arXiv:1004.1437v1 [cs.DS] 8 Apr 2010

A Note on Johnson, Minkoff and Phillips' Algorithm for the Prize-Collecting Steiner Tree Problem

Paulo Feofiloff * Cristina G. Fernandes * Carlos E. Ferreira * José Coelho de Pina *

revised October 2009 †

Abstract

The primal-dual scheme has been used to provide approximation algorithms for many problems. Goemans and Williamson gave a $(2-\frac{1}{n-1})$-approximation for the Prize-Collecting Steiner Tree Problem that runs in $\mathrm{O}(n^3 \log n)$ time. Johnson, Minkoff and Phillips proposed a faster implementation of Goemans and Williamson's algorithm. We give a proof that the approximation ratio of this implementation is exactly 2.

1 Introduction

Consider a graph $G = (V, E)$, a function c from E into the set $\mathbb{Q}_{\geq}$ of non-negative rationals and a function π from V into $\mathbb{Q}_{\geq}$. The **Prize-Collecting Steiner Tree Problem** (PCST) asks for a tree T in G such that $\sum_{e \in E_T} c_e + \sum_{v \in V \setminus V_T} \pi_v$ is minimum. (We denote by V_T and E_T, respectively, the vertex and edge sets of a graph T.) The rooted variant of the problem requires T to contain a given root vertex.

Goemans and Williamson [2, 3] used a primal-dual scheme to derive a $(2-\frac{1}{n-1})$-approximation for the rooted variant of PCST, where $n := |V|$. By trying all possible choices for the root, they obtained a $(2-\frac{1}{n-1})$-approximation for the unrooted PCST. The resulting algorithm runs in time $\mathrm{O}(n^3 \log n)$. Johnson, Minkoff and Phillips [4] proposed a modification of the algorithm that runs the primal-dual scheme only once, resulting in a running-time of $\mathrm{O}(n^2 \log n)$. They claimed their algorithm — which we refer to as JMP — achieves an approximation ratio of $2-\frac{1}{n-1}$. Unfortunately, their claim does not hold.

This note does two things. First, it proves that the JMP algorithm is a 2-approximation (the proof involves some non-trivial technical details). Second, it shows an example where the approximation ratio achieved by the JMP algorithm is exactly 2, thereby contradicting the claim by Johnson, Minkoff and Phillips.

†This paper was originally published as `http://www.ime.usp.br/~cris/publ/jmp-analysis.ps.gz` in 2006. The present version makes explicit a stronger statement, implicit in the original version: that the addressed implementation is a Lagrangean preserving 2-approximation. It also introduces some cosmetic changes in notation and corrects a technical error in the proof of one of the invariants.

*Departamento de Ciência da Computação, Instituto de Matemática e Estatística, Universidade de São Paulo, Rua do Matão 1010, 05508-090 São Paulo/SP, Brazil. E-mail: `{pf,cris,cef,coelho}@ime.usp.br`. Research supported in part by PRONEX/CNPq 664107/1997-4 (Brazil).

2 Notation and preliminaries

For any subset F of E, let $c(F) := \sum_{e \in F} c_e$. For any subset X of V, let $\pi(X) := \sum_{v \in X} \pi_v$ and let $\overline{X} := V \setminus X$. If T is a subgraph of G, we shall abuse notation and write $\pi(T)$ and $\pi(\overline{T})$ to mean $\pi(V_T)$ and $\pi(\overline{V_T})$ respectively. Similarly, we shall write $c(T)$ to mean $c(E_T)$. Hence, the goal of $\mathrm{PCST}(G, c, \pi)$ is to find a tree T in G such that $c(T) + \pi(\overline{T})$ is minimum.

