Approximation Algorithms for Node-Weighted Prize-Collecting Steiner Tree Problems on Planar Graphs*

Jarosław Byrka[1], Mateusz Lewandowski[2], and Carsten Moldenhauer[3]

1 **University of Wrocław, Wrocław, Poland**
2 **University of Wrocław, Wrocław, Poland**
3 **EPFL, Lausanne, Switzerland**

Abstract

We study the prize-collecting version of the node-weighted Steiner tree problem (NWPCST) restricted to planar graphs. We give a new primal-dual Lagrangian-multiplier-preserving (LMP) 3-approximation algorithm for planar NWPCST. We then show a 2.88-approximation which establishes a new best approximation guarantee for planar NWPCST. This is done by combining our LMP algorithm with a threshold rounding technique and utilizing the 2.4-approximation of Berman and Yaroslavtsev [6] for the version without penalties. We also give a primal-dual 4-approximation algorithm for the more general forest version using techniques introduced by Hajiaghay and Jain [17].

1998 ACM Subject Classification G.2.2 [Graph Theory] Graph algorithms

Keywords and phrases approximation algorithms, Node-Weighted Steiner Tree, primal-dual algorithm, LMP, planar graphs

Digital Object Identifier 10.4230/LIPIcs.SWAT.2016.2

1 Introduction

In Steiner problems we aim at connecting certain specified vertices (called terminals) by buying edges or nodes of the given graph. The classic edge-weighted setting is well known to have many applications in areas like electronic circuits, computer networking, and telecommunication. The expressive power of the node weighted variants is used to model various settings common to bioinformatics [11], maintenance of electric power networks [16], and computational sustainability [10].

The node weighted setting is a generalization of the edge weighted case. In particular, one may cast the Set Cover problem as an instance of the Node-weighted Steiner Tree problem, which proves hardness of approximation of the general node-weighted setting. In this paper we study a natural special case, namely planar graphs, for which constant factor approximation algorithms are possible.

In the prize-collecting (penalty-avoiding) setting we are given an option not to satisfy a certain connectivity requirement, but to pay a fixed penalty instead. The main focus of this work is to develop efficient primal-dual approximation algorithms for prize-collecting versions of the node-weighted Steiner problems.

* Supported by NCN 2015/18/E/ST6/00456 grant.

15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2016).
Editor: Rasmus Pagh; Article No. 2; pp. 2:1–2:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Summary of best known approximation ratios for Steiner problems. Results of this paper are highlighted.

		edge-weighted		node-weighted	
		tree	**forest**	**tree**	**forest**
	general	1.39 [7]	2 [15]	O(log k) [5]	O(log k) [5]
	planar	PTAS [4]	PTAS [4]	2.4 [6]	2.4 [6]
prize-collecting	**general**	$2-\epsilon$ [2]	3 2.54 (LP) [17]	O(log k) [5, 19]	O(log k) [5]
	planar	PTAS [3]	APX-HARD [3]	**3** **2.88 (LP)**	**4**

1.1 Previous work

The Steiner tree problem is NP-hard even in planar graphs [12]. The most studied version is the standard edge-weighted Steiner tree, for which the best known approximation ratio 1.39 is obtained via a randomized iterative rounding technique [7]. By contrast, the best approximation algorithms for Steiner forest have the so far unbreakable ratio of 2 [1, 18].

For the prize-collecting Steiner tree problem there exists a primal-dual 2-approximation algorithm [15]. It can be shown that it is also Lagrangian-preserving, *i.e.*, that it achieves a 1-approximation on the penalty term. This property was used by Archer et al. to design the currently best $2-\epsilon$ approximation algorithm for PCST [2].

For the prize-collecting Steiner forest problem there is a 3 approximation primal-dual algorithm [17], which introduces a general technique to handle prize-collecting problems. In the same paper the authors use a threshold rounding technique with randomized analysis to obtain a ≈ 2.54 approximation.

There are optimal (up to a constant factor) algorithms for node-weighted Steiner problems. One example is the recent $O(\ln n)$ approximation algorithm for NWPCSF by Bateni et al [5]. Könemann et al [19] gave a Lagrangian-multiplier-preserving (LMP) approximation that achieves the same guarantee. Establishing the LMP property is of crucial importance for the construction of approximation algorithms for quota and budgeted versions of the NWST problem.

Planarity helps significantly in both the edge and node weighted setting. Both ST and SF admit PTAS in planar graphs [4]. Planar PCST can be also approximated with any constant, but PCSF is APX-HARD already on planar graphs [3].

In the case of the node-weighted setting, planarity helps to achieve constant factor approximations. The $NWSF$ can be expressed as the Hitting Set problem for some uncrossing family of cycles and hence solved as a feedback problem. This was exploited by Berman and Yaroslavtsev in [6] where they obtained a 2.4 approximation for $NWSF$ and other problems on planar graphs.

In [21] it was observed that using a threshold rounding technique together with the 2.4-approximation of Berman and Yaroslavtsev [6] for the version without penalties gives a 2.93-approximation algorithm for NWPCST on planar graphs. This was the best approximation guarantee up to date. However, such an algorithm requires solving an LP.

