An Extension of the Nemhauser&Trotter Theorem to Generalized Vertex Cover with Applications

Reuven Bar-Yehuda[1], Danny Hermelin[2], and Dror Rawitz[3]

[1] Department of Computer Science,
Technion IIT, Haifa 32000, Israel.
reuven@cs.technion.ac.il

[2] Department of Computer Science,
University of Haifa, Haifa 31905, Israel.
danny@cri.haifa.ac.il

[3] School of Electrical Engineering,
Tel-Aviv University, Tel-Aviv 69978, Israel.
rawitz@eng.tau.ac.il

Abstract. The Nemhauser&Trotter Theorem provides an algorithm which is frequently used as a subroutine in approximation algorithms for the classical Vertex Cover problem. In this paper we present an extension of this theorem so it fits a more general variant of Vertex Cover, namely the Generalized Vertex Cover problem, where edges are allowed not to be covered at a certain predetermined penalty. We show that many applications of the original Nemhauser&Trotter Theorem can be applied using our extension to Generalized Vertex Cover. These applications include a $(2 - 2/d)$-approximation algorithm for graphs of bounded degree d, a PTAS for planar graphs, a $(2 - \lg \lg n/2 \lg n)$-approximation algorithm for general graphs, and a $2k$ kernel for the parameterized Generalized Vertex Cover problem.

1 Introduction

Given a graph $G = (V, E)$ with vertex weights, the classical Vertex Cover problem asks to find a minimum weight subset of vertices $S \subseteq V$ that covers all edges in G, *i.e.* a subset S with $S \cap e \neq \emptyset$ for all $e \in E$. The Vertex Cover problem is one of the most well-studied problems in theoretical computer science and discrete mathematics in general, a study dating back to König's classical early 1930s result [17], and probably even prior to that. In 1972, Karp listed the decision version of Vertex Cover in his famous list of initial twenty-one NP-complete problems [15].

In 1975, only three years after Karp's famous NP-complete list, Nemhauser and Trotter published their seminal paper [19] which gave the first 2-approximation algorithm for Vertex Cover. In fact, their algorithm gave much more than that: It essentially gave a reduction that reduces the problem of finding a vertex cover in an arbitrary graph G, to that of finding a vertex cover in a subgraph of G whose total weight is not much more than the weight of any of its vertex covers. This by itself gives a 2-approximation since one can take the entire subgraph as a solution, but more then that, it adds additional structure to the Vertex Cover problem in general. Indeed, after applying the Nemhauser&Trotter reduction, one can use the total weight of the graph as a yardstick for analyzing approximate solutions, rather than use the weight of the optimal solution of which there is rarely any knowledge of. Below is a precise statement of the Nemhauser&Trotter Theorem:

Theorem 1 (Nemhauser&Trotter [19]). *Let (G, w) be an instance of* Vertex Cover*, with $G = (V, E)$ and $w : V \to \mathbb{Q}^{\geq 0}$. Then there is a polynomial-time algorithm that partitions the vertices of G into three subsets, V_1, V_0, and $V_{1/2}$, such that:*

(i) if $S_{1/2}$ is an α-approximate solution for $(G[V_{1/2}], w)$, then $V_1 \cup S_{1/2}$ is an α-approximate solution for (G, w), for all $\alpha \geq 1$, and
(ii) the total w-weight of any vertex cover in $G[V_{1/2}]$ is at least half of the total w-weight of all vertices in $V_{1/2}$.

The first condition of the theorem implies that we can restrict our attention to $G[V_{1/2}]$, ignoring vertices of V_1 and V_0 in G. The second condition of the theorem implies that finding a vertex cover of $G[V_{1/2}]$ that excludes some vertices of $V_{1/2}$ is guaranteed to give a vertex cover with weight strictly less then twice the optimum in G. This is the central idea behind Hochbaum's $(2-2/d)$-approximation algorithm for graphs of bounded degree d [12], and also behind the $(2 - \lg\lg n/2\lg n)$-approximation algorithm for general graphs given in [2]. In fact, many known approximation algorithms for Vertex Cover and its special cases use the Nemhauser&Trotter Theorem as a subroutine. Two other good examples are the PTASs of Lipton-Tarjan [18] and Baker [1] for Vertex Cover in planar graphs[1], where one finds an optimal solution in a large fraction of the graph, and adds all remaining vertices to get a solution for the entire graph. We mention also Chen *et al.* [7] who observed that the Nemhauser&Trotter Theorem gives a $2k$ kernel for the parameterized variant of Vertex Cover in the parameterized complexity setting, when the parameter taken is the total weight of required vertex cover (see also [8]).

