Information and Computation 222 (2013) 293–306

Contents lists available at SciVerse ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Primal-dual approximation algorithms for Node-Weighted Steiner Forest on planar graphs

Carsten Moldenhauer

EPFL SB MATH AA, Station 8, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Available online 26 October 2012

Keywords:
Node-Weighted Steiner Forest
Generalized Steiner Tree
Vertex feedback set
Primal-dual algorithm
Approximation algorithms
Planar graphs

ABSTRACT

Node-Weighted Steiner Forest is the following problem: Given an undirected graph, a set of pairs of terminal vertices, a weight function on the vertices, find a minimum weight set of vertices that includes and connects each pair of terminals. We consider the restriction to planar graphs where the problem remains NP-complete. Demaine et al. showed that the generic primal-dual algorithm of Goemans and Williamson is a 6-approximation on planar graphs. We present (1) two different analyses to prove an approximation factor of 3, (2) show that our analysis is best possible for the chosen proof strategy, and (3) generalize this result to feedback problems on planar graphs.

We give a simple proof for the first result using contraction techniques and following a standard proof strategy for the generic primal-dual algorithm. Given this proof strategy our analysis is best possible which implies that proving a better upper bound for this algorithm, if possible, would require different proof methods. Then, we give a reduction on planar graphs of Feedback Vertex Set to Node-Weighted Steiner Tree, and Subset Feedback Vertex Set to Node-Weighted Steiner Forest. This generalizes our result to the feedback problems studied by Goemans and Williamson. For the opposite direction, we show how our constructions can be combined with the proof idea for the feedback problems to yield an alternative proof of the same approximation guarantee for Node-Weighted Steiner Forest.

1. Introduction

We consider the following problem, called Node-Weighted Steiner Forest (NWSF): Given an undirected graph, a set of pairs of terminal vertices, a weight function on the vertices, find a minimum weight set of vertices whose induced graph includes and connects each pair of terminals. This problem is a generalization of Steiner Forest, where the weight function is defined on the edges and the task is to find a minimum weight set of edges. Steiner Forest reduces to our problem by placing a new vertex on every edge and setting its weight to the edge weight. It follows that NWSF is NP-complete and remains so when restricted to planar graphs [18].

Steiner problems are among the oldest NP-complete problems. In fact, the (edge-weighted) Steiner Tree problem, where all terminal pairs have one common terminal, is one of the 21 NP-complete problems from Karp's list [25]. Besides numerous research in the past, Steiner problems have particularly received much attention in the last three years: Steiner Tree is known to be MAX–SNP hard on general graphs and Chlebík and Chlebíková [11] presented the current best lower

E-mail address: carsten.moldenhauer@epfl.ch.

0890-5401/$ – see front matter
http://dx.doi.org/10.1016/j.ic.2012.10.017

bound of 96/95 in 2008. Further, Byrka et al. [10] recently achieved a 1.39-approximation algorithm. On planar graphs, Borradaile et al. [9] gave a polynomial time approximation scheme for Steiner Tree that has been extended to polynomial time approximation schemes for Steiner Forest [6] and the prize-collecting variants [5]. In contrast, for node-weighted Steiner problems on planar graphs, the only known result is a 6-approximation algorithm [14].

Node-weighted Steiner problems have various applications reaching from maintenance of electric power networks [22] to computational sustainability [15]. Particularly, they have been studied as subproblems in more constrained settings [29] and there is an extensive body of research on various variations [30,2,27]. Due to an easy approximation preserving reduction from Set Cover, NWSF cannot be approximated up to $(1 - o(1)) \ln k$ in general graphs where k is the number of terminal pairs, unless NP admits slightly superpolynomial time algorithms [17]. Hence, the node-weighted problem is much harder to approximate than the edge-weighted one. Klein and Ravi [26] presented an algorithm for NWSF which is implicitly a primal-dual algorithm and has an approximation guarantee of $2 \ln k$. Therefore, their algorithm is optimal up to a constant factor on general graphs. Guha and Khuller [21] improved on this work yielding multiple algorithms with even better guarantees for various subproblems. In contrast, on planar graphs substantial improvement is still possible.

Since most practical applications occur in planar or nearly planar environments, we focus on planar graphs. The only algorithm that has been previously applied to Node-Weighted Steiner Forest in this setting is the primal-dual algorithm of Goemans and Williamson [19]. Demaine et al. [14] showed that this algorithm is a 6-approximation. Due to an edge counting argument their analysis also extends to graphs with a fixed forbidden minor.

We also consider the Feedback Vertex Set problem (FVS): Given an undirected graph with weight function on the vertices, select a set of vertices of least weight whose removal renders the graph acyclic, *i.e.*, destroys all cycles. In Subset Feedback Vertex Set (SFVS) we are additionally given a set of special vertices and the task is to destroy only cycles that pass through special vertices. Becker and Geiger [7] and Bafna et al. [3] each developed a 2-approximation algorithm for FVS on general undirected graphs. Both algorithms have been reinterpreted as primal-dual methods [12] using a different integer programming formulation from what we consider here. When restricted to planar graphs, FVS admits a PTAS [13]. In contrast, the best result for SFVS on general graphs is an 8-approximation due to Even et al. [16]. On planar graphs, SFVS allows for a 3-approximation via the primal-dual method [20].

Our contributions

1. We improve the result of Demaine et al. [14] by showing that the generic primal-dual algorithm has an approximation factor of 3 for Node-Weighted Steiner Forest on planar graphs. We outline our proof in Section 3 using contraction techniques. Further, our analysis implies a reduction by a factor of 2/3 in the constants in [14, p. 337] for graphs with a fixed forbidden minor.

2. Our proof follows a standard proof strategy for primal-dual algorithms due to Goemans and Williamson [19]. We show that our analysis is best possible with respect to this proof strategy in Section 4. This means that obtaining a better upper bound requires new proof methods or changes to the algorithm. We further discover similarities with the example that shows tightness of our analysis and the lower bound example for some feedback problems on planar graphs used in [20].

3. We explore these similarities by giving approximation preserving reductions on planar graphs from Feedback Vertex Set to Node-Weighted Steiner Tree, and from Subset Feedback Vertex Set to Node-Weighted Steiner Forest in Section 5. Hence, the approximation guarantee of 3 for NWSF carries over to the feedback problems. Further, we show that not only the integral solutions are equal but also that the polyhedra of the linear programming relaxations of the respective problems and their induced instances used in the reductions are equivalent. This sheds light on the fact that the primal-dual algorithm can also be applied directly as in [20].

4. The above reductions imply that our approximation result for NWSF generalizes to the feedback problems on planar graphs. Alternatively, also a derivation in the opposite direction is possible. We combine our constructions with the proof techniques of Goemans and Williamson [20] to yield an alternative, more involved, proof of the factor of 3 in Section 6.

We remark that the primal-dual method can be reformulated as a local ratio algorithm. For more information, the reader is referred to the survey by Bar-Yehuda et al. [4]. Thus, alternatively to our chosen exposition, the algorithm can be formulated using local ratio, requiring only minimal modifications to the proofs.

Addendum to the conference version

The preceding version of this paper [28] gave an example of a lower bound of 3 for the generic primal-dual algorithm. This example does not account for the reverse delete step of the algorithm and only yields a lower bound of 2. We discuss lower bounds for the algorithm in Section 4.

The primal-dual algorithm is defined by an oracle. [28] devised an improved oracle and claimed an approximation guarantee of 9/4 for the resulting algorithm. The proof is incorrect and we point out the flaw in detail in Section 7. We remark that the paper of Goemans and Williamson [20] seems to have the same problem and that Berman and Yaroslavtsev [8] provide a counter-example for Subset Feedback Vertex Set contradicting both our and their 9/4-approximation results.

