An SDP Primal-Dual Algorithm for Approximating the Lovász-Theta Function

T-H. Hubert Chan
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
Email: hchan@mpi-inf.mpg.de

Kevin L. Chang
Yahoo Labs
Santa Clara, California, USA
Email: kevin.chang@yahoo-inc.com

Rajiv Raman
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
Email: rraman@mpi-inf.mpg.de

***Abstract*—The Lovász ϑ-function [Lov79] on a graph $G = (V, E)$ can be defined as the maximum of the sum of the entries of a positive semidefinite matrix $\mathbf{X}$, whose trace $\mathbf{Tr}(\mathbf{X})$ equals 1, and $\mathbf{X}_{ij} = 0$ whenever $\{i, j\} \in E$. This function appears as a subroutine for many algorithms for graph problems such as maximum independent set and maximum clique. We apply Arora and Kale's primal-dual method for SDP to design an approximate algorithm for the ϑ-function with an additive error of $\delta > 0$, which runs in time $O(\frac{\alpha^2 n^2}{\delta^2} \log n \cdot M_e)$, where $\alpha = \vartheta(G)$ and $M_e = O(n^3)$ is the time for a matrix exponentiation operation. Moreover, our techniques generalize to the *weighted* Lovász ϑ-function, and both the maximum independent set weight and the maximum clique weight for vertex weighted perfect graphs can be approximated within a factor of $(1+\epsilon)$ in time $O(\epsilon^{-2} n^5 \log n)$.**

I. Introduction

The Lovász ϑ-function [Lov79] on a graph $G = (V, E)$ can be defined as the maximum of the sum of the entries of an $n \times n$ positive semidefinite matrix $\mathbf{X}$, such that $\mathbf{Tr}(\mathbf{X}) = 1$ and $\mathbf{X}_{ij} = 0$ for $\{i, j\} \in V$, where $n = |V|$ and $m = |E|$. The ϑ-function is a fundamental concept in graph theory that was first defined by Lovász [Lov79] in 1979 to compute the *Shannon capacity* of the pentagon graph C_5. When the input graph is *perfect*, the ϑ-function coincides with the maximum independent set size $\alpha(G)$ of G. Grötschel, Lovász, and Schrijver showed in 1988 that the ϑ-function can be computed in polynomial time using the ellipsoid method [GLS87], thereby obtaining polynomial time algorithms for computing maximum independent sets, maximum cliques and minimum colorings of perfect graphs. In particular, the weighted version of Lovász ϑ-function is an important subroutine in computing minimum colorings of perfect graphs. Despite years of research on perfect graphs and the resolution of the strong perfect graph conjecture [CRST06], the algorithm of Grötschel, Lovász, and Schrijver (via the computation of the ϑ-function) is the only algorithm known to compute maximum independent sets of perfect graphs. The algorithm's reliance on semidefinite program solvers implies a prohibitive running time: a central open question in the theory of perfect graphs is therefore the development of fast algorithms to compute maximum independent sets and minimum colorings of perfect graphs [Sey06].

In this paper, we use the primal-dual framework of Arora and Kale [AK07] to give fast algorithms to approximate the ϑ-function of a general graph; this approximation algorithm will then be sufficient for exact computation of the maximum independent set size $\alpha(G)$ of a perfect graph. For the range $\alpha(G) \leq o(n^{3/4})$, our algorithm is the fastest among existing methods to approximate the weighted Lovász ϑ-function with constant additive error. The main theorem we prove in this paper is as follows.

Theorem I.1 Approximating the Lovász ϑ-Function *Suppose the Lovász ϑ-function on a given graph G is $\alpha = \vartheta(G)$. Then, the value $\vartheta(G)$ can be approximated within an additive error of $\delta > 0$ in time $O(\frac{\alpha^2 n^2 \log n}{\delta^2} \cdot M_e)$, where $M_e = O(n^3)$ is the time to perform a matrix exponentiation. Setting $\epsilon := \frac{\delta}{\alpha}$, a multiplicative $(1+\epsilon)$-approximation can be obtained in time $O(\frac{n^2 \log n}{\epsilon^2} \cdot M_e)$.*

Our techniques generalize to the *weighted* Lovász ϑ-Function as well. Hence, we can obtain approximation algorithms for the weighted versions of maximum independent set and maximum clique.