In this paper we focus on a natural generalization of Vertex Cover which can be thought of as the "prize-collecting" version of the problem. In this variant, each edge in the given graph is allowed to be left uncovered at a certain predetermined penalty. Thus, the input now consists of a graph with vertex and edge weights, and the goal is to minimize the total weight of vertices

[1] Baker's algorithm [1] originally did not use the Nemhauser&Trotter algorithm, however, adding it as a preprocessing step makes the analysis of the algorithm somewhat more simple.

selected to a solution, plus the total weight of edges not covered by the solution. Observe that this is in fact a generalization of Vertex Cover, since we return to the original problem by setting all edge weights to ∞. We call this generalization of Vertex Cover, the Generalized Vertex Cover problem:

Generalized Vertex Cover:
Instance: A graph $G = (V, E)$ and a weight function $w : V \cup E \to \mathbb{Q}^{\geq 0}$.
Solution: A subset S of V.
Measure: $cost(S) = \sum_{v \in S} w(v) + \sum_{e \in E, e \cap S = \emptyset} w(e)$.

The first to consider Generalized Vertex Cover was Hochbaum [13], who gave a 2-approximation algorithm for the problem. The time complexity of this algorithm was later improved in [3], where an d-approximation algorithm for Generalized Vertex Cover in d-hypergraphs is given as well. We further note that other "prize collecting covering" problems were also studied extensively in the literature. This includes the paper by Hassin and Tamir [11] who considered the prize collecting variant of the Facility Location on the Real Line problem, and the work of Goemans and Williamson [10] who presented approximation algorithms for the prize collecting versions for Triangle Inequality Traveling Salesman and Steiner Tree. See also [6, 14] for other prize collecting facility location problems.

Due to the importance that the Nemhauser&Trotter Theorem plays in designing approximation algorithms for Vertex Cover and its special cases, a natural question to ask is whether a similar theorem can be found for Generalized Vertex Cover. Observe that the theorem does not carry-on immediately to the more general case due to the different way that the edges now come into play; in fact, this poses a difficulty even in stating the theorem for the more general case. The main result of this paper overcomes these difficulties and gives an affirmative answer to the question above by proving a slightly different variant of the Nemhauser&Trotter Theorem, which is essentially the same for most algorithmic applications. The following is a precise statement of our result:

Theorem 2. *Let (G, w) be an instance of* Generalized Vertex Cover*, with $G = (V, E)$ and $w : V \cup E \to \mathbb{Q}^{\geq 0}$. Then there is a polynomial-time algorithm that partitions the vertices of G into three subsets, V_1, V_0, and $V_{1/2}$, and constructs another weight function $\widetilde{w} : V \cup E \to \mathbb{Q}^{\geq 0}$, such that:*

(i) if $S_{1/2}$ is an α-approximate solution for $(G[V_{1/2}], \widetilde{w})$, then $V_1 \cup S_{1/2}$ is an α-approximate solution for (G, w), for all $\alpha \geq 1$, and
(ii) the cost of any subset of vertices in $(G[V_{1/2}], \widetilde{w})$ is at least half of $\sum_{v \in V_{1/2}} \widetilde{w}(v)$.

Observe the difference in the second condition of the theorem which is necessary since any subset of vertices is a potential solution in Generalized Vertex Cover. This is what makes the proof of the theorem in the generalized case more challenging. Another challenge is that as the edges in Generalized Vertex Cover play a different role, we are not guaranteed the combinatorial structure provided by the original theorem. For instance, in the original theorem the subset V_0 in the partition had to be an independent set, as otherwise any vertex cover had to include at least one vertex from V_0. In our case we can never require such a condition; we must assume that there can be an edge between any pair of vertices in V_0. Furthermore, in the original theorem, it is clear that V_1 must separate V_0 from $V_{1/2}$ in G. Again, in our case this is not necessarily so, which makes the reduction more difficult since we insisted that the resulting subgraph be induced, *i.e.* obtained only by deleting vertices, a fact that allows carrying-on *hereditary* properties of G through the reduction.