1.1. Preliminaries

Let $G = (V, E)$ be an undirected plane graph, *i.e.*, a planar graph with given embedding. Let $S \subseteq V$ be a set of vertices. We denote the induced subgraph of S in G by $G[S]$. The set of neighbors of S is denoted by $\Gamma(S) = \{v\colon\ u \in S,\ v \notin S,\ uv \in E\}$.

The set of connected components of G is denoted by $\mathcal{C}(G)$. A ρ-approximation algorithm is an algorithm that runs in polynomial time and produces a feasible solution of weight at most ρ times the weight of an optimal solution.

Let $w : V \to \mathbb{R}_{\geqslant 0}$ be a weight function on the vertices. Let $\{(s_1, t_1), \ldots, (s_k, t_k)\}$ be a set of pairs of vertices, called *terminal pairs*. The problem Node-Weighted Steiner Forest is to find a minimum weight set $X \subseteq V$ of vertices such that for all $1 \leqslant i \leqslant k$, s_i and t_i are in the same component of $G[X]$. The usual undirected cut-formulation of the problem as integer linear program is the following [19,14,31,23]. Let $\mathcal{T}$ be the family of cuts that separate a terminal pair:

$$\mathcal{T} = \bigl\{S \subseteq V\colon \bigl|\{s_i, t_i\} \cap S\bigr| = 1 \text{ for some } 1 \leqslant i \leqslant k\bigr\}.$$

Then, Node-Weighted Steiner Forest is the problem

$$\text{(NWSF)} \qquad \begin{aligned} &\text{minimize} && \sum_{v \in V} w(v)\, x_v && (1)\\ &\text{subject to} && \sum_{v \in \Gamma(S)} x_v \geqslant 1 \quad \forall S \in \mathcal{T} && (2)\\ & && x_v \in \{0,1\} \quad \forall v \in V. && (3)\end{aligned}$$

The constraints (2) ensure that for every set that separates terminals the solution has to select one of the neighbors of this set. This guarantees connectedness of the terminal pairs.

The problem Node-Weighted Steiner Tree (NWST) is the restriction of Node-Weighted Steiner Forest where all terminal pairs have one common terminal. Thus, the terminal pairs effectively represent one set of terminals that all have to be connected to each other.

Since all the terminals have to be in the solution, we can assume w.l.o.g. that they have zero weight. Let $\mathrm{OPT}_{\mathrm{gen}}$ and OPT denote the optimal solutions to the general problem and the problem with terminal weights equal to zero, respectively. A ρ-approximation algorithm (with value ALG) for the problem with zero weight terminals induces a ρ-approximation for the general problem, because

$$\mathrm{ALG} + \sum_{\text{terminal } v} w(v) \leqslant \rho\, \mathrm{OPT} + \sum_{\text{terminal } v} w(v) \leqslant \rho\Bigl(\mathrm{OPT} + \sum_{\text{terminal } v} w(v)\Bigr) = \rho\, \mathrm{OPT}_{\mathrm{gen}}.$$

2. The primal-dual algorithm

In this section, we will briefly describe the generic primal-dual method, systematically developed by Goemans and Williamson. Their method was originally designed for edge weighted problems. To apply the same algorithm to Node-Weighted Steiner Forest, Demaine et al. [14] introduced a slight modification in the initialization, that we describe here. For more information on primal-dual algorithms, we refer to the surveys [19,23] and the exposition in [31]. We remark that the primal-dual algorithm for Steiner problems can also be reformulated as a local ratio algorithm [4].

Consider the LP relaxation of (NWSF) that replaces the integrality constraints (3) by $x_v \geqslant 0$. The dual of the relaxation is

$$\text{(DNWSF)} \qquad \begin{aligned} &\text{maximize} && \sum_{S \in \mathcal{T}} y_S && (4)\\ &\text{subject to} && \sum_{S \in \mathcal{T}\colon v \in \Gamma(S)} y_S \leqslant w(v) \quad \forall v \in V && (5)\\ & && y_S \geqslant 0 \quad \forall S \in \mathcal{T}. && (6)\end{aligned}$$

Intuitively, the dual packing constraints (5) ensure that the weight distributed among the sets adjacent to v is not more than the weight of v.

The algorithm simultaneously constructs a feasible solution X corresponding to (NWSF) and a feasible solution $\mathbf{y}$ to the dual of the LP relaxation (DNWSF). The initial infeasible solution to X equals the set of all terminals (this is the modification by [14] indicated above) and $\mathbf{y} \equiv 0$. The algorithm is defined by an oracle which, given an infeasible solution X, selects a set of violated constraints in the primal program (NWSF). We denote this oracle by $\mathrm{VIOL}(X)$ and it returns the sets from $\mathcal{T}$ associated with the selected violated constraints (2) of X. While X is not a feasible solution, the algorithm simultaneously and uniformly increases the dual variables on the sets returned by $\mathrm{VIOL}(X)$ (note that the design rule of increasing all dual variables with the same speed was first used in [1]). When one of the dual packing constraints (5) becomes tight for some vertex $v \in V$, v is added to X. Observe that after v is added to X, no variable y_S occurring in v's packing constraint (5) will be increased again since the respective constraint (2) for S is satisfied. After a feasible solution is obtained, the algorithm performs a clean-up step. It considers all vertices in X in the reverse order in which they were added and removes any vertex that is not necessary for feasibility of X.

Primal-Dual Algorithm

(i) $X \leftarrow \{s_i\colon\ 1 \leqslant i \leqslant k\} \cup \{t_i\colon\ 1 \leqslant i \leqslant k\}$.
(ii) $\mathbf{y} \leftarrow 0$.
(iii) While X is not feasible:
  (a) Increase y_S uniformly for all $S \in \mathrm{VIOL}(X)$ until $\exists S \in \mathrm{VIOL}(X)$, $v \in \Gamma(S)$: $\sum_{S' \in \mathcal{T}: v \in \Gamma(S')} y_{S'} = w(v)$.
  (b) $X \leftarrow X \cup \{v\}$.
(iv) For each $v \in X$ in the reverse order in which they were added in step (iii):
  (a) If $X \setminus \{v\}$ is feasible then set $X \leftarrow X \setminus \{v\}$.
(v) Return X (and $\mathbf{y}$).

Note that the algorithm runs in polynomial time if the oracle VIOL can be computed in polynomial time and returns only polynomially many sets. Even though there might be exponentially many variables y_S, only polynomially many will be changed in the course of the algorithm. In fact, the algorithm can be implemented to encode the values of $\mathbf{y}$ implicitly. We refer the reader to [19, p. 164, Fig. 4.5] for details. In the following sections we will present two different oracles that can easily be computed in polynomial time. In fact, for a given infeasible set X they each return a subset of the components $\mathcal{C}(X)$. Recall, that a solution X is feasible if and only if all terminals are selected and all its components do not separate any terminal pair. Thus, for our oracles, the while loop in the algorithm is executed until the oracle does not return any sets anymore.

The above algorithm differs slightly from the primal-dual algorithm for edge-weighted problems because in (i) X is initialized with the set of all terminals. Since we can assume that all terminals have zero weight (see above), this is no restriction. In [19] it is proved that the performance guarantee can be obtained by using the theorem below. A *minimal augmentation* F of X is a feasible solution F containing X such that for any $v \in F \setminus X$, $F \setminus \{v\}$ is not feasible.

Theorem 1. *(See [19].) Let* OPT *denote the weight of an optimal solution. Let γ satisfy that for any infeasible set $X \subseteq V$, containing all the terminals, and any minimal augmentation F of X*

$$\sum_{S \in \mathrm{VIOL}(X)} \left|F \cap \Gamma(S)\right| \leqslant \gamma \left|\mathrm{VIOL}(X)\right|.$$

Then, the primal-dual algorithm delivers a solution of weight at most $\gamma \sum_{S \in \mathcal{T}} y_S \leqslant \gamma\,\mathrm{OPT}$.

A compact version of the proof can be found in Appendix A.