A collection $\mathcal{L}$ of nonnull subsets of V is **laminar** if, for any two elements L_1 and L_2 of $\mathcal{L}$, either $L_1 \cap L_2 = \emptyset$ or $L_1 \subseteq L_2$ or $L_1 \supseteq L_2$. For any subset X of V, let

$$\mathcal{L}[X] := \{L \in \mathcal{L} : L \subseteq X\} \quad \text{and} \quad \mathcal{L}_X := \{L \in \mathcal{L} : L \supseteq X\}\,.$$

For every L in $\mathcal{L}$ that is not in $\mathcal{L}[X] \cup \mathcal{L}[\overline{X}] \cup \mathcal{L}_X$, the sets $L \cap X$, $L \setminus X$ and $X \setminus L$ are all nonempty. For any subgraph T of G, we shall abuse notation and write $\mathcal{L}[T]$, $\mathcal{L}[\overline{T}]$, and $\mathcal{L}_T$ in place of $\mathcal{L}[V_T]$, $\mathcal{L}[\overline{V_T}]$, and $\mathcal{L}_{V_T}$ respectively.

The union of all sets in $\mathcal{L}$ shall be denoted by $\bigcup \mathcal{L}$. The set of all maximal elements of $\mathcal{L}$ shall be denoted by $\mathcal{L}^*$. If $\mathcal{L}$ is laminar, the elements of $\mathcal{L}^*$ are pairwise disjoint. If, in addition, $\bigcup \mathcal{L} = V$ then $\mathcal{L}^*$ is a partition of V.

For any laminar collection $\mathcal{L}$ of subsets of V and any edge e of G, let $\mathcal{L}(e) := \{L \in \mathcal{L} : e \in \delta_G L\}$, where $\delta_G L$ stands for the set of edges of G with one end in L and the other in $\overline{L}$.

Let y be a function from $\mathcal{L}$ into $\mathbb{Q}_{\geq}$. For any subcollection $\mathcal{L}'$ of $\mathcal{L}$, let $y(\mathcal{L}') := \sum_{L \in \mathcal{L}'} y_L$. We say that y **respects** c if

$$y(\mathcal{L}(e)) \;\leq\; c_e \quad \text{for each } e \text{ in } E\,. \tag{1}$$

We say an edge e is **tight for** y if equality holds in (1). We say y **respects** π if

$$y(\mathcal{L}[X]) \;\leq\; \pi(X) \quad \text{for each } X \text{ in } \mathcal{L}\,. \tag{2}$$

We shall say that y **saturates** an element X of $\mathcal{L}$ if equality holds in (2). The following lemma summarizes the effect of the two "respects" constraints on y:

Lemma 2.1 *Let $\mathcal{L}$ be a laminar collection of subsets of V and y a function from $\mathcal{L}$ into $\mathbb{Q}_{\geq}$. If y respects c and π then*

$$y(\mathcal{L} \setminus \mathcal{L}_T) \;\leq\; c(T) + \pi(\overline{T})$$

for any connected subgraph T of G.

Proof. For $\mathcal{M} := \{L \in \mathcal{L} : \delta_T L \neq \emptyset\}$, we have $y(\mathcal{M}) \;\leq\; \sum_{L \in \mathcal{M}} |\delta_T L| y_L \;=\; \sum_{e \in E_T} y(\mathcal{L}(e)) \;\leq\; \sum_{e \in E_T} c_e \;=\; c(T)$. For $\mathcal{N} := \mathcal{L}[\overline{T}]$, we have $y(\mathcal{N}) \;=\; \sum_{L \in \mathcal{N}^*} y(\mathcal{L}[L]) \;\leq\; \sum_{L \in \mathcal{N}^*} \pi(L) \;\leq\; \pi(\overline{T})$. The lemma follows from the two inequalities since $\mathcal{L} = \mathcal{M} \cup \mathcal{N} \cup \mathcal{L}_T$. ∎

Let $\mathrm{opt}(\mathrm{PCST}(G, c, \pi))$ denote the minimum value of the sum $c(T) + \pi(\overline{T})$ when T is a tree in G. Then the following corollary establishes the relevant lower bound for $\mathrm{opt}(\mathrm{PCST}(G, c, \pi))$:

Corollary 2.2 *Let $\mathcal{L}$ be a laminar collection of subsets of V and y a function from $\mathcal{L}$ into $\mathbb{Q}_{\geq}$. If y respects c and π then $y(\mathcal{L} \setminus \mathcal{L}_O) \;\leq\; \mathrm{opt}(\mathrm{PCST}(G, c, \pi))$ for any optimal solution O of $\mathrm{PCST}(G, c, \pi)$.* ∎

Before we state the algorithm, a few more definitions are needed. Let $\mathcal{L}$ be a laminar collection of subsets of V such that $\bigcup \mathcal{L} = V$. We say that an edge is **internal to** $\mathcal{L}^*$ if both of its ends are in the same element of $\mathcal{L}^*$. All other edges are **external to** $\mathcal{L}^*$. For any external edge, there are two elements of $\mathcal{L}^*$ containing its ends. We call these two elements the **extremes** of the edge in $\mathcal{L}^*$.

Given a forest F in G and a subset L of V, we say that F is L**-connected** if $V_F \cap L = \emptyset$ or the induced subgraph $F[V_F \cap L]$ is connected. In other words, F is L-connected if the following property holds: for any two vertices x and y of F in L, there exists a path from x to y in F and that path never leaves L. If F spans G (as is the case during the first phase of the algorithm below), the condition "$F[V_F \cap L]$ is connected" can, of course, be replaced by "$F[L]$ is connected".

For any collection $\mathcal{L}$ of subsets of V, we shall say that F is $\mathcal{L}$**-connected** if F is L-connected for each L in $\mathcal{L}$.

For any collection $\mathcal{S}$ of subsets of V, we say a tree T **has no bridge in** $\mathcal{S}$ if $|\delta_T S| \neq 1$ (whence $\delta_T S = \emptyset$ or $|\delta_T S| \geq 2$) for all S in $\mathcal{S}$. We say that a tree T in G **is wrapped in** $\mathcal{S}$ if $V_T \subseteq S$ for some S in $\mathcal{S}$.

3 Johnson, Minkoff and Phillips' algorithm

The JMP algorithm is a 2-approximation for the PCST. It receives G, c, π and returns a tree T in G such that $c(T) + 2\,\pi(\overline{T}) \leq 2\,\mathrm{opt}(\mathrm{PCST}(G, c, \pi))$. (For our purposes, it would be enough to have $c(T) + \pi(\overline{T})$ on the left side of the inequality. The factor 2 multiplying π is a bonus, and, because of it, the JMP algorithm is said to be a *Lagrangean preserving* 2-approximation [1].)

The algorithm has two phases, the second one operating on the output of the first.

Phase I: Each iteration in phase I starts with a spanning forest F in G, a laminar collection $\mathcal{L}$ of subsets of V such that $\bigcup \mathcal{L} = V$, a subcollection $\mathcal{S}$ of $\mathcal{L}$, and a function y from $\mathcal{L}$ into $\mathbb{Q}_{\geq}$ such that the following invariants hold:

(i1) F is $\mathcal{L}$-connected;

(i2) y respects c and π;

(i3) each edge of F is tight for y;

(i4) y saturates every element of $\mathcal{S}$;

(i5) no element of $\mathcal{L}^* \setminus \mathcal{S}$ is the union of elements of $\mathcal{S}$;

(i6) for any $\mathcal{L}$-connected tree T in G, if T has no bridge in $\mathcal{S}$ and is not wrapped in $\mathcal{S}$ then

$$\sum_{e \in E_T} y(\mathcal{L}(e)) + 2\,y(\mathcal{L}[\overline{T}]) \;\leq\; 2\,y(\mathcal{L} \setminus \mathcal{L}_{\{o\}}) \tag{3}$$

for any vertex o of G.

The first iteration starts with $F = (V, \emptyset)$, $\mathcal{L} = \{\{v\} : v \in V\}$, $\mathcal{S} = \emptyset$, and $y = 0$. Each iteration consists of the following:

Case I.1: $|\mathcal{L}^* \setminus \mathcal{S}| > 1$.

