Linear Programming Hierarchies Suffice for Directed Steiner Tree

Zachary Friggstad[1], Jochen Könemann[⋆,2], Young Kun-Ko[3], Anand Louis[⋆⋆,4], Mohammad Shadravan[⋆,2], and Madhur Tulsiani[⋆⋆⋆,5]

[1] Department of Computing Science, University of Alberta
[2] Department of Combinatorics and Optimization, University of Waterloo
[3] Department of Computer Science, Princeton University
[4] College of Computing, Georgia Tech
[5] Toyota Technical Institute at Chicago.

Abstract. We demonstrate that ℓ rounds of the Sherali-Adams hierarchy and 2ℓ rounds of the Lovász-Schrijver hierarchy suffice to reduce the integrality gap of a natural LP relaxation for Directed Steiner Tree in ℓ-layered graphs from $\Omega(\sqrt{k})$ to $O(\ell \cdot \log k)$ where k is the number of terminals. This is an improvement over Rothvoss' result that 2ℓ rounds of the considerably stronger Lasserre SDP hierarchy reduce the integrality gap of a similar formulation to $O(\ell \cdot \log k)$.

We also observe that Directed Steiner Tree instances with 3 layers of edges have only an $O(\log k)$ integrality gap in the standard LP relaxation, complementing the known fact that the gap can be as large as $\Omega(\sqrt{k})$ in graphs with 4 layers.

1 Introduction

In the Directed Steiner Tree (DST) problem, we are given a directed graph $G = (V, E)$ with edge costs $c_e \geq 0, e \in E$. Furthermore, we are given a root node $r \in V$ and a collection of terminals $X \subseteq V$ and the goal is to find the cheapest collection of edges $F \subseteq E$ such that there is an $r - t$ path using only edges in F for every terminal $t \in X$. The nodes in $V - (X \cup \{r\})$ are called *Steiner nodes*. Throughout (except in Section 2.1), we will let $n = |V|$, $m = |E|$, and $k = |X|$ and we let OPT_G denote the optimum solution cost to the DST instance in graph G.

If $X \cup \{r\} = V$, then the problem is simply the minimum-cost arborescence problem which can be solved efficiently. However, the general case is well-known to be NP-hard. In fact, the problem can be seen to generalize the Group Steiner Tree problem, which cannot be approximated within $O(\log^{2-\epsilon}(n))$ for any constant $\epsilon > 0$ unless NP $\subseteq$ DTIME$(n^{\mathrm{polylog}(n)})$ [7].

⋆ Supported by NSERC grant no. 288340.
⋆⋆ Supported by NSF award CCF-1217793. Part of work done while the author was a summer intern at TTI-Chicago.
⋆⋆⋆ Research supported by NSF Career Award CCF-1254044.

Definition 1. *Say that an instance $G = (V, E)$ of DST with terminals X is* ℓ-layered *if V can be partitioned as $V_0, V_1, \ldots, V_\ell$ where $V_0 = \{r\}, V_\ell = X$ and every edge $uv \in E$ has $u \in V_i$ and $v \in V_{i+1}$ for some $0 \le i < \ell$.*

For any DST instance G and any integer $\ell \ge 1$, Zelikovsky showed that there is an ℓ-layered DST instance H with at most $\ell \cdot n$ nodes such that $OPT_G \le OPT_H \le \ell \cdot k^{1/\ell} \cdot OPT_G$ and that a DST solution in H naturally corresponds to a DST solution in G with the same cost [11, 1]. Charikar et al. [2] exploit this fact and present an $O(\ell^2 k^{1/\ell} \log k)$-approximation[6] with running time poly(n, k^ℓ) for any integer $\ell \ge 1$. In particular, this can be used to obtain an $O(\log^3 k)$-approximation in quasi-polynomial time and for any constant $\epsilon > 0$ a polynomial-time $O(k^\epsilon)$-approximation. Finding a polynomial-time polylogarithmic approximation is an important open problem.

A natural linear programming (LP) relaxation for Directed Steiner Tree is given by LP (P0).

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e & & \text{(P0)} \\ \text{s.t.} \quad & x(\delta^{\text{in}}(S)) \ge 1 & \forall\, S \subseteq V - r, S \cap X \ne \emptyset \quad & (1) \\ & x_e \in [0,1] & \forall e \in E & \end{aligned}$$

Zosin and Khuller [12] demonstrated that the integrality gap of this relaxation can, unfortunately, be as bad as $\Omega(\sqrt{k})$, even in instances where G is a 4-layered graph. Recently, Rothvoss [9] showed that 2ℓ rounds of the Lasserre semidefinite programming (SDP) hierarchy suffice to reduce the integrality gap of a similar LP relaxation to only $O(\ell \cdot \log k)$ in ℓ-layered graphs. The LP he considers is an extended formulation of (P0) with polynomially many constraints plus additional constraints of the form $x(\delta^{in}(v)) \le 1$ for each non-root node v.

A related problem that will appear frequently throughout this paper is the Group Steiner Tree (GST) problem mentioned above. In this, we are given an undirected graph with edge costs, a root node r, and a collection of subsets $X_1, X_2, \ldots, X_k$ of nodes called *terminal groups*. The goal is to find the cheapest subset of edges F such that for every group X_i, there is a path from r to some node in X_i using only edges in F. Unlike DST, the integrality gap of the natural LP relaxation (GST-LP) (introduced in Section 3) is polylogarithmically bounded.

