Operations Research Letters 38 (2010) 567–570

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Integrality gap of the hypergraphic relaxation of Steiner trees: A short proof of a 1.55 upper bound

Deeparnab Chakrabarty[a], Jochen Könemann[b], David Pritchard[c,*]

[a] University of Pennsylvania, United States
[b] University of Waterloo, Canada
[c] École Polytechnique Fédérale de Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 12 June 2010
Accepted 6 September 2010
Available online 17 September 2010

Keywords:
Hypergraph
Integrality gap
Randomized algorithm
Steiner tree

ABSTRACT

Recently, Byrka, Grandoni, Rothvoßand Sanità gave a 1.39 approximation for the Steiner tree problem, using a hypergraph-based linear programming relaxation. They also upper-bounded its integrality gap by 1.55. We describe a shorter proof of the same integrality gap bound, by applying some of their techniques to a randomized loss-contracting algorithm.

1. Introduction

In the Steiner tree problem, we are given an undirected graph $G = (V, E)$ with costs c on edges and its vertex set partitioned into terminals (denoted $R \subseteq V$) and Steiner vertices ($V \setminus R$). A *Steiner tree* is a tree spanning all of R plus any subset of $V \setminus R$, and the problem is to find a minimum-cost such tree. The Steiner tree problem is APX-hard, thus the best we can hope for is a constant-factor approximation algorithm.

The best known ratio is obtained by Byrka et al. [1]: their randomized iterated rounding algorithm gives approximation ratio $\ln(4) + \epsilon \approx 1.39$. The prior best was a $1 + \frac{\ln 3}{2} + \epsilon \approx 1.55$ ratio, via the deterministic loss-contracting algorithm of Robins and Zelikovsky [6]. The algorithm of [1] differs from previous work in that it uses a linear programming (LP) relaxation; the LP is based on hypergraphs, and it has several different looking but equivalent [2,5] nice formulations. A second result of [1] concerns the LP's *integrality gap*, which is defined as the worst-case ratio (max over all instances) of the optimal Steiner tree cost to the LP's optimal value. Byrka et al. show that the integrality gap is at most 1.55, and their proof builds on the analysis of [6]. In this note we give a shorter proof of the same bound using a simple LP-rounding algorithm.

We now describe one formulation for the hypergraphic LP. Given a set $K \subseteq R$ of terminals, a *full component* on K is a tree whose leaf set is K and whose internal nodes are Steiner vertices. Without loss of generality, Steiner trees have no Steiner nodes of degree 1, and under this condition they decompose in a unique edge-disjoint way into full components; Fig. 1(i) and (ii) show an example. Moreover, one can show that a set of full components on sets $\{K_1, \dots, K_r\}$ forms a Steiner tree if and only if the hypergraph $(V, \{K_1, \dots, K_r\})$ is a *hyper-spanning tree*. Here, a hyper-spanning tree means that there is a unique path (simple alternating vertex-hyperedge sequence of incidences) connecting every pair of vertices. Let $\mathrm{F}(K)$ denote a minimum-cost full component for terminal set $K \subseteq R$, and let C_K be its cost. The hypergraphic LP is as follows:

$$\begin{aligned}
&\min && \sum_K C_K x_K : && (\mathcal{S})\\
&\forall \emptyset \neq S \subseteq R : && \sum_{K : K \cap S \neq \emptyset} x_K(|K \cap S| - 1) \le |S| - 1 &&\\
& && \sum_K x_K(|K| - 1) = |R| - 1 &&\\
&\forall K : && x_K \ge 0. &&
\end{aligned}$$

The integral solutions of ($\mathcal{S}$) correspond to the full component sets of Steiner trees. As an aside, the *r-restricted full component* method (e.g. [4]) allows us to assume that there are a polynomial number of full components while affecting the optimal Steiner tree cost by a $1 + \epsilon$ factor. Then, it is possible to solve ($\mathcal{S}$) in polynomial time [1,8]. Here is our goal.

* Corresponding address: EPFL SB IMA DISOPT, Station 8, CH-1015 Lausanne, Switzerland.
E-mail address: david.pritchard@epfl.ch (D. Pritchard).

0167-6377/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2010.09.004

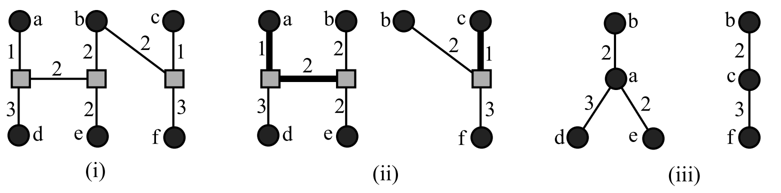

Fig. 1. In (i) we show a Steiner tree; circles are terminals and squares are Steiner nodes. In (ii) we show its decomposition into full components, and their losses in bold. In (iii) we show the full components after loss contraction.

Theorem 1 ([1]). *The integrality gap of the hypergraphic LP* ($\mathcal{S}$) *is at most* $1 + (\ln 3)/2 \approx 1.55$.

2. Randomized loss-contracting algorithm

In this section we describe the algorithm. We introduce some terminology first. The *loss* of full component $\mathtt{F}(K)$, denoted by $\mathtt{Loss}(K)$, is a minimum-cost subset of $\mathtt{F}(K)$'s edges that connects the Steiner vertices to the terminals. For example, Fig. 1(ii) shows the loss of the two full components in bold. We let $\mathtt{loss}(K)$ denote the total cost of all edges in $\mathtt{Loss}(K)$. The *loss-contracted full component of* K, denoted by $\mathtt{LC}(K)$, is obtained from $\mathtt{F}(K)$ by contracting its loss edges (see Fig. 1(iii) for an example).

