nesterov and Nemirovski solve (4) within an additive error of ϵ and have a running time of $O(\epsilon^{-1} L^{1/2})$ where L represents the *Lipschitz constant* of the underlying objective function. Roughly speaking, L represents the maximum distance between points in the feasible region of the saddle-point problem, and thus, Nesterov's and Nemirovski's algorithms are not polynomial in the length of the input. Subsequently, Bienstock and Iyengar [5] adapted the Nesterov approach to develop algorithms that compute solutions for vector packing and covering programs that come within an ϵ *multiplicative* factor of optimality. Our algorithm represents an analog of the Bienstock and Iyengar approach in the semidefinite setting.

We obtain the following results:

- We introduce the natural SDP analog of the vector packing problem and show that it includes the SDP relaxations for MAXCUT [10], coloring [18], and Shannon capacity [22] as special cases.
- By adapting the technique of Nesterov [27], we develop an algorithm, **SDP-Pack**, that requires $O(n\epsilon^{-1})$ iterations, where each iteration requires the calculation of a *matrix exponential*. Note that our algorithm is *not* constrained by the extension of the Klein-Young lower bound.
- **SDP-Pack** has a worst-case running time that leverage sparsity in the underlying objective and constraint matrices.

Algorithm	MAXCUT	Coloring	Lovász ϑ-function
Interior point	$O(\log(\epsilon^{-1})n^{3.5})$	$O(\log(\epsilon^{-1})n^{.5}(m^3+n^3))$	$O(\log(\epsilon^{-1})n^{.5}(m^3+n^3))$
Klein-Lu	$O(\epsilon^{-2}\log(\epsilon^{-1})nm)$	$O(\epsilon^{-4}\log(\epsilon^{-1})nm)$	n/a
SDP-Pack	$O(\epsilon^{-1}\log^3(\epsilon^{-1})n^2(n+m)\log(n))$		$O(\epsilon^{-2}\log^3(\epsilon^{-1})n^2(n+m)\log(n))$

Figure 1: Running time of SDP solvers
n = number of nodes, m = number of edges

The running time of our algorithm, (dual) barrier interior point methods, and the specialized methods of Klein and Lu are shown in Figure 1. In particular, note that the running time for **SDP-Pack** is the same for both the MAXCUT and the Coloring SDP relaxations. Further, the running time of **SDP-Pack** depends on the number of edges in the graph rather than the number of constraints. For the MAXCUT case, this results in an improvement over the barrier interior point method when m is smaller than n^2. For the coloring problem, this represents an improvement over the barrier interior point method in both the sparse ($m = O(n)$) and "dense" case ($m = \Omega(n^2)$). Also, the dependence on the multiplicative approximation factor, ϵ, is improved over the Klein-Lu algorithm as **SDP-Pack** requires $O(\epsilon^{-1})$ iterations (ignoring log factors) for both MAXCUT and coloring whereas the Klein-Lu algorithm requires $O(\epsilon^{-2})$ iterations for MAXCUT, and $O(\epsilon^{-4})$ iterations for coloring. For the Lovász ϑ-function, our method requires $O(\epsilon^{-2})$ iterations in order to deal with the trace equality constraint (see Appendix D).

1.1 Notation and preliminaries

Throughout this paper, we denote vectors in lowercase bold, e.g., $\mathbf{x}$, scalars in italics, e.g., x, matrices in uppercase bold, e.g., $\mathbf{X}$, and sets in uppercase calligraphic font, e.g., $\mathcal{X}$. We overload this notation, when necessary, for objects that do not fit this categorization (e.g., tensors). We use $\mathbf{1}_n$ and $\mathbf{0}_n$ to denote n dimensional vectors of all ones and zeros respectively, and omit the subscript n when the dimension is clear. For a given matrix, $\mathbf{M} \in \mathbb{R}^{n\times n}$, we define $\lambda_k(\mathbf{M})$ as the kth largest eigenvalue, $\lambda_{\max}(\mathbf{M}) := \lambda_1(\mathbf{M})$ and $\lambda_{\min}(\mathbf{M}) = \lambda_n(\mathbf{M})$. For a given vector $\mathbf{v} \in \mathbb{R}^n$, we let $\|\mathbf{v}\|_1 = \sum_{i=1}^n |v_i|$, $\|\mathbf{v}\|_\infty = \max_{i=1,\ldots,n} |v_i|$, and $\|\mathbf{v}\| = \sqrt{\sum_{i=1}^n v_i^2}$, i.e., the ℓ_1, ℓ_∞ and ℓ_2 norms, respectively. We define the $\mathcal{L}_1, \mathcal{L}_\infty$, and $\mathcal{L}_2$ in the usual fashion [16]: For a $\mathbf{A} \in \mathbb{R}^{m\times n}$,

$$\|\mathbf{A}\|_1 = \max_{\mathbf{v}:\|\mathbf{v}\|_1=1} \|\mathbf{A}\mathbf{v}\|_1, \quad \|\mathbf{A}\|_\infty = \max_{\mathbf{v}:\|\mathbf{v}\|_\infty=1} \|\mathbf{A}\mathbf{v}\|_1, \quad \|\mathbf{A}\| = \max_{\mathbf{v}:\|\mathbf{v}\|=1} \|\mathbf{A}\mathbf{v}\|.$$

For a given SDP of form (1), we denote the components of the kth constraint matrix, $\mathbf{A}_k$ by $a_{ij}^{(k)}$ for $k = 1, \ldots, p$.

For an optimization problem on a given graph, G, and associated SDP relaxation of form (1), recall that we use m to denote the number of edges in G, p to denote the number of constraints in the SDP and r to denote the number of nonzero entries in the matrix $|\mathbf{C}| + \sum_{i=1}^p |\mathbf{A}_i|$ (as in (3)). In the combinatorial problems we examine, we often have the case where $p = O(m)$ and/or $r = O(m)$.

The following Lemma, proved in Appendix A, is necessary for the analysis of our algorithm.

Lemma 1. *Suppose $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a given symmetric matrix. Then*

$$\max\{\langle \mathbf{X}, \mathbf{A}\rangle : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq 1\} = \max\{0, \lambda_{\max}(\mathbf{A})\}. \tag{5}$$

If $\mathbf{A}$ has at least one positive eigenvalue, the solution to the optimization problem (5) is $\mathbf{X}^* = \mathbf{u}_1\mathbf{u}_1^\top$ where $\mathbf{u}_1$ is the eigenvector corresponding to the maximum eigenvalue of $\mathbf{A}$. If $\mathbf{A}$ has no positive eigenvalues, then the optimal solution to (5) is to set $\mathbf{X}^*$ to the zero matrix.

The linear programming analog to Lemma 1 is as follows: Let $\boldsymbol{\gamma} \in \mathbf{R}^n$. Then the following linear program has a simple solution:

$$\max_{\mathbf{y}} \Big\{ \sum_{i=1}^{n} \gamma_i y_i : \mathbf{y} \geq 0, \sum_{i=1}^{n} y_i \leq 1 \Big\} = \max \big\{ 0, \max_{1 \leq i \leq n} \{\gamma_i\} \big\}. \tag{6}$$

We formally define additive and multiplicative approximation. Given a function $h : \Theta \to \mathbb{R}$ that we wish to minimize, let $\mathbf{z}^*$ be the minimum-valued solution. We say that $\bar{\mathbf{z}}$ is ϵ-optimal in the *absolute* sense if $h(\bar{\mathbf{z}}) \leq h(\mathbf{z}^*) + \epsilon$, i.e. $h(\bar{\mathbf{z}})$ is within an *additive* error ϵ to the optimal value. We say that $\bar{\mathbf{z}}$ is ϵ-optimal in the *relative* sense if $h(\bar{\mathbf{z}}) \leq (1+\epsilon)h(\mathbf{z}^*)$, i.e. $h(\bar{\mathbf{z}})$ is within a $(1+\epsilon)$ *multiplicative* factor of the optimal value.

Lemma 2. *Let $h : \Theta \to \mathbb{R}$ and suppose $h(\mathbf{z}^*) = \min\{h(\mathbf{y}) : \mathbf{y} \in \Theta\}$. Suppose $h(\mathbf{z}^*) \geq C > 0$, and that $h(\bar{\mathbf{z}}) \leq h(\mathbf{z}^*) + \epsilon$. Then, $\bar{\mathbf{z}}$ has a relative error of at most $\frac{\epsilon}{C}$, i.e.,*

$$h(\bar{\mathbf{z}}) \leq (1 + \epsilon/C)h(\mathbf{z}^*). \tag{7}$$

Proof. $h(\bar{\mathbf{z}}) \leq h(\mathbf{z}^*) + \epsilon = h(\mathbf{z}^*) + C(\epsilon/C) \leq (1 + \epsilon/C)h(\mathbf{z}^*)$. □

2 The Packing SDP

In this section, we first define the Packing SDP and show it possesses strong duality. In Section 2.1 we show that several SDPs arising in the context of combinatorial optimization are Packing SDPs.

Definition 1. *The Packing SDP is an SDP of the form* (1) *that satisfies the following conditions:*

(a) The objective matrix $\mathbf{C} \succeq \mathbf{0}$.

(b) All right-hand sides are strictly positive, i.e., $b_i > 0, i = 1, \ldots, p$.

(c) All the constraint matrices $\mathbf{A}_i \succeq \mathbf{0}, i = 1, \ldots, p$.

(d) For all feasible $\mathbf{X}$, $\mathbf{Tr}(\mathbf{X}) \leq \tau$ for some $\tau > 0$, or, equivalently, there is a $\mathbf{v} \geq \mathbf{0}$ such that $\sum_{i=1}^{p} v_i \mathbf{A}_i \succeq \mathbf{I}$.

Note that (d) implies that the constraint $\langle \mathbf{A}_0, \mathbf{X} \rangle \leq 1$ is valid where $\mathbf{A}_0 = \frac{1}{\tau}\mathbf{I}$. Moreover, since $b_i > 0$ for all i, we assume, without loss of generality, that $b_i = 1$ for all i. Thus, the packing SDP can be written as follows,

$$\begin{array}{rlll} \rho^* = & \max & \langle \mathbf{C}, \mathbf{X} \rangle & \\ & \text{subject to} & \langle \mathbf{A}_i, \mathbf{X} \rangle \leq 1, & i = 0, \ldots, p, \\ & & \mathbf{X} \succeq \mathbf{0}, & \end{array} \tag{8}$$

where $\mathbf{A}_i, i = 0, \ldots, p$, satisfy condition (c), and $\mathbf{C}$ satisfies condition (a).

The dual to (8) is

$$\begin{array}{ll} \min & \sum_{i=0}^{p} v_i \\ & \sum_{i=0}^{p} v_i \mathbf{A}_i \succeq \mathbf{C}, \\ & v_i \geq 0, i = 0, \ldots, p, \end{array} \tag{9}$$

We note immediately that (8) and (9) satisfy *weak duality*, since, for feasible $\mathbf{X}$ and $\mathbf{v}$, we have that:

$$\langle \mathbf{C}, \mathbf{X} \rangle \leq \left\langle \sum_{i=0}^{p} v_i \mathbf{A}_i, \mathbf{X} \right\rangle = \sum_{i=0}^{p} v_i \langle \mathbf{A}_i, \mathbf{X} \rangle \leq \sum_{i=0}^{p} v_i.$$

Moreover, "well-behaved" SDPs also satisfy strong duality. One sufficient condition for strong duality to hold is for (8) to be *strictly feasible* (see Theorem 3.1 in Boyd and Vandenberghe [36] and also [28, 31]), i.e., there exists a matrix $\mathbf{X} \succ \mathbf{0}$ with $\langle \mathbf{A}_i, \mathbf{X} \rangle < 1, i = 0, \ldots, p$.

Lemma 3. *Suppose that a SDP is feasible, bounded and can be written in the form of* (8)*, i.e., satisfies conditions* (a)*,* (b)*, and* (c)*. Then* (8) *and* (9) *have the same objective value at optimality.*

Proof. Let $\mathbf{X} = \delta\mathbf{I}$ where $\delta = \frac{1}{2}\min_{i=0,\ldots,p}\{\frac{1}{\mathbf{Tr}(\mathbf{A}_i)}\}$. By definition, $\langle \mathbf{A}_i, \mathbf{X}\rangle < 1$ and $\mathbf{X} \succ \mathbf{0}$ and (8) has an interior point. Thus, strong duality exists by Theorem 3.1 in [36]. □

Note that the well-known *Slater condition* involves showing the existence of an interior point for both (9) as well as (8). However, consider the SDP with

$$\mathbf{C} = \mathbf{A}_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

where $n = 2$ and $p = 1$. Note that the dual is feasible for all $v_1 \geq 1$, but $\lambda_{\min}(v_1\mathbf{A}_1 - \mathbf{C}) = 0, \forall v_1 \geq 1$. Thus, no interior point exists for the dual. Such an SDP is *not* a Packing SDP, however, as it does not satisfy condition (d). Packing SDPs satisfy the Slater condition. Let $v_0 = 2\tau \max\left\{1, \lambda_{\min}(\sum_{i=1}^{p} \mathbf{A}_i - \mathbf{C})\right\}$ and $v_i = 1, i = 1, \ldots, p$. By definition, $\sum_{i=0}^{p} v_i\mathbf{A}_i \succ \mathbf{C}$ and $v_i > 0, i = 0, \ldots, p$. Thus, (9) satisfies the Slater condition.

2.1 Packing SDP in combinatorial optimization

We now show that the SDP relaxations associated with several combinatorial problems are Packing SDPs. Reformulating these problems as Packing SDPs is generally a matter of transforming some equality constraints to inequality constraints and "shifting" some matrices to positive semidefinite.