Corollary I.2 Approximating the Weighted Lovász ϑ-Function *Given a non-negative weight function $w : V \to \mathbb{R}_+$ on the vertices of a perfect graph $G = (V, E)$, the maximum independent set weight and the maximum clique weight can be approximated within a multiplicative factor of $(1 + \epsilon)$ in time $O(\epsilon^{-2} n^5 \log n)$.*

Our Techniques. Out of the various formulations [GLS87] of the Lovász ϑ-function, we utilize the one for which Arora and Kale's primal-dual method [AK07] for semidefinite programming can be readily applied. The main technical difficulty here is to design an Oracle with small width that, given some "candidate" primal solution $\mathbf{X}$ (feasible or not), decides whether to improve $\mathbf{X}$ and return an appropriate primal feasible solution, or to return a dual solution that can be used to improve $\mathbf{X}$. Thus, given candidate primal solution $\mathbf{X}$, our Oracle checks how far it is from feasible by inspecting an "error" matrix $\mathbf{X}_E$. The magnitude of the error is measured by the Frobenius norm $\|\mathbf{X}_E\|_F$, and if it is less than some threshold, then the Oracle performs a rounding procedure on $\mathbf{X}$ to return a feasible primal solution $\widehat{\mathbf{X}}$. Otherwise, the Oracle returns a dual solution $\mathbf{Y}$, which is a scaled version of $\mathbf{X}_E$.

A. History of the ϑ-Function and Perfect Graphs

The Shannon capacity of a graph [Sha56], is defined as $\lim_{n\to\infty} \sqrt[n]{\alpha(G^n)}$, where G^n is the *strong product* of G with itself n times[1], and $\alpha(G)$ denotes the size of a maximum independent set in G. In general,

$$\lim_{n\to\infty} \sqrt[n]{\alpha(G^n)} \geq \alpha(G). \tag{1}$$

For general graphs, there are no known algorithms (even exponential time) to compute the Shannon capacity. In a breakthrough paper [Lov79], Lovász computed the Shannon capacity of C_5 exactly by introducing the ϑ-function. The ϑ-function of a graph is an upper bound on the Shannon capacity, but is exact for C_5. Further, the ϑ-function can be computed in polynomial time via semi-definite programming (See Knuth's paper [Knu94] for a deeper discussion on the ϑ-function). Shannon observed that the class of graphs for which equality holds in (1) are exactly those graphs for which $\alpha(H) = \chi(\overline{H})$ holds for all induced subgraphs H of G. where $\overline{G}$ denotes the complement graph of G, and $\chi(\overline{G})$ is its chromatic number. While both $\chi(\overline{G})$ and $\alpha(G)$ are inapproximable, it was shown by Lovász that $\alpha(G) \leq \vartheta(G) \leq \chi(\overline{G})$.

Recently, the recognition problem of perfect graphs was shown to be polynomial time solvable [CCL$^+$05]. Despite this progress on the understanding of the structural aspects of perfect graphs, no method faster than that of Grötschel, Lovász, and Schrijver is known for computing maximum independent sets or optimal colorings of perfect graphs. A central open question in the theory of perfect graphs is the development of fast algorithms for such computations [Sey06].

B. Related work

Primal-Dual Algorithms for Solving Mathematical Programming Problems. Many primal-dual algorithms for approximately solving linear programs have appeared in the algorithms literature (see the survey paper [Vaz95] by Vazirani). For certain types of linear programs, these primal-dual algorithms can be faster than general-purpose LP solvers based on the ellipsoid or interior point methods [Ali95]. Some of these algorithms are loosely based on the multiplicative update algorithm associated with the weighted majority algorithm [AHK]. Plotkin, Shmoys and Tardos [PST91] gave the first such approximation algorithm for linear programming. The running time of this algorithm depends upon a quantity known as the width, which is a property of the specific problem at hand. For problems with sufficiently small width, their framework can be an improvement over general LP solvers. For the important special case of packing and covering linear programs, Garg and Könemann [GK98] gave fast approximation algorithms that were later improved by Koufogiannakis and Young [KY07].

A few studies have generalized the above primal-dual techniques to solving semidefinite programs. Klein and Lu [KL96] showed that the framework from [PST91] can be applied to solve SDP relaxations of Max Cut and Vector Coloring faster than general SDP solvers based on interior point algorithms. Arora, Hazan and Kale [AHK04] suggested primal-dual algorithms for solving SDPs related to sparsest cut, and later Arora and Kale [AK07] presented a framework for solving general SDPs, on which our algorithm is based.

Approximating the Lovász ϑ-Function. As noted by Klein and Lu [KL96], the straightforward approach[2] to solve the SDP takes time $O(\sqrt{n}m^3)$, which can be $O(n^{6.5})$ for dense graphs. In the same paper [KL96], they used the framework [PST91] by Plotkin et al. to approximate the Vector Coloring SDP, whose optimal solution is coincides with the Lovász ϑ-function on the complement graph in the case where graph is perfect. To obtain a multiplicative error of $(1+\epsilon)$ for the ϑ-function on perfect graphs, the running time using their algorithm is $\tilde{O}(\epsilon^{-5}\alpha^3 n\overline{m})$,[3] where $\alpha = \vartheta(G)$ and $\overline{m}$ is the number of edges in the complement graph. Iyengar et al. [IPS] has improved the time to approximate the Vector Coloring SDP to $\tilde{O}(\epsilon^{-1}n^2\overline{m})$.