Therefore, given an oracle VIOL we have to calculate the value γ given by Theorem 1. For the oracle presented in this paper we will show that the value of γ is 3.

3. A 3-approximation on planar graphs

We now consider planar graphs and the oracle AVC that returns all components of the current solution that separate a terminal pair. Recall that a set S separates a terminal pair (s_i, t_i) if $|S \cap \{s_i, t_i\}| = 1$. We call a component that separates a terminal pair *violated*. Let X be the current solution in any iteration of the algorithm. The set of all violated components is $\mathrm{AVC}(X) = \mathcal{C}(X) \cap \mathcal{T}$.

Note that $\mathrm{AVC}(X)$ returns the inclusion-wise minimal sets from $\mathcal{T}$ for which the constraint (2) is violated: Let S be a set with violated constraint (2) for X. Let (s_i, t_i) be any terminal pair that is separated by S and w.l.o.g. $s_i \in S$. Due to the initialization in step (i), every terminal is included in a component of X. Denote the component of X containing s_i by C. Clearly, $C \subseteq S$ because otherwise S has a neighbor in $C \setminus S \subseteq X$ and its respective constraint would not have been violated. Now, $C \in \mathrm{AVC}(X)$ since C is a component of X and separates (s_i, t_i).

Clearly, AVC can be computed in polynomial time by checking each component of X if it separates two terminals. We will show that the corresponding value of γ in Theorem 1 is 3 and therefore the primal-dual algorithm is a 3-approximation.

Theorem 2. *Let G be planar, X be an infeasible solution containing all the terminals and F be any minimal feasible augmentation of X. Then,*

$$\sum_{S \in \mathrm{AVC}(X)} \left|F \cap \Gamma(S)\right| \leqslant 3\left|\mathrm{AVC}(X)\right|. \tag{7}$$

Proof. The left-hand side of (7) counts the number of adjacencies between the minimal augmentation F and the violated components of X. Note that the sum is over all violated components. That means, a vertex in F that is adjacent to multiple violated components will be counted multiple times. But, a vertex in F is counted at most once per violated component, *i.e.*, adjacencies to multiple vertices in the same violated component are counted only once.

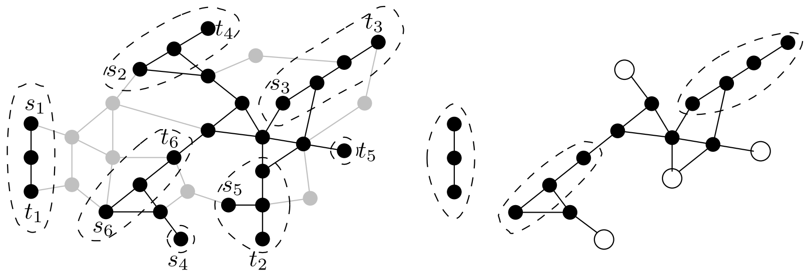
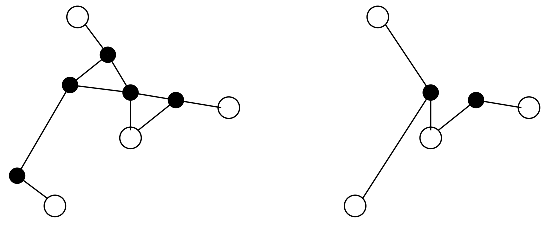


Fig. 1. Construction of $\overline{G}$ from $G[F]$. (a) G in gray; $G[F]$ in black; components in $\mathcal{C}(X)$ indicated by dashed ovals. (b) $\overline{G}$ with non-violated components of X indicated by dashed ovals.



(a) $\hat{G}$. (b) G'.

Fig. 2. Construction of $\hat{G}$ and G' from $\overline{G}$ in Fig. 1.

We will prove Theorem 2 by contracting $G[F]$ to the smaller graphs $\overline{G}$, $\hat{G}$ and G'. During the contraction steps, we will ensure that the adjacencies counted in (7) remain the same. Recall that contraction of an edge in a planar graph preserves planarity.

Construction $\overline{G}$. Let $\overline{G}$ be the graph obtained from $G[F]$ by contracting each violated component of X to a single vertex and discarding multiple copies of edges (Fig. 1).

In the following, we will refer to vertices corresponding to violated components of X as *white* vertices. All other vertices are called *black*. Thus, we want to bound the number of edges between white and black vertices in $\overline{G}$ by three times the number of white vertices.

Construction $\hat{G}$. Construct $\hat{G}$ from $\overline{G}$ in the following way (Fig. 2(a)). Let $C \in \mathcal{C}(X)$ be any component that is not violated and has therefore not been contracted. If C is isolated, remove it from $\hat{G}$. If C has a neighbor v in $\overline{G}$, then contract $C \cup \{v\}$ to a black vertex in $\hat{G}$ and identify it with v.

If $\hat{G}$ is not connected, we apply the following arguments to each component. Since we want to bound the number of edges between white and black vertices the claim then follows for the entire graph. Hence, we may assume that $\hat{G}$ is connected.

Lemma 3. *The number of edges between black and white vertices, and the number of white vertices are the same in $\hat{G}$ and $\overline{G}$. Further, $\hat{G}$ has the following properties*:

(a) *it is planar and connected, and*
(b) *removing a black vertex splits the graph into multiple components.*

Proof. Let $C \in \mathcal{C}(X)$ be any component in $\overline{G}$ that is not violated. Since all white vertices in $\overline{G}$ correspond to other components of X, C does not have a white neighbor in $\overline{G}$. It therefore does not affect the number of edges between black and white vertices directly. If C is isolated, there are no connections from C to any white vertex and it can be safely removed. Otherwise, contracting C into a neighboring black vertex maintains the number of edges between black and white vertices.

The number of white vertices remains the same. As observed before, $\overline{G}$ is planar. $\hat{G}$ was constructed from $\overline{G}$ contracting or removing connected components. This preserves planarity. Connectedness follows by the above assumption. Property (b) follows from the minimality of F. $\square$

Construction G'. Construct G' from $\hat{G}$ in the following way (Fig. 2(b)). Until there are no more edges between two black vertices, iteratively perform one of the following two operations:

- Contract an edge between two black vertices with no common white neighbor.
- Remove an edge between two black vertices with a common white neighbor.

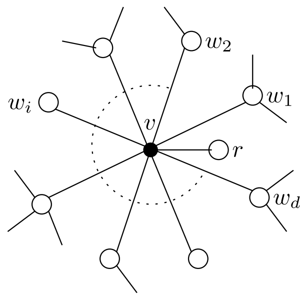


Fig. 3. Earring r in H with adjacent vertex v and its other neighbors $w_1, \dots, w_d$.

For the sequel, the order of these operations can be arbitrary. However, different orderings of these operations can result in different graphs.

Lemma 4. *G' is bipartite and has the same properties stated in Lemma 3 for $\hat{G}$.*

Proof. Edges between white and black vertices are not affected because edges between black vertices with a common white neighbor are not contracted. White vertices remain unchanged.

G' is bipartite since all edges between two black vertices have been contracted or removed. Since white vertices correspond to components of X there are no edges between them in $\overline{G}$ and consequently in $\hat{G}$ and in G'.

G' obeys property (a): contraction along an edge maintains planarity and connectedness. Removing an edge maintains planarity. Since an edge is only removed when the two black vertices have a common white neighbor, the removal also maintains connectedness.