For clarity we make two observations. First, for each K the edges of $\mathtt{LC}(K)$ correspond to the edges of $\mathtt{F}(K) \setminus \mathtt{Loss}(K)$. Second, for terminals u, v, a uv edge may appear in $\mathtt{LC}(K_1)$ and $\mathtt{LC}(K_2)$ for distinct full components K_1 and K_2; but we think of them as distinct parallel edges.

Our randomized rounding algorithm, RLC, is shown below. We choose M to have value at least $\sum_K x_K$ such that $t = M \ln 3$ is integral. $\mathtt{MST}(\cdot)$ denotes a minimum spanning tree and $\mathtt{mst}$ its cost.

Algorithm RLC.

```
1: Let $T_1$ be a minimum spanning tree of the induced graph $G[R]$.
2: $x \leftarrow$ Solve ($\mathcal{S}$)
3: for $1 \le i \le t$ do
4:     Sample a full component $K_i$: with probability $x_K/M$ it is the full component $K$, with probability $1 - \sum_K x_K/M$ it is the empty set (we sample with replacement)
5:     $T_{i+1} \leftarrow \mathtt{MST}(T_i \cup \mathtt{LC}(K_i))$
6: end for
7: Output any Steiner tree in $ALG := T_{t+1} \cup \bigcup_{i=1}^{t} \mathtt{Loss}(K_i)$.
```

To prove that *ALG* actually contains a Steiner tree, we must show that all terminals are connected. To see this, note that each edge uv of T_{t+1} is either a terminal–terminal edge of $G[R]$ in the input instance, or else $uv \in \mathtt{LC}(K_i)$ for some i and therefore a u–v path is created when we add in $\mathtt{Loss}(K_i)$.

3. Analysis

In this section we prove that the tree's expected cost is at most $1 + \frac{\ln 3}{2}$ times the optimum value of ($\mathcal{S}$). Each iteration of the main loop of algorithm RLC first samples a full component K_i in step 4, and subsequently recomputes a minimum-cost spanning tree in the graph obtained by adding the loss-contracted part of K_i to T_i. The new spanning tree T_{i+1} is no more expensive than T_i; some of its edges are replaced by newly added edges in $\mathtt{LC}(K_i)$. Bounding the drop in cost will be the centerpiece of our analysis, and this step will in turn be facilitated by the elegant *Bridge Lemma* of Byrka et al. [1]. We describe this lemma first.

We first define the *drop* of a full component K with respect to a terminal spanning tree T (it is just a different name for the bridges of [1]). Let T/K be the graph obtained from T by identifying the terminals spanned by K. Then let

$$\mathtt{Drop}_T(K) := E(T) \setminus E(\mathtt{MST}(T/K)),$$

be the set of edges of T that are not contained in a minimum spanning tree of T/K, and $\mathtt{drop}_T(K)$ be its cost. We illustrate this in Fig. 2. We state the Bridge Lemma here and present its proof for completeness.

Lemma 1 (*Bridge Lemma [1]*). *Given a terminal spanning tree* T *and a feasible solution* x *to* ($\mathcal{S}$),

$$\sum_K x_K \mathtt{drop}_T(K) \ge c(T). \tag{1}$$

Proof. The proof needs the following theorem [3]: given a graph $H = (R, F)$, the extreme points of the polytope

$$\left\{ z \in \mathbb{R}^F_{\ge 0} : \sum_{e \in \gamma(S)} z_e \le |S| - 1; \forall S \subseteq R, \sum_{e \in F} z_e = |R| - 1 \right\} \tag{$\mathcal{G}$}$$

are the indicator variables of spanning trees of H, where $\gamma(S) \subseteq F$ is the set of edges with both endpoints in S. The proof strategy is as follows. We construct a multigraph $H = (R, F)$ with costs c, and $z \in \mathbb{R}^F$ such that the cost of z equals the left-hand side of (1); $z \in$ ($\mathcal{G}$), and all spanning trees of H have cost at least $c(T)$. Edmonds' theorem then immediately implies the lemma. In the rest of the proof we define H and supply the three parts of this strategy.

For each full component K with $x_K > 0$, consider the edges in $\mathtt{Drop}_T(K)$. Contracting all edges of $E(T) \setminus \mathtt{Drop}_T(K)$, we see that $\mathtt{Drop}_T(K)$ corresponds to edges of a spanning tree of K. These edges are copied (with the same cost c) into the set F, and the copies are given weight $z_e = x_K$. Using the definition of drop, one can show that each $e \in F$ is a maximum-cost edge in the unique cycle of $T \cup \{e\}$.

Having now defined F, we see that

$$\sum_{e \in F} c_e z_e = \sum_K x_K \mathtt{drop}_T(K).$$

Note that we introduce $|K| - 1$ edges for each full component K, and that, for any $S \subseteq R$, at most $|S \cap K| - 1$ of these have both ends in S. These two observations together with the fact that x is feasible for ($\mathcal{S}$) directly imply that z is feasible for ($\mathcal{G}$).

To show all spanning trees of H have cost at least $c(T)$, it suffices to show that T is an MST of $T \cup H$. In turn, this follows (e.g. [7, Theorem 50.9]) from the fact that each $e \in F$ is a maximum-cost edge in the unique cycle of $T \cup \{e\}$. □

We also need two standard facts that we summarize in the following lemma. They rely on the input costs satisfying the triangle inequality (i.e. metricity), and that internal nodes of full components have degree at least 3, both of which hold without loss of generality.

Lemma 2. (a) *The value* $\mathtt{mst}(G[R])$ *of the initial terminal spanning tree computed by algorithm RLC is at most twice the optimal value of* ($\mathcal{S}$).(b) *For any full component* K, $\mathtt{loss}(K) \le C_K/2$.