It would seem that for perfect graphs, it would be better to consider the Vector Coloring SDP than the Lovász ϑ-function. However, in applications like finding the optimal coloring of perfect graphs [GLS87], the weighted version of the ϑ-function is used. Unfortunately, there is no straightforward method to generalize the Vector Coloring SDP to the weighted version. Hence, to approximate the weighted Lovász ϑ-function, our algorithm performs better than the straightforward approach when $\alpha = o(n^{3/4})$.

Iyengar et al. [IPS] considered subgradient methods for approximating the ϑ-function SDP, which runs in time $O(\epsilon^{-2}\log^3(\epsilon^{-1})n^4\log n)$ in the worst case. However, their methods can only give nearly feasible solutions due to the equality trace constraint, and hence their result does not compare directly with ours.

II. Notation and Preliminaries

Given an undirected graph $G = (V, E)$ of vertex size $n = |V|$, let $\alpha(G)$ denote its maximum independent set size, and $\overline{\chi}(G) := \chi(\overline{G})$ denote the chromatic number of its complement graph. All the matrices in this paper are symmetric and have dimensions $n\times n$. The sum of the diagonal entries of a square matrix $\mathbf{X}$ is denoted by the trace $\mathbf{Tr}(\mathbf{X})$. Given two matrices $\mathbf{X}$ and $\mathbf{Y}$, let $\mathbf{X} \bullet \mathbf{Y} := \mathbf{Tr}(\mathbf{X}^T\mathbf{Y})$, where $\mathbf{X}^T$ is the transpose of $\mathbf{X}$. We index the rows and the columns of a matrix with the vertices V of the graph G. The Lovász ϑ-function [Lov79] can be formulated as the solution to a semidefinite program.

[1] For graphs G, H, the strong product graph $G \times H$ is defined as follows. Its vertex set is the cartesian product of the vertex sets of G and H. The edges $\{(u,a),(v,b)\} \in E(G \times H)$ are all edges such that $\{u,v\} \in E(G)$ and $\{a,b\} \in E(H)$, or $u = v$ and $\{a,b\} \in E(H)$, or $\{u,v\} \in E(G)$ and $a = b$

[2] In Alizadeh's paper [Ali95], it is stated that the interior point method for SDP takes $O(\sqrt{n})$ iterations; and Nayakkankuppam and Overton [NO96] explicitly analyzed the running time of each iteration to be $O(m^3)$.

[3] We use the notation $\tilde{O}(\cdot)$ to suppress logarithmic factors.

Definition II.1 (Lovász ϑ-Function) *Given a graph $G = (V, E)$, the Lovász ϑ-function $\vartheta(G)$ on G is the optimal value of the following semidefinite program, together with its dual.*

$$\begin{array}{lrcl} \text{(P)} & \max \mathbf{J} \bullet \mathbf{X} & & \\ & \mathbf{I} \bullet \mathbf{X} & = & 1 \\ & \forall \{i,j\} \in E : \mathbf{E}_{ij} \bullet \mathbf{X} & = & 0 \\ & \mathbf{X} & \succeq & 0 \end{array}$$

$$\begin{array}{lrcl} \text{(D)} & \min z & & \\ & z\,\mathbf{I} + \sum_{\{i,j\} \in E} y_{ij} \mathbf{E}_{ij} & \succeq & \mathbf{J} \\ & z,\, y_{ij} & \in & \mathbb{R} \end{array}$$

Here, the identity matrix is denoted by $\mathbf{I}$, and $\mathbf{J}$ is the matrix in which every entry is 1. For each edge $\{i,j\} \in E$, $\mathbf{E}_{ij}$ is the matrix in which both the (i,j)-th and (j,i)-th entries are 1, and every other entry is 0. For ease of notation, we sometimes write $\mathbf{Y} := \sum_{\{i,j\} \in E} y_{ij} \mathbf{E}_{ij}$. We say that $\mathbf{X}$ is a primal solution, and $(z, \mathbf{Y})$ is a dual solution, though not necessarily feasible.

There are alternative formulations for $\vartheta(G)$; for example, it is equivalent to the strict vector chromatic number [KMS98] of the graph's complement $\overline{G}$. It is well-known that the Lovász ϑ-function satisfies the Sandwich Property: $\alpha(G) \leq \vartheta(G) \leq \overline{\chi}(G)$; in the case where G is a perfect graph, both equalities hold. Recall that a graph G is *perfect* if for all its induced subgraphs H, $\alpha(H) = \overline{\chi}(H)$.