For ε in $\mathbb{Q}_{\geq}$, let y^ε be the function defined as follows: $y^\varepsilon_L = y_L + \varepsilon$ if $L \in \mathcal{L}^* \setminus \mathcal{S}$ and $y^\varepsilon_L = y_L$ otherwise. Let ε be the largest number in $\mathbb{Q}_{\geq}$ such that the function y^ε respects c and π.

Subcase I.1.A: y^ε saturates some element L of $\mathcal{L}^* \setminus \mathcal{S}$.

Start a new iteration with $\mathcal{S} \cup \{L\}$ and y^ε in the roles of $\mathcal{S}$ and y respectively. (The forest F and the collection $\mathcal{L}$ do not change.)

Subcase I.1.B: some edge e external to $\mathcal{L}^*$ is tight for y^ε and has at least one of its extremes in $\mathcal{L}^* \setminus \mathcal{S}$.

Let L_1 and L_2 be the extremes of e in $\mathcal{L}^*$. Set $y^\varepsilon_{L_1 \cup L_2} := 0$ and start a new iteration with $F + e$, $\mathcal{L} \cup \{L_1 \cup L_2\}$, and y^ε in the roles of F, $\mathcal{L}$, and y respectively. (The collection $\mathcal{S}$ does not change.)

Case I.2: $|\mathcal{L}^* \setminus \mathcal{S}| = 1$.

This is the end of phase I. Start phase II.

Phase II: During this phase, the collections $\mathcal{L}$ and $\mathcal{S}$ and the function y remain unchanged. Let M be the only element of $\mathcal{L}^* \setminus \mathcal{S}$. Each iteration begins with a subgraph T of F such that

(i7) T is an $\mathcal{L}$-connected tree;

(i8) $M \setminus V_T$ admits a partition into elements of $\mathcal{S}$.

The first iteration begins with $T = F[M]$. Each iteration does the following:

Case II.1: $|\delta_T Z| = 1$ for some Z in $\mathcal{S}$.

Start a new iteration with $T - Z$ in place of T.

Case II.2: $|\delta_T Z| \neq 1$ for each Z in $\mathcal{S}$.

Return T and stop.

4 Analysis of the algorithm

Suppose, for the moment, that invariants (i1) to (i8) are correct. At the end of phase II, T is a tree by virtue of (i7). As T is a subgraph of F, due to (i3),

$$c(T) = \sum_{e \in E_T} c_e = \sum_{e \in E_T} y(\mathcal{L}(e)) \,.$$

On the other hand, $\mathcal{L}^* \cap \mathcal{S}$ is a partition of $\overline{M}$ and, by (i8), there is a partition of $M \setminus V_T$ into elements of $\mathcal{S}$. Therefore, some subcollection $\mathcal{Z}$ of $\mathcal{S}$ is a partition of $\overline{V_T}$. Hence,

$$\pi(\overline{T}) = \sum_{S \in \mathcal{Z}} \pi(S) = \sum_{S \in \mathcal{Z}} y(\mathcal{L}[S]) \leq y(\mathcal{L}[\overline{T}]) \,.$$

Here, the second equality follows from (i4). Therefore,

$$c(T) + 2\pi(\overline{T}) \leq \sum_{e \in E_T} y(\mathcal{L}(e)) + 2\, y(\mathcal{L}[\overline{T}]) \,. \tag{4}$$

