Basic Research in Computer Science

BRICS RS-99-39 R. Rizzi: On the Steiner Tree $\frac{3}{2}$-Approximation for Quasi-Bipartite Graphs

On the Steiner Tree $\frac{3}{2}$-Approximation for Quasi-Bipartite Graphs

Romeo Rizzi

BRICS Report Series **RS-99-39**

ISSN 0909-0878 **November 1999**

See back inner page for a list of recent BRICS Report Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark
Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `RS/99/39/`

On the Steiner tree $\frac{3}{2}$-approximation for quasi-bipartite graphs

Romeo Rizzi

BRICS*
Department of Computer Science
University of Aarhus
Ny Munkegade
DK-8000 Aarhus C, Denmark
e-mail: romeo@cwi.nl

Abstract

Let $G = (V, E)$ be an undirected simple graph and $w : E \mapsto \mathbb{R}_+$ be a non-negative weighting of the edges of G. Assume V is partitioned as $R \cup X$. A *Steiner tree* is any tree T of G such that every node in R is incident with at least one edge of T. The *metric Steiner tree problem* asks for a Steiner tree of minimum weight, given that w is a metric. When X is a stable set of G, then (G, R, X) is called *quasi-bipartite*. In [1], Rajagopalan and Vazirani introduced the notion of quasi-bipartiteness and gave a $(\frac{3}{2} + \epsilon)$ approximation algorithm for the metric Steiner tree problem, when (G, R, X) is quasi-bipartite. In this paper, we simplify and strengthen the result of Rajagopalan and Vazirani. We also show how classical bit scaling techniques can be adapted to the design of approximation algorithms.

Key words: Steiner tree, local search, approximation algorithm, bit scaling.

1 Introduction

Let $G = (V, E)$ be an undirected simple graph and $w : E \mapsto \mathbb{R}_+$ be a non-negative weighting of the edges of G. Assume V is partitioned into R and $X = \overline{R}$. (From set theory, $\overline{R} = V \setminus R$). A *Steiner tree* is any tree T of G such

*Basic Research in Computer Science,
Centre of the Danish National Research Foundation.

that every node in R is incident with at least one edge of T. The *metric Steiner tree problem* asks for a Steiner tree of minimum weight, given that w is a metric. (The weight of a tree T is defined as $w(T) = \sum_{e \in T} w_e$). When no edge of G has both endpoints in X, then (G, R) is called *quasi-bipartite*. In [1], Rajagopalan and Vazirani introduced the notion of quasi-bipartiteness and gave a $(\frac{3}{2} + \epsilon)$ approximation algorithm for the metric Steiner tree problem, when (G, R) is quasi-bipartite. In this paper, we greatly simplify the analysis of Rajagopalan and Vazirani and give a local search $\frac{3}{2}$ approximation algorithm for the same problem. We also show how classical bit scaling techniques can be adapted to the design of approximation algorithms.

Denote by $MST(G, w)$ the minimum weight of a spanning tree for (G, w). When $S \subseteq V$, then $\langle S \rangle$ denotes the set of those edges in E with both endpoints in S and $G[S]$ denotes the *subgraph of G induced by S*. (That is, $G[S] = (S, \langle S \rangle)$).

The graph $G[V \setminus X]$, where $X \subseteq V$, is also denoted by $G \setminus X$ or by $G[\overline{X}]$. Let T be a minimum weight spanning tree for $(G[\overline{X}], w)$ and x be a node of X. Our arguments base on the following simple and fundamental lemma: if $MST(G[\overline{X} \cup \{x\}], w) \geq w(T)$ for every $x \in X$ and X is a stable set of G, then $MST(G[\overline{X} \cup X'], w) \geq w(T)$ for every $X' \subseteq X$.

2 An useful lemma

Let $G = (V, E)$ be an undirected simple graph and $w : E \mapsto \mathbb{R}_+$ be a nonnegative weighting of the edges of G. Let $X \subseteq V$ be a stable set of G. In the next section, the following lemma is put into use.

Lemma 1 *Let X be a stable set of G. Assume $MST(G[\overline{X} \cup \{x\}], w) \geq MST(G[\overline{X}], w)$ for every node $x \in X$. Then $MST(G[\overline{X} \cup X'], w) \geq MST(G[\overline{X}], w)$ for every $X' \subseteq X$.*

To prove Lemma 1, we need to introduce some notations and facts. Usually, we consider a tree T to be just a set of edges. Sometimes however, and depending on our convenience, a tree T will be regarded as the graph $(V(T), T)$, where $V(T)$ is the set of endnodes of edges in T. Let $V_1, \ldots, V_k$ be a partition of V. Denote by $G < V_1, \ldots, V_k >$ the graph obtained from G by identifying all nodes of V_i into a single node (for $i = 1, \ldots, k$).