Given a matrix $\mathbf{X}$, we denote its operator norm by $\|\mathbf{X}\|_2$, which is also the largest eigenvalue of $\mathbf{X}$ in absolute value. We denote its Frobenius norm by $\|\mathbf{X}\|_F := \sqrt{\sum_{ij} \mathbf{X}_{ij}^2} = \sqrt{\mathbf{X} \bullet \mathbf{X}}$. The following proposition states some well-known properties about the mentioned norms.

Proposition II.2 *Suppose $\mathbf{X}$ and $\mathbf{Y}$ are square matrices.*

(a) **Cauchy-Schwarz Inequality.** *We have $\mathbf{X} \bullet \mathbf{Y} \leq \|\mathbf{X}\|_F \|\mathbf{Y}\|_F$. If there exists $\kappa \geq 0$ such that $\mathbf{Y} = \kappa \mathbf{X}$, then equality holds.*

(b) We have $\|\mathbf{X}\|_2 \leq \|\mathbf{X}\|_F$.

III. Arora and Kale's Primal-Dual Method

We describe how we apply Arora and Kale's framework[4] [AK07] to approximate the SDP stated in Definition II.1 with arbitrarily small additive error $\delta > 0$. The algorithm uses binary search, and given a candidate value α, each binary step either produces a primal feasible solution $\mathbf{X}$ with value at least α (thereby producing a lower bound for the optimal value) or a dual feasible solution $(z, \mathbf{Y} := \sum_{\{i,j\} \in E} y_{ij} \mathbf{E}_{ij})$ with value at most $\alpha + \delta$ (and hence giving an upper bound for the optimal value). We next describe how this step is performed for a fixed α.

Oracle with width ρ. Arora and Kale's method requires an auxiliary algorithm known as the Oracle, whose specification in our case is as follows. Given $\mathbf{W} \succeq 0$,[5] the Oracle either (1) produces a primal feasible solution $\widehat{\mathbf{X}}$ with value at least α; or (2) produces a dual solution $(z, \mathbf{Y})$ (not necessarily feasible) such that $\mathbf{W} \bullet (z\,\mathbf{I} + \mathbf{Y} - \mathbf{J}) \geq 0$, and $z = \alpha$.[6] The Oracle is said to have *width* ρ if any dual solution $(z, \mathbf{Y})$ that it produces satisfies $\|z\,\mathbf{I} + \mathbf{Y} - \mathbf{J}\|_2 \leq \rho$. The width ρ depends on the current tested value α, and we sometimes use the subscript ρ_α to make this dependence explicit. The details of the Oracle are given in Section V.

```
Primal-Dual Algorithm for the Lovász $\vartheta$-Function SDP
Set $\mathbf{W}^{(1)} := \mathbf{I}$. Let $\varepsilon := \frac{\delta}{2\rho}$, $\varepsilon' := -\ln(1-\varepsilon)$ and $T := \frac{8\rho^2 \ln n}{\delta^2}$.
For $t$ from 1 to $T$, do:
  1. Run the Oracle with $\mathbf{W}^{(t)}$.
  2. If the Oracle returns a primal feasible solution $\widehat{\mathbf{X}}$,
     output $\widehat{\mathbf{X}}$ and terminate.
  3. Otherwise, let $\mathbf{Y}^{(t)} := \sum_{\{i,j\} \in E} y_{ij}^{(t)} \mathbf{E}_{ij}$ be the dual
     solution returned by the Oracle.
  4. Let $\mathbf{M}^{(t)} := (\alpha \mathbf{I} + \mathbf{Y}^{(t)} - \mathbf{J} + \rho \mathbf{I})/2\rho$.
  5. Compute $\mathbf{W}^{(t+1)} := (1-\varepsilon)^{\sum_{\tau=1}^{t} \mathbf{M}^{(\tau)}} =$
     $\exp(-\varepsilon' \sum_{\tau=1}^{t} \mathbf{M}^{(\tau)})$.
Return the dual solution $(\overline{z} = \alpha + \delta, \overline{\mathbf{Y}} := \frac{1}{T} \sum_{t=1}^{T} \mathbf{Y}^{(t)})$.
```

Proposition III.1 (Arora and Kale's Primal-Dual Method [AK07, Theorem 1]) *Suppose in the Primal-Dual algorithm, the Oracle with width ρ has not returned a primal feasible solution $\widehat{\mathbf{X}}$ after $T := \frac{8\rho^2 \ln n}{\delta^2}$ iterations. Then, $(\overline{z} = \alpha + \delta, \overline{\mathbf{Y}} := \frac{1}{T} \sum_{t=1}^{T} \mathbf{Y}^{(t)})$ is a feasible dual solution with objective value $\alpha + \delta$.*

IV. Analyzing the Running Time

If we run the binary search step for candidate value α with additive error δ_α, it takes $T_\alpha = O(\frac{\rho_\alpha^2}{\delta_\alpha^2} \log n)$ iterations.[7] In each iteration of the binary step, the Oracle is accessed once and one matrix exponentiation is performed. As we shall see later, each Oracle access takes time $O(n^2)$ and hence the time for each iteration is dominated by the *time for one matrix*

[4] Arora and Kale described a framework that works for the type of SDPs in which the primal has inequality constraints, and the dual has non-negative variables. However, it is straight forward to apply the same argument and technique to an SDP in which the primal has equality constraints and the dual has unrestricted variables.