With the help of Theorem 2, we can show that most algorithms for VERTEX COVER which use the Nemhauser&Trotter Theorem as a subroutine, can be modified so that they apply also for the more general case. In particular, we obtain the following new results for GENERALIZED VERTEX COVER as almost immediate corollaries of Theorem 2:

1. A $(2-2/d)$-approximation algorithm for graphs of bounded degree d.
2. A PTAS for planar graphs.
3. A $(2-\lg\lg n/2\lg n)$-approximation algorithm for general graphs.
4. A $2k$ kernel for the parameterized version of GENERALIZED VERTEX COVER.

The reader should not be mislead into thinking our results imply that VERTEX COVER and GENERALIZED VERTEX COVER are in fact the same in any graph class. To see that this is not so, note that while VERTEX COVER is polynomial-time solvable in complete graphs, GENERALIZED VERTEX COVER in complete graphs is essentially as hard to approximate as VERTEX COVER in any general graph (and thus it cannot be approximated within $10\sqrt{5}-21 \approx 1.36$, unless P=NP [9]). This can be seen by the following reduction from VERTEX COVER in general graphs to GENERALIZED VERTEX COVER in complete graphs: Given a graph G, transform G into a complete graph G' by adding all necessary edges, and a assign a weight to these edges such that their total weight is substantially smaller than the weight of any vertex in the graph. All original edges are assigned a weight of ∞, and the vertex weights remain the same. It is not difficult to see that any α-approximate vertex cover for G is also an $(\alpha+\varepsilon)$-approximate generalized vertex cover of G', for any $\varepsilon > 0$ as small as we want, and vice versa.

Our work is also related to the recent work of Könemann *et al.* [16], who presented a reduction from partial covering to prize collecting covering, or in our context, from GENERALIZED VERTEX COVER to PARTIAL VERTEX COVER. The PARTIAL VERTEX COVER problem is another natural generalization of VERTEX COVER, where now the goal is to find a minimum weight subset of vertices that covers a prespecified number of edges in the graph. Könemann *et al.* [16] showed how to transform a specific class of α-approximation algorithms for GENERALIZED VERTEX COVER into an $(\frac{4}{3}\cdot\alpha+\epsilon)$-approximation algorithms for PARTIAL VERTEX COVER. The algorithm in our Theorem 2 is actually of this specific class, and so Theorem 2 combined with Könemann *et al.* gives a more refined reduction from GENERALIZED VERTEX COVER to PARTIAL VERTEX COVER.

The rest of the paper is devoted to proving Theorem 2 along with all of its applications mentioned above. In the next section, we discuss some preliminaries necessary for our proof, and in particular we review the local-ratio method which plays an important part in many of our results. In Section 3 we provide all details of the proof of Theorem 2, and in Section 4 we discuss all applications mentioned above. Due to space limitations, the proof of some of the lemmas in the paper are omitted to the appendix.

2 Preliminaries

In this section we discuss notation and previous work that is necessary for presenting our results. In particular, we introduce terminology that will be used for proving Theorem 2, and briefly review the local-ratio technique which we will use throughout the paper.

We will deal with graphs that have weights assigned to their vertices and edges. Let $G=(V,E)$ be a graph given along with a weight function $w: V\cup E \to \mathbb{Q}^{\geq 0}$. For any edge $\{u,v\}\in E$, we write $w(u,v)$ as a shorthand for $w(\{u,v\})$. For a subset of vertices $S\subseteq V$, we let $w(S)=\sum_{v\in S} w(v)$, and for a pair of subsets $S_1,S_2\subseteq V$, we use $w(S_1,S_2)$ for the weight of edges with one end-point in S_1 and one end-point in S_2, *i.e.* $w(S_1,S_2)=\sum_{u\in S_1,v\in S_2} w(u,v)-\sum_{u,v\in S_1\cap S_2} w(u,v)$. Recall that the cost of a subset $S \subseteq V$ in G is $\text{cost}(S) = w(S) + w(V \setminus S, V \setminus S)$. Let $\text{opt}(G, w)$ denote the cost of the optimal generalized vertex cover in (G, w). For an $\alpha > 0$, we say that S is *α-approximate*, if $\text{cost}(S) \leq \alpha \cdot \text{opt}(G, w)$. Also, we call any subset $S \subseteq V$ *feasible*, if it has cost less then ∞.