We summarize the current best known results in Table 1.

1.2 Our contribution

We propose a new LMP 3-approximation algorithm for NWPCST on planar graphs. The algorithm is an adaptation of the original technique developed by Goemans and Williamson in [15] for PCST to the node-weighted version. However, we change the pruning phase of the algorithm. This enables us to analyze the connection and penalty costs separately which is the key ingredient. In particular, we can directly charge the penalty costs to a part of the dual solution yielding Langrangian-multiplier-preservation. Further, the connection costs can be bounded using a slightly adapted analysis from [20] for NWSF. The approximation ratio of 3 is slightly higher than the previously best approximation ratio but the primal-dual algorithm does not require solving an LP.

Next, we establish a new best approximation ratio by exploiting the asymmetry of our primal-dual algorithm. A combination of the new LMP algorithm with a threshold rounding technique with the underlying 2.4-approximation from [6] yields a 2.88-approximation for NWPCST on planar graphs.

Furthermore, we obtain an efficient, direct primal-dual 4-approximation algorithm for NWPCSF on planar graphs building up on ideas for edge-weighted PCSF from [17]. This approach was previously indicated by Demaine et al. [9], but we give a better constant.

2 The LMP primal-dual 3-approximation algorithm

Consider an undirected graph $G = (V, E)$ with non-negative cost function and penalties on the vertices denoted by $w : V \to Q_+$ and $\pi : V \to Q_+$, respectively. In the NWPCST problem we are allowed to purchase a *connected* subgraph F of G that connects vertices to a prespecified root $r \in V$. Every bought vertex induces a cost according to w. Every vertex that is not included induces a penalty according to π. The objective is to minimize the sum of the purchase and penalty costs, *i.e.*, $\sum_{v\in F} w_v + \sum_{v\notin F} \pi_v$.

By a standard transformation we can assume that for every vertex v either its cost or its penalty is zero. To see this consider a single vertex v with both strictly positive cost and penalty. Add an additional vertex v', set its cost to zero and penalty to π_v, add an edge from v' to v and set the penalty of v to zero. Now, any solution in the original graph can be transformed to a solution of the same cost in the modified graph and vice-versa.

In the sequel, we call a vertex with a positive penalty a *terminal*. Terminals and the root can be purchased for free. Other vertices do not have a penalty and we call them non-terminals or Steiner vertices.

Let $\Gamma(S)$ denote the set of neighbors of S, *i.e.*, the set of vertices in $V \setminus S$ incident to vertices from $S \subseteq V$. Let also $\Pi(X) = \sum_{v\in X} \pi_v$. Thus, NWPCST is the following problem:

$$\min \sum_{v\in V} w_v x_v + \sum_{X\subseteq V\setminus\{r\}} \Pi(X) z_X \qquad (IP_{PCST})$$

$$s.t.$$

$$\sum_{v\in\Gamma(S)} x_v + \sum_{X:S\subseteq X} z_X \geq 1 \qquad \forall S \subseteq V \setminus \{r\}$$

$$x_v \in \{0,1\} \qquad \forall v \in V$$

$$z_X \in \{0,1\} \qquad \forall X \subseteq V \setminus \{r\}$$

By relaxing the integrality constraints to non-negativity constraints we obtain the standard linear relaxation. The dual of this relaxation is

$$\max \sum_{S\subseteq V\setminus\{r\}} y_S \qquad (DLP_{PCST})$$

$$s.t.$$

$$\sum_{S: v\in\Gamma(S)} y_S \le w_v \qquad \forall v\in V \qquad (1)$$

$$\sum_{S\subseteq X} y_S \le \Pi(X) \qquad \forall X\subseteq V\setminus\{r\} \qquad (2)$$

$$y_S \ge 0 \qquad \forall S\subseteq V\setminus\{r\}$$

2.1 Algorithm

Now we shortly describe our primal-dual algorithm which is an adaptation of the generic moat-growing approach of Goemans and Williamson [15]. In each iteration i we maintain a set of already bought nodes F. We say that some vertex was bought at time i if it was bought in iteration i[1]. At the beginning F contains all terminals (including the root). We maintain also the set of connected components C of subgraph $G[F]$ induced by the vertices bought so far. We call each of these connected components a moat. Moats can be active or inactive. The moat containing root r is always inactive. In each iteration we increase (grow) dual variables corresponding to *all active* moats uniformly until one of the following two events happen:

- a vertex v goes tight (constraint (1) becomes equality), or
- a set X goes tight (constraint (2) becomes equality).

In the first case we buy vertex v and possibly merge moats incident to v. If we merge to a moat containing the root r, this moat becomes inactive, otherwise it is declared active.

In the second case, we declare the moat corresponding to set X inactive. Moreover, we mark all unmarked terminals inside X with the current time.

The growth phase terminates when there are no more active moats. After that, we have a pruning phase. In the pruning phase we restrict to the connected component of F containing the root and discard everything else. Let $F^{(r)}$ denote this component. Then, we consider vertices in $F^{(r)}$ in the reverse order of purchase. We delete vertex v (bought at time t) if it does not disconnect from r any terminal which was unmarked at time t. When we delete v, we further discard all vertices that become disconnected from r. As a result we output the set of bought vertices F' that survived pruning.