Lemma 2 *Let $v_1, \dots, v_k$ the neighbors of a node x in G and $C_1, \dots, C_k$ a partition of $V \setminus \{x\}$ such that $v_i \in C_i$ (for $i = 1, \dots, k$). Assume $w(\delta(x)) < MST(G[V \setminus \{x\}] < C_1, \dots, C_k >, w)$. Then $MST(G, w) < MST(G \setminus \{x\}, w)$.*

Proof: Let T be any spanning tree for $(G \setminus \{x\}, w)$. It suffices to show that there always exists a spanning tree F of $T < C_1, \dots, C_k >$ such that $T \setminus F \cup \delta(x)$ is a spanning tree of G. Let v be a leaf of T and let vu be the edge of T incident with v. W.l.o.g. assume $v \in C_1$.
Case 1: Assume that $v \neq v_1$. Let $T' = T \setminus \{vu\}$ and G' be the graph obtained from G by identifying node v with node u. (All edges previously incident with v will become incident with u, and v is removed). Note that T' is a spanning tree for $G' \setminus \{x\}$. Let F be a spanning tree of $T' < C_1 \setminus \{v\}, C_2, \dots, C_k >$ such that $T' \setminus F \cup \delta(x)$ is a spanning tree of G'. But then, F is a spanning tree of $T < C_1, \dots, C_k >$ such that $T \setminus F \cup \delta(x)$ is a spanning tree of G.
Case 2: Assume therefore that $v = v_1$. This time G' and T' are obtained from G and T as follows. First remove edge vu from T, and in G, remove xv and identify v with u. Next, as long as T contains an edge ab with $a \in C_1$, then remove ab from T and in G identify a with b. Let G' and T' be the graph and the tree so obtained. Note that T' is a spanning tree for $G' \setminus \{x\}$. Let F be a spanning tree of $T' < C_2, \dots, C_k >$ such that $T' \setminus F \cup \delta(x)$ is a spanning tree of G'. But then, $F \cup \{vu\}$ is a spanning tree of $T < C_1, \dots, C_k >$ such that $T \setminus F \cup \delta(x)$ is a spanning tree of G. □

Proof of Lemma 1: It suffices to show that if $MST(G[\overline{X} \cup X'], w) < MST(G \setminus \{x\}, w)$ for some $X' \subseteq X$ with $|X'| \geq 2$, then $MST(G[\overline{X} \cup X''], w) < MST(G \setminus \{x\}, w)$ for some proper subset X'' of X'. Let T' be a spanning tree of $G[\overline{X} \cup X']$ with $w(T') < MST(G \setminus \{x\}, w)$. In a minimal counterexample, we can always assume that $X' = X$ and that every edge of G with an endpoint in X is contained in T'. Let x be any node in X and $X'' = X \setminus \{x\}$. Let $v_1, \dots, v_k$ be the neighbors of x in G. Consider the connected components $\tilde{C}_1, \dots, \tilde{C}_k$ of the graph obtained by removing node x from the graph T'. (Assume w.l.o.g. that $v_i \in \tilde{C}_i$, for $i = 1, \dots, k$). If $MST(G[\overline{X} \cup X''], w) \geq w(T')$, then $w(\delta(x)) < MST(G[V \setminus \{x\}] < \tilde{C}_1, \dots, \tilde{C}_k >, w)$. For $i = 1, \dots, k$, let $C_i = \tilde{C}_i \cap \overline{X}$. Since X is a stable set of G, then $G[\overline{X}] < C_1, \dots, C_k >= G[V \setminus \{x\}] < \tilde{C}_1, \dots, \tilde{C}_k >$. Hence $MST(G[\overline{X}] < C_1, \dots, C_k >, w) > w(\delta(x))$. By Lemma 2, $MST(G[\overline{X} \cup \{x\}], w) < MST(G[\overline{X}], w)$. □

3 The result

Let $G = (V, E)$ be an undirected simple graph and $w : E \mapsto \mathbb{R}_+$ be a non-negative weighting of the edges of G. In this section, we also assume that w satisfies the triangle inequality and that (G, R) is quasi-bipartite.