Theorem 1 (Garg, Konjevod, and Ravi [4]). *The integrality gap of LP (*GST-LP*) is $O(\min\{\ell, \log n\} \cdot \log k)$ in GST instances that have n nodes, k terminal groups, and are trees with height ℓ when rooted at r.*

[6] The algorithm in [2] is presented as an $O(\ell k^{1/\ell} \log k)$-approximation and relied on an incorrect claim in [11]. A correction to this claim was made in [1] which gives the stated DST approximation bound.

Only the bound of $O(\log n \log k)$ is explicitly shown in [4] but the bound $O(\ell \cdot \log k)$ easily follows from their techniques[7].

Hierarchies of convex programming relaxations, a.k.a. "lift-and-project" methods, have recently been used successfully in the design of approximation algorithms. In this paper, we only include the specifics of the Sherali-Adams hierarchy and Lovász-Schrijver LP hierarchy as needed to describe our result. For more information, we direct the reader to an introduction and survey by Chlamtáč and Tulsiani [3] and note that a more recent application of the Sherali-Adams hieararchy by Gupta, Talwar and Witmer obtains a 2-approximation for the non-uniform Sparsest Cut problem in graphs with bounded treewidth [5] .

1.1 Our Results and Techniques

Using the ellipsoid method, it is possible to design a separation oracle for the ℓ-th level lift of (P0) in the Sheral-Adams and Lovász-Schrijver hierarchies with running time being polynomial in m^ℓ and k^ℓ. However, we will start with a much simpler LP relaxation with only polynomially many constraints.

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e && && \text{(P1)} \\
\text{s.t.} \quad & x(\delta^{\text{in}}(t)) \geq 1 && t \in X && (2) \\
& x(\delta^{in}(v)) \leq 1 && \forall\, v \in V - r && (3) \\
& x(\delta^{in}(v)) \geq x_e && \forall\, v \in V - (X \cup \{r\}), e \in \delta^{out}(v) && (4) \\
& x_e \in [0,1] && \forall e \in E &&
\end{aligned}
$$

This is a relaxation in the sense that integer solutions corresponding to *minimal* DST solutions are feasible. That is, any minimal DST solution F is a branching, so every node has indegree 1 in F which justifies the inclusion of Constraints (3). Similarly, if some Steiner node v has no incoming edges in F then, by minimality of F, v also has outdegree 0 which justifies Constraints (4).

Our main result is the following. The notation $\mathrm{SA}_t(\mathcal{P})$ and $\mathrm{LS}_t(\mathcal{P})$ (defined properly in Section 2.1 and Section 2.2 respectively) refers to the t-th level lift of polytope $\mathcal{P} \subseteq [0,1]^m$ in the Sherali-Adams hierarchy and Lovász-Schrijver hierarchy respectively, which can be optimized over in time that is polynomial in the size of LP (P1) and m^ℓ. Thus, we consider the following LPs.

$$
\min \left\{ \sum_{e \in E} c_e \cdot y_{\{e\}} : y \in \mathrm{SA}_\ell(\mathcal{P}) \right\} \qquad \text{(SA-LP)}
$$

$$
\min \left\{ \sum_{e \in E} c_e \cdot y_e : y \in \mathrm{LS}_{2\ell}(\mathcal{P}) \right\} \qquad \text{(LS-LP)}
$$

[7] [4] groups nodes together in their analysis so that the tree has height $h = O(\log n)$. They then prove the gap is $O(h \cdot \log k)$. One could skip the grouping argument to directly prove the $O(\ell \cdot \log k)$ bound.

where $\mathcal{P}$ is the polytope given by the constraints of the LP relaxation (P1).

Theorem 2. *Then the integrality gap of LP (* SA-LP*) is $O(\ell \cdot \log k)$ in ℓ-layered instances of DST.*

Theorem 3. *Then the integrality gap of LP (* LS-LP*) is $O(\ell \cdot \log k)$ in ℓ-layered instances of DST.*

Note that Theorems 2 and 3 are incomparable; fewer rounds are used in the stronger Sherali-Adams hierarchy. For the sake of space, we will only present the proof of Theorem 2 in this extended abstract. The proof of We can also find feasible DST solutions witnessing these integrality gap upper bounds.

Theorem 4. *Given oracle access to some fixed $y \in \mathrm{LS}_{2\ell}(\mathcal{P})$ or $y \in \mathrm{SA}_{\ell}(\mathcal{P})$, with high probability we can find a Directed Steiner Tree solution in time $O(\mathrm{poly}(n))$ of cost at most $O(\ell \cdot \log k)$ times the cost of y^*.*

Rothvoss proved an analogous result for the Lasserre SDP hieararchy [9], but his arguments relied on a particular decomposition theorem proven by Karlin, Mathieu, and Nguyen [8]. This decomposition theorem does not hold in weaker LP hierarchies.

The algorithm for rounding a point in $\mathrm{SA}_{\ell}(\mathcal{P})$ lifted LP is quite different from the algorithm for rounding a point in $\mathrm{LS}_{2\ell}(\mathcal{P})$. At a high level, we prove Theorem 2 by mapping a point y^* in the Sherali-Adams lifted polytope into an LP solution with the same cost as y^* for a related Group Steiner Tree instance. Using Theorem 1, we find a GST solution with cost $O(\ell \cdot \log k)$ times the cost of y^* and this will naturally correspond to a DST solution in G. This construction does not need to be made explicit; one can emulate the GST rounding algorithm in [4] in an expected $O(\mathrm{poly}(n))$ steps given oracle access to $y \in \mathrm{SA}_{\ell}(\mathcal{P})$.