Our algorithm can be implemented with a notion of so-called potentials. Let $P(X) = \Pi(X) - \sum_{S\subseteq X} y_S$ be the potential of set X. Intuitively, we pay for the growth of moats (increase of dual variables) with potentials of these moats. If the potential of a moat goes to zero, the corresponding constraint becomes tight, so we have to make this moat inactive. When we merge moats to a new moat S by buying a vertex, we compute the potential of S by summing the potentials of the old moats.

[1] When we refer to time we always have in mind the number of the current iteration. Note that it implies that the speed of the uniform growth of dual budgets is not constant across iterations, but it does not affect our description of the algorithm.

2.2 Analysis

▶ **Theorem 1** (Lagrangian Multiplier Preservation). *Let G be planar. The algorithm described in the previous section outputs a set of vertices F' such that*

$$\sum_{v\in F'} w_v + 3\Pi(V\setminus F') \le 3 \sum_{S\subseteq V\setminus\{r\}} y_S \le 3\ OPT\,.$$

In the proof we want to use the obtained dual solution y to account for the connection costs and penalties of the primal solution F'. We will partition the y_S into two sets. The first set will yield a bound on the connection costs and the second a bound on the penalties.

The key ingredient in the analysis is the partition that is based on the following lemma. Consider any iteration i and the active moats A_i before this iteration. Let $S \in A_i$ be an active moat that was not included in the final solution, *i.e.*, $S \cap F' = \emptyset$. Then, the dual variable of S did not contribute to buying any vertex in F'. This means that y_S does not contribute to the left-hand-side of the constraints (1) for any $v \in F'$. More formally, this means that S does not have a neighbor in F'.

▶ **Lemma 2.** *Let $S \in A_i$ be such that $S \subseteq V \setminus F'$. Then, the moat S does not have any neighbor in the solution, i.e. $F' \cap \Gamma(S) = \emptyset$.*

Proof of Lemma 2. Note that $S \in A_i$ means that S is active in iteration i and therefore there is an unmarked (before time i) terminal in S. Now, assume for a contradiction that $F' \cap \Gamma(S) \neq \emptyset$ and let $U \subseteq S$ be the set of vertices having a neighbor in F'. Note that all vertices in U were bought before iteration i because S is a connected component of the vertices bought before iteration i and $U \subseteq S$. Since S is not part of F', all the vertices in U must have been deleted in the pruning phase. A contradiction, since this would disconnect the unmarked (before time i) terminal in S. ◀

Following Lemma 2, we can partition all dual variables into the variables that contributed to buying the vertices of F' and the dual variables that account for the penalties induced by F'. Let CC be the set of all moats $S \subseteq V \setminus \{r\}$ that include a vertex of F' or have a neighbor in F', *i.e.*, $(S \cup \Gamma(S)) \cap F' \neq \emptyset$ and $y_S > 0$. Let PC be the set of all other moats, *i.e.*, sets S with $y_S > 0$ but $S \notin CC$. We will show that

$$\sum_{v\in F'} w_v \le 3 \sum_{S\in CC} y_S \qquad \text{and} \qquad \Pi(V\setminus F') = \sum_{S\in PC} y_S$$

which yields Theorem 1.

To show the bound on the connection cost we make a degree counting argument. Here, we can leverage the analysis of the primal-dual algorithm for node-weighted Steiner forest given in [20]. Recall that our algorithm can also deactivate moats due to the penalty constraints. However, this fact does not generate problems. Intuitively, deactivating a moat corresponds to satisfying a demand pair in the forest problem. The proof of the following lemma only requires a minor change to the analysis.

▶ **Lemma 3** (Analog of Analysis in [20]). *Let F' be the output of the algorithm and A_i be the set of active moats before running iteration i. Then,*

$$\sum_{S\in A_i\cap CC} |F' \cap \Gamma(S)| \le 3|A_i \cap CC|.$$

Proof. We outline the proof of Lemma 3. As indicated this proof is, except for a minor change, analogous to the proof used in [20] to show that the generic primal-dual algorithm for node-weighted Steiner forest on planar graphs has an approximation guarantee of 3.

Let F' be the output of the algorithm and A_i be the set of active moats before running iteration i. We want to show that

$$\sum_{S \in A_i \cap CC} |F' \cap \Gamma(S)| \leq 3|A_i \cap CC|. \tag{3}$$

In (3) we count the adjacencies between active moats at iteration i and vertices from F'. Let F_i be the set of vertices bought by the algorithm before iteration i. Consider a graph G' obtained from G in the following way:

1. take the subgraph of G induced by vertices from $F_i \cup F'$
2. keep only the connected component containing root r
3. contract each inactive moat (at iteration i) in this subgraph with a neighboring vertex (excluding the moat containing root)
4. contract each active moat in this component
5. contract the moat containing the root

Next, color the vertices of G' with three colors:

- white color for vertices obtained from contracting active moats
- blue color for the single vertex representing the moat containing the root
- black color for all other vertices, i.e. $F' \setminus F_i$

Observe now that deleting a black vertex in G' disconnects some white vertex from the blue vertex, because otherwise it would be deleted in the pruning phase. G' remains planar, since deletions and contractions preserve planarity. Moreover, it is easy to see that the number of adjacencies $\sum_{S \in A_i} |F' \cap \Gamma(S)|$ in G is the same as the number of edges between white and black vertices in G'.