The MAXCUT SDP

The MAXCUT SDP of Goemans and Williamson [10] can be written as:

$$\begin{array}{ll} \max & \langle \mathbf{L}, \mathbf{X}\rangle \\ & \left\langle \mathbf{X}, (\mathbf{e}_i\mathbf{e}_i^\top)\right\rangle = 1, i = 1, \ldots, n, \\ & \mathbf{X} \succeq \mathbf{0}, \end{array} \tag{10}$$

where $\mathbf{L}$ is the Laplacian of the associated graph G and $\mathbf{e}_i$ is the ith column of the identity matrix. The Laplacian of a weighted graph with nonnegative edge weights $w_{ij}, (i,j) \in A$ is a matrix $\mathbf{L} = [\ell_{ij}]$ where

$$\ell_{ij} = \begin{cases} -w_{ij}, & i \neq j, \\ \sum_{k=1}^{n} w_{ik}, & i = j. \end{cases} \tag{11}$$

Then, (10) can be solved with the Packing SDP:

$$\begin{array}{ll} \max & \langle \mathbf{L}, \mathbf{X}\rangle \\ & \left\langle \mathbf{X}, (\mathbf{e}_i\mathbf{e}_i^\top)\right\rangle \leq 1, i = 1, \ldots, n, \\ & \mathbf{X} \succeq \mathbf{0}, \end{array} \tag{12}$$

Note that any feasible solution to (12) can be transformed into a feasible solution to (10) with a possibly increased objective. This is true, since a positive semidefinite matrix remains positive semidefinite after increasing the diagonal. Moreover, since $\mathbf{L} \succeq \mathbf{0}$, increasing the main diagonal of the solution can only increase the objective value.

The Lovász ϑ function SDP

In order to find the largest stable set, Lovász [22] defined the function $\vartheta(G)$ on a graph, $G = (V, A)$ to be the objective value at the optimal solution of the following SDP:

$$\begin{array}{rl} \max & \langle \mathbf{1}\mathbf{1}^\top, \mathbf{X} \rangle \\ \text{s.t.} & x_{ij} = 0, \quad (i,j) \in E, \\ & \mathbf{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}. \end{array} \tag{13}$$

We can solve (13) via:

$$\begin{array}{rl} \max & \langle \mathbf{1}\mathbf{1}^\top, \mathbf{X} \rangle \\ \text{s.t.} & \langle \mathbf{E}_{ij}, \mathbf{X} \rangle \leq 1, \quad (i,j) \in A, \\ & \langle \mathbf{F}_{ij}, \mathbf{X} \rangle \leq 1, \quad (i,j) \in A, \\ & \mathbf{Tr}(\mathbf{X}) = 1, \\ & \mathbf{X} \succeq \mathbf{0}, \end{array} \tag{14}$$

where

$$\mathbf{E}_{ij} = \mathbf{I} + [z^{ij}_{k\ell}], \quad \mathbf{F}_{ij} = \mathbf{I} - [z^{ij}_{k\ell}]$$

and

$$z^{ij}_{k\ell} = \begin{cases} 1 & k = i, \ell = j \text{ or } k = j, \ell = i, \\ 0 & \text{otherwise} \end{cases}$$

Note that $\mathbf{E}_{ij}$ and $\mathbf{F}_{ij}$ are positive semidefinite as they are rank-one matrices with all diagonal elements equal to one. However, the constraint $\mathbf{Tr}(\mathbf{X}) = 1$ implies that (14) is *not* quite a Packing SDP. In Appendix D, we show how our method generates an ϵ-optimal solution that is within ϵ of satisfying the trace constraint, or, alternatively, satisfies the trace constraint and is within ϵ of satisfying the packing constraints.

The coloring SDP

Karger et al [18] described an SDP relaxation for graph coloring a graph $\mathcal{G} = (V, A)$ that can be formulated as

$$\begin{array}{rl} \max & \zeta \\ \text{s.t.} & x_{ii} = 1, \quad i = 1, \ldots, n, \\ & \zeta \leq -x_{ij}, \quad (i,j) \in A, \\ & \mathbf{X} \succeq \mathbf{0}. \end{array} \tag{15}$$

Here, ζ is the reciprocal of the minimum colors required to color the graph, and for k-colorable graphs (15) is feasible with $(k \geq 2)$, $1 \geq \zeta^* \geq \frac{1}{k}$ and $\zeta \leq 1$ as a valid constraint [18]. Moreover, the optimization problem is actually a saddle-point problem. To see this, let

$$\mathbf{G}_{ij} = \mathbf{e}_i \mathbf{e}_i^\top + \mathbf{e}_j \mathbf{e}_j^\top - \frac{1}{2}(\mathbf{e}_i \mathbf{e}_j^\top + \mathbf{e}_j \mathbf{e}_i^\top)$$

Then,

$$\zeta = \min_{(i,j) \in A} x_{ij} = \min\{ \sum_{(i,j) \in A} w_{i,j} \langle \mathbf{E}_{ij}, \mathbf{X} \rangle : \sum_{(i,j) \in A} w_{(i,j)} = 1 \},$$

the formulation (15) is equivalent to

$$\begin{array}{rl} \max & 2 + \min \left\{ \sum_{(i,j) \in A)} w_{(i,j)} \langle \mathbf{G}_{ij}, \mathbf{X} \rangle \right\}, \\ \text{s.t.} & x_{ii} \leq 1, \quad i = 1, \ldots, n, \\ & \mathbf{X} \succeq \mathbf{0}. \end{array} \tag{16}$$

Note that the constraint $x_{ii} = 1$ can be relaxed to $x_{ii} \leq 1$ by the same argument as in MAXCUT, since increasing the diagonal only increases the objective as $\mathbf{G}_{ij} \succeq \mathbf{0}$. Our techniques can solve (16) at the cost of an additional m dual variables–see Appendix C for the details.

3 Computing feasible solutions from ϵ-approximate saddle points

In this section, we describe our algorithm, **SDP-Pack**, for solving the Packing SDP. In Section 3.1, we derive a Lagrangian saddle point problem from the Packing SDP and describe how to convert an approximate Lagrangian saddle point back to a feasible primal-dual pair for the Packing SDP with a "good" approximation guarantee. In Section 3.3, we present our algorithm for solving the Lagrangian saddle point problem and analyze its running time. In Section 3.4, we discuss how to improve the complexity of our algorithm by speeding up how we compute a matrix exponential for symmetric matrices, which is the bottleneck subroutine for our algorithm.

3.1 Packing SDP as a saddle-point problem

Define the Lagrangian function $\varphi_* : \mathcal{S}^n_+ \to \mathbb{R}$ as follows:

$$\varphi_*(\mathbf{X}) := \langle \mathbf{C}, \mathbf{X} \rangle + \min_{\mathbf{v}} \{ \sum_{i=0}^{p} v_i (1 - \langle \mathbf{A}_i, \mathbf{X} \rangle) : \mathbf{v} \geq \mathbf{0} \}, \tag{17}$$

where v_i is the penalty multiplier of the constraint $\langle \mathbf{A}_i, \mathbf{X} \rangle \leq 1$. Since (8) satisfies strong duality, it follows that

$$\rho^* = \max\{\varphi_*(\mathbf{X}) : \mathbf{Tr}(\mathbf{X}) \leq \tau, \mathbf{X} \succeq \mathbf{0}\}. \tag{18}$$

Switching the max and min in (18) we obtain

$$\rho^* = \min_{\mathbf{v}} \left\{ \sum_{i=0}^{p} v_i + \max_{\mathbf{X} \succeq \mathbf{0}} \left\{ \langle \big(\mathbf{C} - \sum_{i=0}^{p} v_i \mathbf{A}_i \big), \mathbf{X} \rangle \right\} \right\}. \tag{19}$$

Our algorithm focuses on solving (19)). Our method requires that we restrict ourselves to bounded $\mathbf{X}$ and $\mathbf{v}$. Condition (d) of Definition 1 implies that $\mathbf{Tr}(\mathbf{X}) \leq \tau$ and

$$\rho^* \leq \omega \leq \tau \, \mathbf{Tr}(\mathbf{C}). \tag{20}$$

We consider the following "restricted" saddle-point problem.

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{\mathbf{v} \in \mathcal{V}} \phi(\mathbf{X}, \mathbf{v}), \tag{21}$$

where

$$\begin{aligned} \phi(\mathbf{X}, \mathbf{v}) = & \quad \tau \langle \mathbf{C}, \mathbf{X} \rangle - \omega \sum_{i=0}^{p} v_i (\tau \langle \mathbf{A}_i, \mathbf{X} \rangle - 1) \\ = & \quad \omega \sum_{i=0}^{p} v_i - \tau \left\langle \omega \mathbf{v}^\top \mathcal{A} - \mathbf{C}, \mathbf{X} \right\rangle, \\ \mathcal{X} = & \quad \{ \mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq 1 \}, \\ \mathcal{V} = & \quad \left\{ \mathbf{v} : \mathbf{v} \geq \mathbf{0}, \textstyle\sum_{j=0}^{p} v_j \leq 1 \right\}, \end{aligned} \tag{22}$$

and $\mathcal{A} : \mathbb{R}^p \to \mathcal{S}^n$ denote the linear operator

$$\mathbf{v}^\top \mathcal{A} = \sum_{i=0}^{p} v_i \mathbf{A}_i. \tag{23}$$

The width of $\mathcal{V}$ is easy to calculate:

$$\max\{ \|\mathbf{u} - \mathbf{v}\|_1 : \mathbf{u}, \mathbf{v} \in \mathcal{V} \} = 2. \tag{24}$$

By Lemma 1, and Equations (6), and (22), we note that for a given $\mathbf{X} \in \mathcal{X}$,

$$\begin{aligned} \min_{\mathbf{v}\in\mathcal{V}} \phi(\mathbf{X}, \mathbf{v}) &= \tau \langle \mathbf{C}, \mathbf{X} \rangle \\ &\quad -\omega \max\big\{0, \max_{i=0,\ldots,p}\{\tau \langle \mathbf{A}_i, \mathbf{X}\rangle - 1\}\big\}, \end{aligned} \tag{25}$$

and for a given $\mathbf{v} \in \mathcal{V}$,

$$\max_{\mathbf{X}\in\mathcal{X}} \phi(\mathbf{X}, \mathbf{v}) = \omega\big(\sum_{i=0}^{p} v_i\big) + \tau \max\{0, \lambda_{\max}\big(\mathbf{C} - \omega \mathbf{v}^\top \mathcal{A}\big)\}. \tag{26}$$

For a function $\Phi : \Theta \times \Upsilon \to \mathbb{R}$ consider the *saddle-point* problem

$$\max_{\mathbf{z}\in\Theta} \min_{\mathbf{p}\in\Upsilon} \Phi(\mathbf{z}, \mathbf{p}). \tag{27}$$

For a given $\epsilon > 0$, we say that the pair $(\bar{\mathbf{z}}, \bar{\mathbf{p}}) \in \Theta \times \Upsilon$ is an *ϵ-saddle-point* if,

$$0 \le \max_{\mathbf{z}\in\Theta} \Phi(\mathbf{z}, \bar{\mathbf{p}}) - \min_{\mathbf{p}\in\Upsilon} \Phi(\bar{\mathbf{z}}, \mathbf{p}) \le \epsilon. \tag{28}$$

For problems with strong duality (e.g., convex programs satisfying the Slater condition) where Φ represents a Lagrangian saddle function, if the primal and dual feasible vector pair $(\bar{\mathbf{z}}, \bar{\mathbf{p}})$ satisfy (28), then they are within ϵ of optimality since strong duality implies that $\Phi(\mathbf{z}^*, \mathbf{p}^*) = \max_{\mathbf{z}\in\Theta} \Phi(\mathbf{z}, \mathbf{p}^*) = \min_{\mathbf{p}\in\Upsilon} \Phi(\mathbf{z}^*, \mathbf{p})$.

Theorem 1. *Fix $\epsilon > 0$. Suppose $(\bar{\mathbf{X}}, \bar{\mathbf{v}}) \in \mathcal{X} \times \mathcal{V}$ is an ϵ-saddle-point. Let*

$$\bar{d} = \tau \max_{i=1,\ldots,p} \{\langle \mathbf{A}_i, \bar{\mathbf{X}}\rangle\}, \tag{29}$$

and

$$\widehat{\mathbf{X}} = \begin{cases} \tau\bar{\mathbf{X}}, & \bar{d} \le 1, \\ \frac{\tau\bar{\mathbf{X}}}{\bar{d}}, & \bar{d} > 1, \end{cases} \qquad \widehat{v}_i = \begin{cases} \tau \max\{0, \lambda_{\max}(\mathbf{C} - \omega\bar{\mathbf{v}}^\top \mathcal{A})\}, & i = 0, \\ \omega\bar{\mathbf{v}}, & i = 1, \ldots, p, \end{cases} \tag{30}$$

Then $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{v}}$ are a feasible ϵ-optimal primal-dual pair for (8) and (9), respectively.

Proof. To prove this, we need to show that $\widehat{\mathbf{v}}$ and $\widehat{\mathbf{X}}$ are dual and primal feasible and that their respective objective functions are the correct values.

Note that $\widehat{\mathbf{v}}$ is feasible to (9). Since both $\bar{\mathbf{X}}$ and $\mathbf{A}_i$ are positive semidefinite, $\widehat{\mathbf{X}}$ is feasible to (8).

Then, by (26) and (30),

$$\begin{aligned} \max_{\mathbf{X}\in\mathcal{X}} \phi(\mathbf{X}, \bar{\mathbf{v}}) &= \omega \sum_{i=0}^{p} \bar{v}_i + \tau \max\{0, \lambda_{\max}(\mathbf{C} - \omega\bar{\mathbf{v}}^\top \mathcal{A})\} \\ &\ge \sum_{i=0}^{p} \widehat{v}_i + \end{aligned}$$

All that remains is to show that $\widehat{\mathbf{X}}$ induces an objective that is lower bounded by $\min_{\mathbf{v}\in\mathcal{D}} \phi(\bar{\mathbf{X}}, \mathbf{v})$. To show this, we examine the two cases:

(a) $\bar{d} \le 1$. Then $\left\langle \mathbf{C}, \widehat{\mathbf{X}} \right\rangle = \left\langle \mathbf{C}, (\tau\bar{\mathbf{X}}) \right\rangle$, and

$$\max\{0, \tau(\mathbf{Tr}(\bar{\mathbf{X}}) - 1), \max_{i=0,\ldots,p} (\tau \left\langle \bar{\mathbf{X}}, \mathbf{A}_i \right\rangle - 1)\} = 0,$$

whence $\left\langle \mathbf{C}, \widehat{\mathbf{X}} \right\rangle = \min_{\mathbf{v}\in\mathcal{V}\times\mathcal{V}} \phi(\bar{\mathbf{X}}, \mathbf{v})$.