The Local-Ratio Technique [2] is central in our proof of Theorem 2, and is also used in its applications. The technique in most part is based on the Local-Ratio Lemma, which in our terms can be stated as follows:

Lemma 1 (Local-Ratio [2]). *Let (G, w_1) and (G, w_2) be two instances of* Generalized Vertex Cover*, with $G = (V, E)$ a graph and $w_1, w_2 : V \cup E \to \mathbb{Q}^{\geq 0}$ two weight functions. If $S \subseteq V$ is α-approximate both in (G, w_1), and in (G, w_2), then S is also α-approximate in $(G, w_1 + w_2)$.*

A typical local ratio algorithm is recursive. In each recursive step, the algorithm first collects all zero elements to its solution. It then defines a weight function w_1 in such a way that $w_2 = w - w_1$ still assigns non-negative weights to all elements, and at least one element with non-zero weight with respect to w will get zero weight with respect to w_2. The algorithm then recursively solves the instance of the problem with w_2 as the given weight function, and fixes the returned solution so it will be a good approximate solution with respect to both w_1 and w_2. By the Local-Ratio theorem, this solution is guaranteed to be a good approximation with respect to w as well. The following definition hints on how to select a good initial weight function w_1.

Definition 1 (α-effectiveness). *A weight function w_ε is said to be α-*effective* in G, if when a subset of vertices is feasible with respect to w_ε, it is also α-approximate with respect to w_ε.*

Below we give the a variation of the Local Ratio Lemma which combines the notion of α-effectiveness, and will be the variation that will actually be used in the paper. Its proof is immediate from the Local-Ratio Lemma and the definition of α-effectiveness, and is the left to the reader.

Lemma 2. *Let (G, w) be an instance of* Generalized Vertex Cover*, and let w_ε a weight function which is α-effective in G. If S is an β-approximate solution for $(G, w - w_\varepsilon)$, then S is an $\max\{\alpha, \beta\}$-approximate solution for (G, w).*

3 The Main Proof

In this section we present the central result of this paper, namely, the proof of Theorem 2. Our proof consists of two main steps: In the first, similar to the proof of the original Nemhauser&Trotter Theorem, we obtain an initial partition of the vertices of our graph G into three classes according to an optimal solution for an appropriate bipartite graph constructed from G. However, unlike the original proof, in our case we can have edges between all classes, and inside each class. We will show that the only really problematic edges are those that are between two particular classes. These edges are taken care of in the second step by several applications of the Local-Ratio Lemma, at the end of which we obtain our desired partition of the vertices of G, and the desired weight function $\widetilde{w}$. Before describing both steps in actual detail, we start with the following lemma which will later be used in our proof, but is also of independent interest.

Lemma 3. Generalized Vertex Cover *is polynomial-time solvable in bipartite graphs.*

Proof. Let $B = (V, V', F)$ be a bipartite graph, and let $w : V \cup V' \cup F \to \mathbb{Q}^{\geq 0}$ be a weight function. Construct a flow-network N from B by adding to B a source vertex s and a target vertex t, with s connected to all vertices in V, and vertices in V' connected to t. Define the capacities of the edges in N by:

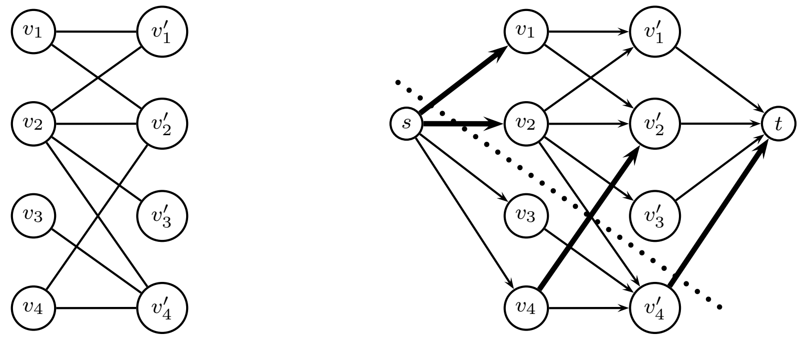

Fig. 1: A bipartite graph and the corresponding network: The dotted line represents an s,t-cut; the thick edges that cross the s,t-cut correspond to the cover $\{v_1, v_2, v_4'\}$.