To bound this number we will use the following result that is implicit in [20].

▶ **Lemma 4.** *Consider a simple connected planar graph $H = (V, E)$ in which vertices are colored with two colors: black and white, i.e. $V = B \cup W$. If for this graph the two following conditions hold*

- *there is no edge between any two white vertices*
- *removing any black vertex disconnects the graph*

then the number of edges between black and white vertices ($|E'|$) is at most 3 times greater than the number of white vertices, i.e., $|E'| \leq 3(|W| - 1)$

Before we prove Lemma 4, let us remark how it yields the claim. Consider for a moment the color of the blue vertex in G' to be white (resulting in graph H). Now removing a black vertex clearly splits the graph into multiple components, since it disconnects at least two white vertices (one of them is this recolored blue vertex). All other conditions of the lemma are satisfied. Applying Lemma 4 finishes the proof of Lemma 3, since $|A_i| = |W| - 1$. ◀

Proof of the Lemma 4. We follow the proof given in [21]. Consider the following operation on the graph H. Take any edge $e = (u, v)$ between two black vertices u and v in H.

- If u and v share a common white neighbor, then delete edge e.
- Otherwise contract u and v.

Observe that this operation preserves conditions of the lemma. Moreover it does not change the number of adjacencies between black and white vertices. Consider now the graph H' obtained by performing as many above operations as possible. The H' is bipartite since we contracted or deleted all edges between any two black vertices. The goal is now to bound the number of edges in H'. The idea is to use the Euler's formula for planar graphs. But first we have to show a few claims about H'.

Let W and B denote the set of white and black vertices of H', respectively.

▶ **Fact 5.** $|B| \leq |W| - 1$.

Proof. Consider a breadth-first search tree T in H' rooted at any white vertex r_w. Since removing a black vertex splits the graph, all leaves of T are white. Recall that H' is bipartite. Thus each black vertex has at least one unique white child in T. Furthermore, r_w is the only white vertex that does not have a parent. This concludes the proof of Fact 5. ◀

Now, using Fact 5 instead of Claim 1.4 of [21] in the proof of Lemma 1.3 of [21] yields Lemma 4 ◀

To conclude the upper bound on the connection costs, note that constraint (1) is tight for all vertices $v \in F'$. This gives

$$\sum_{v \in F'} w_v = \sum_{v \in F'} \sum_{S : v \in \Gamma(S)} y_S = \sum_{S \subseteq V \setminus \{r\}} |F' \cap \Gamma(S)| \; y_S = \sum_{S \in CC} |F' \cap \Gamma(S)| \; y_S.$$

We will show that $\sum_{S \in CC} |F' \cap \Gamma(S)| \; y_S \leq 3 \; \sum_{S \in CC} y_S$ by induction on the number of iterations. At the beginning all dual variables are equal to 0 and the inequality holds. In iteration i we grow each active moat from $A_i \cap CC$ by ϵ_i. This increases the left-hand side by $\epsilon_i \sum_{S \in A_i \cap CC} |F' \cap \Gamma(S)|$ and the right-hand side by $3\epsilon_i |A_i \cap CC|$. Then, Lemma 3 concludes the proof of the bound on the connection costs.

In order to prove the bound on the penalties we employ the following lemma.

▶ **Lemma 6.** *Let F' and y_S be the primal and dual solution constructed by the algorithm. The set of vertices $X = V \setminus F'$ not spanned by the final solution can be partitioned into sets $X_1, X_2, \ldots X_l$ such that the potential of each set is 0, i.e., $P(X_k) = 0$ for each k.*

Proof. Observe that there are two ways for a vertex v to be in X: either it was never a part of the root component ($v \in V \setminus F^{(r)}$) or it was deleted in the pruning phase ($v \in F^{(r)}$). It is easy to see that $P(V \setminus F^{(r)}) = 0$. Each vertex in $V \setminus F^{(r)}$ was at the end a part of some inactive component not containing the root and hence the potentials of these components were 0. Or, it was never in any moat.

It remains to show that the set S of vertices disconnected from F' by pruning a vertex v can be partitioned into sets X_k for which $P(X_k) = 0$. Let t be the time when v was bought. Observe that every vertex u in the neighborhood $\Gamma(S)$ of S has been bought after time t or was not bought at all. Now, S contains only marked terminals at time t, otherwise v would not have been pruned. Hence, S is a union of inactive moats at time t. This gives the desired partition. ◀

Observe that PC is the set of all $S \subseteq X_i$ with $y_S > 0$. To conclude the bound on the penalties note that since all X_k have zero potential we have

$$\Pi(V \setminus F') = \sum_{k=1}^{l} \Pi(X_k) = \sum_{k=1}^{l} \sum_{S \subseteq X_k} y_S = \sum_{S \in PC} y_S.$$

3 Combination with threshold rounding

A standard technique to generalize primal-dual algorithms from Steiner tree problems to their price-collecting variations is to use *threshold rounding* (see Section 5.7 of [22] or [13]). Here, in the first step an LP formulation for the price-collecting version is solved over fractional variables. Then, we pick a threshold α and consider the vertices that are bought with value at least α to be terminals. In the second step, the primal-dual algorithm for the original Steiner tree problem is run on this set of terminals to obtain the final solution. We note that the resulting algorithm is deterministic because we can try all possible thresholds (at most one for every vertex). However, the analysis uses a randomization argument.

