Cuts and Orderings: On semidefinite relaxations for the linear ordering problem

Alantha Newman*

MIT CSAIL, Cambridge, MA 02139
alantha@theory.csail.mit.edu

Abstract. The linear ordering problem is easy to state: Given a complete weighted directed graph, find an ordering of the vertices that maximizes the weight of the forward edges. Although the problem is NP-hard, it is easy to estimate the optimum to within a factor of 1/2. It is not known whether the maximum can be estimated to a better factor using a polynomial-time algorithm. Recently it was shown [NV01] that widely-studied polyhedral relaxations for this problem cannot be used to approximate the problem to within a factor better than 1/2. This was shown by demonstrating that the integrality gap of these relaxations is 2 on random graphs with uniform edge probability $p = 2^{\sqrt{\log n}}/n$. In this paper, we present a new semidefinite programming relaxation for the linear ordering problem. We then show that if we choose a random graph with uniform edge probability $p = \frac{d}{n}$, where $d = \omega(1)$, then with high probability the gap between our semidefinite relaxation and the integral optimal is at most 1.64.

1 Introduction

Vertex ordering problems comprise a fundamental class of combinatorial optimization problems that, on the whole, is not well understood. For the past thirty years, combinatorial methods and linear programming techniques have failed to yield improved approximation guarantees for many well-studied vertex ordering problems such as the linear ordering problem and the traveling salesman problem. Semidefinite programming has proved to be a powerful tool for solving a variety of cut problems, as first exhibited for the maximum cut problem [GW95]. Since then, semidefinite programming has been successfully applied to many other problems that can be categorized as cut problems such as coloring k-colorable graphs [KMS98], maximum-3-cut [GW04], maximum k-cut [FJ97], maximum bisection and maximum uncut [HZ01], and correlation clustering [CGW03], to name a few. In contrast, there is no such comparably general approach for approximating vertex ordering problems.

In this paper, we focus on a well-studied and notoriously difficult combinatorial optimization problem known as the linear ordering problem. Given a

* Supported in part by NSF Grant CCR0307536.

complete weighted directed graph, the goal of the linear ordering problem is to find an ordering of the vertices that maximizes the weight of the forward edges. Although the problem is NP-hard [Kar72], it is easy to estimate the optimum to within a factor of $\frac{1}{2}$: In any ordering of the vertices, either the set of forward edges or the set of backward edges accounts for at least half of the total edge weight. It is not known whether the maximum can be estimated to a better factor using a polynomial-time algorithm. Approximating the problem to within better than $\frac{65}{66}$ is NP-hard [NV01].

The linear ordering problem is also known as the maximum acyclic subgraph problem. Given a weighted directed graph, the maximum acyclic subgraph problem is that of finding the maximum weight subgraph that contains no cycles. The forward edges in any linear ordering comprise an acyclic subgraph and a topological sort of an acyclic subgraph yields a linear ordering of the vertices in which all edges in the acyclic subgraph are forward edges.

Recently it was shown that several widely-studied polyhedral relaxations for the linear ordering problem each have an integrality gap of 2, showing that it is unlikely these relaxations can be used to approximate the problem to within a factor greater than $\frac{1}{2}$ [NV01,New00]. The graphs used to demonstrate these integrality gaps are random graphs with uniform edge probability of approximately $2^{\sqrt{\log n}}/n$, where n is the number of vertices. For sufficiently large n, such a random graph has a maximum acyclic subgraph close to half the edges with high probability. However, each of the polyhedral relaxations studied provide an upper bound for these graphs that is asymptotically close to all the edges, which is off from the optimal by a factor of 2.

In this paper, we first present a new semidefinite programming relaxation for the linear ordering problem. A vertex ordering for a graph $G = (V, E)$ with n vertices can be fully described by a series of $n - 1$ cuts. We use this simple observation to relate cuts and orderings. We derive a semidefinite program for the linear ordering problem that is related to the semidefinite program used in the Goemans-Williamson algorithm to approximate the maximum cut problem [GW95]. We note that by using different objective functions, our semidefinite programming relaxation can be used to obtain semidefinite relaxations for many other vertex ordering problems.