- $c(s, v) = w(v)$ for all $v \in V$,
- $c(u, v) = w(u, v)$ for all $\{u, v\} \in F$, and
- $c(v, t) = w(v)$ for all $v \in V'$.

Observe that there is a one-to-one correspondence between edges in B and s,t-paths in N, and that the edges on a s,t-path in N correspond to the three ways of covering the corresponding edge in B: Either adding one of its endpoints to the generalized vertex cover, or not covering this edge at all. Specifically, given a subset $U \subseteq V \cup V'$ the corresponding s,t-cut is (S, T), where $S = \{s\} \cup (V \setminus U) \cup (V' \cap U)$ and $T = (V \cup V') \setminus S$, and conversely, given an s,t-cut (S, T) the corresponding cover is $U = (S \cap V') \cup (T \cap V)$ (see Fig. 1.) Hence, there is a one-to-one correspondence between generalized vertex covers in B and s,t-cuts in N. Furthermore, by our selection of capacities, each generalized vertex cover corresponds to an s,t-cut in with equal capacity. Since one can compute minimum s,t-cuts by standard flow techniques, the lemma is proven. □

3.1 Step I

Given an instance (G, w) for Generalized Vertex Cover, with $G = (V, E)$ and $w : V \cup E \to \mathbb{Q}^{\geq 0}$, we construct a bipartite graph $B = (V, V', F)$, along with a weight function $w_B : V \cup V' \cup F \to \mathbb{Q}^{\geq 0}$ for B, as follows: The set V' contains a *duplicate vertex* for each vertex in V, and is defined by $V' = \{v' \mid v \in V\}$. The set F of edges in B includes the pair of edges $\{u, v'\}$ and $\{u', v\}$, for each edge $\{u, v\} \in E$. We define w_B by $w_B(v), w_B(v') = w(v)$ for all $v, v' \in V \cup V'$, and $w_B(u', v), w_B(u, v') = w(u, v)$ for all $\{u, v'\}, \{u', v\} \in F$. Here, and throughout the remainder of this section, we denote by S' the set of duplicates of some subset $S \subseteq V$. That is, $S' = \{v' \mid v \in S\}$.

We next compute an optimal solution S_B^* in B, using the algorithm implied by Lemma 3. According to the computed solution S_B^*, we partition V into the following three subsets:

$$U_1 = \{v \mid v, v' \in S_B^*\}\ , \qquad U_0 = \{v \mid v, v' \notin S_B^*\}\ , \qquad \text{and} \qquad U_{1/2} = V \setminus (U_1 \cup U_0)\ .$$

We first show that we may assume S_B^* does not include any vertices of $U_{1/2}'$.

Claim 1. $\text{cost}_B(U_1 \cup U_1' \cup U_{1/2}) \leq \text{cost}_B(S_B^*)$.

Proof. By a simple manipulation of the cost of S_B^* in (B, w_B), we get:

$$\begin{aligned}\text{cost}_B(S_B^*) &= w_B(S_B^*) + w_B(U_0 \cup U_{1/2} \setminus S_B^*, U_0' \cup U_{1/2}' \setminus S_B^*) \\ &\geq w_B(S_B^*) + w_B(U_0, U_0') + w_B(U_0, U_{1/2}' \setminus S_B^*) + w_B(U_0', U_{1/2} \setminus S_B^*).\end{aligned}$$

Thus, since $w_B(U_0, U'_{1/2} \setminus S^*_B) + w_B(U'_0, U_{1/2} \setminus S^*_B) = w_B(U_0, U'_{1/2})$, we have

$$\begin{aligned}\mathrm{cost}_B(S^*_B) &\geq w_B(S^*_B) + w_B(U_0, U'_0) + w_B(U_0, U'_{1/2} \setminus S^*_B) + w_B(U'_0, U_{1/2} \setminus S^*_B) \\ &= w_B(S^*_B) + w_B(U_0, U'_0) + w_B(U_0, U'_{1/2}) \\ &= \mathrm{cost}_B(U_1 \cup U'_1 \cup U_{1/2}),\end{aligned}$$

and the claim is proven. □

Next, using the simple observation given by Claim 1, we show that we are on the right direction with our initial partition of the vertices of G, since there is an optimal solution which includes all vertices of U_1 and no vertex of U_0.