[5] In Arora and Kale's description, it is suggested that the Oracle is run with a "candidate" primal solution $\mathbf{X} := \frac{\mathbf{W}}{\mathbf{Tr}(\mathbf{W})}$, but this is not entirely necessary, and we incorporate this rescaling operation in the Oracle itself to simplify the description. Moreover, in their description of the Oracle, the primal solution returned is always $\mathbf{X}$ itself. However, in their applications, the Oracle can also return a slightly modified version of $\mathbf{X}$, in a manner which is more in accordance with our description.

[6] According to Arora and Kale's framework, it is sufficient to have $z \leq \alpha$. However, in our case, it is to our advantage for the Oracle to produce a dual solution with $z = \alpha$.

[7] We remind the reader here that the width ρ_α of the Oracle depends on the candidate value α currently being tested.

exponentiation, which we denote by M_e. [8]

Theorem IV.1 *Suppose the Lovász ϑ-function on a given graph G is α_*. Then, the value $\vartheta(G)$ can be approximated within an additive error of $\delta > 0$ in time $O(\frac{\rho^2_{\alpha_*}}{\delta^2}\log n \cdot M_e)$, where M_e is the time for matrix exponentiation and ρ_{α_*} is the width of the* ORACLE *applied with candidate value α_*.*

Proof: We describe how the binary search can be performed carefully to avoid an extra factor of $O(\log\frac{\alpha_*}{\delta})$ (the number of binary search steps) in the total running time. In the first phase, we double the value of α until we obtain an upper bound $\overline{\alpha}$ for α_*. In this phase, we use additive error $\delta_\alpha = 1$. Observe that $\overline{\alpha} \le 2\alpha_*$ and as we shall see in Section V, the width of the ORACLE is $\rho_\alpha = O(n\alpha)$. Hence, the running time in this phase is dominated by $T_{\overline{\alpha}} = O(\rho^2_{\overline{\alpha}}\log n) = O(\rho^2_{\alpha_*}\log n)$.

In the second phase, we gradually decrease the additive error δ_α in each step of the binary search. Observe that if the length of the *active interval* in binary search is currently L, then it is enough to set the additive error $\delta := \frac{L}{4}$ in order for the active interval to decrease geometrically after each binary step. Hence, it follows the total running time is dominated by the final binary search step, which takes time $O(\frac{\rho^2_{\alpha_*}}{\delta^2}\log n \cdot M_e)$. ■

V. ORACLE FOR THE LOVÁSZ ϑ-FUNCTION

Specifications for the ORACLE **with width ρ_α**
Input: A matrix $\mathbf{W} \succeq 0$ and a candidate value α.
Output: Returns either

1. a primal feasible solution $\widehat{\mathbf{X}}$ with objective value at least α, or
2. a dual solution $(z, \mathbf{Y} := \sum_{\{i,j\}\in E} y_{ij}\mathbf{E}_{ij})$ with objective value at most α such that $(z\,\mathbf{I}+\mathbf{Y}-\mathbf{J})\bullet\mathbf{W} \ge 0$ and $\|z\,\mathbf{I}+\mathbf{Y}-\mathbf{J}\|_2 \le \rho_\alpha$.

Theorem V.1 (ORACLE **for the Primal-Dual Method)** *There exists such an* ORACLE *with access time $O(n^2)$, and for candidate value α, it has width $\rho_\alpha = O(\alpha n)$.*

We first describe the ORACLE, and then show that it meets the above specifications.

Implementation for the ORACLE
Input: A matrix $\mathbf{W} \succeq 0$ and a candidate value α.
If $\mathbf{W} = 0$, returns $(z = \alpha, \mathbf{Y} := 0)$; otherwise, compute $\mathbf{X} := \frac{\mathbf{W}}{\mathbf{Tr}(\mathbf{W})}$ and $\sigma := \mathbf{J}\bullet\mathbf{X} - \alpha$.