Second, we show that for sufficiently large n, if we choose a random directed graph on n vertices with uniform edge probability $p = \frac{d}{n}$ (i.e. every edge in the complete directed graph on n vertices is chosen with probability p), where $d = \omega(1)$, our semidefinite relaxation will have an integrality gap of no more than 1.64 with high probability. In particular, the graphs used in [NV01] to demonstrate integrality gaps of 2 for the widely-studied polyhedral relaxations fall into this category of random graphs. The main idea is that our semidefinite relaxation provides a "good" bound on the value of an optimal linear ordering for a graph if it has no small roughly balanced bisection. With high probability, a random graph with uniform edge probability contains no such small balanced bisection.

2 Relating Cuts and Orderings

Given an undirected weighted graph $G = (V, E)$, the maximum cut (maxcut) problem is to find a bipartition of the vertices that maximizes the weight of the edges crossing the partition. In 1993, Goemans and Williamson used a semidefinite programming relaxation to obtain a .87856-approximation algorithm for this fundamental graph optimization problem [GW95]. The goal of the Goemans-Williamson algorithm for the maxcut problem is to assign each vertex $i \in V$ a vector $v_i \in \{1, -1\}$ so as to maximize the weight of the edges (i, j) such that $v_i \neq v_j$.

A closely related graph optimization problem is the maximum directed cut (dicut) problem. Given a directed weighted graph $G = (V, A)$, the dicut problem is to find a bipartition of the vertices—call these disjoint sets V_1 and V_2—that maximizes the weight of the directed edges (i, j) such that vertex i is in set V_1 and vertex j is in set V_2. Note that the edges in a directed cut form an acyclic subgraph. We can generalize the dicut problem to that of dividing the vertices into k labeled sets $V_1, V_2, \ldots, V_k$ so as to maximize the weight of the edges (i, j) such that vertex i is in set V_k and vertex j is in set V_h and $k < h$. We call this the k-acyclic dicut problem. The linear ordering problem is equivalent to the n-acyclic dicut problem.

2.1 A Relaxation for the Linear Ordering Problem

We can generalize the semidefinite programming relaxation for the dicut problem [FG95,GW95] to obtain a new semidefinite programming relaxation for the linear ordering problem. The basic idea behind this formulation is a particular description of a vertex ordering that uses $n+1$ unit vectors for each vertex. Each vertex $i \in V$ has $n + 1$ ($n = |V|$) associated unit vectors: $v_i^0, v_i^1, v_i^2, \ldots v_i^n$. In an integral solution, we enforce that $v_i^0 = -1$, $v_i^n = 1$ and that v_i^h and v_i^{h+1} differ for only one value of h, $0 \leq h < n$. Constraint (1) enforces that in an integral solution, v_i^h and v_i^{h+1} differ for only one such value of h. This position h denotes vertex i's position in the ordering. For example, suppose we have a graph G that has four vertices, arbitrarily labeled 1 through 4. Consider the vertex ordering in which vertex i is in position i. An integral description of this vertex ordering is:

$$\begin{aligned}
\{v_1^0,\ v_1^1,\ v_1^2,\ v_1^3,\ v_1^4\} &= \{-1,\ \ 1,\ \ 1,\ \ 1,\ \ 1\},\\
\{v_2^0,\ v_2^1,\ v_2^2,\ v_2^3,\ v_2^4\} &= \{-1, -1,\ \ 1,\ \ 1,\ \ 1\},\\
\{v_3^0,\ v_3^1,\ v_3^2,\ v_3^3,\ v_3^4\} &= \{-1, -1, -1,\ \ 1,\ \ 1\},\\
\{v_4^0,\ v_4^1,\ v_4^2,\ v_4^3,\ v_4^4\} &= \{-1, -1, -1, -1,\ \ 1\}.
\end{aligned}$$

Let $G = (V, A)$ be a directed graph. The following is an integer quadratic program for the linear ordering problem. For the sake of convenience, we assume that n is odd since this simplifies constraint (2). By $P(G)$, we denote the optimal value of the integer quadratic program P on the graph G.