We observed in [21] that using threshold rounding in combination with the primal-dual 2.4-approximation for node-weighted Steiner forest by Berman and Yaroslavtsev [6] yields a 2.93-approximation for NWPCST on planar graphs.

In this section, we combine the previous LMP algorithm with the threshold rounding technique to gain an improved approximation factor of 2.88. Our approach is inspired by an idea of Goemans [14]. Intuitively, such an improvement is possible because the LMP approximation improves over the factor of 3 if the optimal solution induces a high penalty cost. In contrast, if the penalties are only a small part of the optimal solution's cost, threshold rounding can leverage the robustness of the underlying 2.4-approximation. Thus, by combining the two algorithms we can hedge their weaknesses.

3.1 Threshold rounding

We use the standard threshold rounding technique (cf. [22]). Consider the following LP

$$
\begin{aligned}
\min \sum_{v\in V} w_v x_v + \sum_{u\in V\setminus\{r\}} \pi_u y_u & & (LP_{thr})\\
s.t. & & \\
\sum_{v\in\Gamma(S)} x_v + y_u \geq 1 & & \forall S\subseteq V\setminus\{r\},\quad u\in S\\
x_v \geq 0 \quad \forall v\in V & & y_u \geq 0 \quad \forall u\in V
\end{aligned}
$$

This LP is equivalent to the LP used in the construction of the primal-dual LMP 3-approximation from Section 2. This was shown by Williamson for the edge-weighted variant (see section 7.4.1 of [23]), however arguments are identical in our case. This is due to the fact, that the mapping between feasible solutions leaves variables related to connection costs unchanged and constructs variables z based solely on y and vice-versa.

In the sequel, let (x^*, y^*) be the optimum solution to LP_{thr} with objective value OPT_{LP}. Further, if T is a solution to NWPCST, let $w(T)$ be the total connection and $\pi(V\setminus T)$ be the total penalties of T. We also use this notation for (fractional) solutions: $w(x)$, $\pi(z)$ and $\pi(y)$.

Let $\beta\in(0,1)$ be a constant to be determined later. For every possible value α of y^* that is at most β, let $Q=\{u : y^*_u\leq\alpha\}$. Consider the instance I_{NWST_Q} of the $NWST$ problem which is derived from I_{NWPCST} by keeping only terminals from Q. Let LP_{NWST_Q} be the following linear program

$$\min \sum_{v \in V} w_v x_v \qquad\qquad (LP_{NWST_Q})$$
$$\sum_{v \in \Gamma(S)} x_v \geq 1 \qquad\qquad \forall S \subseteq V \setminus \{r\}, \quad Q \cap S \neq \emptyset$$
$$x_v \geq 0 \qquad\qquad \forall v \in V$$

Let OPT_{LP_Q} be the optimum objective function value of LP_{NWST_Q}. We run the 2.4-approximation algorithm for I_{NWST_Q} by Berman and Yaroslavtsev [6] which returns a solution F such that its cost is no greater than $2.4 \cdot OPT_{LP_Q}$. Finally, return the best of all obtained solutions F (due to different values of α).

Though the algorithm is deterministic its analysis is based on a randomized argument. Instead of trying all possible values of α, consider α to be chosen uniformly at random from $[0, \beta]$. Consider $x' = \frac{1}{1-\alpha} x^*$. It follows that x' is a feasible solution to LP_{NWST_Q}. We bound the expected connection and penalty costs of F.

$$\begin{aligned}
\mathbb{E}\left[\sum_{v \in F} w_v\right] &\leq \mathbb{E}\left[2.4 \cdot OPT_{LP_Q}\right] \leq \mathbb{E}\left[2.4 \sum_{v \in V} x'_v \cdot w_v\right] \leq \mathbb{E}\left[\frac{2.4}{1-\alpha}\right] \sum_{v \in V} x^*_v \cdot w_v \\
&= \left(\int_0^\beta \frac{1}{\beta} \cdot \frac{2.4}{1-\alpha} d\alpha\right) w(x^*) \\
&= \frac{2.4}{\beta} \ln\left(\frac{1}{1-\beta}\right) w(x^*)
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}\left[\sum_{u \notin Q} \pi_u\right] &= \mathbb{E}\left[\sum_{u : y^*_u > \alpha} \pi_u\right] \leq \sum_u \pi_u Pr\left[y^*_u \geq \alpha\right] \leq \sum_u \pi_u \int_0^{y^*_u} \frac{1}{\beta} d\alpha \\
&= \sum_u \pi_u \frac{1}{\beta} y^*_u = \frac{1}{\beta} \pi(y^*)
\end{aligned}$$

3.2 Combining the two algorithms

To combine the LMP approximation with threshold rounding we require a slight modification of the instance submitted to the LMP approximation.