However, these techniques do not seem to help in proving Theorem 3. We prove Theorem 3 by employing a different algorithm to round LP (LS-LP). Roughly speaking, we start from the terminals, then iteratively extend the paths by adding edges in a bottom-up fashion guided by probabilities given by the LP.

As a warmup, we also obtain the following interesting bound that shows lift-and-project techniques are not necessary for graphs having 3 layers.

Theorem 5. *The integrality gap of LP (P0) is $O(\log k)$ in 3-layered graphs.*

As with Theorem 2, this is obtained by mapping a point in LP (P0) to an LP solution for the corresponding GST instance. However, the restriction to only 3 layers allows us to accomplish this without the use of hierarchies. In contrast, the integrality gap of LP (P0) is $\Omega(\sqrt{k})$ in some graphs with 4 layers [12].

The paper is organized as follows. Section 2 describes the hierarchies and introduces some additional notation. The proof of Theorem 5 is presented in Section 3. The proof of Theorem 2 is outlined in Section 4. Finally, the rounding algorithms for both hierarchies are described in Section 5.

2 Preliminaries

2.1 The Sherali-Adams Hierarchy

Consider a polytope $\mathcal{P} \subseteq \mathbb{R}^n$ specified by m linear constraints $\sum_{i=1}^n A_{j,i} \cdot x_i \geq b_j, 1 \leq j \leq m$. Suppose the "box constraints" $0 \leq x_i$ and $x_i \leq 1$ (equivalently, $-x_i \geq -1$) appear among these constraints for each $1 \leq i \leq n$.

For $t \geq 0$, let $\mathscr{P}_t([n]) = \{S \subseteq \{1, \ldots, n\} : |S| \leq t\}$ denote the collection of subsets of $\{1, \ldots, n\}$ of size at most t. We also let $\mathbb{R}^{\mathscr{P}_t([n])}$ denote $\mathbb{R}^\alpha$ where $\alpha = |\mathscr{P}_t([n])| = n^{O(t)}$. We index a vector $y \in \mathbb{R}^{\mathscr{P}_t([n])}$ by sets in $\mathscr{P}_t([n])$. The Sherali-Adams hieararchy (introduced in [10]) is described as follows.

Definition 2. $\mathrm{SA}_t(\mathcal{P})$ *is the set of vectors* $y \in \mathbb{R}^{\mathscr{P}_{t+1}([n])}$ *satisfying* $y_\emptyset = 1$ *and*

$$\sum_{H \subseteq J} (-1)^{|H|} \cdot \left(\sum_{i=1}^n A_{j,i} \cdot y_{I \cup H \cup \{i\}} - b_j \cdot y_{I \cup H} \right) \geq 0 \tag{5}$$

for each $j = 1, \ldots, m$ *and each pair of subsets of indices* $I, J \subseteq \{1, \ldots, n\}$ *having* $|I| + |J| \leq t$.

If $\mathcal{P}$ is described by m linear constraints over n variables, then $\mathrm{SA}_t(\mathcal{P})$ has $n^{O(t)}$ variables and $n^{O(t)} m$ constraints. So, we can solve the LP

$$\min \left\{ \sum_{i=1}^n c_i \cdot y_{\{i\}} : y \in \mathrm{SA}_\ell(\mathcal{P}) \right\}$$

with only $\mathrm{poly}(n^\ell)$ overhead over the running time of solving $\min \left\{ c^T x : x \in \mathcal{P} \right\}$.

We only use some of the many well-known properties of the Sherali-Adams hierarchy.

Lemma 1. *Suppose* $y \in \mathrm{SA}_t(\mathcal{P})$ *for some* $t \geq 0$. *Then the following hold.*

- *For any* $A \in \mathscr{P}_t([n])$ *such that* $y_A > 0$, *let* $y' \in \mathbb{R}^{\mathscr{P}_{t+1-|A|}([n])}$ *be defined by* $y'_I = \frac{y_{I \cup A}}{y_A}$. *Then* $y' \in \mathrm{SA}_{t-|A|}(\mathcal{P})$.
- *For any* $A \subseteq B \subseteq [n]$ *with* $|B| \leq t+1$, *we have* $y_B \leq y_A$.

Furthermore, $\mathcal{P}$ *and the projection of* $\mathrm{SA}_t(\mathcal{P})$ *to the singletons have the same integral solutions.*

In particular, the last statement implies that if $\mathcal{P}$ is an LP relaxation of a $\{0,1\}$ integer program, then $\mathrm{SA}_t(\mathcal{P})$ is also a relaxation for the same integer program for any $t \geq 0$.

Proof. For the first statement, we have $y'_\emptyset = \frac{y_A}{y_A} = 1$. Furthermore, for any pair of indices I', J' with $|I'| + |J'| \leq t - |A|$ we are given that $y \in \mathrm{SA}_t(\mathcal{P})$ satisfies (5) with $I = I' \cup A$ and $J = J'$. Scaling this bound by $\frac{1}{y_A}$ shows that (5) holds for y' using $I = I'$ and $J = J'$.

The second part can be proved by induction on $|B \setminus A|$ with the base case being $|B| = |A| + 1$. If $B = A \cup \{i\}$ then using constraint $x_i \leq 1$ and (5) with $I = A$ and $J = \emptyset$ directly shows $y_B \leq y_A$.