(P)

$$\max \sum_{ij \in A} \sum_{1 \le h < \ell \le n} w_{ij}(v_i^h \cdot v_j^\ell + v_i^{h-1} \cdot v_j^{\ell-1} - v_i^h \cdot v_j^{\ell-1} - v_i^{h-1} \cdot v_j^\ell) \quad (1)$$

$$\begin{aligned} v_i^h \cdot v_j^\ell + v_i^{h-1} \cdot v_j^{\ell-1} - v_i^h \cdot v_j^{\ell-1} - v_i^{h-1} \cdot v_j^\ell &\ge 0 \quad \forall i,j \in V,\ h,\ell \in [n] \\ v_i^h \cdot v_i^h &= 1 \qquad \forall i \in V,\ h \in [n] \\ v_i^0 \cdot v_0 &= -1 \quad\ \forall i \in V \\ v_i^n \cdot v_0 &= 1 \qquad \forall i \in V \\ \sum_{i,j \in V} v_i^{\frac{n}{2}} \cdot v_j^{\frac{n}{2}} &= 0 \qquad\qquad\qquad (2) \\ v_i^h &\in \{1,-1\} \qquad \forall i,h \in [n]. \quad (3) \end{aligned}$$

We obtain a semidefinite programming relaxation for the linear ordering problem by relaxing constraint (3) to: $v_i^h \in \mathcal{R}^n,\ \forall i,h$. We denote the optimal value of the relaxation of P on the graph G as $P_R(G)$.

2.2 Cuts and Uncuts

Suppose we have a directed graph $G = (V, A)$ and we are given a set of unit vectors $\{v_i\} \in \mathcal{R}^n,\ 0 \le i \le n$. We will define the *forward* value of this set of vectors as the value obtained if we compute the value of the dicut semidefinite programming relaxation [GW95,FG95] using these vectors. Specifically, the forward value for this set of vectors is:

$$\max \sum_{ij \in A} \frac{1}{4}(1 - v_i \cdot v_j - v_0 \cdot v_i + v_0 \cdot v_j). \quad (4)$$

In an integral solution for the dicut problem, there will be edges that cross the cut in the backward direction, i.e. they are not included in the dicut. For a specified set of unit vectors, we can view the dicut semidefinite programming relaxation as having forward and *backward* value. We define the backward value of the set of vectors $\{v_i\}$ as:

$$\max \sum_{ij \in A} \frac{1}{4}(1 - v_i \cdot v_j - v_0 \cdot v_j + v_0 \cdot v_i). \quad (5)$$

The *difference* between the forward and backward value of a set of vectors $\{v_i\}$ is:

$$\sum_{ij \in A} \frac{1}{2}(v_j \cdot v_0 - v_i \cdot v_0). \quad (6)$$

Lemma 1. *If a directed graph $G = (V, A)$ has a maximum acyclic subgraph of $(\frac{1}{2} + \delta)|A|$ edges, then there is no set of vectors $\{v_i\}$ such that the difference between the forward and backward value of this set of vectors exceeds $2\delta|A|$.*

Proof. We will show that given a vector solution $\{v_i\}$ to the semidefinite program in which the objective function is (6) and all the v_i vectors are unit vectors, we can find an integral (i.e. an actual cut) solution in which the difference of the forward and backward edges crossing the cut is exactly equal to the objective value. If the difference of an actual cut exceeds $2\delta|A|$, e.g. suppose it is $(2\delta+\epsilon)|A|$, then we can find an ordering with $(\frac{1}{2}+\delta+\epsilon/2)|A|$ forward edges, which is a contradiction. This ordering is found by taking the cut that yields $(2\delta+\epsilon)|A|$ more forward than backward edges and ordering the vertices in each of the two sets greedily so as to obtain at least half of the remaining edges.