In order to show that (3) holds, we must verify that T satisfies the hypotheses of (i6). By (i7), T is $\mathcal{L}$-connected. Due to (i5), M is not the union of elements of $\mathcal{S}$. Hence, by virtue (i8), T is not

wrapped in $\mathcal{S}$. Since we are in Case II.2, T has no bridge in $\mathcal{S}$. Hence, T satisfies the hypotheses of (i6). Now, by (3) coupled with (4),

$$c(T) + 2\,\pi(\overline{T}) \leq 2\,y(\mathcal{L} \setminus \mathcal{L}_{\{o\}}) \tag{5}$$

for any vertex o. Now, let o be an arbitrary vertex of an optimal solution O of $\mathrm{PCST}(G, c, \pi)$. Since y respects c and π, as stated in (i2), Corollary 2.2 implies

$$c(T) + 2\,\pi(\overline{T}) \;\leq\; 2\,y(\mathcal{L} \setminus \mathcal{L}_{\{o\}}) \;\leq\; 2\,y(\mathcal{L} \setminus \mathcal{L}_O) \;\leq\; 2\,\mathrm{opt}(\mathrm{PCST}(G, c, \pi))\,.$$

This proves the following theorem (which is the correct version of Theorem 3.2 by Johnson, Minkoff and Phillips [4]):

Theorem 4.1 *The* JMP *algorithm is a* 2*-approximation for the* PCST*.*

To complete the proof of the theorem we must only verify the invariants of the algorithm, something we shall do in the next section.

The example in Figure 1 shows that the approximation ratio of the JMP algorithm can be arbitrarily close to 2, regardless of the size of the graph. So, Theorem 4.1 is tight.

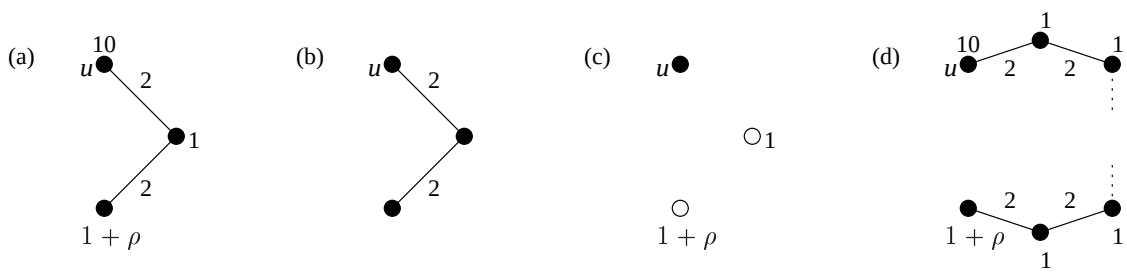

Figure 1: (a) An instance of the PCST. (b) The solution produced by the JMP algorithm when $\rho > 0$. Its cost is 4. (c) The optimal solution, consisting of vertex u alone, has cost $2 + \rho$. (d) A similar instance of arbitrary size consists of a long path.

5 Proofs of the invariants

Invariants (i1) to (i4) obviously hold at the beginning of each iteration of phase I. We must only verify the other four invariants.

Proof of (i5). Obviously (i5) holds at the beginning of the first iteration. Now consider an iteration where Case I.1 occurs. If Subcase I.1.A occurs, then (i5) remains trivially true at the beginning of the next iteration. Next, suppose Subcase I.1.B occurs. Adjust notation so that $L_1 \notin \mathcal{S}$. Since (i5) holds at the beginning of the current iteration, L_1 is not the union of elements of $\mathcal{S}$. Hence, $L_1 \cup L_2$ is not the union of elements of $\mathcal{S}$. Therefore, (i5) remains trivially true at the beginning of the next iteration. ∎

The verification of (i6) depends on the following lemma:

Lemma 5.1 *Let $\mathcal{P}$ be a partition of V and $(\mathcal{A},\mathcal{B})$ a bipartition of $\mathcal{P}$. Let T be a tree in G. If T is $\mathcal{P}$-connected, has no bridge in $\mathcal{B}$, and is not wrapped in $\mathcal{B}$, then*

$$\frac{1}{2}\sum_{A\in\mathcal{A}} |\delta_T A| + |\mathcal{A}[\overline{T}]| \;\le\; |\mathcal{A}| - 1\,. \tag{6}$$