Recall that for an instance I the LMP 3-approximation returns a solution T such that $w(T) + 3\pi(V \setminus T) \leq 3OPT_{LP}$. Consider now instance I' with has its penalties scaled by $1/3$, *i.e.*, $\pi'_v = \frac{1}{3}\pi_v$. Run the LMP approximation on I' to obtain a solution T' satisfying $w(T') + \pi(V \setminus T') = w(T') + 3\pi'(V \setminus T') \leq 3OPT'_{LP}$, where OPT'_{LP} is the value of the optimum solution to program LP' derived from LP_{thr} by taking scaled penalties π'. Observe that (x^*, y^*) is also feasible to LP', because this program differs only in the objective function. Hence we have that

$$w(T') + \pi(V \setminus T') \leq 3OPT'_{LP} \leq 3\left(w(x^*) + \pi'(y^*)\right) = 3w(x^*) + \pi(y^*)$$

Now, our final algorithm returns the best solution among T' and the solution produced by the threshold rounding technique in the previous section. Note that this is a deterministic procedure. However, the analysis uses a randomized argument inspired by Goemans [14]:

pick one solution with probability p and the other with probability $1-p$. Let SOL be the returned solution.

$$\mathbb{E}[SOL] \leq \left[3p + (1-p)\frac{2.4}{\beta}\ln\left(\frac{1}{1-\beta}\right)\right] w(x^*) + \left[p + (1-p)\frac{1}{\beta}\right]\pi(y^*)$$
$$\leq \left[\left(3p+(1-p)\frac{2.4}{\beta}\ln\left(\frac{1}{1-\beta}\right)\right) w(x^*) + \left(p+(1-p)\frac{1}{\beta}\right)\pi(y^*)\right]$$

Finally, optimizing constants we obtain for $\beta = 1 - e^{-\frac{5}{36}}$ and $p = \frac{1}{4-3e^{-5/36}}$ the claimed result

$$\mathbb{E}[SOL] \leq \frac{4}{4-3e^{-5/36}}(w(x^*) + \pi(y^*))$$
$$\leq \frac{4}{4-3e^{-5/36}} OPT \approx 2.8797 \cdot OPT$$

4 The primal-dual 4-approximation for forest

In this section we use a general combinatorial approach for solving prize-collecting problems introduced by Hajiaghayi and Jain [17]. In their work they obtained the primal-dual 3-approximation algorithm for edge-weighted prize-collecting Steiner forest problem. We adapt their argumentation to the planar node-weighted setting resulting in the 4-approximation algorithm. We provide here only a sketch - the more detailed description and proofs can be found in the full version of the paper [8].

Consider a graph $G = (V, E)$ with a non-negative cost function on nodes $w : V \to Q_+$, a set of pairs of vertices (demands) $D = (s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$ and a non-negative penalty function $\pi : D \to Q_+$. In the node-weighted prize-collecting Steiner forest problem we are asked to find a set of vertices $F \subseteq V$ which minimizes the sum of costs of vertices in F plus penalties for pairs of vertices which are not connected in a subgraph of G induced by F.

Note that we can give an equivalent definition of demands and penalties by specifying penalties for each unordered pair of vertices. Simply set penalties for pairs of vertices which are not in D to 0. From now on we will use values π_{ij} to denote penalties. Let also $\Gamma(S)$ denote the set of vertices in $V \setminus S$ incident to vertices from $S \subseteq V$ and let $S \odot (i,j)$ means that $|(i,j) \cap S| = 1$ (i.e., S separates vertices i and j) Using this notation, we can formulate our problem with the following integer program

$$\min \sum_{v\in V} w_v x_v + \sum_{(i,j)\in V\times V} \pi_{ij} z_{ij} \qquad (IP_{SF})$$
$$s.t.$$
$$\sum_{v\in\Gamma(S)} x_v + z_{i,j} \geq 1 \qquad \forall S \subseteq V, \quad \forall (i,j) \in V \times V: \quad S \odot (i,j)$$
$$x_v \in \{0,1\} \qquad \forall v \in V$$
$$z_{i,j} \in \{0,1\} \qquad \forall (i,j) \in S \times S$$

Setting $x_v = 1$ corresponds to buying a vertex v (including v into solution F) and setting $z_{i,j} = 1$ corresponds to paying a penalty instead of connecting vertices i and j.

Unfortunately, the dual of the linear relaxation of IP_{SF} is not suitable for obtaining a primal-dual algorithm. However, following the framework in [17], we can replace it with the following LP:

$$\max \sum_{S\subset V} y_S \qquad (DLP_{SF}4)$$

$$s.t.$$

$$\sum_{S:v\in\Gamma(S)} y_S \le w_v \qquad \forall v \in V \qquad (4)$$

$$\sum_{S\in\mathbb{S}} y_S \le \sum_{(i,j)\in V\times V, \mathbb{S}\odot(i,j)} \pi_{i,j} \qquad \forall \mathbb{S} \in 2^{2^V} \qquad (5)$$

$$y_S \ge 0 \qquad \forall S \subset V$$

where $\mathbb{S} \odot (i,j)$ denotes that there exists $S \in \mathbb{S}$ such that $S \odot (i,j)$ (we say that family $\mathbb{S}$ separates vertices i and j if and only if there exists at least one set $S \in \mathbb{S}$ which separates vertices i and j).