(b) $\bar{d} > 1$. From the fact that $\frac{1}{d} \geq 1 - (d-1)$ for all $d > 0$,

$$\begin{aligned} \tfrac{\tau}{\bar{d}} \left\langle \mathbf{C}, \bar{\mathbf{X}} \right\rangle &\geq \tau \left\langle \mathbf{C}, \bar{\mathbf{X}} \right\rangle - (\bar{d}-1)\tau \left\langle \mathbf{C}, \bar{\mathbf{X}} \right\rangle \\ &\geq \tau \left\langle \mathbf{C}, \bar{\mathbf{X}} \right\rangle - \omega(\bar{d}-1) \qquad \text{by (20)} \\ &\geq \tau \left\langle \mathbf{C}, \bar{\mathbf{X}} \right\rangle - \omega \max\left\{0, \tau(\mathbf{Tr}(\bar{\mathbf{X}}) - 1), \max_{i=0,\ldots,p} (\tau \left\langle \mathbf{A}_i, \bar{\mathbf{X}} \right\rangle - 1)\right\} \\ &= \min_{\mathbf{v} \in \mathcal{V}} \phi(\mathbf{X}, \mathbf{v}). \end{aligned}$$

We can conclude that

$$\sum_{i=0}^{p} \widehat{v}_i - \left\langle \mathbf{C}, \widehat{\mathbf{X}} \right\rangle \quad \leq \max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \bar{\mathbf{v}}) - \min_{\mathbf{v} \in \mathcal{V}} \phi(\mathbf{X}, \mathbf{v}) \leq \epsilon.$$

Thus, we have established that $\widehat{\mathbf{X}}$ and $\widehat{\mathbf{v}}$ are a feasible ϵ-optimal primal-dual pair. □

The trace bound in Condition (d) of Definition 1 is required to ensure that the shifting in (30) yields a feasible dual point, and the bound (20) is required to ensure that the dual generated is "close" to the original dual in terms of objective value. This bound (i.e., the width) is directly tied to the running time.

3.2 Computing an approximate saddle point

Define

$$f(\mathbf{v}) := \max_{\mathbf{X} \in \mathcal{X}} \phi(\mathbf{X}, \mathbf{v}). \tag{31}$$

Theorem 1, the lowerbound on the objective from (20) and the known conversion from absolute to relative approximation detailed in Lemma 2 imply that an absolute ϵ-approximate solution to

$$\min_{\mathbf{v} \in \mathcal{V}} f(\mathbf{v}), \tag{32}$$

is an ϵ/ω-relative approximate solution to (9). However, from (26) we have that

$$f(\mathbf{v}) = \omega \left(\sum_{i=0}^{p} v_i \right) + \tau \max \left\{ 0, \lambda_{\max} \left(\mathbf{C} - \omega \sum_{i=0}^{p} v_i \mathbf{A}_i \right) \right\}. \tag{33}$$

Thus, $f(\mathbf{v})$ is not differentiable, which is unfortunate since we would like to use a gradient direction in our algorithm. We approximate f with the following function

$$f_\alpha(\mathbf{v}) := \omega \left(\sum_{i=0}^{p} v_i \right) + \frac{1}{\alpha} \log \left(1 + \sum_{i=0}^{n} e^{\alpha \tau \lambda_i} \right), \tag{34}$$

where

$$\lambda_i = \lambda_i \left(\mathbf{C} - \omega \sum_{j=1}^{p} v_j \mathbf{A}_j \right); \lambda_{\max} = \max_{i=1,\ldots,n} \lambda_i. \tag{35}$$

f_α is also known as a *potential function*. The following result verifies that f_α is a good approximation:

Lemma 4. *For any* $\mathbf{v} \in \mathcal{V}$ $f(\mathbf{v}) \leq f_\alpha(\mathbf{v}) \leq f(\mathbf{v}) + \frac{\log(n+1)}{\alpha}$.

Proof. The linear terms in f and f_α are identical, so we need only show that:

$$\tau \max\{0, \lambda_{\max}\} \leq \frac{1}{\alpha} \log \left(1 + \sum_{i=1}^{n} e^{\alpha \tau \lambda_i} \right) \leq \tau \max\{0, \lambda_{\max}\} + \frac{\log(n+1)}{\alpha}. \tag{36}$$

Define $\lambda_0 = 0$ whence $1 + \sum_{i=1}^{n} e^{\alpha\tau\lambda_i} = \sum_{i=0}^{n} e^{\alpha\tau\lambda_i}$, and

$$\max_{i=1,\ldots,n} \lambda_i = \max\{0, \lambda_{\max}\}.$$

To see the first inequality in (36),

$$\frac{1}{\alpha}\log\left(\sum_{i=0}^{n} e^{\alpha\tau\lambda_i}\right) \geq \frac{1}{\alpha}\log\left(e^{\alpha\tau \max_{i=0,\ldots,n} \lambda_i}\right) = \tau\max\{0, \lambda_{\max}\}.$$

To see the second inequality in (36)

$$\frac{1}{\alpha}\log\left(\sum_{i=0}^{n} e^{\alpha\tau\lambda_i}\right) \leq \frac{1}{\alpha}\log\left((n+1)e^{\alpha\tau \max_{i=0,\ldots,n} \lambda_i}\right) = \frac{\log(n+1)}{\alpha} + \tau\max\{0, \lambda_{\max}\}.$$

□

Thus, we can compute an absolute ϵ-approximation to (32) by setting $\alpha = \frac{\log(n+1)}{2\epsilon}$ and computing an absolute $\epsilon/2$-solution to

$$\min\{f_\alpha(\mathbf{v}) : \mathbf{v} \in \mathcal{V}\}. \tag{37}$$

Using potential functions in conjunction with Lagrangian penalization is an idea from the vector packing and covering literature [30, 13]. In this approach, (37) is solved by a first-order gradient descent method. Klein and Lu [19] and Arora and Kale [1] also use first-order methods to approximately solve SDPs. We use a second-order quasi-Newton-type method to solve (37), which will require the following Taylor series-like expansion.

Lemma 5. *For all* $\mathbf{v}, \mathbf{w} \in \mathcal{V}$,

$$f_\alpha(\mathbf{v}) \leq f_\alpha(\mathbf{w}) + \nabla f_\alpha(\mathbf{w})^\top(\mathbf{v} - \mathbf{w}) + \frac{\alpha\tau\omega}{2}\|\mathbf{v} - \mathbf{w}\|_1^2. \tag{38}$$

We prove Lemma 5 in Appendix B. Since $\|\mathbf{v} - \mathbf{w}\|_1$ is non-differentiable, we approximate it with a differentiable function such that (38) remains valid.
Define

$$d(\mathbf{v}, \mathbf{v}') := \sum_{i=0}^{p} v_i \log\left(\frac{v_i}{v_i'}\right). \tag{39}$$

d is called the *Kullback-Leibler*, or entropy, distance, and bounds the norm as follows (see Appendix B for the proof).

Lemma 6. *For* $\mathbf{v}, \mathbf{v}' \in \mathcal{V}$,

$$d(\mathbf{v}, \mathbf{v}') \geq \frac{1}{2}\|\mathbf{v} - \mathbf{v}'\|_1^2.$$

Lemma 6 implies that $\frac{1}{2}\|\mathbf{v} - \mathbf{w}\|_1^2$ can be replaced by the differentiable function d. The optimization problems that we encounter are of the form $\min\{\mathbf{c}^\top\mathbf{w} + \gamma d(\mathbf{v} - \mathbf{w})\}$. The solution to such problems can be found in closed form:

Lemma 7. *Consider the optimization problem*

$$\begin{array}{ll} \min & \gamma d(\mathbf{v}, \mathbf{v}') + \mathbf{c}^\top\mathbf{v}, \\ s.\ t. & \sum_{i=0}^{p} v_i \leq 1, \\ & \mathbf{v} \geq \mathbf{0}, \end{array} \tag{40}$$

where $\mathbf{v}'$ *is on the interior of* $\mathcal{V}$ *and* $\gamma \in \mathbb{R}$ *is a given parameter. The optimal solution,* $\mathbf{v}^*$*, of (40) is given by:*

$$v_i^* = \frac{v_i' e^{-(c_i/\gamma)}}{1 + \sum_{j=0}^{p} v_j'(e^{-(c_j/\gamma)} - 1)}, \quad i = 0, \ldots, p.$$

We want to use Lemma 7 with an approximation to $\mathbf{c}$. If we instead use a vector $\mathbf{c}_\delta$, where $\|\mathbf{c} - \mathbf{c}_\delta\|_\infty \leq \delta$, then the optimal solution to

$$\begin{array}{ll} \min & \gamma d(\mathbf{v}, \mathbf{v}') + \mathbf{c}_\delta^\top \mathbf{v}, \\ \text{s. t.} & \sum_{i=0}^{p} v_i \leq 1, \\ & \mathbf{v} \geq \mathbf{0}, \end{array} \tag{41}$$

is an absolute δ-approximation to (40). To prove this, let $\mathbf{v}^*$ be the optimal solution to (40) and $\bar{\mathbf{v}}$ be the solution to the (41), and note that

$$\gamma d(\bar{\mathbf{v}}, \mathbf{v}') + \mathbf{c}_\delta^\top \bar{\mathbf{v}} \leq \gamma d(\mathbf{v}^*, \mathbf{v}') + \mathbf{c}_\delta^\top \mathbf{v}^* \leq \gamma d(\mathbf{v}^*, \mathbf{v}') + \mathbf{c}^\top \mathbf{v}^* + \delta,$$

where the last inequality follows since $\sum_{i=0}^{p} v_i^* \leq 1$.

We now discuss the calculation of the gradient of f_α. which is given by (e.g., see Theorem 1.1 in Lewis [21]):

$$\nabla f_\alpha(\mathbf{v}) = \mathbf{1} + \tau \mathbf{X}_\mathbf{v} \otimes \mathcal{A}, \tag{42}$$

where

$$\mathbf{X}_\mathbf{v} = \frac{e^{\alpha\tau\left(\mathbf{C} - \omega \sum\limits_{i=0}^{p} v_i \mathbf{A}_i\right)}}{\left(1 + \sum\limits_{j=1}^{n} e^{\alpha\tau\lambda_j}\right)},$$

with λ_i as defined in (35). $\mathcal{A}$ was defined in (23) and we define its *adjoint* operator as $\mathbf{X}_\mathbf{v} \otimes \mathcal{A} = \mathbf{v}$, where $v_i = \langle \mathbf{A}_i, \mathbf{X}_\mathbf{v} \rangle$. The term $e^\mathbf{A}$ denotes the *matrix exponential* of $\mathbf{A}$ (see Moler and Van Loan [23, 24]). It is well known that

$$\mathbf{X}_\mathbf{v} = \frac{\sum\limits_{j=1}^{n} e^{\alpha\tau\lambda_j} \mathbf{u}_j \mathbf{u}_j^\top}{\left(1 + \sum\limits_{j=1}^{n} e^{\alpha\tau\lambda_j}\right)}.$$

where $\mathbf{u}_j$ are the eigenvectors of $\mathbf{C} - \omega \sum\limits_{i=0}^{p} v_i \mathbf{A}_i$.

Thus, using the eigenvalue decomposition to calculate $\mathbf{X}_\mathbf{v}$ requires $O(n^3)$ time. In Section 3.4, we discuss faster methods for computing the matrix exponential. However, all of these methods only compute an approximation of the matrix exponential. We therefore call a matrix $\mathbf{E}$ a δ-approximation of $\mathbf{X}_\mathbf{v}$ if $\mathbf{E} \succeq \mathbf{0}$, and

$$\|\mathbf{X}_\mathbf{v} - \mathbf{E}\|_1 = \|\mathbf{X}_\mathbf{v} - \mathbf{E}\|_\infty \leq \delta. \tag{43}$$

Note the equality follows since $\mathbf{X}_\mathbf{v}$ and $\mathbf{E}$ are symmetric. Let

$$\kappa := \max_{k=1,\ldots,p} \left\{ \max_{ij} \{|a_{ij}^{(k)}|\} \right\} \tag{44}$$

where $\mathbf{A}_k = [a_{ij}^{(k)}]$.

Lemma 8. *Suppose* $\mathbf{E}$ *is a* δ*-approximation of* $\mathbf{X}_\mathbf{v}$ *and* $\mathbf{v} \in \mathcal{V}$ *is infeasible for* (9). *Let* $\mathbf{g}_\mathbf{v} = \frac{\mathbf{E} \otimes \mathcal{A}}{1 + \mathbf{Tr}(\mathbf{E})}$. *Then* $\|\nabla f_\alpha(\mathbf{v}) - \mathbf{g}_\mathbf{v}\|_\infty \leq 2\tau\delta\kappa$.