Finally, if $z \in \mathcal{P}$ is integral then it is straightforward to check that $y_S = \prod_{i \in S} z_i$ is an integer point in $\mathrm{SA}_t(\mathcal{P})$ with $y_{\{i\}} = z_i$. Conversely, the restriction of *any* point $y \in \mathrm{SA}_t(\mathcal{P})$ to the singletons is a point of $\mathcal{P}$: take $I = J = \emptyset$ in (5).

2.2 The Lovász-Schrijver Hierarchy

Given a convex set $\mathcal{P} \subseteq [0,1]^n$, we convert it to a cone in $\mathbb{R}^{n+1}$ as follows.

$$\mathit{cone}(\mathcal{P}) = \{\mathbf{y} = (\lambda, \lambda x_1, \ldots, \lambda x_n) \mid \lambda \geq 0, (x_1, \ldots, x_n) \in \mathcal{P}\}$$

With a linear program given by constraints $\sum_{i=1}^n A_{j,i} \cdot x_i \geq b_j, 1 \leq j \leq m$, this is accomplished by *homogenizing* the constraints with a new variable x_0, yielding the cone $\{x \in \mathbb{R}^{d+1} : \sum_{i=1}^n A_{j,i} \cdot x_i \geq b_j \cdot x_0, 1 \leq j \leq m \text{ and } x_0 \geq 0\}$.

Definition 3. *For a cone $K \subseteq \mathbb{R}^{d+1}$ we define the set $N(K)$ (also a cone in $\mathbb{R}^{d+1}$) as follows: a vector $\mathbf{y} = (y_0, \ldots, y_d) \in \mathbb{R}^{d+1}$ is in $N(K)$ if and only if there is a matrix $Y \in \mathbb{R}^{(d+1)\times(d+1)}$ such that*

1. *Y is symmetric*
2. *For every $i \in \{0, 1, \ldots, d\}$, $Y_{0,i} = Y_{i,i} = y_i$*
3. *Each row Y_i is an element of K*
4. *Each vector $Y_0 - Y_i$ is an element of K*

The matrix Y is said to be a protection matrix *for $y \in N(K)$. For $t \geq 0$ we recursively define the cone $N^t(K)$ as $N^0(K) = K$ and $N^t(K) = N(N^{t-1}(K))$.*

Then we project the cone back to the desired space.

Definition 4. $\mathrm{LS}_t(\mathcal{P})$ *is the set of vectors $\mathbf{y} \in N^t(\mathit{cone}(\mathcal{P}))$ with $y_0 = 1$.*

Lovász and Schrijver [13] showed that if we start from a LP relaxation of a 0-1 integer program with n variables, then $\mathrm{LS}_n(\mathcal{P})$ is a tight relaxation in the sense that the only feasible solutions are convex combinations of integral solutions. In addition, if we start with a LP relaxation with poly(n) inequalities, we can obtain an optimal solution over the set of solutions given by t levels of LS in $n^{O(t)}$ time.

One key fact that is derived easily from Definition 3 is the following.

Lemma 2. *If $\mathbf{y} = (1, \mathbf{x}) \in \mathrm{LS}_t(\mathcal{P})$ with protection matrix Y, for any $i \in \{1, \ldots, n\}$ such that $x_i > 0$ consider the vector $\mathbf{y}' = \frac{1}{x_i} Y_i$. Then $\mathbf{y}' \in \mathrm{LS}_{t-1}(\mathcal{P})$ with $y'_i = 1$.*

2.3 Notation

Suppose G is an ℓ-layered instance of Directed Steiner Tree with root r, terminals X, and layers $\{r\} = V_0, V_1, \dots, V_\ell = X$. We will assume every $v \in V$ can be reached by r. In particular, for every $v \in V_1$ we have $rv \in E$.

Say a path in G is *rooted* if it begins at r. The notation $\langle v_j, v_{j+1}, v_{j+2}, \dots, v_i \rangle$ refers to a path in G that follows edges $v_j v_{j+1}, v_{j+1} v_{j+2}, \dots, v_{i-1} v_i \in E$ in succession. The subscript of a vertex in this notation will always indicate which layer the node lies in. The notation $\langle e_j, e_{j+1}, e_{j+2}, \dots, e_i \rangle$ refers to a path in G that follows edges $e_j, e_{j+1}, \dots, e_i \in E$ in succession. The subscript of an edge in this notation will always indicate which layer the (directed) edge starts from.

For any node $v \in V(G)$ we let

$$Q(v) = \{\langle r, v_1, v_2, \dots, v_i \rangle : v_i = v\}$$

and for any $e \in E(G)$ we let

$$Q(e) = \{\langle r, v_1, v_2, \dots, v_i \rangle : v_{i-1} v_i = e\}$$

denote all rooted paths ending at node v or ending with edge e, respectively. More generally, for a vertex v and another vertex u or an edge e, we let $Q(v, u)$ and $Q(v, e)$ denote all paths starting at v and ending at u or ending with edge e, respectively. We let $Q(e, v)$ denote all paths starting with edge e and ending at v. It will also sometimes be convenient to think of a path as a set of edges $\{v_j v_{j+1}, \dots, v_{i-1} v_i\}$.