This new formulation allows us to obtain a natural primal-dual algorithm which is described below.

The algorithm starts with an initial solution F in which there are all vertices of cost 0 (hence all terminals). In each iteration the algorithm maintains moats which are the connected components of graph G induced by the vertices of the current solution F. Demands can be marked (meaning that we decide to pay a penalty for them) or unmarked. At the beginning all demands are unmarked. Once demand is marked, it stays marked forever. A moat (denoted by the corresponding set $S \subseteq V$) is active in the current iteration if and only if there is at least one unmarked demand (i,j) such that $S \odot (i,j)$. Now in each iteration we simultaneously grow each active moat until one of the following two events occur:

- a vertex v goes tight (constraint (4) becomes equality), or
- a family $\mathbb{S}$ goes tight (constraint (5) becomes equality).

In the first case we simply add v to our solution F (which may make some moats inactive) and continue to the next iteration.

In the second case, we mark each demand (i,j) such that $\mathbb{S} \odot (i,j)$. Hence in the following iterations all moats from $\mathbb{S}$ will be inactive, and we will not violate any constraint during the growth process. We repeat this process until all moats become inactive.

After that we have an additional pruning phase in which we process all vertices of F in the reverse order of buying. We remove a vertex v from F if after its removal from F, all unmarked demands are still connected in the graph induced by F. We output this pruned set of vertices as F' which is our final solution.

Obtaining ϵ_1 and a tight vertex in line 7 is straightforward. On the other hand obtaining ϵ_2 in line 8 and a tight family $\mathbb{S}$ seems to be much harder, since the number of corresponding constraint is doubly exponential. Fortunately Hajiaghayi and Jain in section 4 of [17] gave a polynomial time algorithm for computing ϵ_2 and the corresponding tight family $\mathbb{S}$.

Since the algorithm terminates after at most $2|V| - 1$ iterations (in each iteration the number of active moats or the number of connected components decreases), the running time of this algorithm is polynomial.

We can combine proofs from [20] and [17] in order to obtain the following result.

▶ **Theorem 7.** *The algorithm outputs a set of vertices F' and a set of demands Q' which are not connected via F' such that*

$$\sum_{v\in F'} w_v + \sum_{(i,j)\in Q'} \pi_{ij} \le 4 \sum_{S\subseteq V} y_S \le 4\ OPT\,.$$

Input : A planar graph $G = (V, E)$ with non-negative weights w_i on the nodes and non-negative penalties π_{ij} between each pair of vertices such that if $\pi_{ij} > 0$ then $w_i = 0$ and $w_j = 0$

Output : A set of vertices F' representing a forest and a set of pairs Q' representing not connected demands

begin

- $F \leftarrow \{v_i \in V : w_i = 0\}$;
- $Q \leftarrow \emptyset$ `// set all demands unmarked`
- $y_S \leftarrow 0$ `// implicitly`
- $AM \leftarrow \left\{ S \subseteq V : S \in SCC\,(G[F]) \wedge \underset{(i,j) \in V \times V - Q}{\exists} \pi_{ij} > 0 \wedge S \odot (i,j) \right\}$;
- `// identify active moats as components of subgraph of` G `induced by vertices` F `for which there is at least one unmarked demand` (i,j) `which is separated by the corresponding set`
- **while** $AM \neq \emptyset$ **do**
  - find minimum ϵ_1 s.t if we increase y_S for each $S \in AM$ by ϵ_1 we get a new tight vertex v;
  - find minimum ϵ_2 s.t if we increase y_S for each $S \in AM$ by ϵ_2 we get a new tight family $\mathbb{S}$;
  - $\epsilon \leftarrow min(\epsilon_1, \epsilon_2)$;
  - $y_S \leftarrow y_S + \epsilon$ for all $S \in AM$;
  - **if** $\epsilon = \epsilon_1$ **then**
    - $F \leftarrow F \cup \{v\}$;
  - **else**
    - $Q \leftarrow Q \cup \{(i,j) \in V \times V : \mathbb{S} \odot (i,j)\}$
  - **end**
  - $AM \leftarrow \left\{ S \subseteq V : S \in SCC\,(G[F]) \wedge \underset{(i,j) \in V \times V - Q}{\exists} \pi_{ij} > 0 \wedge S \odot (i,j) \right\}$;
- **end**
- `// pruning phase`
- Derive F' from F by removing vertices in reverse order of purchase so that every unmarked demand is connected in F'.
- Let Q' be all demands not connected via F'

end

Algorithm 1: Primal-dual algorithm for NWPCSF on planar graphs.

We need to show that:

$$\sum_{v \in F'} w_v \leq 3 \sum_{S \subseteq V} y_S \qquad \text{and} \qquad \sum_{(i,j) \in Q'} \pi_{ij} \leq \sum_{S \subseteq V} y_S$$

The bound on the connection cost is shown in a similar way as in the tree version, *i.e.*, using a degree counting argument for each iteration. This is captured by the lemma below.