Suppose we have a set of unit vectors $\{v_i\}$ such that the value of equation (6) is at least $(2\delta+\epsilon)|A| = \beta|A|$. We will show that we can find an actual cut such that the difference between the forward and the backward edges is at least $\beta|A|$. Note that $v_0 \cdot v_i$ is a scalar quantity since v_0 is a unit vector that without loss of generality is $(1, 0, 0, \ldots)$. Thus, our objective function can be written as $\sum_{ij\in A} \frac{1}{2}(z_j - z_i)$ where $1 \geq z_i \geq -1$. We transform the z_i variables into x_i variables that range between 0 and 1 by letting $z_i = 2x_i - 1$. Then we have that $\sum_{ij\in A} \frac{1}{2}(z_j - z_i) = \sum_{ij\in A}(x_j - x_i)$. This results in a linear program. We claim that an optimal solution to the following linear program is integral.

$$\sum_{ij\in A}(x_j - x_i) \tag{7}$$
$$0 \leq x_i \leq 1, \quad \forall i \in V.$$

To show this, consider rounding the variables by letting i be 1 with probability x_i and 0 otherwise. Then the expected value of the solution is exactly the objective value. However, note that the value of the solution cannot be less than the expected value, since then there must exist a solution with value greater than the expected value, which contradicts the optimality of the expected value. Thus, the integral solution obtained must have difference of forward and backward edges that is equal to the objective value (7). $\square$

We will also find a discussion of the following problem useful. Consider the problem of finding a balanced partition of the vertices of a given graph (i.e. each partition has size $\frac{n}{2}$) that maximizes the weight of the edges that do *not* cross the cut. This problem is referred to as the max-$\frac{n}{2}$-uncut problem by Halperin and Zwick [HZ01]. Below is a integer quadratic program for the max-$\frac{n}{2}$-uncut problem.

(T)

$$\max \sum_{ij\in A} \frac{1 + v_i \cdot v_j}{2}$$
$$\sum_{i,j\in V} v_i \cdot v_j = 0$$
$$v_i \cdot v_i = 1 \qquad \forall i \in V$$
$$v_i \in \{1, -1\} \quad \forall i \in V. \tag{8}$$

We obtain a semidefinite programming relaxation for the max-$\frac{n}{2}$-uncut problem by relaxing constraint (8) to: $v_i \in \mathcal{R}^n, \ \forall i$. We denote the value of the relaxation of T on the graph G as $T_R(G)$.

Lemma 2. *Let $G = (V, A)$ and ϵ, δ be positive constants. Suppose the maximum acyclic subgraph of G is $(\frac{1}{2} + \delta)|A|$. If $P_R(G) \geq (1 - \epsilon)|A|$, then $T_R(G) \geq (1 - 2\epsilon - 2\delta)|A|$.*

Proof. For each edge $ij \in A$, we have:

$$\sum_{h<\ell} v_i^h \cdot v_j^\ell + v_i^{h-1} \cdot v_j^{\ell-1} - v_i^h \cdot v_j^{\ell-1} - v_i^{h-1} \cdot v_j^\ell = \tag{9}$$

$$\sum_{h<\ell} (v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}) \leq$$

$$\sum_{h \leq \frac{n}{2}, \ell \leq \frac{n}{2}} (v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}) + \tag{10}$$

$$\sum_{h > \frac{n}{2}, \ell > \frac{n}{2}} (v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}) + \tag{11}$$

$$\sum_{h \leq \frac{n}{2}, \ell > \frac{n}{2}} (v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}). \tag{12}$$

For each edge, we refer to the quantity (9) as the forward value for that edge with respect to $P_R(G)$. The same term summed instead over $h > \ell$ is referred to as the *backward* value of the edge with respect to $P_R(G)$. We can simplify the terms above. Let $v_i = v_i^{\frac{n}{2}}$.

$$\sum_{h \leq \frac{n}{2}, \ell \leq \frac{n}{2}} (v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}) = \frac{1}{4}(v_i + v_0) \cdot (v_j + v_0),$$

$$\sum_{h > \frac{n}{2}, \ell > \frac{n}{2}} (v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}) = \frac{1}{4}(v_0 - v_i) \cdot (v_0 - v_j),$$

$$\sum_{h \leq \frac{n}{2}, \ell > \frac{n}{2}} (v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}) = \frac{1}{4}(v_i + v_0) \cdot (v_0 - v_j).$$

Since $P_R(G) \geq (1 - \epsilon)|A|$, we have:

$$\sum_{ij \in A} \sum_{h > \frac{n}{2}, \ell \leq \frac{n}{2}} (v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}) = \sum_{ij \in A} \frac{1}{4}(v_0 - v_i) \cdot (v_0 + v_j) \ \leq \ \epsilon |A|.$$