Proof. Without loss of generality, assume that $\mathbf{Tr}(\mathbf{E}) > \mathbf{Tr}(\mathbf{X}_\mathbf{v})$ and note that (43) implies that $\mathbf{Tr}(\mathbf{E}) - \mathbf{Tr}(\mathbf{X}_\mathbf{v}) \leq \delta$.

$$\|\nabla f_\alpha(\mathbf{v}) - \mathbf{g}_\mathbf{v}\|_\infty = \tau \|(\mathbf{E} - \mathbf{X}_\mathbf{v}) \otimes \mathcal{A}\|_\infty \leq 2\tau\delta\kappa.$$

The result follows since $\mathbf{Tr}(\mathbf{X}_\mathbf{v}) \geq 0$. □

Note that the complexity of computing $\mathbf{X_v} \otimes \mathcal{A}$ is dependent on the number of non-zero entries in each of the matrices $\mathbf{A}_i$. Such a computation will be determined by the number of nonzeros in the matrix $\sum_{i=1}^{p} |\mathbf{A}_i|$ which is at most $O(rp)$, and sometimes significantly less. For example, in the MAXCUT problem, $\mathbf{X_v} \otimes \mathcal{A} = \mathbf{diag}(\mathbf{X_v})$, which can be computed via n memory look-ups. In both the coloring and ϑ-function SDP, computing $\mathbf{X_v} \otimes \mathcal{A}$ requires $O(m+n)$ time. When $\mathbf{A}_i$ are $0-1$ matrices, as is often the case in combinatorial optimization problems, $\otimes$ has the effect of choosing entries from the $\mathbf{X_v}$ matrix.
For the purpose of computing algorithmic complexity, we define

$$\gamma(\mathcal{A}, \delta) := \text{the running time to compute } \mathbf{E} \otimes \mathcal{A} \text{ and a } \delta\text{-approximation of } \mathbf{X_v}. \tag{45}$$

Lemma 9. *A δ-approximation to the gradient ∇f_α can be computed in $O(\gamma(\mathcal{A}, \frac{\delta}{2\tau\kappa})$ time.*

3.3 SDP-Pack

In this section, we describe our algorithm, **SDP-Pack**(depicted in Figure 2), for finding an absolute approximation to the problem (37). As in Bienstock and Iyengar's algorithm for covering and packing [5], our approach employs Lemma 7 to solve both "global" and "local" optimization problems in a manner similar to Nesterov [27]. In the "global" optimization, we optimize based upon a weighted combination of the previous iterates' gradients. Suppose that we have computed dual iterates $\mathbf{v}^1, \ldots, \mathbf{v}^k$. Then, define

$$\mathcal{G}_{\mathbf{v}}(k) \equiv \min_{\mathbf{v} \in \mathcal{V}} \left\{ \sum_{i=0}^{k} \frac{i+1}{2} \left[f_\alpha(\mathbf{v}^i) + \nabla f_\alpha(\mathbf{v}^i)^\top (\mathbf{v} - \mathbf{v}^i) \right] + \alpha\tau\omega d(\mathbf{v}, \mathbf{v}^0) \right\}.$$

Note that Lemma 7 indicates that the solution to $\mathcal{G}_{\mathbf{v}}(k)$ can be found in closed form. Also, note that using a δ-approximate gradient results in an error of $\delta \sum_{i=0}^{k} \frac{i+1}{2}$. Since we divide the solution through by $\sum_{i=1}^{k} \frac{i+1}{2}$, our overall error is δ.

In order to define the local optimization, we construct a problem based upon the most recent dual and global iterates. We note that the constructed problem integrates both local and global information. Suppose we have computed the solution to $\mathcal{G}_{\mathbf{v}}(k)$, $\mathbf{z}^k$. Then, define

$$\mathcal{L}_{\mathbf{v}}(k) \equiv \min_{\mathbf{v} \in \mathcal{V}} \left\{ \left(\frac{k+2}{2} \right) \nabla f_\alpha(\mathbf{v}^{k+1})^\top (\mathbf{v} - \mathbf{z}^k) + \alpha\tau\omega d(\mathbf{v}, \mathbf{z}^k) \right\},$$

and note that the solution to $\mathcal{L}_{\mathbf{v}}(k)$ can also be found in closed form according to Lemma 7. The overall error is $\frac{k+2}{2}\delta$ when a δ-approximate gradient is used.

We now prove the complexity of **SDP-Pack**. Our proof uses techniques similar to [5] and [27]. First, we require the following technical lemma:

Lemma 10. *Suppose $\mathbf{u}^k, \mathbf{z}^k$, and $\mathbf{v}^k$ are computed according to* **SDP-Pack***, and that the gradients used are all δ-approximations. For all $k \geq 0$, the following inequality holds:*

$$(\mathcal{R}_k) \quad \Gamma_k f_\alpha(\mathbf{u}^k) \leq \psi_k := \min_{\mathbf{v} \in \mathcal{V}} \left\{ \sum_{i=0}^{k} \gamma_i \left[f_\alpha(\mathbf{v}^i) + \nabla f_\alpha(\mathbf{v}^i)^\top (\mathbf{v} - \mathbf{v}^i) \right] + \alpha\tau\omega d(\mathbf{v}, \mathbf{v}^0) \right\} + \Gamma_k \delta,$$

where $\gamma_k = \frac{k+1}{2}$, $\Gamma_k = \sum_{i=0}^{k} \gamma_i = \frac{(k+1)(k+2)}{4}$.

Proof. The base case $(\mathcal{R}_0)$ is established as follows. Let $\mathbf{g}_{\mathbf{v}^i}$ denote the δ-approximate gradient to $\nabla f_\alpha(\mathbf{v}^i)$

Algorithm: SDP-Pack

Inputs: $\mathbf{C}, \mathbf{A}_i, i = 1, \ldots, m, \epsilon, \mathbf{u}^0 = \mathbf{v}^0 := \frac{1}{p+2}\mathbf{1}, \gamma := \alpha\tau\omega$

Outputs: $(\bar{\mathbf{v}}, \bar{\mathbf{X}})$

(a) $k = 0$

(b) Compute $f_\alpha(\mathbf{v}^k)$, and a δ-approximate gradient $\mathbf{g}_{\mathbf{v}^i}$

Do:

(c) Let $\widehat{\mathbf{g}}^k = \sum_{i=0}^k \frac{i+1}{2}\mathbf{g}_{\mathbf{v}^i}$,

Let $\mathbf{w}^k = \sum_{i=0}^k \mathbf{v}^i$

Let $z_i^k = \frac{w_i^k e^{-(\widehat{g}_i^k/\gamma + w_i^k)}}{1+\sum_{j=0}^p w_j^k (e^{-(\widehat{g}_i^k/\gamma + w_j^k)} - 1)}, i = 0, \ldots, p$

(d) Update $\mathbf{v}^{k+1} = \frac{2}{k+3}\mathbf{z}^k + \frac{k+1}{k+3}\mathbf{u}^k$

$\bar{\mathbf{X}} = \bar{\mathbf{X}} + \frac{k+2}{1+\mathbf{Tr}(\mathbf{X}_{\mathbf{v}^{k+1}})}\mathbf{X}_{\mathbf{v}^{k+1}}$

(e) Let $\bar{\mathbf{g}}^k = \frac{k+2}{2}\mathbf{g}_{\mathbf{v}^k}$,

Let $u_i^k = \frac{v_i^k e^{-(\bar{g}_i^k/\gamma + v_i^k)}}{1+\sum_{j=0}^p v_j^k (e^{-(\bar{g}_i^k/\gamma + v_j^k)} - 1)}, i = 0, \ldots, p$

(f) $k \leftarrow k + 1$

(g) While $(k \leq 8\tau\omega \log(p+1)/\epsilon)$

(h) Return $(\mathbf{u}^k, \frac{\bar{\mathbf{X}}}{(k+1)(k+2)})$

Figure 2: **SDP-Pack**: Note that the initial dual point passed is feasible, as $\mathbf{v}^0 \in \mathcal{V}$. Note that computing ∇f_α and $\mathbf{X}_{\mathbf{v}}$ rely on the problem-specific oracle and may result in an approximate rather than exact computation (see Sections 3.1 and 3.4).

for all i. Since $\gamma_0 = \frac{1}{2} < 1$,

$$\begin{aligned}
&\min_{\mathbf{v}\in\mathcal{V}} \left\{\tau\omega\alpha d(\mathbf{v}, \mathbf{v}^0) + \gamma_0[f_\alpha(\mathbf{v}^0) + \mathbf{g}_{\mathbf{v}^0}^T(\mathbf{v} - \mathbf{v}^0)]\right\} \\
&\quad \geq \quad \gamma_0\left(\min_{\mathbf{v}\in\mathcal{V}} \left\{\tau\omega\alpha d(\mathbf{v}, \mathbf{v}^0) + f_\alpha(\mathbf{v}^0) + \nabla f_\alpha(\mathbf{v}^0)^T(\mathbf{v} - \mathbf{v}^0)\right\} + \delta\right) \\
&\quad = \quad \gamma_0\left(\tau\omega\alpha d(\mathbf{u}^0, \mathbf{v}^0) + f_\alpha(\mathbf{v}^0) + \nabla f_\alpha(\mathbf{v}^0)^T(\mathbf{u}^0 - \mathbf{v}^0) + \delta\right), \qquad (46) \\
&\quad \geq \quad \gamma_0 f_\alpha(\mathbf{u}^0).
\end{aligned}$$

where (46) follows from Lemma 6 and since $\mathbf{u}^0 = \mathbf{v}^0$ and $\delta > 0$.

Assume $(\mathcal{R}_k)$ is true. Note that by definition of $\mathcal{G}_{\mathbf{v}}$,

$$\mathbf{z}^k \quad = \quad \arg\min_{\mathbf{v}\in\mathcal{V}} \left\{\tau\omega\alpha d(\mathbf{v}, \mathbf{v}^0) + \sum_{i=0}^k \gamma_i[f_\alpha(\mathbf{v}^i) + \mathbf{g}_{\mathbf{v}^i}^T(\mathbf{v} - \mathbf{v}^i)]\right\}.$$

Then,

$$\psi_k = \sum_{i=0}^k \gamma_k \left[f_\alpha(\mathbf{v}^i) + \nabla f_\alpha(\mathbf{v}^i)^\top(\mathbf{z}^k - \mathbf{v}^i)\right] + \Gamma_k\delta + \alpha\tau\omega d(\mathbf{z}^k, \mathbf{v}^0), \qquad (47)$$

and

$$\left(\tau\omega\alpha\nabla d(\mathbf{z}^k, \mathbf{v}^0) + \sum_{i=1}^k \gamma_i \mathbf{g}_{\mathbf{v}^i}\right)^T(\mathbf{v}^i - \mathbf{z}^k) \geq 0, \qquad (48)$$

Since $\mathbf{v}^0 = \frac{1}{p+2}\mathbf{1}$ and $\mathbf{z}^k, \mathbf{v} \in \mathcal{V}$, it follows that

$$\begin{aligned} d(\mathbf{v},\mathbf{v}^0) - d(\mathbf{v},\mathbf{z}^k) &= \sum_{i=0}^{p} u_i \log\left(\frac{z_i^k}{u_i^0}\right) \\ &= \sum_{i=0}^{p} z_i^k \log\left(\frac{z_i^k}{u_i^0}\right) + \sum_{i=0}^{p}(u_i - z_i^k)\log\left(\frac{z_i^k}{u_i^0}\right) \\ &= d(\mathbf{z}^k,\mathbf{v}^0) + \left(\nabla d(\mathbf{z}^k,\mathbf{v}^0)\right)^\top (\mathbf{v}-\mathbf{z}^k). \end{aligned}$$

Thus, we have that

$$\begin{aligned} & \tau\omega\alpha d(\mathbf{v},\mathbf{v}^0) + \sum_{i=0}^{k}\gamma_i[f_\alpha(\mathbf{v}^i) + \nabla f_\alpha(\mathbf{v}^i)^T(\mathbf{v}-\mathbf{v}^i) + \delta] \\ &= \tau\omega\alpha d(\mathbf{v},\mathbf{z}^k) + \underbrace{\left(\tau\omega\alpha\nabla d(\mathbf{z}^k,\mathbf{v}^0) + \sum_{i=0}^{k}\gamma_i\nabla f_\alpha(\mathbf{v}^i)\right)^T(\mathbf{v}-\mathbf{z}^k) + \delta\sum_{i=0}^{k}\gamma_i}_{\geq 0} \\ &\quad + \underbrace{\left(\tau\omega\alpha d(\mathbf{z}^k,\mathbf{v}^0) + \sum_{i=0}^{k}\gamma_i[f_\alpha(\mathbf{v}^i) + \nabla f_\alpha(\mathbf{v}^i)^T(\mathbf{z}^k-\mathbf{v}^i)]\right)}_{=\psi_k}, \\ &\geq \Gamma_k f_\alpha(\mathbf{u}^k) + \tau\omega\alpha d(\mathbf{v},\mathbf{z}^k), \end{aligned} \tag{49}$$

where the last inequality follows from the induction hypothesis.

Thus, we have that

$$\begin{aligned} \psi_{k+1} \geq \min_{\mathbf{v}\in\mathcal{V}}\big\{ & \tau\omega\alpha d(\mathbf{v},\mathbf{z}^k) + \Gamma_k f_\alpha(\mathbf{u}^k) \\ & + \gamma_{k+1}[f_\alpha(\mathbf{v}^{k+1}) + \nabla f_\alpha(\mathbf{v}^{k+1})^T(\mathbf{v}-\mathbf{v}^{k+1}) + \delta]\big\}. \end{aligned} \tag{50}$$

From the convexity of f_α and the update rule in Line (d) of **SDP-Pack** we have that

$$\begin{aligned} & \Gamma_k f_\alpha(\mathbf{u}^k) + \gamma_{k+1}[f_\alpha(\mathbf{v}^{k+1}) + \nabla f_\alpha(\mathbf{v}^{k+1})^T(\mathbf{v}-\mathbf{v}^{k+1})] \\ &\geq \Gamma_k[f_\alpha(\mathbf{v}^{k+1}) + \nabla f_\alpha(\mathbf{v}^{k+1})^T(\mathbf{u}^k-\mathbf{v}^{k+1})] + \gamma_{k+1}[f_\alpha(\mathbf{v}^{k+1}) + \nabla f_\alpha(\mathbf{v}^{k+1})^T(\mathbf{v}-\mathbf{v}^{k+1})], \\ &= \Gamma_{k+1} f_\alpha(\mathbf{v}^{k+1}) + \gamma_{k+1}\nabla f_\alpha(\mathbf{v}^{k+1})^T(\mathbf{v} - (1+\frac{k+1}{2})\mathbf{v}^{k+1} + \frac{k+1}{2}\mathbf{u}^k), \\ &= \Gamma_{k+1} f_\alpha(\mathbf{v}^{k+1}) + \gamma_{k+1}\nabla f_\alpha(\mathbf{v}^{k+1})^T(\mathbf{v}-\mathbf{z}^k). \end{aligned} \tag{51}$$

From Line (e) of **SDP-Pack**:

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{v}\in\mathcal{V}}\left\{\gamma_{k+1}\nabla f_\alpha(\mathbf{v}^{k+1})^\top(\mathbf{v}-\mathbf{z}^k) + \alpha\tau\omega d(\mathbf{v},\mathbf{z}^k)\right\}.$$