Definition 5. *Suppose $G = (V, E)$ is an ℓ-layered instance of DST with root r and k terminals X. Then we consider the Group Steiner Tree instance on a tree $\mathcal{T}(G)$ with terminal groups $X_t, t \in X$ defined as follows.*

- *The vertex set of $\mathcal{T}(G)$ consists of all rooted paths $\cup_{v \in V} Q(v)$ in G.*
- *For any rooted path $P \neq \langle r \rangle$, we connect P to its maximal proper rooted subpath and give this edge cost c_e, where $P \in Q(e)$. Denote this edge in $\mathcal{T}(G)$ by $m(P)$.*
- *For each terminal $t \in X$, we let $X_t = Q(t)$: the set of all $r - t$ paths in G.*

This construction is illustrated in Figure 1. We will not explicitly construct $\mathcal{T}(G)$ in our rounding algorithm described in Section 5. It is simply a tool for analysis.

The following is immediate from the construction of $\mathcal{T}(G)$.

Lemma 3. *Let $|V| = n$. The graph $\mathcal{T}(G)$ constructed from an ℓ-layered Directed Steiner Tree instance G is a tree with height ℓ when rooted at $\langle r \rangle$. For every GST solution in $\mathcal{T}(T)$ there is a DST solution in G of no greater cost, and vice-versa.*

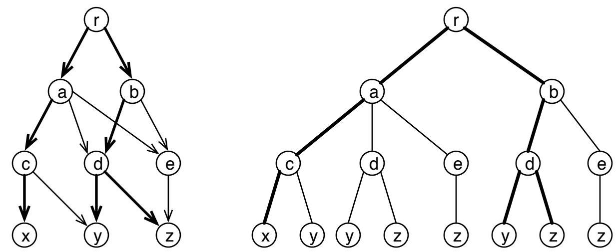

Fig. 1. A 3-layered DST instance with terminals $X = \{x, y, z\}$ (left) and the corresponding GST instance $\mathcal{T}(G)$ (right). Each node in $\mathcal{T}(G)$ corresponds to a path P in G and is labelled in the figure with the endpoint of P in G. A terminal group in $\mathcal{T}(G)$ in the figure consists of all leaf nodes with a common label. A DST solution and its corresponding GST solution are drawn with bold edges.

3 Rounding for 3-Layered Graphs

We first demonstrate that the natural LP relaxation (P0) for Directed Steiner Tree has an integrality gap of $O(\log k)$ in 3-layered graphs without using any lift-and-project machinery. As mentioned earlier, this complements the observation of Zosin and Khuller [12] that the integrality gap is $\Omega(\sqrt{k})$ in some 4-layered instances.

We show this by directly embedding a solution to the Directed Steiner Tree LP relaxation (P0) for some 3-layered instance G into a feasible LP solution to the Group Steiner Tree LP (GST-LP) on instance $\mathcal{T}(G)$. The reason we can do this with 3-layered instances is essentially due to the fact that for any edge $e = uv$ that either $v \in X$ or $|Q(e)| = 1$ (Figure 1 also helps illustrate this). This property does not hold in general for instances with at least 4 layers.

Consider a Group Steiner Tree instance $H = (V, E)$ with root r, terminal groups $X_1, X_2, \ldots, X_k \subseteq V$, and edge costs $c_e, e \in E$. The LP relaxation we consider for Group Steiner Tree is the following.

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e z_e && \text{(GST-LP)} \\ \text{s.t.} \quad & z(\delta(S)) \geq 1 \qquad \forall S \subseteq V - r, X_i \subseteq S \text{ for some group } X_i && (6) \\ & z \geq 0 \end{aligned}$$

Now we can prove Theorem 5.

Proof. Let $G = (V, E)$ be a 3-layered instance of Directed Steiner Tree with layers $\{r\} = V_0, V_1, V_2, V_3 = X$ and $\mathcal{T}(G)$ the corresponding Group Steiner Tree instance. Let x^* be an optimal solution to LP (P0). Note that for edge $uv \in E$ with $v \notin X$ there is a unique rooted path in G ending with e (i.e. $|Q(e)| = 1$).

We construct a feasible solution z^* to LP relaxation (GST-LP) for the Group Steiner Tree instance $\mathcal{T}(G)$. For every edge $e = uv$ of G where $v \notin X$, set

$z^*_{m(P)} := x^*_e$ where $Q(e) = \{P\}$. All that is left to set is the the z^*-value for the leaf edges of $\mathcal{T}(G)$.

To do this, fix a terminal $t \in X$. By the max-flow/min-cut theorem and Constraints (1), there is a flow f^t sending 1 unit of flow from r to t satisfying $f^t_e \leq x^*_e$ for every edge e. Furthermore, for each $e \in \delta^{in}(t)$ we may assume that $x^*_e = f^t_e$, otherwise we could reduce x^*_e while maintaining feasibility. Consider any path decomposition of f^t and say that this decomposition places weight w^t_P on a path $P \in Q(t)$. That is, $f^t_e = \sum_{P \in Q(t): e \in P} w^t_P$ for every edge $e \in G$. Then we set $z^*_{m(P)} := w^t_P$ for each $P \in Q(t)$.

We claim that z^* is a feasible solution for LP (GST-LP) with cost equal to $\sum_{e \in E} c_e x^*_e$. If so, then by Theorem 1, there is a Group Steiner Tree solution of cost at most $O(\log k)$ times the cost of x^*. We conclude by using Lemma 3 to note that there is then a Directed Steiner Tree solution of cost at most $O(\log k)$.