Claim 2. There is an optimal solution S for (G, w) with $U_1 \subseteq S$ and $U_0 \cap S = \emptyset$.

Proof. Let S be any subset of vertices in G, and let $S_z = U_z \cap S$ for $z \in \{1, \frac{1}{2}, 0\}$. Also, let $T = V \setminus S$ and $T_z = U_z \setminus S_z$ for $z \in \{1, \frac{1}{2}, 0\}$. To prove the claim, we will argue that the solution $U_1 \cup S_{1/2}$ does not have greater cost than S. The claim will then immediately follow by taking S to be optimal.

For this, consider the difference between the cost of S and the cost of $U_1 \cup S_{1/2}$. The only advantage the former has over the latter is that it does not pay for any vertex in T_1, nor for any edge with between S_0 and $U_0 \cup T_{1/2}$, all elements which are paid for by $U_1 \cup S_{1/2}$. However, S has to pay for all vertices in S_0, and all edges between T_1 and T, while $U_1 \cup S_{1/2}$ does not. (See depiction in Figure 2a.) Since this is the only difference between the two solution, we have

$$\mathrm{cost}_G(S) - \mathrm{cost}_G(U_1 \cup S_{1/2}) = w(S_0) + w(T_1, T) - w(T_1) - w(S_0, U_0 \cup T_{1/2}) \ .$$

Now, let us construct a solution S_B for the bipartite graph $B = B(G)$ described above, defined by $S_B = (V \setminus T_0) \cup S'_1$, and let us compare this solution to S^*_B. The first observation is that there is no loss of generality in assuming that $S^*_B = U_1 \cup U_{1/2} \cup U'_1$, *i.e.* that S^*_B does not include any vertices of $U'_{1/2}$. Now S_B does not pay for any vertex in T'_1, while S^*_B does, nor does it pay for any edges between S_0 and $U'_0 \cup U'_{1/2}$, all of which are paid for by S^*_B. On the other hand, S^*_B does not pay for any vertex in S_0, nor for any edge between T'_1 and T_0. Noting that this is the exact difference between their costs, and that all weights are positive, we get

$$\begin{aligned}\mathrm{cost}_B(S_B) - \mathrm{cost}_B(S^*_B) &= w_B(S_0) + w_B(T'_1, T_0) - w_B(T'_1) - w_B(S_0, U'_0 \cup U'_{1/2}) \\ &= w(S_0) + w(T_1, T_0) - w(T_1) - w(S_0, U_0 \cup U_{1/2}) \\ &= w(S_0) + w(T_1, T) - w(T_1, T_1 \cup T_{1/2}) \\ &\quad - w(T_1) - w(S_0, U_0 \cup T_{1/2}) - w(S_0, S_{1/2}) \\ &= \mathrm{cost}_G(S) - \mathrm{cost}_G(U_1 \cup S_{1/2}) - w(T_1, T_1 \cup T_{1/2}) - w(S_0, S_{1/2}) \\ &\leq \mathrm{cost}_G(S) - \mathrm{cost}_G(U_1 \cup S_{1/2}) \ .\end{aligned}$$

As S^*_B is optimal in B, we know that $\mathrm{cost}_B(S_B) - \mathrm{cost}_B(S^*_B) \geq 0$, which implies by the above that $\mathrm{cost}(S) - \mathrm{cost}(U_1 \cup S_{1/2}) \geq 0$, and so the claim is proven. □

Claim 2 implies that we can safely restrict ourselves to solutions for (G, w) which includes all vertices of U_1, and no vertex of U_0. Therefore, all edges which have at least one endpoint in U_1 are redundant to us in this sense. Also, edges with both endpoints in U_0 are redundant, since we can safely leave these uncovered. The same is not true for edges between U_0 and $U_{1/2}$, as these still might need to be covered. We take care of these edges in the second step of our algorithm, but