G' obeys property (b): Fix a black vertex v of G'. Due to the above contractions, v is the replacement for a set of black vertices in $\hat{G}$. Due to the minimality of F, deleting this set in $\hat{G}$ splits $\hat{G}$ into components each containing at least one white vertex. Since all white vertices are equivalent in $\hat{G}$ and G', the deletion of v also splits G' into components that each contain at least one white vertex. $\square$

Lemma 5. *Let W and B be the sets of white and black vertices in G', respectively. Then, $|B| \leqslant |W|$.*

Proof. Fix any $r \in W$. Construct a breadth-first search tree T in G' rooted at r. Due to the minimality property (b), all the leafs of T are white. Since each white vertex has at most one black parent in the tree and G' is bipartite, the number of black vertices is at most the number of white vertices. $\square$

Lemma 6. *Given a bipartite graph H on black and white vertices that satisfies properties* (a) *and* (b) (*Lemma* 3)*. Then, there exists a white vertex that is only connected to one black vertex in H. We call such a vertex an* earring*.*

Proof. Consider a breadth-first search tree T and a leaf w of maximum depth with its parent v. Recall that v is black and all its children are white. Due to property (b), v has a child in T that is only connected to v in H; otherwise removing v would leave H connected. $\square$

Lemma 7. *The number of faces in G' is at most the number of white vertices, i.e., $|W|$.*

Proof. The argument proceeds by induction on the number of black vertices. Consider a bipartite graph H that satisfies (a) and (b), and has b black vertices and any number of white vertices. Let $b = 1$. Then, H is a star with at least two white vertices, yielding the induction base. Suppose H has $b + 1$ black vertices. Due to Lemma 6, there exists an earring r in H (Fig. 3). Let v denote its black neighbor. Further, let $(d + 1)$ be the degree of v in H and let $r, w_1, \dots, w_d$ denote the neighbors of v. Since r is an earring, v is adjacent to at most d faces of H.

Let H' denote the graph obtained from H by contracting $v, r, w_1, \dots, w_d$ to a new white vertex identified with r. Clearly, H' has b black vertices and is bipartite, planar and connected, and thus obeys (a). Consider any black vertex $v' \neq v$ in H. Since H obeys (b) we know that removing v' from H splits H into multiple components. Here, $v, r, w_1, \dots, w_d$ are in the same component. Hence, there must be a second component of $H \setminus \{v'\}$ that is also contained in H'. Therefore, v' disconnects r from this component in H' and therefore H' also obeys (b). By induction hypothesis we have that in H' the number of faces is at most the number of white vertices. H has at most d more faces than H' and exactly d more white vertices. Thus, the number of faces in H is at most the number of white vertices in H. $\square$

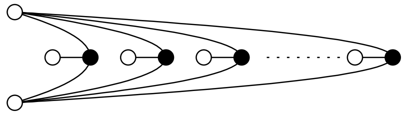


Fig. 4. Example graph that shows that the factor of 3 is best possible in Theorem 2.

We now finish the proof of Theorem 2. Let E' and F' denote the edges and faces of G', respectively. Since G' is a connected planar graph, Euler's formula yields $|E'| = |B| + |W| + |F'| - 2$, where B and W denote the set of black and white vertices, respectively. Due to Lemma 5 and Lemma 7 we have $|B| \leqslant |W|$ and $|F'| \leqslant |W|$ which yields

$$\left|E'\right| = |B| + |W| + \left|F'\right| - 2 \leqslant 3|W| - 2 \leqslant 3|W|. \qquad \square$$

We remark that Demaine et al. [14] gave a bound on the approximation guarantee of the primal-dual algorithm for graphs with forbidden minor.

Corollary 8. *(See [14].) Let G be an H-minor free graph, X be an infeasible solution containing all the terminals and F be any minimal feasible augmentation of X. Then,*

$$\sum_{S \in \mathrm{AVC}(X)} \left|F \cap \Gamma(S)\right| = O\left(m\sqrt{\log m}\right)\left|\mathrm{AVC}(X)\right|, \tag{8}$$

where m is the number of vertices in H.

If G is H-minor free then by construction $\overline{G}$, $\hat{G}$ and G' are H-minor free. This implies that the number of edges in G' is $O((|B| + |W|)m\sqrt{\log m})$ where m is the number of vertices in H (see [14] for references). Given a bound on $|B|$ in terms of $|W|$ then gives Corollary 8. Demaine et al. [14, Lemma 3, p. 337] showed a bound of $|B| \leqslant 2|W|$. Using the bound from Lemma 5 reduces the constant in Corollary 8, and hence the approximation guarantee of the primal-dual algorithm, by a factor of $2/3$.

4. Tightness of analysis and lower bounds

The following example shows that the factor of 3 is best possible in Theorem 2. Consider the graph in Fig. 4 as an instance of NWST. The white circles indicate terminals. Let X be the set of all terminals. Clearly X is infeasible and there is only one feasible minimal augmentation F. This gives

$$\sum_{S \in \mathrm{AVC}(X)} \left|F \cap \Gamma(S)\right| = \sum_{v \in X} \left|F \cap \Gamma(\{v\})\right| = 3\left(|X| - 2\right).$$

Hence, the factor of 3 is tight when the number of terminals grows. This implies that proving a better upper bound for the algorithm, if possible, would require a different proof strategy than Theorem 1.

An example where the algorithm outputs a solution of 2 times the cost of the optimal solution is given in [28]. We remark that such an example can also be obtained easier. Create a circle of k terminals, subdivide every edge by putting a vertex of cost 2 on it, and create a central vertex of cost $k + \epsilon$ that is connected with all terminals. The algorithm will return the circle at a cost of $2(k - 1)$, whereas the optimal solution would be the central vertex at cost $k + \epsilon$. We are not aware of a better lower bound for the algorithm with respect to NWST. However, a better lower bound is known for NWSF [8] that we discuss in Section 7.

We found a striking similarity of the graph in Fig. 4 with the lower bound example for feedback problems on planar graphs [20, Fig. 2, p. 47]. Replace every terminal v in the graph by a (directed) cycle such that every adjacent vertex to v is a member of this cycle. The result is the graph used to show that the approximation factor of 3 is best possible for the generic primal-dual algorithm for some of the feedback problems on planar graphs, e.g., SUBSET FEEDBACK VERTEX SET.

We remark that the reductions of Section 5 do not generalize the lower bound example by Goemans and Williamson [20] for SUBSET FEEDBACK VERTEX SET to NWSF. This is because the primal-dual algorithm on the respective LP formulations of the two problems can produce different results on the original and reduced instance.

5. Reduction of feedback to node-weighted Steiner problems on planar graphs

The previous section suggests a connection between the feedback problems studied in [20] and our problem. We will show that on planar graphs

1. FEEDBACK VERTEX SET reduces to NODE-WEIGHTED STEINER TREE, and
2. SUBSET FEEDBACK VERTEX SET reduces to NODE-WEIGHTED STEINER FOREST.

In particular, this implies that the approximation guarantee of the primal-dual algorithm for NWSF generalizes to the feedback problems, as already proved separately in [20]. Further, we show that the polyhedra given by the linear programming relaxations of the respective problems are essentially equal. This is conform with the fact that the primal-dual algorithm can also be directly applied to the feedback problems circumventing the reductions.

Feedback Vertex Set (FVS) is the following problem: Given an undirected graph $G = (V, E)$ with non-negative weight function w defined on the vertices, select a set of vertices $X \subseteq V$ such that $G[V \setminus X]$ is acyclic, *i.e.*, hit all cycles.

In Subset Feedback Vertex Set (SFVS) we are additionally given a set of special vertices and the task is to hit only cycles that pass through special vertices. Here, we consider the restrictions to planar graphs only.

Theorem 9. *Let $G = (V, E)$ be a plane graph and $w : V \to \mathbb{R}^+$. Let G' be a copy of G, $T = \emptyset$. For each face f of G create a vertex t in G', connect it to all vertices on f and add t to T. Let $w'(t) = 0$ for all $t \in T$ and $w'(v) = w(v)$ otherwise.*

Then, for any solution X to Feedback Vertex Set *on (G, w), $X \cup T$ is a solution to* Node-Weighted Steiner Tree *on (G', w', T) and vice versa,*

Proof. First, let $X \cup T$ denote the solution to NWST on (G', w', T). Assume X is not a feasible solution to FVS on (G, w), *i.e.*, there is a cycle in $G[V \setminus X]$. This cycle partitions the faces of G into two disjoint sets F_1 and F_2. Consequently, the terminals associated with the faces of F_1 and F_2 are not connected by X which contradicts the feasibility of $X \cup T$.