1. If $\sigma \le 0$, returns dual solution $(z := \alpha, \mathbf{Y} := 0)$.
2. Let $\mathbf{X}_E$ be the matrix such that its (i,j)-th entry is the same as that of $\mathbf{X}$ if $\{i,j\}\in E$, and otherwise 0.
3. If $\|\mathbf{X}_E\|_F \ge \frac{\sigma}{\alpha n}$, returns dual solution $(z := \alpha, \mathbf{Y} := \frac{\sigma}{\|\mathbf{X}_E\|_F^2}\cdot\mathbf{X}_E)$.
4. Otherwise, compute $\tilde{\mathbf{X}} := \mathbf{X} - \mathbf{X}_E + \frac{\sigma}{\alpha n}\mathbf{I}$, and returns primal solution $\widehat{\mathbf{X}} := \frac{\tilde{\mathbf{X}}}{\mathbf{Tr}(\tilde{\mathbf{X}})}$.

From the description of the ORACLE, it always terminates with either a dual solution $(z, \mathbf{Y})$ or a primal solution $\widehat{\mathbf{X}}$. Moreover, the ORACLE runs in time $O(n^2)$, since the Frobenius norm can be computed in quadratic time. Hence, it suffices to show that the solutions returned by the ORACLE satisfy the specifications.

Lemma V.2 (Dual Solution) *Suppose the* ORACLE *returns a dual solution $(z, \mathbf{Y})$.*

Then, $(z\,\mathbf{I}+\mathbf{Y}-\mathbf{J})\bullet\mathbf{W} \ge 0$ and $\|z\,\mathbf{I}+\mathbf{Y}-\mathbf{J}\|_2 = O(n\alpha)$.

Proof: We first consider the case when the dual solution returned is of the form $(z := \alpha, \mathbf{Y} := 0)$. In this case, since $\|\mathbf{J}\|_2 = n$, the norm clearly satisfies $\|z\,\mathbf{I}+\mathbf{Y}-\mathbf{J}\|_2 \le \|\alpha\mathbf{I}\|_2 + \|\mathbf{J}\|_2 \le \alpha + n \le O(\alpha n)$.

For the trivial case where $\mathbf{W} = 0$, we have $(z\,\mathbf{I}+\mathbf{Y}-\mathbf{J})\bullet\mathbf{W} = 0$. Otherwise, we have $\mathbf{X} := \frac{\mathbf{W}}{\mathbf{Tr}(\mathbf{W})}$ and $\sigma := \mathbf{J}\bullet\mathbf{X} - \alpha$. Since $\mathbf{Y} := 0$ is returned, it must be the case that $\sigma \le 0$. Hence, observing that $\mathbf{I}\bullet\mathbf{X} = \mathbf{Tr}(\mathbf{X}) = 1$, we have $(z\mathbf{I}+\mathbf{Y}-\mathbf{J})\bullet\mathbf{W} = \mathbf{Tr}(\mathbf{W})\cdot(\alpha\mathbf{I}\bullet\mathbf{X}+0\bullet\mathbf{X}-\mathbf{J}\bullet X) = -\sigma\,\mathbf{Tr}(\mathbf{W}) \ge 0$.

We next consider the case in which a non-trivial dual solution is returned. The diagonal and non-edge entries of the matrix $\mathbf{X}$ are dropped to form the matrix $\mathbf{X}_E$. The dual solution $\mathbf{Y} := \frac{\sigma}{\|\mathbf{X}_E\|_F^2}\cdot\mathbf{X}_E$ returned is a scalar multiple of $\mathbf{X}_E$. Hence, in the matrix $\mathbf{Y}$, only the (i,j)-th entry for which $\{i,j\}\in E$ can be non-zero, and so $\mathbf{Y}$ has the correct form.

Observing that $\mathbf{Y}\bullet\mathbf{X} = \mathbf{Y}\bullet\mathbf{X}_E$, we have $(z\mathbf{I}+\mathbf{Y}-\mathbf{J})\bullet\mathbf{W} = \mathbf{Tr}(\mathbf{W})\cdot(\mathbf{Y}\bullet\mathbf{X}_E - \sigma) = \mathbf{Tr}(\mathbf{W})\cdot\left(\frac{\sigma}{\|\mathbf{X}_E\|_F^2}\cdot(\mathbf{X}_E\bullet\mathbf{X}_E) - \sigma\right) = 0$.

Finally, we need to show that the width of the ORACLE is small. In this case, $\|\mathbf{X}_E\|_F \ge \frac{\sigma}{\alpha n}$, and hence it follows that $\|\mathbf{Y}\|_F = \frac{\sigma}{\|\mathbf{X}_E\|_F} \le \alpha n$. Observing that $\|\mathbf{Y}\|_2 \le \|\mathbf{Y}\|_F$, it follows that $\|z\mathbf{I}+\mathbf{Y}-\mathbf{J}\|_2 \le \alpha + \alpha n + n = O(\alpha n)$. ■

Lemma V.3 (Primal Solution) *Suppose the* ORACLE *returns a primal solution $\widehat{\mathbf{X}}$. Then, $\widehat{\mathbf{X}}$ is primal feasible and has value $\mathbf{J}\bullet\widehat{\mathbf{X}}$ at least α.*

Proof: In this case, we have $\sigma := \mathbf{J}\bullet\mathbf{X} - \alpha > 0$ and $\|X_E\|_F < \frac{\sigma}{\alpha n}$. The solution returned is of the form $\widehat{\mathbf{X}} := \frac{\tilde{\mathbf{X}}}{\mathbf{Tr}(\tilde{\mathbf{X}})}$, where $\tilde{\mathbf{X}} := \mathbf{X} - \mathbf{X}_E + \frac{\sigma}{\alpha n}\mathbf{I}$.