Let $\overline{T}$ be an optimal Steiner tree for (G, R, w). If we knew which of the nodes of X, say $\overline{I} \subseteq X$, are actually in $V(\overline{T})$, then we could find an optimal Steiner tree by computing a minimum spanning tree of $(G[R \cup \overline{I}], w)$. Moreover, since w is a metric, there always exists an optimal solution $\overline{T}$ such that no node in $\overline{I}$ is incident with less than 3 edges in $\overline{T}$. As observed by Rajagopalan and Vazirani in [1], the following local search algorithm returns a $\frac{3}{2}$-*optimal* Steiner tree, i.e. a Steiner tree $\tilde{T}$ with $w(\tilde{T}) \leq \frac{3}{2} w(\overline{T})$.

Algorithm 1 LOCAL_STEINER_TREE (G, R, w)

1. $I \leftarrow \emptyset$; $T \leftarrow$ any minimum spanning tree of $(G[R \cup I], w)$;
2. **while** $\exists x \in X \setminus I$ such that $MST(G[R \cup I \cup \{x\}], w) < w(T)$ **do**
3. $I \leftarrow I \cup \{x\}$; $T \leftarrow$ any minimum spanning tree of $(G[R \cup I], w)$;
4. remove from I all nodes with degree one in T; update T accordingly; (drop the corresponding leafs);
5. remove from I all nodes with degree two in T; update T accordingly; (shortcut the pairs of consecutive edges si and it with the single edges st);
6. **return** T;

We offer a direct and simple proof of the following result.

Theorem 3 *The Steiner tree output by Algorithm 1 is within a factor of* $\frac{3}{2}$ *from optimum.*

Proof: Let $\tilde{T}$ be the Steiner tree output by Algorithm 1 and let $\overline{T}$ be an optimal Steiner tree for (G, R, w). Let $\tilde{I} = V(\tilde{T}) \setminus R$ and $\overline{I} = V(\overline{T}) \setminus R$. In $\tilde{T}$, consider the stars of the nodes in $\tilde{I}$. Since (G, R) is quasi-bipartite, then these stars are all disjoint. Moreover, by steps 4 and 5, each star contains at least three edges. For every $x \in \tilde{I}$, let e_x be any edge of $\tilde{T}$ incident with x and with smallest possible weight. By the above remarks, $\sum_{x \in \tilde{I}} w(e_x) \leq \frac{1}{3} w(\tilde{T})$. Since for every $x \in \tilde{I}$ one of the two endpoints of e_x is in R, then there exists a spanning tree T of $G[R \cup \overline{I} \cup \tilde{I}]$ with $w(T) \leq w(\overline{T}) + \frac{1}{3} w(\tilde{T})$. (Take any spanning tree in $\overline{T} \cup \{e_x : x \in \tilde{I}\}$). By step 3, $MST(G[R \cup \tilde{I} \cup \{x\}], w) \geq w(\tilde{T})$ for every $x \in X \setminus \tilde{I}$. By Lemma 1,

$w(T) \geq w(\tilde{T})$. Combining, $w(\overline{T}) + \frac{1}{3}w(\tilde{T}) \geq w(\tilde{T})$. So, $w(\tilde{T}) \leq \frac{3}{2}w(\overline{T})$. □

4 Running time

In this section, we show how a bit-scaling technique can be employed to derive an implementation of Algorithm 1 with running time polynomial in the size of the input.

Consider the sequence of weightings $w = w_0, w_1, \ldots$, where, for $i > 0$, w_i is defined as follows: $w_i(e) = \lfloor \frac{w_{i-1}(e)}{2} \rfloor$. Let k be the smallest index for which $w_k(e) \leq 1$ for every edge e of G. Therefore $k \leq log_2(\max\{w(e) \ : e \in E\})$. When Algorithm 1 is executed on (G, R, w_k) as input, then loop 2–5 will cycle at most n times, since w_k is a $0,1$-vector. The output will be a tree ${T^k}_{APX}$. Note that ${T^k}_{APX}$ is a $\frac{3}{2}$-optimal Steiner tree for (G, R, w_k).

For $i = 0, 1, \ldots, k$, let ${T^i}_{OPT}$ be an optimal and ${T^i}_{APX}$ be a $\frac{3}{2}$-optimal Steiner tree in (G, w_i). Hence,

$$w_i({T^i}_{APX}) - w_i({T^{i-1}}_{OPT}) \leq w_i({T^i}_{APX}) - w_i({T^i}_{OPT}) \leq \frac{1}{2}w_i({T^i}_{OPT})$$

Moreover, since every tree has less that n edges, we have:

$$w_{i-1}({T^i}_{OPT}) - w_{i-1}({T^{i-1}}_{OPT}) \leq \Big(2w_i({T^i}_{OPT}) + n\Big) - 2w_i({T^{i-1}}_{OPT}) \leq n$$