The above inequality says that the *backward* value of the vectors $\{v_i\}$ (i.e. quantity (5)) is at most the *backward* value of $P_R(G)$. By Lemma 1, the difference of the edges crossing the cut in the forward direction and the edges crossing the cut in the backward direction is at most $2\delta|A|$.

$$\sum_{ij\in A} \frac{1}{4}(v_i+v_0)\cdot(v_0-v_j) \ - \sum_{ij\in A}\frac{1}{4}(v_0-v_i)\cdot(v_0+v_j) = \sum_{ij\in A}\frac{1}{2}(v_i\cdot v_0 - v_j\cdot v_0) \le 2\delta|A|.$$

This implies that the forward value cannot exceed the backward value by more than $2\delta|A|$. Thus, we can bound the forward value as follows:

$$\sum_{ij\in A}\sum_{h\le\frac{n}{2},\ell>\frac{n}{2}} (v_i^h - v_i^{h-1})\cdot(v_j^\ell - v_j^{\ell-1}) = \sum_{ij\in A}\frac{1}{4}(v_0-v_i)\cdot(v_0+v_j) \le (\epsilon+2\delta)|A|.$$

This implies that if we sum the quantities (10) and (11) over all edges in A, then the total value of this sum is at least $(1-2\epsilon-2\delta)|A|$. The sum of (10) and (11) taken over all the edges is:

$$\sum_{ij\in A}\frac{1+v_i\cdot v_j}{2}. \tag{13}$$

□

3 Balanced Bisections of Random Graphs

A *bisection* of a graph is a partition of the vertices into two equal (or with cardinality differing by one if n is odd) sets. We use a related definition in this section.

Definition 1. *A γ-bisection of a graph for $\gamma \le \frac{1}{2}$ is the set of edges that cross a cut in which each set of vertices has size at least γn.*

Suppose we choose an undirected random graph on n vertices in which every edge is present with probability $p=\frac{2d}{n}$. The expected degree of each vertex is $2d$ and the expected number of edges is dn. We will call such a class of graphs G_p.

Lemma 3. *For any fixed positive constants ϵ,γ, if we choose a graph $G\in G_p$ on n vertices for a sufficiently large n with $p=\frac{2d}{n}$ and $d=\omega(1)$, then the minimum γ-bisection contains at least $(1-\epsilon)\gamma(1-\gamma)2nd$ edges with high probability.*

Proof. We will use the principle of deferred decisions. First, we will choose a $\gamma n,(1-\gamma)n$ partition of the vertices. Thus $\gamma(1-\gamma)n^2$ edges from the complete graph on n vertices cross this cut. Then we can choose the random graph G by picking each edge with probability $p=\frac{2d}{n}$. The expected number of edges from G crossing the cut is $\mu=(\gamma(1-\gamma)n^2)(\frac{2d}{n})=\gamma(1-\gamma)2dn$. For each edge in the complete graph that crosses the cut, we have the indicator random variable X_i

such that $X_i = 1$ if the edge crosses the cut and $X_i = 0$ if the edge does not cross the cut. Let $X = \sum X_i$, i.e. X is the random variable for the number of edges that cross the cut. By Chernoff Bound, we have:

$$\Pr[X < (1-\epsilon)\gamma(1-\gamma)2dn)] < e^{-\frac{\epsilon^2\gamma(1-\gamma)2dn}{2}}.$$

We can union bound over all the possible γ-bisections. There are less than 2^n ways to divide the vertices so that at least γn are in each set. Thus, the probability that the minimum γ-bisection of G is less than a $(1-\epsilon)$ fraction of its expectation is:

$$\Pr[\min\ \gamma\text{-bisection}(G) < (1-\epsilon)\gamma(1-\gamma)2nd] < \frac{2^n}{e^{\frac{\epsilon^2\gamma(1-\gamma)2dn}{2}}}.$$

Let $d = \omega(1)$. Then for any fixed positive constants γ, ϵ, this probability will be arbitrarily small for sufficiently large n. □

4 A Contradictory Cut

In this section, we will prove our main theorem. Suppose we choose a directed random graph on n vertices in which every edge in the complete directed on n vertices is included with probability p. Let $p = \frac{d}{n}$ and let $d = \omega(1)$. We will call this class of graphs $\vec{G}_p$. Note that if we randomly choose a graph from $\vec{G}_p$, the underlying undirected graph is randomly chosen from G_p.