For the converse, let $X \subseteq V$ be the solution to FVS on (G, w). Assume $X \cup T$ is not a solution to NWST on (G', w', T). Then, there exist two terminals t_1 and $t_2 \in T$ that are not connected. Let F_1 and F_2 be the faces of $G[V \setminus X]$ including the associated faces of t_1 and t_2, respectively. Since there is no connection between t_1 and t_2 in G', F_1 and F_2 are disjoint. In particular, this means that X does not include any of the vertices on the perimeter of F_1 and F_2. Since these vertices induce cycles in G, this is a contradiction to the fact that X is a feasible solution to FVS. □

Theorem 9 shows that the integral solutions to FVS and NWST are equivalent. Next, we derive a stronger statement, namely, that the polyhedra of the linear programming relaxations for the two problems are equivalent. Let Cyc denote the set of cycles in G. Then, the linear programming relaxation of Feedback Vertex Set is the problem

$$\text{minimize} \quad \sum_{v \in V} w(v) x_v \tag{9}$$

$$(\text{FVS}) \qquad \text{subject to} \quad \sum_{v \in C} x_v \geqslant 1 \quad \forall C \in \text{Cyc} \tag{10}$$

$$x_v \geqslant 0 \quad \forall v \in V. \tag{11}$$

Further, let $\text{Cut} = \{S \subseteq V \cup T\colon\ |S \cap T| \geqslant 1,\ |(V \setminus S) \cap T| \geqslant 1\}$ denote the set of all terminal cuts in G'. Then, the linear programming relaxation of Node-Weighted Steiner Tree is the problem

$$\text{minimize} \quad \sum_{v \in V \cup T} w(v) x_v \tag{12}$$

$$\text{subject to} \quad \sum_{v \in \Gamma(S)} x_v \geqslant 1 \quad \forall S \in \text{Cut} \tag{13}$$

$$x_v \geqslant 0 \quad \forall v \in V \cup T. \tag{14}$$

Theorem 10. *Under the assumptions of Theorem 9, a solution $x \in \mathbb{R}^{|V|}$ to* (FVS) *is feasible if and only if*

$$x'_v = \begin{cases} x_v, & v \in V \\ 1, & v \in T \end{cases}$$

is feasible for (NWST).

Proof. Instead of looking at the feasible solutions, we show that the constraints in (FVS) and (NWST) are the same. First, consider a cycle $C \in \text{Cyc}$ and the respective constraint (10). The cycle partitions the plane in two regions F_1 and F_2. Construct the set S as all the vertices in G' in the interior of F_1 (excluding the vertices from C; including all terminals t associated with the faces in F_1). Clearly, S is a terminal cut in G' since there is at least one face of G in F_1, hence at least one terminal in S, and at least one face in F_2, hence at least one terminal outside S. Since C consists of all the neighbors of S in the NWST instance the constraint (13) for S has the same coefficients as the constraint (10) for C.

Consider a terminal cut $S \in \text{Cut}$ and the respective constraint (13). We can assume w.l.o.g. that $\Gamma(S) \cap T = \emptyset$, otherwise the constraint is fulfilled automatically since all terminals are included in the solution x'. We show that $\Gamma(S)$ has an induced cycle concluding that there is an associated constraint (10) on a subset of the coefficients. By the above, this associated constraint exists also in (NWST) and therefore the constraint for S is either redundant or equivalent to this associated constraint. Assume, $\Gamma(S)$ does not have an induced cycle, *i.e.*, is a forest. Recall that $\Gamma(S)$ is a cut separating at least one terminal inside S and at least one outside S. Since there is a vertex in G' in every face of G, two vertices in G' cannot be separated by a forest consisting only of vertices in V. Hence, $\Gamma(S)$ has an induced cycle. □

Note that it seems unlikely that SFVS can be reduced to NWST as well by creating terminals in the faces adjacent to special vertices. This is because special vertices could be far apart from each other in the graph and the solution might therefore be locally bounded around the special vertices, not inducing a connected solution to NWST. However, SFVS can be reduced to NWSF by creating terminals in the faces adjacent to special vertices and only requiring that the terminals around the *same* special vertex have to be connected to each other. We give the formal reduction below.

Theorem 11. *Let $G = (V, E)$ be a plane graph, $w : V \to \mathbb{R}^+$ and $S \subseteq V$ be a set of special vertices. Let G' be a copy of G, $N = \emptyset$ and $P = \emptyset$. For each face f in G create a new vertex n, connect it to all vertices on f and add it to N. Further, for each special vertex $s \in S$ (that is at least adjacent to two faces) do the following: Let $f_1, \dots, f_k$ be the faces adjacent to s and $n_1, \dots, n_k$ be the new vertices associated with these faces. Add the terminal pairs $(n_1, n_2), \dots, (n_1, n_k)$ to P. Let $w'(n) = 0$ for all $n \in N$ and $w'(v) = w(v)$ otherwise.*

Then, for any solution X to Subset Feedback Vertex Set *on (G, w, S), $X \cup N$ is a solution to* Node-Weighted Steiner Forest *on (G', w', P) and vice versa.*

Proof. The proof is similar to the proof of Theorem 9. Let $X \cup N$ be the solution to the NWSF instance. Assume X is not feasible for the SFVS instance, *i.e.*, there is a cycle through a special vertex s that is not hit. This cycle separates two adjacent faces of s and therefore these faces are not connected in $X \cup N$; a contradiction.

For the converse, let X be a solution to SFVS. Assume $X \cup N$ is not a solution to the respective NWSF instance. Then, there is a terminal pair (t_1, t_i) associated with some special vertex $s \in S$ that is not connected in $X \cup N$. Recall that N is the set of zero weight vertices in G' that represent the faces of G. Since the NWSF solution includes all of N, every separator consists of vertices in V only. However, two vertices in G' can only be separated by a cycle (or any superset of a cycle) of vertices in V. This cycle includes s, otherwise it does not separate t_1 from t_i. Thus, there is a cycle, passing through s, that is not hit by the solution X; a contradiction. □

Again, using the same argumentation as above we can also prove that the polyhedra given by the linear programming relaxations for SFVS and the reduced instance for NWSF are equal. The linear programming relaxation for SFVS is obtained by using (FVS) with Cyc as the set of all cycles passing through the special vertices.

Theorem 12. *Under the assumptions of Theorem 11, a solution $x \in \mathbb{R}^{|V|}$ to* (FVS)*, where* Cyc *denotes the set of cycles passing through special vertices, is feasible if and only if*

$$x'_v = \begin{cases} x_v, & v \in V \\ 1, & v \in N \end{cases}$$

is feasible for (NWSF).

6. An alternative proof of the 3-approximation

Due to the above reductions, the approximation guarantee of 3 for NWSF carries over to FVS and SFVS on planar graphs. In this section we show that also a derivation in the opposite direction is possible: after constructing a certain family of sets we leverage the proof techniques of Goemans and Williamson [20], developed for feedback problems on planar graphs, to yield an alternative, more involved, proof of Theorem 2.

Alternative proof of Theorem 2 adapted from Goemans and Williamson [20]. As before, let X be the infeasible solution containing all the terminals and F be the minimal feasible augmentation of X. Again, construct the graph $\hat{G}$ and let W and B denote the sets of white and black vertices in $\hat{G}$, respectively. Recall that we would like to bound the number of edges between W and B in terms of $|W|$.