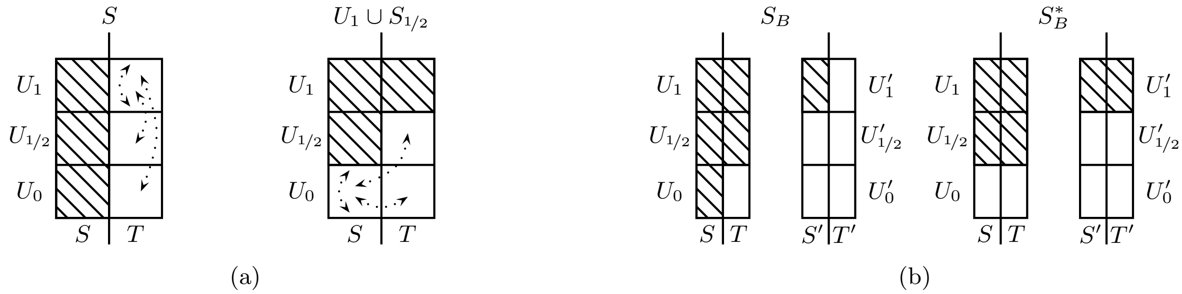

Fig. 2: Depiction of the differences beween S and $U_1 \cup S_{1/2}$ and between S_B and S_B^*; The dotted lines correspond to edges that are not covered by one cover, but are covered by the other.

for now consider the graph H obtained by deleting all edges between vertices of U_0 in the induced subgraph $G[U_0 \cup U_{1/2}]$ of G. Define a weight function w_H for H which equals w on all edges of H and all vertices of $U_{1/2}$ and assigns ∞ to all vertices in U_0. In the following we argue that a good approximation for (H, w_H) gives a good approximation for (G, w).

Claim 3. If S is α-approximate for (H, w_H), then $U_1 \cup S$ is α-approximate for (G, w).

Proof. Let S_H^* denote an optimal solution in (H, w_H), and assume w.l.o.g. that $\mathrm{cost}_H(S_H^*) < \infty$. Then $\mathrm{cost}_H(S) \leq \alpha \cdot \mathrm{cost}_H(S_H^*) \leq \infty$. Now, according to Claim 2, there is an optimal solution for (G, w) which includes all vertices of U_1 and no vertex of U_0, so let S^* be such a solution, with $S_{1/2}^* = S^* \cap U_{1/2}$ and $T_{1/2}^* = U_{1/2} \setminus S_{1/2}^*$. Hence,

$$\begin{aligned}
\mathrm{cost}_G(U_1 \cup S) &= w(U_1) + w(U_0, U_0) + \mathrm{cost}_H(S) \\
&\leq w(U_1) + w(U_0, U_0) + \alpha \cdot \mathrm{cost}_H(S_H^*) \\
&\leq \alpha \cdot (w(U_1) + w(U_0, U_0) + \mathrm{cost}_H(S_{1/2}^*)) \\
&= \alpha \cdot (w(U_1) + w(U_0, U_0) + w_H(S_{1/2}^*) + w_H(U_0 \cup T_{1/2}^*, U_0 \cup T_{1/2}^*)) \\
&= \alpha \cdot (w(U_1) + w(U_0, U_0) + w(S_{1/2}^*) + w(U_0 \cup T_{1/2}^*, U_0 \cup T_{1/2}^*)) \\
&= \alpha \cdot \mathrm{cost}_G(S^*) \ ,
\end{aligned}$$

and the claim is proven. □

Furthermore, we show that the total weight of elements in H with finite weight is at most twice the cost of any solution of (H, w_H):

Claim 4. $w_H(U_{1/2}) + w_H(U_0, U_{1/2}) \leq 2 \cdot \mathrm{cost}_H(S)$ for every $S \subseteq U_0 \cup U_{1/2}$.

Proof. If $S \nsubseteq U_{1/2}$, then $\mathrm{cost}_H(S) = \infty$, and the claim is trivial. Assume therefore that $S \subseteq U_{1/2}$, and denote $T = U_{1/2} \setminus S$. Consider the solution $S_B = U_1 \cup U_1' \cup S \cup S'$ for the bipartite graph B constructed above. The cost of this solution is:

$$\begin{aligned}
\mathrm{cost}_B(S_B) &= w_B(U_1 \cup U_1') + w_B(S \cup S') + w_B(U_0 \cup T, U_0' \cup T') \\
&= 2 \cdot w(U_1) + 2 \cdot w(S) + w(U_0, U_0) + w(T, T) + 2 \cdot w(U_0, T) \\
&= 2 \cdot w(U_1) + 2 \cdot \mathrm{cost}_H(S) + w(U_0, U_0) - w(T, T)
\end{aligned}$$