Proof. Let us say that two elements of $\mathcal{P}$ are *adjacent* if there is an edge of T with these two elements as extremes. This adjacency relation defines a graph $\mathcal{H}$ having $\mathcal{P}$ as set of vertices. Since T is $\mathcal{P}$-connected, the edges of $\mathcal{H}$ are in one-to-one correspondence with the edges of T external to $\mathcal{P}$. Hence, the degree of any vertex P of $\mathcal{H}$ is exactly $|\delta_T P|$, and therefore $\frac{1}{2}\sum_{P\in\mathcal{P}} |\delta_T P| = |E_{\mathcal{H}}|$. Since T is connected, $\mathcal{H}$ has $1+|\mathcal{P}[\overline{T}]|$ components (all are singletons, except at most one). Since T has no cycles and is $\mathcal{P}$-connected, $\mathcal{H}$ is a forest. Hence $|E_{\mathcal{H}}| = |\mathcal{P}| - 1 - |\mathcal{P}[\overline{T}]|$ and therefore

$$\frac{1}{2}\sum_{P\in\mathcal{P}} |\delta_T P| = |\mathcal{P}| - 1 - |\mathcal{P}[\overline{T}]|\,. \tag{7}$$

Now consider the vertices of $\mathcal{H}$ that are in $\mathcal{B}$. Since T has no bridge in $\mathcal{B}$ and is not wrapped in $\mathcal{B}$, each B in $\mathcal{B}$ is such that either $|\delta_T B| \ge 2$ or $B \subseteq \overline{V_T}$. Hence $\sum_{B\in\mathcal{B}} |\delta_T B| \ge 2\,|\mathcal{B}\setminus\mathcal{B}[\overline{T}]|$, and therefore

$$\frac{1}{2}\sum_{B\in\mathcal{B}} |\delta_T B| \;\ge\; |\mathcal{B}| - |\mathcal{B}[\overline{T}]|\,. \tag{8}$$

The difference between (7) and (8) is the claimed inequality (6). ∎

Proof of (i6). It is clear that (i6) holds at the beginning of the first iteration. Now assume that it holds at the beginning of some iteration where Case I.1 occurs.

Suppose, first, that Subcase I.1.A occurs. At the end of the subcase, let $\mathcal{S}' := \mathcal{S}\cup\{L\}$, let o be any vertex, and let T be an $\mathcal{L}$-connected tree that has no bridge in $\mathcal{S}'$, is not wrapped in $\mathcal{S}'$, and such that all its edges are tight for y^ε. Of course all edges of T are tight for y. Since T has no bridge in $\mathcal{S}$ and is not wrapped in $\mathcal{S}$, (3) holds. We must show that (3) also holds when y^ε is substituted for y. Let $\mathcal{P} := \mathcal{L}^*$, $\mathcal{A} := \mathcal{L}^*\setminus\mathcal{S}$, and $\mathcal{B} := \mathcal{L}^*\cap\mathcal{S}$. Since $|\mathcal{A}_{\{o\}}| \le 1$, Lemma 5.1 implies

$$\sum_{A\in\mathcal{A}} |\delta_T A|\,\varepsilon + 2\,|\mathcal{A}[\overline{T}]|\,\varepsilon \;\le\; 2\,|\mathcal{A}\setminus\mathcal{A}_{\{o\}}|\,\varepsilon\,.$$

The addition of this inequality to (3) produces

$$\sum_{e\in E_T} y^\varepsilon(\mathcal{L}(e)) + 2\,y^\varepsilon(\mathcal{L}[\overline{T}]) \;\le\; 2\,y^\varepsilon(\mathcal{L}\setminus\mathcal{L}_{\{o\}})\,,$$

since y^ε differs from y only in $\mathcal{A}$. Hence, (i6) remains true at the beginning of the next iteration.