A family of sets $\mathcal{S}$ is said to be *laminar* if for any two sets $S_1, S_2 \in \mathcal{S}$ we have $S_1 \subseteq S_2$, $S_2 \subseteq S_1$ or $S_1 \cap S_2 = \emptyset$. For each black vertex $v \in B$ we will select a set S_v of vertices of $\hat{G}$, called the *witness set* for v, with the properties:

(c) For all $v \in B$, S_v contains at least one white vertex.
(d) For all $v \in B$, v is the only neighbor of S_v in $\hat{G}$.
(e) The family $\{S_v\colon\ v \in B\}$ is laminar.

Let $w \in W$ and T be a breadth-first search tree in $\hat{G}$ rooted at w. Denote the subtree of T below a vertex v and including v by $T(v)$. We will use the same notation for the vertex sets of these trees. Recall that, due to property (b), all leafs in T are white. Now, fix a vertex $v \in B$. Clearly, v is neither a leaf nor the root of T and therefore has a parent and at least one child

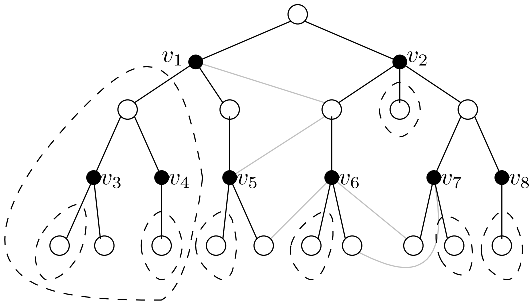


Fig. 5. Construction of S_v's from $\hat{G}$. T is marked with black edges. S_v is marked with dashed lines for all $v \in B$.

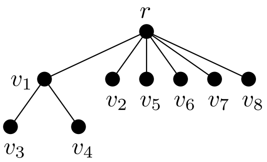


Fig. 6. Induced $\overline{T}$ for the family of S_v's from Fig. 5.

in T. Due to property (b), removing v from $\hat{G}$ splits $\hat{G}$ into multiple components. Let C_1 denote the component containing the parent of v. Let $C_2 \neq C_1$ be any other component of $\hat{G} \setminus \{v\}$. Set $S_v = C_2$ (see Fig. 5).

Since $T \setminus T(v)$ is connected and the parent of v is in C_1 we have $T \setminus T(v) \subseteq C_1$. Hence, C_2 is the union of subtrees of some children of v in T. Since all the leafs of T are white, (c) follows. Further, v is the only neighbor of S_v in $\hat{G}$ because S_v is a component of $\hat{G} \setminus \{v\}$, which gives (d). To show (e) consider any other vertex $v' \in B$, $v' \neq v$. If neither v is a predecessor of v' in T nor v' a predecessor of v, then $S_v \subseteq T(v)$ and $S_{v'} \subseteq T(v')$ are disjoint. W.l.o.g. let v be the predecessor of v'. If $v' \in S_v$, then $S_{v'} \subseteq S_v$ since $S_{v'}$ is the union of subtrees of some children of v' which are all in S_v. If $v' \notin S_v$, then $S_v \cap T(v') = \emptyset$ since $T(v')$ is connected and therefore entirely contained in a component of $\hat{G} \setminus \{v\}$. Since $S_{v'} \subseteq T(v')$, S_v and $S_{v'}$ are disjoint. Hence, (e) holds.

The proof now continues with the arguments from Goemans and Williamson [20, pp. 45–47]. Intuitively, the idea is to count the number of edges of some special bipartite subgraphs of $\hat{G}$ and thereby double counting all black vertices. Replace "witness cycles" by witness sets: the property that is exploited by the proof is property (d). We have shown in (e) that our family of witness sets is laminar. Property (c) is equivalent to the "Minimal Cycle Property 2" [20, top of p. 46]. The "Minimal Cycle Property 1" does not make sense in our context since the violated components of X are the inclusion-wise minimal sets with violated constraints (2) and are therefore contracted to (white) vertices in $\hat{G}$. We outline the adapted proof for completeness.

The laminar family $\{S_v\colon v \in B\}$ can be represented by a forest by considering the partial order induced by inclusion. For ease of exposition, add a black vertex r to B and set S_r to be all the vertices of $\hat{G}$. Clearly, the new family induces a tree $\overline{T}$ where every vertex in $\overline{T}$ represents a set S_v for some $v \in B$. Therefore, we identify the vertices of $\overline{T}$ with their respective vertices in B. The root of $\overline{T}$ is r (see Fig. 6).

Assign white vertices to black vertices in the following way. Let $w \in W$. Assign w to the vertex $v \in B$ corresponding to the (inclusion-wise) smallest set S_v containing w. For a vertex $v \in B$, let W_v denote the white vertices assigned to it. Observe that W_r is non-empty since it contains at least the root of the breadth-first-search tree T.

Now, given a $v \in B$ we would like to bound the number of edges having an endpoint in W_v. Denote for any $w \in W$ the degree of w in $\hat{G}$ by $\deg(w)$, *i.e.*, the number of edges incident with w. Hence, we would like to bound $\sum_{w \in W_v} \deg(w)$.

Consider a vertex $v \in B$ that is neither r nor a leaf of $\overline{T}$. Let $\overline{\deg}(v)$ be the degree of v in $\overline{T}$, $c_1, \ldots, c_{\overline{\deg}(v)-1}$ be the children of v in $\overline{T}$, and $w \in W_v$ (see Fig. 7 for an illustration). Due to (d), v is the only neighbor of S_v. Thus, w cannot have an edge to any vertex outside S_v. Further, w is first included in S_v and can therefore only have edges to $c_1, \ldots, c_{\overline{\deg}(v)-1}$. There are no edges between w and vertices in S_{c_i} because c_i is the only neighbor of this set. Thus, all the edges of w are included in the bipartite graph induced by W_v and $v, c_1, \ldots, c_{\overline{\deg}(v)-1}$. The number of edges in a simple bipartite planar graph is at most twice the number of vertices minus four, unless the graph consists of a single vertex or of two vertices with one edge. Here, the bipartite graph has at least two black vertices, v and c_1, and therefore

$$\sum_{w \in W_v} \deg(w) \leqslant 2|W_v| + 2\,\overline{\deg}(v) - 4. \tag{15}$$

Consider $v \in B$ that is a leaf of $\overline{T}$. Then, $\overline{\deg}(v) = 1$. Due to the construction of S_v we know that S_v consists of exactly one white vertex: S_v does not contain any other black vertices because v is a leaf in $\overline{T}$. Therefore S_v contains only white

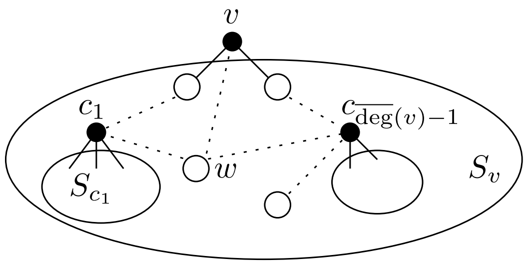


Fig. 7. Argumentation when $v \in B$ is neither r nor a leaf of $\overline{T}$.