Theorem 1. *For sufficiently large n, $d = \omega(1)$, and $p = \frac{d}{n}$, if we randomly choose a graph $\vec{G} \in \vec{G}_p$, then with high probability, the ratio $P_R(\vec{G})/P(\vec{G}) \leq 1.64$.*

Let E represent the edges in the complete undirected graph K_n for some fixed n. Let $A \subseteq E$ represent the edges in an undirected graph G chosen at random from G_p. Let ϵ_1 be a small positive constant whose value can be arbitrarily small for sufficiently large n. We weight the edges in E as follows:

$$\begin{aligned} w_{ij} &= -\frac{n}{(1-\epsilon_1)2d} && \text{if} \quad ij \in A, \\ w_{ij} &= 1 && \text{if} \quad ij \in E - A. \end{aligned}$$

We will refer to this weighted graph as G'.

Lemma 4. *The minimum γ-bisection of G' has negative value with high probability.*

Proof. By Lemma 3 with high probability the minimum γ-bisection of G has at least $(1-\epsilon_1)\gamma(1-\gamma)2nd$ edges. Thus, with high probability the total weight of the edges in the minimum γ-bisection of G' is at most:

$$\begin{aligned}\gamma(1-\gamma)n^2 - (1-\epsilon_1)\gamma(1-\gamma)2nd + (1-\epsilon_1)\gamma(1-\gamma)2nd(-\frac{n}{(1-\epsilon_1)2d}) &= \\ \gamma(1-\gamma)\left(n^2 - (1-\epsilon_1)2nd + (1-\epsilon_1)2nd(-\frac{n}{(1-\epsilon_1)2d})\right) &= \\ \gamma(1-\gamma)\left(-(1-\epsilon_1)2nd\right) &< 0.\end{aligned}$$

□

Lemma 5. *Let $\{v_i\}$, $i \in V$ be a set of unit vectors that satisfy the following constraints:*

$$\sum_{i,j \in V} v_i \cdot v_j = 0 \tag{14}$$

$$\sum_{ij \in A} \frac{1 + v_i \cdot v_j}{2} \geq (1-\epsilon_2)|A|. \tag{15}$$

If $\epsilon_2 < .36$, then we can find a γ-bisection of G' with a strictly positive value.

To prove Lemma 5, we will use the following theorem from [GW95].

Theorem 2.7 [GW] Let $W_- = \sum_{i<j} w_{ij}^-$, where $x^- = \min(0, x)$. Then

$$\{E[W] - W_-\} \geq \alpha \left\{ \frac{1}{2} \sum_{i<j} w_{ij}(1 - v_i \cdot v_j) - W_- \right\}.$$

Proof of Lemma 5: We will use Goemans-Williamson's random hyperplane algorithm to show that we can find a cut that is roughly balanced and has a strictly positive value. Let W represent the total weight of the edges that cross the cut obtained from a random hyperplane. Let W_- denote the sum of the negative edges weights, i.e. $W_- = A$. Applying Theorem 2.7 from [GW], we have:

$$\begin{aligned}E[W] &\geq \alpha \left\{ \frac{1}{2} \sum_{i<j} w_{ij}(1 - v_i \cdot v_j) - W_- \right\} + W_- \\ &\geq \alpha \left\{ \sum_{i<j: w_{ij}>0} w_{ij} \frac{1 - v_i \cdot v_j}{2} + \sum_{i<j: w_{ij}<0} |w_{ij}| \frac{1 + v_i \cdot v_j}{2} \right\} + W_-.\end{aligned}$$

We want to calculate the value of $\sum_{i<j: w_{ij}>0} \frac{1 - v_i \cdot v_j}{2}$. By condition (14), we have that $\sum_{i,j \in V} v_i \cdot v_j = 0$ and therefore $\sum_{i<j} \frac{1 - v_i \cdot v_j}{2} = \frac{n^2 - 2n}{4}$.