Thus, $\delta > 0$, (50) and (51) imply that

$$\begin{aligned} & \psi_{k+1} \\ &\geq \tau\omega\alpha d(\mathbf{u}^{k+1},\mathbf{z}^k) + \Gamma_{k+1} f_\alpha(\mathbf{v}^{k+1}) + \gamma_{k+1}\left(\nabla f_\alpha(\mathbf{v}^{k+1})^T\left(\mathbf{u}^{k+1}-\mathbf{z}^k\right) + \delta\right), \\ &\geq \frac{\tau\omega\alpha}{2}\left\|\mathbf{u}^{k+1}-\mathbf{z}^k\right\|_1^2 + \Gamma_{k+1} f_\alpha(\mathbf{v}^{k+1}) + \gamma_{k+1}\nabla f_\alpha(\mathbf{v}^{k+1})^T\left(\mathbf{u}^{k+1}-\mathbf{z}^k\right), \\ &\geq \Gamma_{k+1}\Big(f_\alpha(\mathbf{v}^{k+1}) + \nabla f_\alpha(\mathbf{v}^{k+1})^T\left(\tau_k(\mathbf{u}^{k+1}-\mathbf{z}^k)\right) + \frac{\tau\omega\alpha}{2}\left\|\tau_k(\mathbf{u}^{k+1}-\mathbf{z}^k)\right\|_1^2\Big), \qquad (52) \\ &\geq \Gamma_{k+1}\Big(f_\alpha(\mathbf{v}^{k+1}) + \nabla f_\alpha(\mathbf{v}^{k+1})^T\left(\mathbf{u}^{k+1}-\mathbf{v}^{k+1}\right) + \frac{\tau\omega\alpha}{2}\left\|\mathbf{u}^{k+1}-\mathbf{v}^{k+1}\right\|_1^2\Big), \qquad (53) \\ &\geq \Gamma_{k+1} f_\alpha(\mathbf{u}^{k+1}), \qquad (54) \end{aligned}$$

where (52) follows from the fact that $\tau_k^2 \leq 1/\Gamma_{k+1}$, (53) follows from the rule in Line (e) in **SDP-Pack**, and (54) follows from the Lemma 5. Thus, we have that $(\mathcal{R}_k)$ holds for all $k \geq 0$. □

We can now prove the complexity of **SDP-Pack**.

Theorem 2. *Assume we can approximate the gradient ∇f_α to within $\epsilon/4$. For all $T \geq \frac{8\tau\omega\log(p+1)}{\epsilon}$, we have that $f(\mathbf{u}^T) \leq f^* + \epsilon$ and*

$$\min_{\mathbf{v}\in\mathcal{V}} \phi(\widehat{\mathbf{X}}, \mathbf{v}) + \epsilon \geq f_\alpha(\mathbf{u}^T) \geq f(\mathbf{u}^T) \geq f^*.$$

Proof. Let $\mathbf{v}^* = \text{argmin}\{f_\alpha(\mathbf{v})\}$. From the convexity of f_α we have that

$$f_\alpha(\mathbf{v}^i) + \nabla f_\alpha(\mathbf{v}^i)^\top(\mathbf{v}^* - \mathbf{v}^i) \leq f_\alpha(\mathbf{v}^*).$$

Thus, $(\mathcal{R}_k)$ implies that

$$\begin{aligned} f_\alpha(\mathbf{u}^k) &\leq \frac{\tau\omega\alpha\max_{\mathbf{v}\in\mathcal{V}}\{d(\mathbf{v},\mathbf{v}^0)\}}{\Gamma_k} + \frac{1}{\Gamma_k}\sum_{i=1}^{k}\gamma_i[f_\alpha(\mathbf{v}^i) + \nabla f_\alpha(\boldsymbol{\delta}_i)^\top(\mathbf{v}^* - \mathbf{v}^i)] + \epsilon/4, \\ &\leq \frac{\tau\omega\alpha\log(n)}{\Gamma_k} + \epsilon/4 + f_\alpha^*, \end{aligned} \tag{55}$$

where (55) follows from that that $\max_{\mathbf{v}\in\mathcal{V}}\{d(\mathbf{v},\mathbf{v}^0)\}$ is achieved at one of the extreme points of $\mathcal{V}$. From the bounds $f(\mathbf{v}) \leq f_\alpha(\mathbf{v}) \leq f(\mathbf{v}) + \frac{\log(p+1)}{\alpha}$, we have that

$$f(\mathbf{u}^T) \leq f_\alpha(\mathbf{u}^T) \leq \frac{\tau\omega\alpha\log(n)}{\Gamma_T} + \epsilon/4 + f_\alpha^* \leq \frac{\tau\omega\alpha\log(n)}{\Gamma_T} + \epsilon/4 + \frac{\log(p+1)}{\alpha} + f^*.$$

Then, $f(\mathbf{u}^T) \leq f^* + \epsilon$ now follows by substituting $\alpha = \frac{2\log(p+1)}{\epsilon}$.

Thus, we have established that the dual variable $\bar{\mathbf{v}}$ produced by **SDP-Pack** is close to optimal. All that remains to be shown is that the primal variable $\bar{\mathbf{X}}$ is also close to optimal. From the definition of $f_\alpha(\mathbf{v})$ in (34), we have that

$$\begin{aligned} f_\alpha(\mathbf{v}) &= \mathbf{1}^\top\mathbf{v} + \frac{1}{\alpha}\log\Big(1 + \sum_{i=1}^{n}\exp(\alpha\lambda_i(\mathbf{X}_u))\Big), \\ &\leq \mathbf{1}^\top\mathbf{v} + \langle\mathbf{C}, \mathbf{X}_\mathbf{v}\rangle + \frac{\log(p+1)}{\alpha}. \end{aligned}$$

Then, the induction hypothesis, Line (e) and Line (h) of **SDP-Pack** imply that

$$\begin{aligned} \Gamma_k f_\alpha(\mathbf{u}^k) &\leq \tau\omega\alpha\log(n) + \min_{\mathbf{v}\in\mathcal{U}}\left\{\sum_{i=0}^{k}\gamma_i[f_\alpha(\mathbf{v}^i) + \nabla f_\alpha(\mathbf{v}^i)^T(\mathbf{v} - \mathbf{v}^i)]\right\} + \Gamma_k\epsilon/4, \\ &\leq \tau\omega\alpha\log(n) + \frac{\Gamma_k\log(p+1)}{\alpha} + \Gamma_k\min_{\mathbf{v}\in\mathcal{V}}\left\{\tau\left\langle\mathbf{C}, \bar{\mathbf{X}}\right\rangle - \mathbf{v}^\top\mathcal{A}\right\} + \Gamma_k\epsilon/4, \\ &= \tau\omega\alpha\log(n) + \frac{\Gamma_k\log(p+1)}{\alpha} + \Gamma_k\min_{\mathbf{v}\in\mathcal{V}}\phi(\bar{\mathbf{X}}, \mathbf{v}) + \Gamma_k\epsilon/4. \end{aligned}$$

□

Note that for both the MAXCUT problem and the graph coloring problem, $\tau = n$ and $\omega = 2$. Also, the bottleneck step in **SDP-Pack** is the computation of the matrix exponential in order to determine ∇f_α. Using an ordinary eigendecomposition results in $\gamma(\mathcal{A}, \delta) = O(n^3)$. Thus, we have the following corollary.

Corollary 1. SDP-Pack *has a running time of $O(\frac{1}{\epsilon}n^4\log n)$ for the Packing SDP when an ordinary eigendecomposition is used to compute the matrix exponential.*

In the Section 3.4, we investigate faster methods for computing the matrix exponential. We first examine how the constraint $\mathbf{Tr}(\mathbf{X}) = \tau$ (as opposed to a trace inequality) simplifies our smoothing.

3.4 Matrix exponential via Lanczos iterations

The most expensive step in **SDP-Pack** is the computation of:

$$\mathbf{X_v} = \frac{\mathbf{V}\mathbf{D}(\alpha\tau\boldsymbol{\lambda})\mathbf{V}^\top}{\left(1+\sum_{i=1}^n e^{\alpha\tau\lambda_i}\right)} = \frac{e^{\alpha\tau\mathbf{A}(\mathbf{v})}}{1+\mathbf{Tr}(e^{\alpha\tau\mathbf{A}(\mathbf{v})})},$$

where $e^{\alpha\tau\mathbf{A}(\mathbf{v})}$ denotes the *matrix exponential* [23, 24] of

$$\mathbf{A}(\mathbf{v}) = \mathbf{C} - \omega\mathbf{v}^\top\mathcal{A}.$$

The direct method for computing $e^{\tilde{r}\alpha\mathbf{A}(\mathbf{v})}$ involves computing an eigendecomposition of the matrix $\mathbf{A}(\mathbf{v})$ and then using this to compute $\mathbf{X}_v$ via (3.4). In this section we discuss the *Krylov subspace method* which computes an ϵ-approximation for $e^{\tau\alpha\mathbf{A}(\mathbf{v})}$ without first computing the eigendecomposition.

In Section 3.4.1, we present the Krylov subspace methods for computing the matrix exponential, and in Section 3.4.2, we present an improvement to this method.

3.4.1 Krylov subspace methods

In this section we describe the Krylov subspace method for computing a matrix exponential, which takes advantage of the sparsity of the underlying matrix. Since normalizing $e^{\tau\alpha\mathbf{A}(\mathbf{v})}$ by $\mathbf{Tr}(e^{\tau\alpha\mathbf{A}(\mathbf{v})})$ is an $O(n)$ operations, we focus on bounding the complexity of computing the matrix exponential. We compute the matrix $e^{\tau\alpha\mathbf{A}(\mathbf{v})}$ one column at time. Thus, our problem reduces to computing $e^{\tau\alpha\mathbf{A}(\mathbf{v})}\mathbf{w}$ for some unit vector $\mathbf{w}$. The Krylov subspace methods are the most efficient numerical methods for computing $e^{\alpha\mathbf{A}(\mathbf{v})}\mathbf{w}$ (see Gallopoulos [9] and Hochbruck and Lubich [15]. For each $k = 1, \ldots, n$, the Krylov subspace

$$\mathcal{K}_k = \mathrm{span}\{\mathbf{v}, \mathbf{A}(\mathbf{v})\mathbf{w}, \mathbf{A}(\mathbf{v})^2\mathbf{w}, \ldots, \mathbf{A}(\mathbf{v})^{k-1}\mathbf{w}\}.$$

Let $\mathbf{Q}_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k] \in \mathbf{R}^{n\times k}$ denote any orthonormal basis for $\mathcal{K}_k$ with $\mathbf{q}_1 = \mathbf{w}$ such that

$$\mathbf{T}_k = \mathbf{Q}_k^\top\mathbf{A}\mathbf{Q}_k,$$

is a *tridiagonal* matrix, i.e. a matrix with non-zero elements only along the main diagonal and the principal subdiagonal. Using the Lanczos iterations [24], the complexity of computing $\mathbf{Q}_k$ is bounded by $O(k(r+n))$.

We approximate $e^{\tau\alpha\mathbf{A}(\mathbf{v})}\mathbf{w}$ by $\mathbf{Q}_p e^{\tau\alpha\mathbf{T}_k}\mathbf{e}_1$. Results in [15] establish that

$$\frac{\left\|e^{\tau\alpha\mathbf{A}(\mathbf{v})}\mathbf{w} - \mathbf{Q}_k e^{\tau\alpha\mathbf{T}_k}\mathbf{e}_1\right\|}{\left\|e^{\tau\alpha\mathbf{A}(\mathbf{v})}\mathbf{w}\right\|} \le 10e^{-\frac{k^2}{\omega\tau\rho\alpha}}, \qquad \sqrt{4\tau\rho\alpha} \le k \le 2\tau\rho\alpha, \tag{56}$$

where

$$\rho = \lambda_{\max}(\mathbf{A}(\mathbf{v})) - \lambda_{\min}(\mathbf{A}(\mathbf{v})) \le \omega/\tau + \omega \max_{i=1,\ldots,p}\{\lambda_{\max}(\mathbf{A}_i)\}$$

The next step is to compute $\mathbf{Q}_k e^{\alpha\mathbf{T}_k}\mathbf{e}_1$. Combining the results in [37] and [9], we have that the complexity of computing $\widehat{\mathbf{y}}$ such that

$$\frac{\left\|\widehat{\mathbf{y}} - \mathbf{Q}_k e^{\tau\alpha\mathbf{T}_k}\mathbf{e}_1\right\|}{\left\|\mathbf{Q}_k e^{\tau\alpha\mathbf{T}_k}\mathbf{e}_1\right\|} \le \eta$$

is bounded by $O\big(kn + k\log(1/\eta)\big)$.