Therefore,

$$\begin{aligned}
&w_{i-1}({T^i}_{APX}) - w_{i-1}({T^{i-1}}_{OPT}) \leq (2w_i({T^i}_{APX}) + n) - 2w_i({T^{i-1}}_{OPT}) \leq \\
&n + 2\left(w_i({T^i}_{APX}) - w_i({T^{i-1}}_{OPT})\right) \leq n + 2\left(\tfrac{1}{2}w_i({T^i}_{OPT})\right) \leq \\
&n + \tfrac{2}{2}\left(\tfrac{1}{2}w_{i-1}({T^i}_{OPT})\right) \leq \\
&n + \tfrac{1}{2}\Big(\ w_{i-1}({T^i}_{OPT}) + w_{i-1}({T^{i-1}}_{OPT}) - w_{i-1}({T^{i-1}}_{OPT})\ \Big) \leq \\
&n + \tfrac{1}{2}w_{i-1}(T^{i-1}_{OPT}) + \tfrac{1}{2}\Big(\ w_{i-1}({T^i}_{OPT}) - w_{i-1}({T^{i-1}}_{OPT})\ \Big) \leq \\
&n + \tfrac{1}{2}w_{i-1}({T^{i-1}}_{OPT}) + \tfrac{1}{2}\,(n) = \tfrac{3}{2}n + \tfrac{1}{2}w_{i-1}({T^{i-1}}_{OPT})
\end{aligned}$$

We conclude that $w_{i-1}({T^i}_{APX}) \leq \frac{3}{2}n + \frac{3}{2}w_{i-1}({T^{i-1}}_{OPT})$. Therefore, by executing loop 2–5 at most $\frac{3}{2}n$ times, then Algorithm 1 finds a $\frac{3}{2}$-optimal Steiner tree in (G, w_{i-1}) starting from any $\frac{3}{2}$-optimal Steiner tree in (G, w_i).

Acknowledgements

I thank Devdatt Dubashi for stimulating conversations.

References

[1] Sridhar Rajagopalan and Vijay V. Vazirani, On the Bidirected Cut Relaxation for the Metric Steiner Tree Problem. *SODA '99* (1999) 742–751.

Recent BRICS Report Series Publications

RS-99-39 **Romeo Rizzi.** ***On the Steiner Tree $\frac{3}{2}$-Approximation for Quasi-Bipartite Graphs*. November 1999. 6 pp.**

RS-99-38 **Romeo Rizzi.** ***Linear Time Recognition of P_4-Indifferent Graphs*. November 1999. 11 pp.**

RS-99-37 **Tibor Jordán. *Constrained Edge-Splitting Problems*. November 1999. 23 pp. A preliminary version with the title *Edge-Splitting Problems with Demands* appeared in Cornujols, Burkard and Woeginger, editors, *Integer Programming and Combinatorial Optimization: 7th International Conference*, IPCO '99 Proceedings, LNCS 1610, 1999, pages 273–288.**

RS-99-36 **Gian Luca Cattani and Glynn Winskel. *Presheaf Models for* CCS*-like Languages*. November 1999.**

RS-99-35 **Tibor Jordán and Zoltán Szigeti. *Detachments Preserving Local Edge-Connectivity of Graphs*. November 1999. 16 pp.**

RS-99-34 **Flemming Friche Rodler. *Wavelet Based 3D Compression for Very Large Volume Data Supporting Fast Random Access*. October 1999. 36 pp.**

RS-99-33 **Luca Aceto, Zoltán Ésik, and Anna Ingólfsdóttir. *The Max-Plus Algebra of the Natural Numbers has no Finite Equational Basis*. October 1999. 25 pp. To appear in *Theoretical Computer Science*.**

RS-99-32 **Luca Aceto and François Laroussinie. *Is your Model Checker on Time? — On the Complexity of Model Checking for Timed Modal Logics*. October 1999. 11 pp. Appears in Kutyłowski, Pacholski and Wierzbicki, editors, *Mathematical Foundations of Computer Science: 24th International Symposium*, MFCS '99 Proceedings, LNCS 1672, 1999, pages 125–136.**

RS-99-31 **Ulrich Kohlenbach. *Foundational and Mathematical Uses of Higher Types*. September 1999. 34 pp.**

RS-99-30 **Luca Aceto, Willem Jan Fokkink, and Chris Verhoef. *Structural Operational Semantics*. September 1999. 128 pp. To appear in Bergstra, Ponse and Smolka, editors, *Handbook of Process Algebra*, 1999.**

RS-99-29 **Søren Riis. *A Complexity Gap for Tree-Resolution*. September 1999. 33 pp.**