$$\begin{aligned}\sum_{i<j:w_{ij}>0} \frac{1 - v_i \cdot v_j}{2} &= \sum_{i<j} \frac{1 - v_i \cdot v_j}{2} - \sum_{i<j:w_{ij}<0} \frac{1 - v_i \cdot v_j}{2} \\ &= \sum_{i<j} \frac{1 - v_i \cdot v_j}{2} - \frac{nd}{2} + \sum_{i<j:w_{ij}<0} \frac{v_i \cdot v_j}{2} \\ &\geq \sum_{i<j} \frac{1 - v_i \cdot v_j}{2} - \frac{nd}{2} + \frac{(1-2\epsilon_2)nd}{2} \\ &= \frac{n^2 - 2n}{4} - \epsilon_2 nd.\end{aligned}$$

Now we have:

$$E[W] \geq \alpha \left\{ (\frac{n^2 - 2n}{4} - \epsilon_2 nd) + \frac{n}{(1-\epsilon_1)2d}(1-\epsilon_2)nd \right\} - \frac{n}{(1-\epsilon_1)2d} nd.$$

For large enough n, we can choose ϵ_1 to be arbitrarily small. So $E[W]$ can be bounded from below by a value arbitrarily close to the following:

$$(\frac{\alpha}{4} + \frac{\alpha}{2} - \frac{1}{2} - \frac{\alpha\epsilon_2}{2})n^2 - o(n^2) \ \geq \ (.1585 - \frac{\alpha\epsilon_2}{2})n^2 - o(n^2). \tag{16}$$

If the value of ϵ_2 is such that the quantity on line (16) is strictly greater than βn^2 for some positive constant β, then we will have a contradiction for sufficiently large n. Note that if this value is at least βn^2, then each side of the cut contains at least $\sqrt{\beta} n$ vertices, so it is a $\sqrt{\beta}$-bisection. This value will be strictly positive as long as $\epsilon_2 < .36$. Thus, it must be the case that $\epsilon_2 > .36$. □

Proof of Theorem 1: We fix positive constants γ, ϵ_1. Suppose we choose a random directed graph $\vec{G}$ as prescribed and let the graph $G = (V, A)$ be the undirected graph corresponding to the underlying undirected graph of $\vec{G}$. We then weight the edges in the graph K_n as discussed previously and obtain G'. By Lemma 4, the minimum γ-bisection of G' is negative with high probability. Thus, with high probability equation (15) hold only when $\epsilon_2 > .36$.

Suppose the maximum acyclic subgraph of $\vec{G}$, i.e. $P(\vec{G})$ is $(\frac{1}{2} + \delta)|A|$ for some positive constant δ. Then the value of $P_R(\vec{G})$ is upper bounded by the maximum value for some set of unit vectors $\{v_i\}$ of (10), (11), and (12) summed over all edges in A. Note that this is equivalent to the quantity in (13) (which is no more than $.64|A|$) plus the quantity in (4). By Lemma 1, the difference between (4) and (5) must be no more than $2\delta|A|$. Thus, we can upper bound the value of $P_R(\vec{G})$ by $.64|A| + (2\delta + \frac{1}{2}(.36 - 2\delta))|A| = (.82 + \delta)|A|$. Thus, with high probability, we have:

$$\frac{P_R(\vec{G})}{P(\vec{G})} \leq \frac{.82 + \delta}{.5 + \delta} \leq \frac{.82}{.5} = 1.64.$$

□

5 Discussion

In this paper, we make a connection between cuts and vertex ordering of graphs in order to obtain a new semidefinite programming relaxation for the linear ordering problem. We show that the relaxation is "good" on random graphs chosen with uniform edge probability, i.e. the integrality gap is strictly less than 2 for most of these graphs. We note that we can extend this theorem to show that this relaxation is "good" on graphs that have no small γ-bisections for some constant $\gamma > 0$.