Now suppose Subcase I.1.B occurs. At the end of the subcase, let $\mathcal{L}' := \mathcal{L}\cup\{L_1\cup L_2\}$, let o be any vertex, and let T be an $\mathcal{L}'$-connected tree that has no bridge in $\mathcal{S}$ and is not wrapped in $\mathcal{S}$. Since T is $\mathcal{L}$-connected, (3) holds. We must show that (3) remains true when y^ε and $\mathcal{L}'$ are substituted for y and $\mathcal{L}$ respectively. Let $\mathcal{P} := \mathcal{L}^*$, $\mathcal{A} := \mathcal{L}^*\setminus\mathcal{S}$, and $\mathcal{B} := \mathcal{L}^*\cap\mathcal{S}$. Since $|\mathcal{A}_{\{o\}}| \le 1$, Lemma 5.1 implies $\sum_{A\in\mathcal{A}} |\delta_T A|\,\varepsilon + 2\,|\mathcal{A}[\overline{T}]|\,\varepsilon \le 2\,|\mathcal{A}\setminus\mathcal{A}_{\{o\}}|\,\varepsilon$, as in the previous case. The addition of this inequality to (3) produces

$$\sum_{e\in E_T} y^\varepsilon(\mathcal{L}'(e)) + 2\,y^\varepsilon(\mathcal{L}'[\overline{T}]) \;\le\; 2\,y^\varepsilon(\mathcal{L}'\setminus\mathcal{L}'_{\{o\}})\,,$$

since $y^{\varepsilon}_{L_1 \cup L_2} = 0$ and y^{ε} differs from y only in $\mathcal{A}$. Hence, (i6) remains true at the beginning of the next iteration. ∎

Proof of (i7). Suppose we are at the beginning of the first iteration of phase II. Let L be an element of $\mathcal{L}$ such that $L \cap V_T \neq \emptyset$. Since $V_T = M \in \mathcal{L}^*$, we have $L \subseteq V_T$ and therefore $T[V_T \cap L] = T[L] = F[L]$. Since $F[L]$ is connected by virtue of (i1), so is $T[V_T \cap L]$. This argument shows that T is $\mathcal{L}$-connected. In particular, T is M-connected and therefore T is a tree. Hence, (i7) holds at the beginning of the first iteration.

Now suppose (i7) holds at the beginning of some iteration where Case II.1 occurs. Let L be an element of $\mathcal{L}$ and let u and v be vertices in $L \cap (V_T \setminus Z)$. Let P be the unique path from u to v in T. We may assume that P never leaves L. Moreover, P never enters Z, given that $|\delta_T Z| = 1$. Hence, $T - Z$ is L-connected. For the same reason, $T - Z$ is a tree. Hence (i7) holds at the beginning of the next iteration. ∎

Proof of (i8). At the beginning of the first iteration of phase II, (i8) holds because $V_T = M$. Now consider an iteration where Case II.1 occurs. We may assume that there is a partition $\mathcal{U}$ of $M \setminus V_T$ into elements of $\mathcal{S}$. If $Z \subseteq V_T$ then $\mathcal{U} \cup \{Z\}$ is a partition of $M \setminus (V_T \setminus Z)$ into elements of $\mathcal{S}$. Otherwise, Z includes some of the elements of $\mathcal{U}$ and is disjoint from all the others. Hence, $\{Z\} \cup \{U \in \mathcal{U} : U \cap Z = \emptyset\}$ is a partition of $M \setminus (V_T \setminus Z)$ into elements of $\mathcal{S}$. This shows that (i8) holds at the beginning of the next iteration. ∎

References

[1] A. Archer, M. Bateni, M. Hajiaghayi, and H. Karloff. Improved approximation algorithms for prize-collecting Steiner tree and TSP. In *50th Annual Symposium on Foundations of Computer Science*, 2009.

[2] M.X. Goemans and D.P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[3] D.S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems.* PWS Publishing, 1997.

[4] D.S. Johnson, M. Minkoff, and S. Phillips. The prize collecting Steiner tree problem: theory and practice. In *Symposium on Discrete Algorithms*, pages 760–769, 2000.