To see why z^* is feasible, we prove for every group t that there is a flow g^t of value 1 from $\langle r \rangle$ to the nodes in X_t with $g^t_{m(P)} \leq z^*_{m(P)}$ for every edge $m(P), P$ of H. By the max-flow/min-cut theorem, this means every constraint of (GST-LP) is satisfied by z^*. That such a flow exists essentially follows from the path decomposition of the flow f^t. Recall that a path decomposition of f^t placed weight w^t_P on $P \in Q(t)$. So, for each group X_t we define a flow g^t in $\mathcal{T}(G)$ by $g^t_{m(P)} = \sum_{P^* \in Q(t): P \subseteq P^*} w^t_{P^*}$.

Verifying that g^t is one unit of $r{-}X_t$ flow satisfying $g^t \leq z^*$ is straightforward; the details are left to the full version. It is also easy to see that the total z^*-value for paths ending with a copy of an edge e in G is equal to x^*_e, so the x^* and z^* have the same cost.

4 Sherali-Adams Gap for ℓ-Layered Graphs

Our basic approach for proving Theorem 2 is similar to our approach for Theorem 5. Let $\mathcal{P}$ denote the polytope defined by the constraints of LP (P1). We show how to embed a point y^* in the Sherali-Adams lift of LP (P1), namely $\mathrm{SA}_\ell(\mathcal{P})$, for an ℓ-layered instance G to a feasible solution to LP (GST-LP) for the corresponding Group Steiner Tree instance $\mathcal{T}(G)$.

Describing the embedding is straightforward. For every edge $m(P)$ in $\mathcal{T}(G)$, simply set $z^*_{m(P)} := y^*_P$. The rest of our analysis shows that z^* is feasible for LP (GST-LP) for instance $\mathcal{T}(G)$ and the cost of z^* in (GST-LP) is equal to $\sum_{e \in E} c_e \cdot y^*_{\{e\}}$.

Before delving into the proofs of these statements, we note a helpful technical result about the structure of Sherali-Adams solutions.

Lemma 4. *Suppose* $0 \leq i < j \leq \ell$. *For any node* $v \in V_i$, *any edge* $e = uw$ *with* $w \in V_j$, *and any* $y \in \mathrm{SA}_{j-i}(\mathcal{P})$ *we have* $\sum_{P \in Q(v,e)} y_P \leq y_{\{e\}}$. *Furthermore, if* $v = r$ *then this bound holds with equality.*

Note that $|P| = j - i$ for any $P \in Q(v, e)$ so it is valid to index $y \in \mathrm{SA}_{j-i}(\mathcal{P})$ with P in the sum.

Proof. We prove this by induction on $j - i$. The base case $j = i + 1$ is trivial since either $Q(v, e) = \emptyset$ (so the sum in question is 0) or $Q(v, e)$ consists of the singleton path that only uses edge e (so the sum in question is just $y_{\{e\}}$ already). Furthermore, if $v = r$ then $e \in \delta^{out}(r)$ so the bound holds with equality.

Inductively, suppose $j > i + 1$. If $y_{\{e\}} = 0$ then by Lemma 1 we have $y_P \leq y_{\{e\}} = 0$ for every $P \in Q(v, e)$ so the bound holds with equality. Otherwise, define the conditioned solution $y^{(e)} \in \mathrm{SA}_{j-i-1}(\mathcal{P})$ by $y^{(e)}_I = \frac{y_{I \cup \{e\}}}{y_{\{e\}}}$ for every $I \subseteq E, |I| \leq i - j$ (cf. Lemma 1). Then

$$\begin{aligned}\sum_{P \in Q(v,e)} y_P &= \sum_{e' \in \delta^{in}(u)} \sum_{P \in Q(v,e')} y_{P \cup \{e\}} = y_{\{e\}} \sum_{e' \in \delta^{in}(u)} \sum_{P \in Q(v,e')} y^{(e)}_P \\ &\leq y_{\{e\}} \sum_{e' \in \delta^{in}(u)} y^{(e)}_{\{e'\}} = y_{\{e\}}\end{aligned}$$

where the inequality follows by induction (note that the endpoint of e' is in V_{j-1}). The last equality follows by Constraints (3) and (4) of LP (P1) plus the fact that $y^{(e)}_{\{e\}} = 1$. Finally, if $v = r$ then the inequality above holds with equality by induction, so $\sum_{P \in Q(v,e)} y_P = y_{\{e\}}$.

4.1 Cost Analysis

The cost bound is an easy consequence of Lemma 4.

Lemma 5. *The cost of z^* in LP (*GST-LP*) is $\sum_{e \in E(G)} c_e \cdot y^*_{\{e\}}$.*

Proof.

$$\begin{aligned}\sum_{m(P) \in E(\mathcal{T}(G))} c_{m(P)} \cdot z^*_{m(P)} &= \sum_{e \in E(G)} \sum_{P \in Q(e)} c_e \cdot z^*_{m(P)} \\ &= \sum_{e \in E(G)} \sum_{P \in Q(e)} c_e \cdot y^*_P = \sum_{e \in E(G)} c_e \cdot y^*_{\{e\}}\end{aligned}$$

where the last equality is by Lemma 4 applied with $v = r$.

4.2 Feasibility

Similar to the proof of Theorem 5, for every group X_t we construct a unit $\langle r \rangle - X_t$ flow g^t in $\mathcal{T}(G)$ which satisfies the capacities given by z^*. Thus, by the max-flow/min-cut theorem we have that $z^*(\delta(S)) \geq 1$ for every subset $S \subseteq V(\mathcal{T}(G)) - \langle r \rangle$ such that $X_t \subseteq S$ for some group X_t.