vertices. Each of these white vertices become a component in $\hat{G} \setminus \{v\}$ because they have no edges other than to v. Since S_v contains only one such component, S_v is a singleton. Since v is the only neighbor of S_v we have

$$\sum_{w \in W_v} \deg(w) = 1 = 2|W_v| + 2\,\overline{\deg}(v) - 3.$$

Consider $v = r$. Since there is at least one black vertex in $\hat{G}$, $\overline{\deg}(r) \geqslant 1$. Let $c_1, \ldots, c_{\overline{\deg}(r)}$ be the children of r in $\overline{T}$. Analogously to the above, all edges incident with vertices in W_r are contained in the induced bipartite graph of W_r and $c_1, \ldots, c_{\overline{\deg}(r)}$. Note that there are no edges between W_r and r since we added r artificially. In contrast to the above, this bipartite graph can be an edge on two vertices which gives the slightly weaker inequality

$$\sum_{w \in W_r} \deg(w) \leqslant 2|W_v| + 2\,\overline{\deg}(r) - 3.$$

Summing over all black vertices gives

$$\sum_{w \in W} \deg(w) \leqslant 2|W| + 2\sum_{v \in B} \overline{\deg}(v) - 4|B| + \ell + 1,$$

where ℓ is the number of leafs in $\overline{T}$. The sum of degrees in a tree is equal to twice the number of vertices of the tree minus two. Since $|B|$ is the number of vertices in $\overline{T}$,

$$\sum_{w \in W} \deg(w) \leqslant 2|W| + \ell - 3.$$

Since every S_v contains at least one white vertex, and this holds in particular for the leafs v of $\overline{T}$, we have $\ell \leqslant |W|$ which completes the proof

$$\sum_{w \in W} \deg(w) \leqslant 3|W| - 3 \leqslant 3|W|. \qquad \square$$

7. Addendum to the conference version

The primal-dual algorithm heavily depends on the oracle used to specify which dual variables should be increased. An earlier version of this paper [28] devised an improved oracle and claimed an approximation guarantee of 9/4 for the resulting algorithm. Unfortunately, the proof has a flaw and the result is incorrect.

The following counter-example is due to Berman and Yaroslavtsev [8] for the primal-dual algorithm for SFVS. Using the reduction of Section 5 it can be generalized to NWSF. Consider the graph depicted in Fig. 8 as an instance of SFVS. Every intersection of two lines constitutes a vertex. The vertices indicated by white cycles are special. The weight of any black vertex is 3. The weight of a vertex that is not black is its degree plus some negligible $\epsilon > 0$. Now, use the reduction of Section 5 to transfer this instance into an instance of NWSF. Effectively, the reduction places a vertex of weight 0 into each face and with the exception of the vertex on the outer face all these new vertices appear in some terminal pairs.

After initialization, the algorithm will raise the dual variables corresponding to all the interior faces, *i.e.*, to all singular sets consisting of only one terminal. Note that there are no two terminals that surround a third one. The variables will be raised by 1 and then all black vertices become tight. In subsequent iterations, they will all be added to the solution after which the algorithm terminates. The cost of the obtained solution is therefore $18k$ if the graph has k interior gadgets.

However, selecting only the vertices indicated by white cycles also gives a solution to the NWSF problem. This solution induces a weight of $k(7 + 3\epsilon) + 2(2 + \epsilon)$. Therefore, the algorithm only achieves an approximation guarantee of 18/7 for small ϵ and large k.

We point out the flaw in the proof. The proof considers three cases: (1) no two white vertices surround a third one and no two white vertices have a common neighbor, (2) no two white vertices surround a third one but might have common

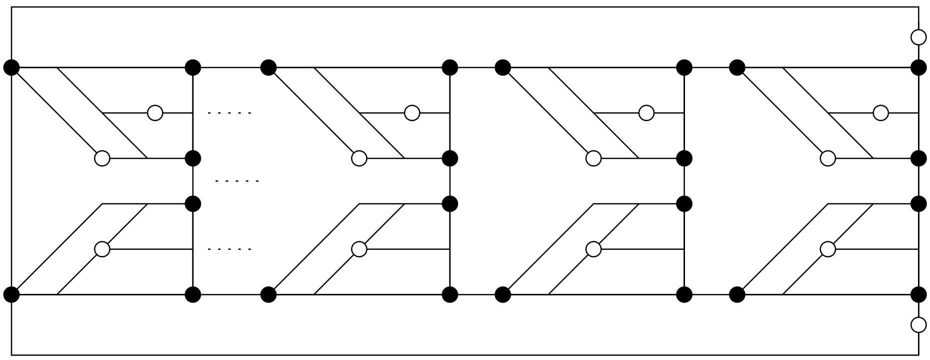


Fig. 8. Counter-example for the 9/4-approximation for SFVS and NWSF on planar graphs.

neighbors, and (3) the oracle selected some white vertices that were surrounded by others. The proof reduces the second case to the first using a defective induction.

We remark that the same flaw exists in the 9/4-approximation result for the feedback problems in [20] and the above example is a counter-example for SFVS.

Furthermore, the above example for NWSF constitutes also a lower bound for the generic primal-dual algorithm for NWSF of 18/7.

8. Open problems

We presented the standard integer linear programming formulation for NWSF in Section 1.1. The integrality gap for this formulation is lower bounded by 2: consider a cycle with alternating terminals and non-terminals. Our result implies an upper bound of 3. However, the precise value of the integrality gap is unknown (presumably 2). An interesting question is if the upper bound of 3 on the primal-dual algorithm can be improved. This might necessitate the use of a different oracle. Since the analyses of the primal-dual method for NWSF and the feedback problems from [20] are very similar, improving the upper bound is essentially as hard as improving on the integrality gap for the feedback problems. The latter is an open question since 1996.

Note that a popular candidate for improving the approximation factor is the iterative rounding technique [24]. However, there exist graphs with a basic solution to (NWSF) that assigns values of 1/3 to every non-terminal: Use the complete graph on 4 vertices K_4, create a terminal in each face and connect the terminal with all the vertices on the respective face. Hence, the best approximation factor one can hope for with a naive iterative rounding technique is 3.

An interesting open question is if there are different integer linear programming formulations of NWSF, than the undirected cut formulation, that lead to better integrality gaps or better approximation guarantees. Such formulations exist for Steiner Tree (see [10] and references therein) and Feedback Vertex Set [12].

Note that for some special graph classes there exist polynomial time approximation schemes (PTAS) for NWST, e.g. unit disk graphs [27]. However, it is not known if PTAS exist for NWST or even NWSF on planar graphs. The edge weighted versions, *i.e.*, Steiner Tree and Steiner Forest, allow for PTAS [9,6]. Furthermore, the vertex weighted problem Feedback Vertex Set that reduces to NWST also allows for a PTAS [13]. Hence, there might be hope for a PTAS for NWST on planar graphs.

Acknowledgments

The author would like to thank Piotr Berman and Grigory Yaroslavtsev for providing the counter-example of Section 7, and Amir Abboud for helpful discussions.

Appendix A. Proof of Theorem 1

Proof of Theorem 1. Let X and $\mathbf{y}$ be the solutions constructed by the algorithm. Clearly, X induces a feasible solution to (NWSF). Further, $\mathbf{y}$ is a feasible solution to (DNWSF). This is because after a vertex v is added to X, no variable y_S occurring in v's packing constraint (5) will be changed since the constraint (2) for S is satisfied.

Let v be a terminal vertex. Recall that $w(v) = 0$. Since the algorithm initializes with all terminals in step (i) no variable y_S corresponding to a set S with $v \in \Gamma(S)$ is ever raised. Hence, $w(v) = 0 = \sum_{S \in \mathcal{T}: v \in \Gamma(S)} y_S$. Due to step (iiia) of the algorithm, we have that for all non-terminals $v \in X$ the packing constraint (5) is also tight. Thus,

$$\sum_{v \in X} w(v) = \sum_{v \in X} \sum_{S \in \mathcal{T}: v \in \Gamma(S)} y_S.$$

Changing the order of summation yields

$$\sum_{v\in X} w(v) = \sum_{S\in\mathcal{T}} \sum_{v\in X\cap\Gamma(S)} y_S = \sum_{S\in\mathcal{T}} \big|X\cap\Gamma(S)\big| y_S.$$