Notice that in the matrix $\tilde{\mathbf{X}}$, all the (i,j)-th entries, for which $\{i,j\}\in E$, are dropped. Hence, for all $\{i,j\}\in E$,

[8] In Arora and Kale's paper [AK07], matrix exponentiation is approximated by using ideas from the Johnson-Lindenstrauss Lemma. However, their matrix exponentiation approximation takes time proportional to the width ρ of the ORACLE times the time for matrix multiplication. Since the width in our case is $\rho = \Omega(n)$ and the matrices involved are dense, it would be better to compute matrix exponentiation directly, which takes time $M_e = O(n^3)$ [GvL96].

$\widehat{\mathbf{X}}_{ij} = 0$. Moreover, since $\|X_E\|_2 \le \|X_E\|_F \le \frac{\sigma}{\alpha n}$, the additive term $\frac{\sigma}{\alpha n}\mathbf{I}$ makes sure that $\tilde{\mathbf{X}} \succeq 0$. Finally, the rescaling ensures that $\mathbf{I} \bullet \widehat{\mathbf{X}} = \mathbf{Tr}(\widehat{\mathbf{X}}) = 1$. Hence, $\widehat{\mathbf{X}}$ is primal feasible.

Next, we check the value of the primal solution. By the Cauchy-Schwarz Inequality, $\mathbf{X}_E \bullet \mathbf{J} \le \|X_E\|_F \cdot \|\mathbf{J}\|_F \le \frac{\sigma}{\alpha}$, observing that $\|\mathbf{J}\|_F = n$. Hence, it follows that $\mathbf{J} \bullet \tilde{\mathbf{X}} \ge \mathbf{J} \bullet \mathbf{X} - \frac{\sigma}{n} + \frac{\sigma}{\alpha n}\mathbf{J} \bullet \mathbf{I} = \sigma + \alpha$. Finally, observing that $\mathbf{Tr}(\tilde{\mathbf{X}}) = 1 + \frac{\sigma}{\alpha}$, it follows that $\mathbf{J} \bullet \widehat{\mathbf{X}} \ge \alpha$, as required. ■

Theorem V.1 now follows from Lemmas V.2 and V.3. Together with Theorem IV.1, we have our main result in Theorem I.1.

VI. Extension to the Weighted Lovász ϑ-Function

Definition VI.1 (Weighted Lovász ϑ-Function) *Given a graph $G = (V, E)$ and a non-negative weight function $w : V \to \mathbb{R}_+$, the* weighted Lovász ϑ-function $\vartheta(G)$ *is the optimal value of the following semidefinite program, together with its dual.*

$$\begin{array}{llcl} \text{(P)} & \max \mathbf{J}_w \bullet \mathbf{X} & & \\ & \mathbf{I} \bullet \mathbf{X} & = & 1 \\ & \forall \{i,j\} \in E : \mathbf{E}_{ij} \bullet \mathbf{X} & = & 0 \\ & \mathbf{X} & \succeq & 0 \end{array}$$

$$\begin{array}{llcl} \text{(D)} & \min z & & \\ & z\,\mathbf{I} + \sum_{\{i,j\} \in E} y_{ij}\mathbf{E}_{ij} & \succeq & \mathbf{J}_w \\ & z,\, y_{ij} & \in & \mathbb{R} \end{array}$$

Here, the notation is the same as that in Definition II.1, except that $\mathbf{J}_w$ is now the matrix whose (i,j)-th entry is $\sqrt{w_i \cdot w_j}$.

Note that the definition is exactly the same as before, except that the matrix $\mathbf{J}$ is now replaced by $\mathbf{J}_w$. An advantage of this formulation of ϑ_w is that the algorithm in Section III and the arguments in Section V remain valid. In fact, we have not used any property specific to $\mathbf{J}$ except in the proofs of Lemmas V.2 and V.3, where we need a bound on $\|\mathbf{J}\|_2$. Let $W := \sum_{i \in V} w_i$. Observe that in this case, $W = \|\mathbf{J}_w\|_2 = \|\mathbf{J}_w\|_F = \mathbf{Tr}(\mathbf{J}_w)$.