For a set of nodes F and the set of unmarked demands $R = D - Q$ define a minimal feasible augmentation F_{aug} of F with respect to R to be a set of vertices F_{aug} containing F as a subset such that every pair of vertices from R is connected in the subgraph of G induced by F_{aug} and such that removal of any $v \in F_{aug} \setminus F$ from F_{aug} disconnects some pair from R.

▶ **Lemma 8** (Analog of Analysis in [20]). *Let G be planar, R be the set of unmarked demands after running the above algorithm, F_j be the set of bought vertices before running iteration j and F_{aug} be a minimal feasible augmentation of F_j with respect to R. Let also A_j be the set of active moats before running iteration j. Then*

$$\sum_{S \in A_j} |F_{aug} \cap \Gamma(S)| \leq 3|A_j|\,.$$

The proof of this lemma is conducted in a similar way as the proof of Lemma 3 and the analysis is essentially the same as in [20].

In turn, the bound on penalties is shown exactly in the same way as in the edge-weighted version [17]. When we mark a pair it belongs to a tight family. It is observed that the union of those tight families is also tight, hence the corresponding constraint gives the bound.

The more detailed proofs of these bounds can be found in in the full version of the paper [8].

Note that we cannot separate dual variables like in the tree version, hence we obtain a factor of 4 instead of 3 as in Section 2. This is essentially due to the same difficulty as in the standard edge-weighted variant for the prize-collecting Steiner forest problem.

References

1 Ajit Agrawal, Philip N. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

2 Aaron Archer, MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Howard J. Karloff. Improved approximation algorithms for prize-collecting steiner tree and TSP. *SIAM J. Comput.*, 40(2):309–332, 2011.

3 MohammadHossein Bateni, Chandra Chekuri, Alina Ene, Mohammad Taghi Hajiaghayi, Nitish Korula, and Dániel Marx. Prize-collecting steiner problems on planar graphs. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1028–1049, 2011.

4 MohammadHossein Bateni, Mohammad Taghi Hajiaghayi, and Dániel Marx. Approximation schemes for steiner forest on planar graphs and graphs of bounded treewidth. *J. ACM*, 58(5):21, 2011.

5 MohammadHossein Bateni, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Improved approximation algorithms for (budgeted) node-weighted steiner problems. In *Automata, Languages, and Programming – 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 81–92, 2013.

6 Piotr Berman and Grigory Yaroslavtsev. Primal-dual approximation algorithms for node-weighted network design in planar graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques – 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 50–60, 2012.

7 Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. An improved lp-based approximation for steiner tree. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 583–592, 2010.

8 Jarosław Byrka, Mateusz Lewandowski, and Carsten Moldenhauer. Approximation algorithms for node-weighted prize-collecting steiner tree problems on planar graphs. *CoRR*, abs/1601.02481, 2016.

9 Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Philip N. Klein. Node-weighted steiner tree and group steiner tree in planar graphs. *ACM Trans. Algorithms*, 10(3):13:1–13:20, 2014.

10 Bistra N. Dilkina and Carla P. Gomes. Solving connected subgraph problems in wildlife conservation. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 7th International Conference, CPAIOR 2010, Bologna, Italy, June 14-18, 2010. Proceedings*, pages 102–116, 2010.

11 Karoline Faust, Pierre Dupont, Jérôme Callut, and Jacques van Helden. Pathway discovery in metabolic networks by subgraph extraction. *Bioinformatics*, 26(9):1211–1218, 2010.

12 M. R. Garey and David S. Johnson. The rectilinear steiner tree problem in NP complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977.

13 Joseph Geunes, Retsef Levi, H. Edwin Romeijn, and David B. Shmoys. Approximation algorithms for supply chain planning and logistics problems with market choice. *Math. Program.*, 130(1):85–106, 2011.

14 Michel X. Goemans. Combining approximation algorithms for the prize-collecting TSP. *CoRR*, abs/0910.0553, 2009.

15 Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995.

16 Sudipto Guha, Anna Moss, Joseph Naor, and Baruch Schieber. Efficient recovery from power outage (extended abstract). In *Proc. of the 31st Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 574–582, 1999.

17 Mohammad Taghi Hajiaghayi and Kamal Jain. The prize-collecting generalized steiner tree problem via a new approach of primal-dual schema. In *Proc. of the 7th Annual ACM-SIAM Symp. on Discrete Algorithms, SODA'06*, pages 631–640, 2006.

18 Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

19 Jochen Könemann, Sina Sadeghian Sadeghabad, and Laura Sanità. An LMP o(log n)-approximation algorithm for node weighted prize collecting steiner tree. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 568–577, 2013.

20 Carsten Moldenhauer. Primal-dual approximation algorithms for node-weighted steiner forest on planar graphs. *Inf. Comput.*, 222:293–306, 2013.

21 Carsten Moldenhauer. *Node-weighted network design and maximum sub-determinants*. PhD thesis, EPFL, 2014.

22 David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

23 David Paul Williamson. *On the design of approximation algorithms for a class of graph problems*. PhD thesis, MIT, Cambridge, MA, September 1993.