We now fix a terminal $t \in X$ and describe the flow g^t by giving a path decomposition of the flow. For each $P \in Q(t)$, we assign a weight of y^*_P to the $\langle r \rangle - P$ path in $\mathcal{T}(X)$. So, the flow $g^t_{m(P)}$ crossing edge $m(P)$ in $\mathcal{T}(G)$ is just $\sum_{P^* \in Q(t) : P \subseteq P^*} y^*_{P^*}$.

Lemma 6. *g^t is one unit of $\langle r \rangle - X_t$ flow in $\mathcal{T}(G)$.*

Proof. It is an $\langle r\rangle - X_t$ flow because we constructed it from a path decomposition using only paths in $Q(t)$. Furthermore,

$$\begin{aligned} g^t(\delta^{out}_{\mathcal{T}(G)}(\langle r\rangle)) &= \sum_{P^* \in Q(t): \langle r\rangle \subseteq P^*} y^*_{P^*} = \sum_{P \in Q(t)} y^*_P \\ &= \sum_{e \in \delta^{in}_G(t)} \sum_{P \in Q(e)} y^*_P = \sum_{e \in \delta^{in}_G(t)} y^*_{\{e\}} = 1. \end{aligned}$$

Here, the second last equality follows from Lemma 4. The last equality follows from combining Constraint (2) with Constraint (3) for $v = t$.

All that is left is to prove that each flow g^t for a terminal group X_t satisfies the capacities given by z^*. The following lemma is the heart of this argument. A similar result was proven in [9] which relied on the strong decomposition property for the Lasserre hierarchy of Karlin, Mathieu, and Nguyen [8]. We emphasize that our proof only uses properties of the Sherali-Adams LP hierarchy.

Lemma 7. *For every rooted path P and every terminal group X_t, we have $\sum_{P^* \in Q(t): P \subseteq P^*} y^*_{P^*} \leq y^*_P$.*

Proof. If $y^*_P = 0$ this is trivial since $y^*_{P^*} \leq y^*_P$ for any $P^* \supseteq P$ by Lemma 1. Otherwise, form the conditioned solution $y^{(P)} \in \mathrm{SA}_{\ell - |P|}(\mathcal{P})$ by $y^{(P)}_I = \frac{y^*_{P \cup I}}{y^*_P}$ for any $|I| \leq \ell + 1 - |P|$.

Say v is the endpoint of P. Note that $y^{(P)} \in \mathrm{SA}_{\ell-|P|}(\mathcal{P})$ by Lemma 1. So, for any $e \in \delta^{in}(t)$ we have $\sum_{P^* \in Q(v,e)} y^{(P)}_{P^*} \leq y^{(P)}_{\{e\}}$ from Lemma 4 (with $i = |P|$ and $j = \ell$). Summing over all $e \in \delta^{in}(t)$ while using Constraints (3) and the fact that the projection of $y^{(P)}$ to the singleton sets is a point in $\mathcal{P}$, we have $\sum_{P^* \in Q(v,t)} y^{(P)}_{P^*} \leq 1$.

Multiplying both sides of this bound by y^*_P, we see that $\sum_{P^* \in Q(v,t)} y^*_{P \cup P^*} \leq y^*_P$. But the left hand side is precisely $\sum_{P^* \in Q(t): P \subseteq P^*} y^*_{P^*}$, which is what we were required to show.

We can now easily verify that the capacity constraints are satisfied.

Corollary 1. *For every terminal group X_t and every edge $m(P)$ of $\mathcal{T}(G)$, $g^t_{m(P)} \leq z^*_{m(P)}$.*

Proof. By Lemma 7 $g^t_{m(P)} = \sum_{\substack{P^* \in Q(t) \\ P \subseteq P^*}} y^*_{P^*} \leq y^*_P = z^*_{m(P)}$.

The proof of Theorem 2 is now complete.

5 Rounding Algorithms

5.1 Sherali-Adams Rounding Algorithm

We bounded the integrality gap of LP (SA-LP) by converting some $y \in \mathrm{SA}_\ell(\mathcal{P})$ to a feasible solution for LP (GST-LP) in $\mathcal{T}(G)$. However, this mapping does not have to be explicitly constructed to round y. Instead, we emulate the GST rounding algorithm in [4] by simply querying the y_P variables as needed. Algorithm 1 describes the main subroutine from [4] in our context.

Algorithm 1 Sherali-Adams Rounding Subroutine

1: $S_0 \leftarrow \{\langle r \rangle\}$
2: **for** $j = 1, \ldots, \ell$ **do**
3: $\quad S_j \leftarrow \emptyset$
4: $\quad$ **for** each $P \in S_{j-1}$ **do**
5: $\quad\quad$ **for** each $e \in \delta^{out}(v)$ where v is the endpoint of P **do**
6: $\quad\quad\quad$ Add $\langle P, e \rangle$ to S_j with probability $\frac{y^*_{P \cup \{e\}}}{y^*_P}$
7: $F \leftarrow$ edges used by some path in S_ℓ
8: **return** F

As in [4], the expected cost of F is $\sum_{e \in E} c_e y_{\{e\}}$ and, for each terminal $t \in X$, the probability that F contains an $r - t$ path is at least $\frac{1}{\ell}$. Iterating this procedure sufficiently many times gives us a feasible DST solution with cost at most $O(\ell \cdot \log k)$ times the cost of y.

In fact, it is easy to see that in one run of Algorithm 1 we have for any rooted path P ending in layer i that $\Pr[P \in S_i] = y^*_P$. This leads to an interesting observation which, ultimately, means the expected running time of Algorithm 1 is polynomial in n.