Fix $\epsilon < 0.5$ and set $\eta = \frac{\epsilon}{15\sqrt{2}n^{3.5}}$. Then the bound (56) implies that

$$k_\eta = \sqrt{5n\rho\alpha\log\left(\frac{150\sqrt{2}n^{3.5}}{\epsilon}\right)} = O\left(\sqrt{n\log(n)}\sqrt{\frac{1}{\epsilon}\log\left(\frac{1}{\epsilon}\right)}\right).$$

Let $\widehat{\mathbf{y}}_i$ denote the vector computed by the above approximation procedure when the initial unit vector $\mathbf{v} = \mathbf{e}_i$. Let $\widehat{\mathbf{Y}} = [\widehat{\mathbf{y}}_1, \ldots, \widehat{\mathbf{y}}_n]$. Then, the choice of p implies that the complexity of computing $\widehat{\mathbf{Y}}$ is

$$O\Big((r + n + \log(\eta^{-1}))nk_\eta\Big) = O\left(rn\sqrt{n\log(n)}\sqrt{\frac{1}{\epsilon}\log\left(\frac{1}{\epsilon}\right)}\right).$$

Let $\mathbf{y}_i = e^{\alpha\tau\mathbf{A}(\mathbf{v})}\mathbf{e}_i$, $i = 1, \ldots, n$ and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$. Then $e^{\tau\alpha\mathbf{A}(\mathbf{v})} = \mathbf{Y}$. In the rest of this section, we show that using $\widehat{\mathbf{Y}}$ instead of $\mathbf{Y}$ results in at most an additional additive error ϵ. It is easy to show that, for all $i = 1, \ldots, n$, $\|\mathbf{y}_i - \widehat{\mathbf{y}}_i\| \leq 3\eta \|\mathbf{y}_i\|$. Thus, we have that

$$\frac{\left\|\mathbf{Y} - \widehat{\mathbf{Y}}\right\|}{\|\mathbf{Y}\|} \leq \frac{\sqrt{n}\left\|\mathbf{Y} - \widehat{\mathbf{Y}}\right\|_F}{\|\mathbf{Y}\|_F} \leq 3\eta\sqrt{n},$$

where $\|\cdot\|_F$ denote the Frobenius norm and $\|\cdot\|$ denotes the $\mathcal{L}_2$-norm. This bound also implies a bound on $\mathbf{Tr}(\widehat{\mathbf{Y}})$ as follows.

$$\begin{aligned}
\mathbf{Tr}(\widehat{\mathbf{Y}}) &= \mathbf{Tr}(\mathbf{Y}) + \mathbf{Tr}(\widehat{\mathbf{Y}} - \mathbf{Y}), \\
&\geq \mathbf{Tr}(\mathbf{Y}) - n\left\|\mathbf{Y} - \widehat{\mathbf{Y}}\right\|, \\
&\geq \mathbf{Tr}(\mathbf{Y}) - 3n^{\frac{3}{2}}\eta\|\mathbf{Y}\|, \\
&\geq (1 - 3n^{\frac{3}{2}}\eta)\,\mathbf{Tr}(\mathbf{Y}) > 0.
\end{aligned}$$

Let $\mathbf{X} = \frac{n\mathbf{Y}}{1+\mathbf{Tr}(\mathbf{Y})}$ and $\widehat{\mathbf{X}} = \frac{n\widehat{\mathbf{Y}}}{1+\mathbf{Tr}(\widehat{\mathbf{Y}})}$. Then we have that

$$\begin{aligned}
\left\|\mathbf{X} - \widehat{\mathbf{X}}\right\| &\leq \frac{n\left\|\mathbf{Y} - \widehat{\mathbf{Y}}\right\|}{1 + \mathbf{Tr}(\widehat{\mathbf{Y}})} + \frac{n|\,\mathbf{Tr}(\widehat{\mathbf{Y}} - \mathbf{Y})|}{1 + \mathbf{Tr}(\widehat{\mathbf{Y}})}, \\
&\leq \frac{3n^{1.5}(1+n)\eta}{(2 - 3n^{\frac{3}{2}}\eta)} \quad \leq \quad 3\tau\omega\eta.
\end{aligned}$$

The gradient of the smoothed function $f_\alpha(\mathbf{v})$ is given by $\mathbf{diag}(\mathbf{X}) - \mathbf{1}$. Thus, using $\widehat{\mathbf{X}}$ instead of $\mathbf{X}$ introduces an error that is bounded by

$$5n\left\|\mathbf{diag}(\mathbf{X}) - \mathbf{diag}(\widehat{\mathbf{X}})\right\| \leq 5n\left\|\mathbf{X} - \widehat{\mathbf{X}}\right\|_F \leq \sqrt{n}\left\|\mathbf{X} - \widehat{\mathbf{X}}\right\| \leq 15n^{3.5}\eta.$$

This error in the gradient results in an additional additive error that is bounded by

$$D\left\|\mathbf{diag}(\mathbf{X}) - \mathbf{diag}(\widehat{\mathbf{X}})\right\| = \sqrt{2}(15n^{3.5}\eta) \leq \epsilon,$$

where D denotes the diameter of the simplex $\{\mathbf{v} : \mathbf{v} \geq \mathbf{0}, \sum_i v_i \leq 1\}$. Thus, we have the following corollary.

Corollary 2. *Suppose* **SDP-Pack** *computes* $\mathbf{X}_u$ *via Lanczos iterations. Then the complexity of* **SDP-Pack** *is* $O\left(\epsilon^{-1}rn^{3.5}\log(n)\sqrt{\frac{1}{\epsilon^3}\log\left(\frac{1}{\epsilon}\right)}\right)$.

3.4.2 Improving Lanczos

A recent development by van den Eshof and Hochbruck [34], has resulted in a method which we refer to as SI-Lanczos, or the *shift-and-invert* Lanczos. The main improvement in [34] results from using Krylov subspaces generated by $(\mathbf{I} + \nu\mathbf{A})^{-1}$, where $\nu > 0$ (see also [15]). Therefore, at each iteration, we need to compute a solution to the linear system $\mathbf{y}_{k+1} = (\mathbf{I} + \nu\mathbf{A})\mathbf{y}_k$. SI-Lanczos works best when $\mathbf{A} \succ 0$, so we

MatExp($\mathbf{A}, n, \delta$)

1 $\mathbf{A} \leftarrow \mathbf{A} + (6n-2)\mathbf{I}$
  for $i = 1$ **to** n
  do
2 $\mathbf{E}_i \leftarrow$ SI-Lanczos($\mathbf{A}, \mathbf{e}_i$)
  return E

Figure 3: Procedure MatExp($\mathbf{A}, p$)

approximate the exponential of $\bar{\mathbf{A}} = \mathbf{A} + 2\theta\mathbf{I}$ for the Packing SDP where $\theta = \omega\lambda_{\max}(\sum_{i=1}^{p} \mathbf{A}_i$. This shift ensures that $\bar{\mathbf{A}} \succ 0$ and the condition number $\kappa(\bar{\mathbf{A}}) = \lambda_{\max}(\bar{\mathbf{A}})/\lambda_{\min}(\bar{\mathbf{A}}) \leq 2$. Note that this shift does *not* change the value of $\mathbf{X}_{\mathbf{v}}$ defined in (3.4). Since the condition number, $\kappa(\bar{\mathbf{A}})$, is bounded, SI-Lanczos can use the well-known conjugate gradient method (see, e.g., Golub and Van Loan [12]) to solve the system of linear equations required at each iteration. The conjugate gradient method computes solutions to linear systems that, at iteration k, have residual error bounded by $[(\sqrt{\kappa(\mathbf{A})}-1)/(\sqrt{\kappa(\mathbf{A})}+1)]^k$, which implies the convergence is geometric. Thus, this solves systems of linear equations in $O((n+m)\log(1/\epsilon))$ iterations where $\epsilon > 0$ is the relative error and m is the number of nonzeros in the matrix $\mathbf{A}$. Overall, Theorem 3.3 of [34] indicates that $O(\log^2(1/\epsilon))$ iterations are required to approximate each column of the exponential. This results an overall complexity of $O(n(n+r)\log^3(1/\epsilon))$. Thus, we have the following corollary.

Corollary 3. *The complexity of computing* $\mathbf{X}_u$ *via* SI-Lanczos *is* $O\big(n(n+r)\log^3(\frac{1}{\epsilon})\big)$. *Therefore, using* SI-Lanczos *to compute the matrix exponential in* **SDP-Pack** *results in a complexity of* $O\Big(\epsilon^{-1}n^2(n+r)\log(n)\log^3(\frac{1}{\epsilon})\Big)$.

4 Conclusions and future directions

In this paper, we have presented **SDP-Pack**, an algorithm that approximately solves a class of semidefinite programs that are significant in combinatorial optimization. **SDP-Pack** represents a synthesis of techniques from several related disciplines including non-smooth convex optimization and combinatorial optimization. We surveyed several key results in these areas in an effort to establish that similar techniques and results have arisen through (sometimes) different approaches. In particular, we investigated the use of potential functions to "smooth" objective functions that were not differentiable. We then presented a new class of semidefinite programs

Some interesting remaining questions include:

- Are there more tractable approximations of the error bound estimates? In particular, is there a spectral function that would avoid or simplify the matrix exponential calculation?
- Is it possible to improve on the current matrix exponential calculation? As the bottleneck of the algorithm, any improvement here would improve the overall running time of **SDP-Pack**. Some possible improvements include:
  - Leverage the fact that only the diagonal of the matrix exponential is required. Moreover, the underlying matrix whose exponential is required changes only according to the sparsity of the constraint matrices. One, or both, of these properties might allow for a useful approximation.
  - Utilize a different smoothing function. This may involve a tradeoff in the terms of calculation in the primal space, however.

- Is it possible to extend the class of SDPs that this method can solve? What are the properties of the Packing SDP that are necessary for these methods to work?
- What other useful SDPs are Packing SDPs?

References

[1] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, 2007.

[2] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings, and graph partitionings. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 222–231, 2004.

[3] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM J. Optim.*, 10(2):443–461 (electronic), 2000.

[4] D. Bienstock. *Potential function methods for approximately solving linear programming problems: theory and practice*. International Series in Operations Research & Management Science, 53. Boston, MA, 2002.

[5] D. Bienstock and G. Iyengar. Solving fractional packing problems in $O^*(\frac{1}{\epsilon})$ iterations. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 146–155, 2004.

[6] S. Burer and R. D. C. Monteiro. A projected gradient algorithm for solving the maxcut SDP relaxation. *Optim. Methods Softw.*, 15(3-4):175–200, 2001.

[7] Jane Cullum, W. E. Donath, and P. Wolfe. The minimization of certain nondifferentiable sums of eigenvalues of symmetric matrices. *Math. Programming Stud.*, (3):35–55, 1975. Nondifferentiable optimization.

[8] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Develop.*, 17:420–425, 1973.

[9] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM J. Sci. Statist. Comput.*, 13(5):1236–1264, 1992.

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

[11] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[12] Gene H. Golub and Charles F. Van Loan. *Matrix computations*, volume 3 of *Johns Hopkins Series in the Mathematical Sciences*. Johns Hopkins University Press, Baltimore, MD, 1983.

[13] M. D. Grigoriadis and L. G. Khachiyan. An exponential-function reduction method for block angular convex programs. *Networks*, 26:59–68, 1995.

[14] M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. In *Topics on perfect graphs*, volume 88 of *North-Holland Math. Stud.*, pages 325–356. North-Holland, Amsterdam, 1984.

[15] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, 1997.

[16] Roger A. Horn and Charles R. Johnson. *Topics in matrix analysis*. Cambridge University Press, Cambridge, 1985. 1999 reprint.

[17] G. Iyengar, D. J. Phillips, and C. Stein. Approximation algorithms for semidefinite packing problems with applications to maxcut and graph coloring. In *Proceedings of the 11th Conference on Integer Programming and Combinatorial Optimization*, pages 152–166, 2005.

[18] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming. *J. ACM*, 45(2):246–265, 1998.

[19] P. Klein and H-I Lu. Efficient approximation algorithms for semidefinite programs arising from MAX CUT and COLORING. In *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pages 338–347, New York, 1996. ACM.

[20] P. Klein and N. Young. On the number of iterations for Dantzig-Wolfe optimization and packing-covering approximation algorithms. In *Integer programming and combinatorial optimization (Graz, 1999)*, volume 1610 of *Lecture Notes in Comput. Sci.*, pages 320–327. Springer, Berlin, 1999.

[21] A. S. Lewis. Derivatives of spectral functions. *Math. Oper. Res.*, 21(3):576–588, 1996.

[22] L. Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(1):1–7, 1979.

[23] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Rev.*, 20(4):801–836, 1978.

[24] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.*, 45(1):3–49 (electronic), 2003.

[25] A. Nemirovski. Prox-method with rate of convergence $o(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems, 2003. Submitted to *SIAM Journal of Optimization*.

[26] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. A Wiley-Interscience Publication. John Wiley & Sons Inc., New York, 1983. Translated from the Russian and with a preface by E. R. Dawson, Wiley-Interscience Series in Discrete Mathematics.

[27] Yu. Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103:127–152, 2005.

[28] Yu. Nesterov and A. Nemirovski. *Interior-point polynomial algorithms in convex programming*, volume 13 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[29] D. J. Phillips. *Optimizing approximately: Theory and practice*. PhD thesis, Columbia University, New York, NY, February 2007.

[30] S. Plotkin, D. B. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20:257–301, 1995.

[31] R. Tyrrell Rockafellar. *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J., 1970.

[32] N. Z. Shor. The cut-off method with stretching of the space for the solution of convex programming problems. *Kibernetika (Kiev)*, (1):94–95, 1977.

[33] M. Skutella. Convex quadratic and semidefinite programming relaxations in scheduling,. *Journal of the ACM*, 48(2):206–242, 2001.

[34] J. van den Eshof and M. Hochbruck. Preconditioning lanczos approximations to the matrix exponential, 2004. To appear in SIAM J. of Sci. Comp.

[35] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

[36] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

[37] R. S. Varga. On higher order stable implicit methods for solving parabolic partial differential equations. *J. Math. and Phys.*, 40:220–231, 1961.

[38] D. B. Yudin and A. S. Nemirovskiĭ. Informational complexity and effective methods for the solution of convex extremal problems. *Èkonom. i Mat. Metody*, 12(2):357–369, 1976.

A Proving Lemma 1

We now present a proof of Lemma 1, which follows from some fundamental linear algebra. We first show some specific properties about the minimum eigenvalue function of a sum of matrices, which is a specific form of a *matrix pencil*. We define $\mathcal{S}^n$ to be the space of symmetric n-dimensional real matrices.