Modifications to the proof of Lemma V.3 (Primal Solution). The same argument still goes through. The only place where the matrix $\mathbf{J}_w$ plays a part is the evaluation of the objective value. Recall that we have $\sigma := \mathbf{J}_w \bullet \mathbf{X} - \alpha > 0$ and $\|\mathbf{X}_E\|_F < \frac{\sigma}{\alpha n}$. The solution returned is of the form $\widehat{\mathbf{X}} := \frac{\tilde{\mathbf{X}}}{\mathbf{Tr}(\tilde{\mathbf{X}})}$, where $\tilde{\mathbf{X}} := \mathbf{X} - \mathbf{X}_E + \frac{\sigma}{\alpha n}\mathbf{I}$. By the Cauchy-Schwarz Inequality, $\mathbf{X}_E \bullet \mathbf{J}_w \le \|\mathbf{X}_E\|_F \cdot \|\mathbf{J}_w\|_F \le \frac{\sigma W}{\alpha n}$. Hence, it follows that $\mathbf{J}_w \bullet \tilde{\mathbf{X}} \ge \mathbf{J}_w \bullet \mathbf{X} - \frac{\sigma W}{\alpha n} + \frac{\sigma}{\alpha n}\mathbf{Tr}(\mathbf{J}_w) = \sigma + \alpha$. Finally, observing that $\mathbf{Tr}(\tilde{\mathbf{X}}) = 1 + \frac{\sigma}{\alpha}$, it follows that $\mathbf{J}_w \bullet \widehat{\mathbf{X}} \ge \alpha$, just like before.

Modifications to the proof of Lemma V.2 (Dual Solution). Here the matrix $\mathbf{J}_w$ plays a role when we bound the norm $\|z\,\mathbf{I} + \mathbf{Y} - \mathbf{J}_w\|_2 \le \|\alpha\mathbf{I}\|_2 + \|\mathbf{Y}\|_2 + \|\mathbf{J}_w\|_2 \le \alpha + \alpha n + W$. At first this appears to be a problem, because the width of the oracle is now $\rho_\alpha = O(\alpha n + W)$, which is worse than before if $W \ge \alpha n$. However, if this is indeed the case, then there exists a vertex $v_i \in V$ such that its weight satisfies $w_i \ge \frac{W}{n} \ge \alpha$. Since $\vartheta_w(G)$ is at least the maximum independent set weight, it follows there exists a primal feasible solution whose value is at least that of the weight of the singleton $\{v_i\}$, which is trivially an independent set. Therefore, it is unnecessary to run the primal-dual algorithm in this case. Hence, our result generalizes to the weighted Lovász ϑ-function in Corollary I.2.

References

[AHK] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: A meta-algorithm and applications. http://www.cs.princeton.edu/~ehazan/papers/MWsurvey.pdf.

[AHK04] Sanjeev Arora, Elad Hazan, and Satyen Kale. $0(\sqrt{\log n})$ approximation to sparsest cut in $\tilde{0}(n^2)$ time. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 238–247, 2004.

[AK07] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 227–236, 2007.

[Ali95] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.

[CCL+05] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vusković. Recognizing berge graphs. *Combinatorica*, 25:143–186, 2005.

[CRST06] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164:51–229, 2006.

[GK98] Naveen Garg and Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 300–309, 1998.

[GLS87] L. Grotchel, L. Lovasz, and A. Schrijver. *Geometric algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1987.

[GvL96] Gene. H. Golub and Charles F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[IPS] G. Iyengar, D. Phillips, and C. Stein. Approximating semidefinite packing programs. *Computational Optimization Research Center Technical Report, Columbia University. TR-2008-01.*

[KL96] Philip N. Klein and Hsueh-I Lu. Efficient approximation algorithms for semidefinite programs arising from max cut and coloring. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 338–347, 1996.

[KMS98] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998.

[Knu94] Donald E. Knuth. The sandwich theorem. In *Electronic Journal of Combinatorics 1*, 1994.

[KY07] Christos Koufogiannakis and Neal E. Young. Beating simplex for fractional packing and covering linear programs. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 494–504, 2007.

[Lov79] L. Lovász. On the shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, 1979.

[NO96] M. V. Nayakkankuppam and M. L. Overton. Primal-dual interior-point methods for semidefinite programming:numerical experience with block-diagonal problems. *Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design*, pages 235–239, 1996.

[PST91] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*, pages 495–504, 1991.

[Sey06] P. Seymour. How the proof of the strong perfect graph conjecture was found. *Gazette des Mathematiciens*, 109:69–83, 2006.

[Sha56] C. E. Shannon. The zero-error capacity of a noisy channel. *IRE Transactions on Information Theory*, IT-2(3):8–19, 1956.

[Vaz95] Vijay V. Vazirani. Primal-dual schema based approximation algorithms (abstract). In *Computing and Combinatorics*, pages 650–652, 1995.