Lemma 8. $\mathrm{E}\left[\sum_{i=0}^{\ell} |S_i|\right] \leq n$

Proof. As noted before, for any edge $e = uv$ with $v \in V_i$ and any $P \in Q(e)$ we have $\Pr[P \in S_i] = y^*_P$. Thus, $\mathrm{E}[|S_i \cap Q(e)|] = \sum_{P \in Q(e)} y^*_P = y^*_{\{e\}}$ where the second equality is by Lemma 4. Summing over all edges e and using Constraints (3) shows $\sum_{e \in E} y^*_{\{e\}} \leq n-1$. Finally, since $y^*_{\langle r \rangle} = y^*_\emptyset = 1$ then $\mathrm{E}\left[\sum_{i=0}^{\ell} |S_i|\right] \leq n$.

This also completes the proof of Theorem 4 for the Sherali-Adams rounding since the total number of iterations of the loop in Step (4) of Algorithm 1 is precisely $\sum_{i=0}^{\ell-1} |S_i|$, which is polynomial in expectation. Therefore, with high probability the running time of the entire rounding algorithm is polynomial in n.

5.2 Lovász-Schrijver Rounding Algorithm

We introduce a bit more notation to describe the rounding algorithm. We start with some $y \in \mathrm{LS}_{2\ell}(\mathcal{P})$ with corresponding protection matrix Y. For $0 < j \leq \ell$, consider some path $P = \langle v_j, v_{j+1}, \ldots, v_\ell \rangle$ ending at some terminal $v_\ell \in X$. We let y^P denote a point in $\mathrm{LS}_{\ell+j}(\mathcal{P})$ and Y^P be a corresponding protection matrix, which we define inductively.

If $j = \ell$ (so $P = \langle v_\ell \rangle$) then we simply let $y^P = y$ and $Y^P = Y$. For $j < \ell$, let y^P be the point obtained by conditioning $y^{P'}$ on $y^{P'}_{v_j v_{j+1}} = 1$ where $P' =$

$\langle v_{j+1}, v_{j+2}, \ldots, v_\ell \rangle$. Then Y^P is the protection matrix witnessing the inclusion of row $Y^{P'}_{v_j v_{j+1}}$ in $N^{\ell+j+1}(cone(\mathcal{P}))$ (scaled by $\frac{1}{y_e^{P'}}$ to ensure $Y_0^P = y^P$). This definition only makes sense if $y_e^{P'} > 0$ for every suffix $\langle e, P' \rangle$ of P; this will be the case for every path P constructed in the algorithm.

The algorithm for rounding the Sherali-Adams relaxation does not does not work for the Lovász-Schrijver hierarchy because a direct analogue of Lemma 4 fails to hold in this case. However, using the constraint that the indegree of every node is at most 1, we are able to prove an analogue when we consider paths going as edge to a particular terminal, instead of paths from the root to an edge. We utilize this by building the tree in a "bottom-up" fashion in our algorithm.

Algorithm 2 contains the main subroutine for the Lovász-Schrijver rounding procedure. As with the Sherali-Adams rounding procedure, we iterate Algorithm 2 until there is an $r - t$ path for every terminal t in the union of the returned sets of edges F.

Algorithm 2 Lovász-Schrijver Rounding Subroutine

```
F ← ∅, C ← ∅
S_t ← ∅ for each t ∈ X
for t ∈ X do
    for each e ∈ δ^in(t) do
        Add ⟨e⟩ to S_t independently with probability y_e.
    for j = 1, ..., l do
        for each u − t path P of length j in S_t do
            if u ∉ C then
                for each e ∈ δ^in(u) do
                    Add ⟨e, P⟩ to S_t with probability y_e^P
    Add edges in S_t to F, and the vertices covered by S_t to C
return F
```

As mentioned before, the proofs of Theorems 3 and 4 for this rounding procedure will appear in the full version.

References

1. G. Calinescu and G. Zelikovsky. The polymatroid Steiner problems. J. Combinatorial Optimization, 9(3):281–294, 2005.
2. M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. J. Algorithms, 33(1):73–91, 1999.
3. E. Chlamtáč and M. Tulsiani. Convex relaxations and integrality gaps. Handbook on Semidefinite, Conic and Polynomial Optimization, Springer, 2012.
4. N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. J. Algorithms, 37(1):66–84, 2000.
5. A. Gupta, K. Talwar, and D. Witmer. Sparsest cut on bounded treewidth graphs: algorithms and hardness results. In proceedings of STOC, 2013.

6. V. Guruswami and A. K. Sinop. Faster SDP hierarchy solvers for local rounding algorithms. In proceedings of FOCS, 2012.
7. E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In proceedings of STOC, 2003.
8. A. Karlin, C. Mathieu, and C. Nguyen. Integrlaity gaps of linear and semidefinite programming relaxations for knapsack. In proceedings of IPCO, 2011.
9. T. Rothvoss. Directed Steiner tree and the Lasserre hierarchy. CoRR abs/1111.5473, 2011.
10. H. Sherali and W. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. SIAM J. Discrete Math., 3:411–430, 1990.
11. A. Zelikovsky. A series of approximation algorithms for the acyclic directed Steiner tree problem. Algorithmica, 18:99–110, 1997.
12. L. Zosin and S. Khuller. On directed Steiner trees. In proceedings of SODA, 2002.
13. L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. SIAM Journal on Optimization, 1:166–190, 1991.