For a matrix $\mathbf{A}$, $\mathbf{diag}(\mathbf{A})$ is the vector $(a_{11}, \ldots, a_{nn})^\top$. For the vector $\mathbf{x}$, $\mathbf{diag}(\mathbf{x})$ is the diagonal matrix with $\mathbf{x}$ along the main diagonal. Thus, $\mathbf{diag}(\mathbf{diag}(\mathbf{x})) = \mathbf{x}$. Recall also that the trace of a square matrix $\mathbf{A} = (a_{ij})$ is simply:

$$\mathbf{Tr}(\mathbf{A}) := \sum_{i=1}^{n} a_{ii}. \tag{57}$$

The trace operator is additive. Consider the trace of two $n \times n$ matrices, $\mathbf{A}$ and $\mathbf{B}$:

$$\mathbf{Tr}(\mathbf{A} + \mathbf{B}) = \sum_{i=1}^{n} (a_{ii} + b_{ii}) = \mathbf{Tr}(\mathbf{A}) + \mathbf{Tr}(\mathbf{B}). \tag{58}$$

We need facts about the Frobenius inner product which follow from the following elementary properties of the trace of a matrix:

Lemma 11. *Suppose that $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{n \times n}$ are given matrices with components a_{ij}, b_{ij}, and c_{ij} respectively. Further suppose that $\alpha \in \mathbb{R}$ is given. Then:*

$$\mathbf{Tr}(\mathbf{AB}) = \mathbf{Tr}(\mathbf{BA}). \tag{59}$$

Moreover, the trace satisfies the cyclic property:

$$\mathbf{Tr}(\mathbf{ABC}) = \mathbf{Tr}(\mathbf{CAB}) = \mathbf{Tr}(\mathbf{BCA}). \tag{60}$$

Suppose that $\mathbf{U} \in \mathbb{R}^{n \times n}$ is an invertible matrix. Then

$$\mathbf{Tr}(\mathbf{A}) = \mathbf{Tr}(\mathbf{U}^{-1}\mathbf{AU}). \tag{61}$$

Finally,

$$\mathbf{Tr}(\alpha\mathbf{A}) = \alpha\,\mathbf{Tr}(\mathbf{A}). \tag{62}$$

Proof. To see (59), note that by ordinary matrix multiplication and (57),

$$\begin{aligned} \mathbf{Tr}(\mathbf{AB}) &= \sum_{i=1}^{n} \sum_{j=1}^{n} (a_{ij} b_{ji}), \\ &= \sum_{j=1}^{n} \sum_{i=1}^{n} (b_{ji} a_{ij}), \\ &= \mathbf{Tr}(\mathbf{BA}). \end{aligned} \tag{63}$$

Further, (59) directly implies:

$$\mathbf{Tr}((\mathbf{AB})\mathbf{C}) = \mathbf{Tr}((\mathbf{CA})\mathbf{B}) = \mathbf{Tr}(\mathbf{B}(\mathbf{CA})). \tag{64}$$

But (60), implies (61) since:

$$\mathbf{Tr}(\mathbf{U}\mathbf{A}\mathbf{U}^{-1}) = \mathbf{Tr}(\mathbf{U}^{-1}\mathbf{U}\mathbf{A}) = \mathbf{Tr}(\mathbf{A}). \tag{65}$$

(62) follows from (57) and distributivity. □

Another useful definition is when two matrices, $\mathbf{A}, \mathbf{B}$ are *similar*, which indicates that there exists an *orthogonal* matrix $\mathbf{U}$ (i.e., $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$), and

$$\mathbf{A} = \mathbf{U}^\top\mathbf{B}\mathbf{U}. \tag{66}$$

Note that if $\mathbf{A} = \mathbf{A}^\top$, then the matrix $\mathbf{A}$ has a eigenvalue-eigenvector decomposition, i.e., there exists $\mathbf{U}$ and Λ where $\mathbf{V}$ is an orthogonal matrix with the eigenvalues of $\mathbf{A}$ as columns and Λ is a diagonal matrix with the eigenvalues of $\mathbf{A}$ along the main diagonal. Moreover, if $\mathbf{B}$ is similar to $\mathbf{A}$ via the orthoganal matrix $\mathbf{U}$, then

$$\mathbf{B} = \mathbf{U}^\top\mathbf{A}\mathbf{U} = \mathbf{U}^\top(\mathbf{V}^\top\Lambda\mathbf{V})\mathbf{U},$$

which indicates that $\mathbf{B}$ has the same eigenvalues as $\mathbf{A}$ since $\mathbf{V}\mathbf{U}$ is itself orthogonal, and eigenvalues are unique up to permutation. We can now use Lemma 11 to prove a particular matrix optimization problem is the same as finding the maximum eigenvalue.

We can now use Lemma 11 to prove Lemma 1:

Proof of Lemma 11. Let $m(\mathbf{A}) := \max\{\max\{\langle \mathbf{X}, \mathbf{A}\rangle : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq 1\}\}$. Consider any $\mathbf{X} \in \{\mathbf{X} \succeq \mathbf{0} : \mathbf{Tr}(\mathbf{X}) \leq 1\}$. Consider the eigenvalue-eigenvector decomposition, $\mathbf{X} = \mathbf{V}^\top\Lambda\mathbf{V}$ where $\mathbf{V}$ is a matrix with the eigenvectors of $\mathbf{X}$ as columns, and Λ is a diagonal matrix with the eigenvalues of $\mathbf{X}$ as entries. Then, from the symmetry of $\mathbf{X}$ and Lemma 11,

$$\begin{aligned}
\langle \mathbf{X}, \mathbf{A}\rangle &= \mathbf{Tr}(\mathbf{X}^\top\mathbf{A}), \\
&= \mathbf{Tr}(\mathbf{X}\mathbf{A}), \\
&= \mathbf{Tr}(\mathbf{V}\mathbf{X}\mathbf{A}\mathbf{V}^\top), \\
&= \mathbf{Tr}(\Lambda\mathbf{V}\mathbf{A}\mathbf{V}^\top), \\
&= \sum_{i=1}^{n} \lambda_i(\mathbf{X})\bar{a}_{ii},
\end{aligned} \tag{67}$$

where $\bar{a}_{ii}$ are the main diagonal components of the matrix $\mathbf{V}\mathbf{A}\mathbf{V}^\top$. Since $\mathbf{V}\mathbf{A}\mathbf{V}^\top$ is similar to $\mathbf{A}$, their eigenvalues, and, therefore, trace are the same. If $\lambda_{\max}(\mathbf{A}) \leq 0$ then the $\mathbf{A} \preceq \mathbf{0}$ and so is $\mathbf{V}\mathbf{A}\mathbf{V}^\top$ whence the main diagonal of $\mathbf{V}\mathbf{A}\mathbf{V}^\top$ is nonpositive. Then, (67) indicates that $m(\mathbf{A}) = 0$.

Suppose then, that $\lambda_{\max}(\mathbf{A}) > 0$. In this case, there must be an i such that $\bar{a}_{ii} > 0$. But, consider the optimization problem:

$$\max\{\sum_{i=1}^{n} c_i x_i : \sum_{i=1}^{n} x_i \leq 1, x_i \geq 0\} = \max\{c_i\}, \tag{68}$$

where at least one $c_i > 0$. The optimal solution to (68) is to set $x_j = 1$ for $j = \arg\max_{i=1,\ldots,n} c_i$. Thus, if $\mathbf{V}$ were to be fixed, and λ_i were chosen so as to maximize (67), but still satisfy $\sum_{i=1}^{n} \lambda_i \leq 1$, then $\mathbf{X}$ should be chosen so that $\lambda_j = 1$ where $j = \arg\max_{i=1}^{n} \bar{a}_{ii}$. However, note that by the definition of eigenvalues, for any symmetric matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$,

$$\lambda_{\max}(\mathbf{B}) = \max_{\mathbf{x}}\{\mathbf{x}^\top\mathbf{B}\mathbf{x} | \mathbf{x}^\top\mathbf{x} = 1\}. \tag{69}$$

In particular, (69) implies that $\bar{a}_{jj} \leq \lambda_{\max}(\mathbf{A})$, for any choice of $\mathbf{X} = \mathbf{V}\Lambda\mathbf{V}$. Thus,

$$m(\mathbf{A}) \leq \lambda_{\max}(\mathbf{A}). \tag{70}$$

So, by setting $\mathbf{X} = \mathbf{u}_1\mathbf{u}_1^\top$, we have that $\langle \mathbf{X}, \mathbf{A}\rangle = \mathbf{Tr}(\mathbf{X}\mathbf{A}) = \mathbf{u}_1^\top\mathbf{A}\mathbf{u}_1 = \lambda_{\max}(\mathbf{A})$ which, by (70) is optimal. □

B Proofs of the Lemmas in Section 3

Proof of Lemma 5. The second-order Taylor series expansion of $f_\alpha(\mathbf{v})$ around $\mathbf{w}$ is:

$$f_\alpha(\mathbf{v}) = f_\alpha(\mathbf{w}) + \nabla f_\alpha(\mathbf{w})^\top(\mathbf{v}-\mathbf{w}) + R(\mathbf{v}),$$

where $R(\mathbf{v}) \leq \frac{\tau\omega}{2}\sup_{\mathbf{c}\in\mathcal{V}}(\mathbf{v}-\mathbf{w})^\top\nabla^2 f_\alpha(\mathbf{c})(\mathbf{v}-\mathbf{w})$. However, since $\nabla^2 f_\alpha(\mathbf{c}) \leq \mathbf{1}\mathbf{1}^\top$ for all $\mathbf{c}\in\mathcal{V}$, we have that:

$$R(\mathbf{v}) \leq \frac{\tau\omega}{2}\left\|\mathbf{v}-\mathbf{w}\right\|_1^2.$$

□

Proof of Lemma 6. Let $H(\mathbf{v}) = \sum_{i=0}^p v_i\log(v_i)$ for $\mathbf{v}\in\mathcal{V}$. Then H is a convex function of $\mathbf{v}$. Also, for any vector $\mathbf{w}\in\mathbb{R}^{p+1}$

$$\begin{aligned}
\mathbf{w}^\top\nabla^2 H(\mathbf{v})\mathbf{w} &= \sum_{i=0}^p \frac{w_i^2}{v_i}, \\
&\geq \Big(\sum_{i=0}^p \frac{w_i^2}{v_i}\Big)\Big(\sum_{i=0}^p v_i\Big), \\
&\geq (\sum_{i=0}^p \frac{|w_i|}{\sqrt{v_i}}\sqrt{v_i})^2, \qquad (71)\\
&= \left\|\mathbf{w}\right\|_1^2, \qquad (72)
\end{aligned}$$

where (71) follows from the Cauchy-Schwartz inequality $|\mathbf{w}^\top\mathbf{v}|^2 \leq \|\mathbf{w}\|\,\|\mathbf{v}\|$. From the second-order Taylor series expansion we have that for some $\theta\in[0,1]$

$$\begin{aligned}
H(\mathbf{v}) - H(\mathbf{v}') &= \nabla H(\mathbf{v}')^\top(\mathbf{v}-\mathbf{v}') + \frac{1}{2}(\mathbf{v}-\mathbf{v}')^\top\nabla^2 H(\theta\mathbf{v}+(1-\theta)\mathbf{v}')(\mathbf{v}-\mathbf{v}'), \\
&\geq \nabla H(\mathbf{v}')^\top(\mathbf{v}-\mathbf{v}') + \frac{1}{2}\left\|\mathbf{v}-\mathbf{v}'\right\|_1^2.
\end{aligned}$$

Simple algebra and the definition of $\mathcal{V}$ shows that the entropy distance

$$d(\mathbf{v},\mathbf{v}') = H(\mathbf{v}) - H(\mathbf{v}') - \nabla H(\mathbf{v}')^\top(\mathbf{v}-\mathbf{v}') \geq \frac{1}{2}\left\|\mathbf{v}-\mathbf{v}'\right\|_1^2. \qquad (73)$$

□

Proof of Lemma 7. Since $d(\mathbf{v},\mathbf{v}')\to\infty$ as one or more components of $\mathbf{v}$ approach zero, it follows that the optimal solution to (40) is in the strictly positive. The Lagrangian for (40) is

$$L(\mathbf{v},\beta,\boldsymbol{\rho}) := \gamma d(\mathbf{v},\mathbf{v}') + \mathbf{c}^\top\mathbf{v} + \beta(1-\sum_{i=0}^{p+1} v_i).$$

Note that we have added the slack variable v_{p+1} in order to to make the problem an equality constraint, with $v'_{p+1} = 1-\sum_{i=0}^p v'_i$ and $c_{p+1}=0$. Then, the KKT conditions are

$$\frac{\partial L}{\partial v_i} = \gamma\log(\frac{v_i}{v'_i}) + \gamma + c_i - \beta = 0,$$

which indicates that the optimal solution is

$$v_i^* = v'_i e^{-(c_i/\gamma+v'_i)}e^{\beta+\rho_i}, \quad i=0,\ldots,p+1,$$

where $\boldsymbol{\rho} \geq \mathbf{0}$ and satisfies the complementary slackness condition $\rho_i v_i^* = 0$, $i = 0, \ldots, p+1$. Since $\mathbf{v}' > \mathbf{0}$, it follows that $\boldsymbol{\rho} = \mathbf{0}$. Since $\sum_{i=0}^{p+1} v_i^* = 1$, we see that β should be chosen so that

$$\exp(\beta) = \frac{1}{\sum_{i=1}^{p+1} v_i' \exp(-(c_i/\gamma + v_i))}.$$

which implies that

$$v_i^* = \frac{v_i' e^{-(c_i/\gamma + v_i')}}{\sum_{j=0}^{p+1} v_i' e^{-(c_j/\gamma + v_j')}}, \quad i = 0, \ldots, p+1.$$

□

C Semidefinite Relaxation of Graph Coloring

In order to use our methods with (16), we dualize the additional $x_{ii} \leq 1$ constraints to obtain

$$\begin{array}{ll} \min & \sum_{i=1}^n y_i \\ \text{s.t.} & \sum_{j=1}^m v_j = 1, \\ & \mathbf{diag}(\mathbf{y}) + \mathcal{G}\mathbf{v} \succeq \mathbf{0}, \\ & \mathbf{y}, \mathbf{v} \geq \mathbf{0}, \end{array} \tag{74}$$

where $m = |E|$ denotes the number of edges in the graph, $\mathcal{G} : \mathbf{R}^m \mapsto \mathbf{R}^{n \times n}$ is the linear operator $\mathcal{G}\mathbf{v} = \frac{1}{2}\sum_{(i,j)\in E} v_{(i,j)} \mathbf{G}_{ij}$ (for convenience, we have scaled each of the $\mathbf{G}_{ij}$ by $1/2$.