Let X_j be the solution of the algorithm after the jth iteration of the while loop (iii). Let $\mathcal{V}_j = \mathrm{VIOL}(X_j)$ and let ϵ_j denote the increase of the dual variables corresponding to $\mathcal{V}_j$ in iteration j. Then, $y_S = \sum_{j:S\in\mathcal{V}_j} \epsilon_j$ for all S. This gives

$$\sum_{S\in\mathcal{T}} y_S = \sum_{j\geqslant 1} |\mathcal{V}_j|\epsilon_j$$

and

$$\sum_{v\in X} w(v) = \sum_{S\in\mathcal{T}} \big|X\cap\Gamma(S)\big| \sum_{j:S\in\mathcal{V}_j} \epsilon_j = \sum_{j\geqslant 1} \Bigg(\sum_{S\in\mathcal{V}_j} \big|X\cap\Gamma(S)\big| \Bigg) \epsilon_j.$$

Due to the reverse delete step (iv), we have that $X \cup X_j$ is a minimal augmentation for X_j. Clearly, it includes X_j. Further, since the delete step considers the vertices in the reverse order they were added to X, every vertex $v \in X \setminus X_j$ is considered for deletion before all the vertices in X_j. Thus, for every vertex $v \in X \setminus X_j$ we have that $(X \setminus X_j) \setminus \{v\}$ is infeasible. Hence, the precondition of the theorem can be applied

$$\sum_{S\in\mathcal{V}_j} \big|(X\cup X_j)\cap\Gamma(S)\big| \leqslant \gamma|\mathcal{V}_j|.$$

Together with $|X\cap\Gamma(S)| \leqslant |(X\cup X_j)\cap\Gamma(S)|$ this yields

$$\sum_{v\in X} w(v) \leqslant \sum_{j\geqslant 1} \Bigg(\sum_{S\in\mathcal{V}_j} \big|(X\cup X_j)\cap\Gamma(S)\big| \Bigg) \epsilon_j \leqslant \sum_{j\geqslant 1} \gamma|\mathcal{V}_j|\epsilon_j = \gamma \sum_{S\in\mathcal{T}} y_S.$$

The claim now follows since **y** is a feasible solution to the dual of the LP relaxation. □

References

[1] Ajit Agrawal, Philip Klein, R. Ravi, When trees collide: An approximation algorithm for the generalized Steiner problem on networks, SIAM Journal on Computing 24 (1995) 440–456.

[2] Spyros Angelopoulos, The node-weighted Steiner problem in graphs of restricted node weights, in: Algorithm Theory – SWAT 2006, vol. 4059, 2006, pp. 208–219.

[3] Vineet Bafna, Piotr Berman, Toshihiro Fujito, Constant ratio approximations of the weighted feedback vertex set problem for undirected graphs, Algorithms and Computations 1004 (1995) 142–151.

[4] Reuven Bar-Yehuda, Keren Bendel, Ari Freund, Dror Rawitz, Local ratio: A unified framework for approximation algorithms. In memoriam: Shimon Even 1935–2004, ACM Computing Surveys 36 (2004) 422–463.

[5] MohammadHossein Bateni, Chandra Chekuri, Alina Ene, MohammadTaghi Hajiaghayi, Nitish Korula, Dániel Marx, Prize-collecting Steiner problems on planar graphs, in: 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, 2010.

[6] MohammadHossein Bateni, MohammadTaghi Hajiaghayi, Dániel Marx, Approximation schemes for Steiner forest on planar graphs and graphs of bounded treewidth, in: 42nd ACM Symposium on Theory of Computing, 2010, pp. 211–220.

[7] Ann Becker, Dan Geiger, Approximation algorithms for the loop cutset problem, Journal of Artificial Intelligence Research (1994) 60–68.

[8] Piotr Berman, Grigory Yaroslavtsev, Personal communication, 2012.

[9] Glencora Borradaile, Philip Klein, Claire Mathieu, An O($n \log n$) approximation scheme for Steiner tree in planar graphs, ACM Transactions on Algorithms 5 (3) (2009) 1–31.

[10] Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoß, Laura Sanità, An improved LP-based approximation for Steiner tree, in: 42nd ACM Symposium on Theory of Computing, 2010, pp. 583–592.

[11] Miroslav Chlebík, Janka Chlebíková, The Steiner tree problem on graphs: Inapproximability results, Theoretical Computer Science 406 (2008) 207–214.

[12] Fabián A. Chudak, Michel X. Goemans, Dorit S. Hochbaum, David P. Williamson, A primal-dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs, Operations Research Letters 22 (4–5) (1998) 111–118.

[13] Erik D. Demaine, MohammadTaghi Hajiaghayi, The bidimensionality theory and its algorithmic applications, The Computer Journal 51 (3) (2008) 292–302.

[14] Erik D. Demaine, MohammadTaghi Hajiaghayi, Philip N. Klein, Node-weighted Steiner tree and group Steiner tree in planar graphs, in: 36th International Colloquium on Automata, Languages and Programming, in: LNCS, vol. 5555, 2009, pp. 328–340.

[15] Bistra Dilkina, Carla P. Gomes, Solving connected subgraph problems in wildlife conservation, in: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 2010, pp. 102–116.

[16] Guy Even, Joseph (Seffi) Naor, Leonid Zosin, An 8-approximation algorithm for the subset feedback vertex set problem, in: 37th Annual Symposium on Foundations of Computer Science, 1996, pp. 310–319.

[17] Uriel Feige, A threshold of $\ln n$ for approximating set cover, Journal of the ACM 45 (1998) 634–652.

[18] Michael R. Garey, David S. Johnson, The rectilinear Steiner tree problem is NP-complete, SIAM Journal on Applied Mathematics 32 (4) (1977) 826–834.

[19] Michel X. Goemans, David P. Williamson, The primal-dual method for approximation algorithms and its application to network design problems, in: Dorit S. Hochbaum (Ed.), Approximation Algorithms for NP-hard Problems, PWS Publishing Co., Boston, MA, USA, 1997, pp. 144–191 (Chapter 4).

[20] Michel X. Goemans, David P. Williamson, Primal-dual approximation algorithms for feedback problems in planar graphs, Combinatorica 18 (1998) 37–59.

[21] Sudipto Guha, Samir Khuller, Improved methods for approximating node weighted Steiner trees and connected dominating sets, Information and Computation 150 (1) (1999) 57–74.

[22] Sudipto Guha, Anna Moss, Joseph Seffi Naor, Baruch Schieber, Efficient recovery from power outage (extended abstract), in: 31st ACM Symposium on Theory of Computing, 1999, pp. 574–582.

[23] Anupam Gupta, Jochen Könemann, Approximation algorithms for network design: A survey, Surveys in Operations Research and Management Science 16 (1) (2011) 3–20.

[24] Kamal Jain, A factor 2 approximation algorithm for the generalized Steiner network problem, Combinatorica 21 (1) (2001) 39–60.

[25] Richard M. Karp, Reducibility among combinatorial problems, Complexity of Computer Computations (1972) 85–103.

[26] Philip Klein, R. Ravi, A nearly best-possible approximation algorithm for node-weighted Steiner trees, Journal of Algorithms 19 (1) (1995) 104–115.

[27] Xianyue Li, Xiao-Hua Xu, Feng Zou, Hongwei Du, Pengjun Wan, Yuexuan Wang, Weili Wu, A PTAS for node-weighted Steiner tree in unit disk graphs, Combinatorial Optimization and Applications 5573 (2009) 36–48.

[28] Carsten Moldenhauer, Primal-dual approximation algorithms for node-weighted Steiner forest on planar graphs, in: 38th International Colloquium on Automata, Languages and Programming, in: LNCS, vol. 6755, 2011, pp. 748–759.

[29] Anna Moss, Yuval Rabani, Approximation algorithms for constrained node weighted Steiner tree problems, SIAM Journal on Computing 37 (2) (2007) 460–481.

[30] Jan Remy, Angelika Steger, Approximation schemes for node-weighted geometric Steiner tree problems, in: Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, vol. 3624, 2005, pp. 642–642.

[31] David P. Williamson, David B. Shmoys, The Design of Approximation Algorithms, Cambridge University Press, 32 Avenue of the Americas, New York, NY 10013-2473, USA, 2011.