In [HZ01], Halperin and Zwick give a .8118-approximation for a related problem that they call the max $\frac{n}{2}$-directed-uncut problem. Given a directed graph, the goal of this problem is to find a bisection of the vertices that maximizes the weight of the edges that cross the cut in the forward direction plus the weight of the edges that do not cross the cut. We note that a weaker version of Theorem 1 follows from their .8118-approximation algorithm. This is because their semidefinite program for the max $\frac{n}{2}$-directed uncut problem is the sum over all edges of terms (10), (11), and (12). If for some directed graph $G = (V, A)$, $P_R(G)$ has value at least $(1 - \epsilon)|A|$, then the value of their semidefinite programmming relaxation also has at least this value. Thus, if ϵ is arbitrarily small, we can obtain a directed uncut of value close to .8118 of the edges, which is a contradiction for a random graph with uniform edge probability. In this paper, our goal was to give a self-contained proof of this theorem.

We would like to comment on the similarity of this work to the work of Poljak and Delorme [DP93] and Poljak and Rendl [PR95] on the maxcut problem. Poljak showed that the class of random graphs with uniform edge probability could be used to demonstrate an integrality gap of 2 for several well-studied polyhedral relaxations for the maxcut problem [Pol92]. These same graphs can be used to demonstrate an integrality gap of 2 for several widely-studied polyhedral relaxations for the linear ordering problem [NV01]. The similarity of these results stems from the fact that the polyhedral relaxations for the maxcut problem are based on odd-cycle inequalities and the polyhedral relaxations for the linear ordering problem are based on cycle inequalities. Poljak and Delorme subsequently studied an eigenvalue bound for the maxcut problem that is equivalent to the bound provided by the semidefinite programming relaxation used in the Goemans-Williamson algorithm [GW95]. Despite the fact that random graphs with uniform edge probability exhibit worst-case behaviour for several polyhedral relaxations for the maxcut problem, Delorme and Poljak [DP93] and Poljak and Rendl [PR95] experimentally showed that the eigenvalue bound provides a "good" bound on the value of the maxcut for these graphs. This experimental evidence was the basis for their conjecture that the 5-cycle exhibited a worst-case integrality gap of $0.88445\ldots$ for the maxcut semidefinite relaxation [DP93,Pol92]. The gap demonstrated for the 5-cycle turned out to be very close to the true integrality gap of $.87856\ldots$ [FS].

Acknowledgments

I would like to thank Prahladh Harsha, Santosh Vempala, and Michel Goemans for many helpful discussions.

References

[CGW03] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 524–533, Boston, 2003.

[DP93] Charles Delorme and Svatopluk Poljak. The performance of an eigenvalue bound in some classes of graphs. *Discrete Mathematics*, 111:145–156, 1993. Also appeared in *Proceedings of the Conference on Combinatorics*, Marseille, 1990.

[FG95] Uriel Feige and Michel X. Goemans. Approximating the value of two prover proof systems with applications to MAX-2-SAT and MAX DICUT. In *Proceedings of the Third Israel Symposium on Theory of Computing and Systems*, pages 182–189, 1995.

[FJ97] Alan Frieze and Mark R. Jerrum. Improved approximation algorithms for MAX-k-Cut and MAX BISECTION. *Algorithmica*, 18:61–77, 1997.

[FS] Uriel Feige and Gideon Schechtman. On the optimality of the random hyperplane rounding technique for MAX-CUT. *Random Structures and Algorithms*. To appear.

[GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[GW04] Michel X. Goemans and David P. Williamson. Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming. *STOC 2001 Special Issue of Journal of Computer and System Sciences*, 68:442–470, 2004.

[HZ01] Eran Halperin and Uri Zwick. A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems. In *Proceedings of Eighth Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 210–225, Utrecht, 2001.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–104. Plenum Press, New York, 1972.

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

[New00] Alantha Newman. Approximating the maximum acyclic subgraph. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, June 2000.

[NV01] Alantha Newman and Santosh Vempala. Fences are futile: On relaxations for the linear ordering problem. In *Proceedings of Eighth Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 333–347, 2001.

[Pol92] Svatopluk Poljak. Polyhedral and eigenvalue approximations of the max-cut problem. *Sets, Graphs and Numbers, Colloqiua Mathematica Societatis Janos Bolyai*, 60:569–581, 1992.

[PR95] Svatopluk Poljak and Franz Rendl. Computing the max-cut by eigenvalues. *Discrete Applied Mathematics*, 62(1–3):249–278, September 1995.