We can define the saddle-point problem:

$$\max_{\mathbf{X} \in \mathcal{X}} \min_{(\mathbf{y},\mathbf{v}) \in \mathcal{Y} \times \mathcal{V}} \phi(\mathbf{X}, \mathbf{y}, \mathbf{v}),$$

where

$$\phi(\mathbf{X}, \mathbf{y}, \mathbf{v}) = n \sum_{i=1}^m y_i - n(n\mathbf{v} + \mathcal{G}\mathbf{v}) \bullet \mathbf{X}, \tag{75}$$

$$\begin{array}{rcl} \mathcal{X} & = & \{\mathbf{X} : \mathbf{X} \succeq \mathbf{0}, \mathbf{Tr}(\mathbf{X}) \leq 1\}, \\ \mathcal{Y} & = & \{\mathbf{y} : \mathbf{y} \geq 0, \sum_{i=1}^n y_i \leq 1\}, \\ \mathcal{V} & = & \left\{\mathbf{v} : \mathbf{v} \geq \mathbf{0}, \sum_{j=1}^m v_j = 1\right\}. \end{array} \tag{76}$$

Theorem 3. *Fix $\epsilon > 0$. Suppose $(\bar{\mathbf{X}}, (\bar{\mathbf{y}}, \bar{\mathbf{v}})) \in \mathcal{X} \times (\mathcal{Y} \times \mathcal{V})$ is an ϵ-saddle-point. Let $\bar{\zeta} = -n \max_{(i,j)\in E}\{\bar{x}_{ij}\}$, $\bar{d} = n \max_{i=1,\ldots,n}\{\bar{x}_{ii}\}$,*

$$\widehat{\mathbf{y}} = n\bar{\mathbf{y}} - \big(\min\{\lambda_{\min}(\mathbf{diag}(\bar{\mathbf{y}}) + \mathcal{G}\bar{\mathbf{v}}), 0\}\big)\mathbf{1}, \qquad \widehat{\mathbf{v}} = \bar{\mathbf{v}}, \tag{77}$$

and

$$\begin{array}{rcl} \widehat{x}_{ii} & = & \begin{cases} n\bar{x}_{ii}, & \bar{d} \leq 1, \\ \frac{n\bar{x}_{ii}}{\bar{d}}, & \bar{d} > 1, \end{cases} \\ \widehat{x}_{ij} & = & \begin{cases} n\bar{x}_{ij}, & \bar{d} \leq 1, \bar{\zeta} > 0, \\ \frac{n\bar{x}_{ij}}{\bar{d}}, & \bar{d} > 1, \bar{\zeta} > 0, \\ 0, & \bar{\zeta} \leq 0. \end{cases} \\ \widehat{\zeta} & = & -\max_{(i,j)\in E}\{\widehat{x}_{ij}\}. \end{array} \tag{78}$$

Then $(\widehat{\zeta}, \widehat{\mathbf{X}})$ and $(\widehat{\mathbf{y}}, \widehat{\mathbf{v}})$ are an ϵ-optimal primal-dual pair for (16) and (74).

Proof. One can check that $(\widehat{\mathbf{y}}, \widehat{\mathbf{v}})$ defined in (77) is feasible for the dual SDP (74). From the definition of $\mathcal{X}$, it follows that

$$\max_{\mathbf{X}\in\mathcal{X}} \phi(X, \bar{\mathbf{y}}, \bar{\mathbf{v}}) = n\sum_{i=1}^{m} y_i - n\big(\min\{\lambda_{\min}(\mathbf{diag}(\bar{\mathbf{y}}) + n\mathcal{G}\bar{\mathbf{v}}), 0\}\big) = \sum_{i=1}^{n} y_i. \tag{79}$$

The pair $(\widehat{\zeta}, \widehat{\mathbf{X}})$ defined in (78) is feasible for the primal SDP (16). Next, we show that the objective value $\widehat{\zeta}$ is lower bounded by

$$\min_{(\mathbf{y},\mathbf{v})\in\mathcal{Y}\times\mathcal{V}} \phi(\bar{\mathbf{X}}, \mathbf{y}, \mathbf{v}) = \bar{\zeta} - n\max\{\bar{d} - 1, 0\}. \tag{80}$$

(a) $\bar{d} \le 1, \bar{\zeta} > 0$: In this case $(\widehat{\zeta}, \widehat{\mathbf{X}}) = (\bar{\zeta}, \bar{\mathbf{X}})$, and $\widehat{\zeta} = \bar{\zeta} = \bar{\zeta} - n\max\{\bar{d}-1, 0\}$, since $\bar{d} \le 1 \Rightarrow \max\{\bar{d}-1, 0\} = 0$.

(b) $\bar{d} \ge 1, \bar{\zeta} > 0$: In this case, $\widehat{\zeta} = \frac{\bar{\zeta}}{\bar{d}} \ge \bar{\zeta}\big(1 - (\bar{d} - 1)\big) \ge \bar{\zeta} - n(\bar{d} - 1)$, where the first inequality follows from the fact that $\frac{1}{d} \ge 1 - (d - 1)$ for all $d > 0$, and the second inequality follows from the bound $|\bar{x}_{ij}| \le \sqrt{\bar{x}_{ii}\bar{x}_{jj}} \le 1$.

(c) $\bar{\zeta} \le 0$. In this case, $\widehat{\zeta} = 0 \ge \bar{\zeta} + n\max\{\bar{d} - 1, 0\}$.

Thus, we have established the existence of an ϵ-optimal primal-dual pair $(\widehat{\lambda}, \widehat{\mathbf{X}})$ and $(\widehat{\mathbf{y}}, \widehat{\mathbf{v}})$. □

D Trace equality constraints

Many problems of interest, such as the Lovász ϑ-function, impose an explicit constraint of form $\mathbf{Tr}(\mathbf{X}) = \tau$. As in the case with vector packing, we cannot find exactly feasible solutions to equality constraints and we show how the methods described in Section 3.1 can slightly modified to return ϵ-feasible solutions to Packing SDPs with the additional trace constraint

$$\mathbf{Tr}(\mathbf{X}) = \tau \Leftrightarrow \mathbf{Tr}(\mathbf{X}) \le \tau, \mathbf{Tr}(\mathbf{X}) \ge \tau.$$

Note that the equality constraint is actually a packing *and* a covering constraint and is difficult to satisfy using vector packing and covering techniques (see [30]). As shown in Lemma 12 below, the runtime of **SDP-Pack** increases by a factor of $O(1/\epsilon)$. The analogous primal-dual pair of (8) and (9) in this case is given by

$$\begin{array}{llll} \rho^* = \max & \langle \mathbf{C}, \mathbf{X}\rangle & \eta^* = \min & \tau v_0 + \sum_{i=1}^{p} v_i \\ (P) & \mathbf{Tr}(\mathbf{X}) = \tau & & v_0\mathbf{I} + \sum_{i=1}^{p} v_i\mathbf{A}_i \succeq \mathbf{C} \quad (D) \\ & \langle \mathbf{A}_i, \mathbf{X}\rangle \le 1 & & v_i \ge 0, i = 1, \ldots, p \\ & \mathbf{X} \succeq \mathbf{0} & & \end{array}$$

In contrast to (9), note that v_0 is unrestricted. Note that we still have strong duality, but now the interior point is in (D) rather than (P). In particular, let $v_i = 1$ for $i = 1, \ldots, p$ and set

$$v_0 = |\lambda_{\max}(\mathbf{C} - \sum_{i=1}^{p} \mathbf{A}_i)| + \epsilon,$$

where $\epsilon > 0$. We also let $M := \lambda_{\max}(\mathbf{C})$ and assume there exists $\widehat{\rho} > 0$ such that $\widehat{\rho} \le \rho^*$. We then consider the following saddle point problem:

$$\min_{\mathbf{X}\in\mathcal{Q}} \max_{\mathbf{v}\in\mathcal{D}} \phi(\mathbf{X}, \mathbf{v}), \tag{81}$$

where

$$\mathcal{Q} = \{\mathbf{X} \succeq \mathbf{0} : \mathbf{Tr}(\mathbf{X}) = 1\}, \quad \mathcal{D} = \{\mathbf{v} \ge \mathbf{0} : \sum_{i=0}^{p} v_i \le 1\},$$

and

$$\phi(\mathbf{X}, \mathbf{v}) = \tau \langle \mathbf{C}, \mathbf{X} \rangle - \tau \left\langle \omega \sum_{i=1}^{p} v_i \mathbf{A}_i, \mathbf{X} \right\rangle + \omega \sum_{i=1}^{p} v_i.$$

In this case, we have that

$$\begin{aligned} \varphi_*(\mathbf{X}) &:= \min_{\mathbf{v} \in \mathcal{D}} \phi(\mathbf{X}, \mathbf{v}) = \tau \langle \mathbf{C}, \mathbf{X} \rangle - \omega \max\{(\bar{d} - 1), 0\}, \\ f_*(\mathbf{v}) &:= \max_{\mathbf{X} \in \mathcal{Q}} \phi(\mathbf{X}, \mathbf{v}) = \omega \sum_{i=1}^{p} v_i + \tau \lambda_{\max}(\mathbf{C} - \omega \sum_{i=1}^{p} v_i \mathbf{A}_i), \end{aligned}$$

where $\bar{d} = \max_{i=1,\ldots,p}\{\langle \mathbf{A}_i, \mathbf{X} \rangle\}$. Note that the trace constraint has *not* been dualized.

Lemma 12. *Suppose, for a given $\epsilon > 0$, $\omega = \max\{1, \frac{\tau M - \widehat{\rho}}{2\epsilon}\}$ that $(\bar{\mathbf{X}}, \bar{\mathbf{v}})$ are primal-dual $\epsilon/2$-optimal pairs to (81). Then let*

$$\widehat{\mathbf{X}} = \tau \bar{\mathbf{X}}, \quad \widehat{u}_i = \begin{cases} \lambda_{\max}(\mathbf{C} - \sum_{i=1}^{p} \omega \bar{u}_i \mathbf{A}_i), & i = 0, \\ \omega \bar{u}_i, & i > 0. \end{cases}$$

Then $\widehat{\mathbf{X}}$ is ϵ-feasible and $\widehat{\mathbf{v}}$ feasible to (D). Moreover,

$$\left\langle \mathbf{C}, \widehat{\mathbf{X}} \right\rangle \geq \tau \widehat{u}_0 + \sum_{i=1}^{p} \widehat{u}_i + \epsilon.$$

Proof. Assume that $\widehat{\mathbf{X}}$ is *not* feasible, whence $\max\{0, \max_{i=1,\ldots,p} \left\langle \mathbf{A}_i, \widehat{\mathbf{X}} \right\rangle\} = \max_{i=1,\ldots,p} \left\langle \mathbf{A}_i, \widehat{\mathbf{X}} \right\rangle$. Then, based on the definition of $\widehat{\mathbf{X}}$ and since $(\bar{\mathbf{X}}, \bar{\mathbf{v}})$ are ϵ-saddle points, it follows that

$$\varphi_*(\bar{\mathbf{X}}) = \left\langle \mathbf{C}, \widehat{\mathbf{X}} \right\rangle - \omega(\max_{i=1,\ldots,p} \left\langle \mathbf{A}_i, \widehat{\mathbf{X}} \right\rangle - 1) \geq f_*(\bar{\mathbf{v}}) - \epsilon/2,$$

Thus,

$$\left\langle \mathbf{A}_i, \widehat{\mathbf{X}} \right\rangle \leq \max_{i=1,\ldots,p} \langle \mathbf{A}_i, \mathbf{X} \rangle \leq 1 + \frac{1}{\omega}(\langle \mathbf{C}, \bar{\mathbf{X}} \rangle - f_*(\bar{\mathbf{v}}) + \epsilon/2) \leq 1 + \frac{\tau M + \epsilon/2}{\omega} \leq 1 + \epsilon,$$

by the definition of ω, so $\widehat{\mathbf{X}}$ is ϵ-feasible. Moreover, based on the definition of $\widehat{\mathbf{v}}$, and whether $\widehat{\mathbf{X}}$ is feasible or not,

$$\left\langle \mathbf{C}, \widehat{\mathbf{X}} \right\rangle \geq f_*(\bar{\mathbf{v}}) - \epsilon/2 + \omega \max\{(\max_{i=1,\ldots,p} \left\langle \mathbf{A}_i, \widehat{\mathbf{X}} \right\rangle - 1), 0\} \geq f_*(\bar{\mathbf{v}}) - \epsilon/2 = \sum_{i=1}^{p} \widehat{u}_i + \tau \widehat{u}_0 - \epsilon/2.$$

□

To use Lemma 12 our methods must return a solution $\mathbf{X}$ which has $\mathbf{Tr}(\mathbf{X}) = \tau$. We show this to be true via (85) below. The Lagrangian function defined in (17) remains the same, but now the domain is $\{\mathbf{X} \succeq \mathbf{0} : \mathbf{Tr}(\mathbf{X}) = \tau\}$. Thus, we are interested in the optimization problem

$$\max\{\varphi_*(\mathbf{X}) : \mathbf{Tr}(\mathbf{X}) = \tau, \mathbf{X} \succeq \mathbf{0}\}. \tag{82}$$

We redefine

$$f_*(\mathbf{v}) := \sum_{i=1}^{p} v_i + \max_{\mathbf{X} \succeq \mathbf{0} : \mathbf{Tr}(\mathbf{X}) = \tau} \{\langle (\mathbf{C} - \sum_{i=1}^{p} v_i \mathbf{A}_i), \mathbf{X} \rangle\}. \tag{83}$$

Strong duality implies that we can solve

$$\min_{\mathbf{v} \geq \mathbf{0}} f_*(\mathbf{v}) \tag{84}$$

instead. Since f_* is nondifferentiable, we approximate f_* by the differentiable function The rest of the method described in Section 3.1 remains the same except for one helpful change. Now, (42) still holds, but with

$$\mathbf{X}_{\mathbf{v}} = \frac{e^{\alpha\tau\left(\mathbf{C}-\omega\sum_{i=1}^{p} v_i\mathbf{A}_i\right)}}{\left(\sum_{i=1}^{p} e^{\alpha\tau\lambda_i}\right)} = \frac{\sum_{i=1}^{n} e^{\alpha\tau\lambda_i}\mathbf{u}\mathbf{u}^\top}{\left(\sum_{i=1}^{p} e^{\alpha\tau\lambda_i}\right)} \tag{85}$$

which implies the following lemma.

Lemma 13. *Suppose f_α is defined as in* (34)*. Then, $\nabla f_\alpha(\mathbf{v})$ is given by* (42) *with $\mathbf{X}_{\mathbf{v}}$ described in* (85)*, whence $\mathbf{Tr}(\tau\mathbf{X}_{\mathbf{v}}) = \tau$.*