Now let us compare this solution to S_B^*, the optimal solution of B. Recall that we can assume that S_B^* includes all vertices of $U_{1/2}$, and no vertex of $U_{1/2}'$. The cost of S_B^* equals the total weight of its vertices plus the total weight of all edges between U_0 and $U'_0 \cup U'_{1/2}$. We have,

$$\mathrm{cost}_B(S^*_B) = 2 \cdot w(U_1) + w(U_{1/2}) + w(U_0, U_0) + w(U_0, U_{1/2}) \ .$$

The claim can now be easily proven by combining the two equalities above with the fact that $\mathrm{cost}_B(S^*_B) \leq \mathrm{cost}_B(S_B)$. □

3.2 Step II

Note that while the instance (H, w_H) is close to what we aimed to achieve, it is not quite there. One reason is that H is not an induced subgraph of G, it contains vertices of U_0 without the edges between them. Another reason is that the U_0 vertices have w_H-weight equal to ∞, and therefore any feasible solution for (H, w_H) will not satisfy the second condition of Theorem 2. We cannot simply discard these U_0 vertices, since some of them might be connected to vertices in $U_{1/2}$. For this reason, we apply the Local-Ratio lemma to eliminate edges between U_0 and U_1. This is done by applying the following procedure that iteratively subtracts 1-effective weight functions from w_H, in order to obtain the weight function $\widetilde{w}$ promised by Theorem 2:

While there is an edge $e_0 = \{u, v\}$ in H with $u \in U_0$, $v \in U_{1/2}$, and $w_H(e_0), w_H(v) > 0$, do:
a. Let $\varepsilon = \min\{w_H(e), w_H(v)\}$.
b. Define the weight function w_ε for H by:
   – $w_\varepsilon(v), w_\varepsilon(e_0) = \varepsilon$, and
   – $w_\varepsilon(u), w_\varepsilon(e) = 0$ for all $u \neq v$ and $e \neq e_0$.
c. $w_H = w_H - w_\varepsilon$.

It is easy to see that the above procedure terminates after polynomial time, since at each iteration, either a vertex or an edge get their w_H-weight reduced to zero. The weight function $\widetilde{w}$ is defined to be w_H at the end of the procedure. We define the partition of the vertices in G which is promised in Theorem 2 using $\widetilde{w}$:

$$V_1 = U_1 \cup \{v \in U_{1/2} \,|\, \widetilde{w}(v) = 0\} \ , \qquad V_0 = U_0 \ , \qquad \text{and} \qquad V_{1/2} = V \setminus (V_1 \cup V_0) \ .$$

To complete the proof, we argue that a good approximation for $(G[V_{1/2}], \widetilde{w})$ gives a good approximation for (H, w_H), and that the total weight of vertices in $V_{1/2}$ is at most twice the cost of any solution for $(G[V_{1/2}], \widetilde{w})$.

Claim 5. If S is α-approximate for $(G[V_{1/2}], \widetilde{w})$ then S is α-approximate for (H, w_H).

Proof. We prove the claim using induction on the number of steps applied in this procedure. According to Lemma 2, it suffices to show that any weight function w_ε subtracted in the procedure above is 1-effective. But this is immediate since any solution with cost less than ∞ in (H, w_ε) has cost exactly ε: It either pays for not covering the edge e_0, or for its endpoint in $U_{1/2}$, but never for both. Finally, since $G[V_{1/2}]$ is obtained by removing vertices from H with either 0 or ∞ $\widetilde{w}$-weights, an α-approximate solution for $(G[V_{1/2}], \widetilde{w})$ implies an α-approximate solution for $(H, \widetilde{w})$, and the claim follows. □

Claim 6. $\widetilde{w}(V_{1/2}) \leq 2 \cdot \mathrm{cost}_{G[V_{1/2}]}(S)$ for every $S \subseteq V_{1/2}$.

Proof. By Claim 4, we have for any $S \subseteq U_0 \cup U_{1/2}$,

$$w_H(U_{1/2}) + w_H(U_0, U_{1/2}) \leq 2 \cdot \text{cost}_H(S) - 2 \cdot w_H(U_0, U_0) \ .$$

Since at each iteration in the procedure above, we subtract exactly 2ε from each side of this inequality, at the end of which $G[V_{1/2}]$ includes all positive weighted vertices of H, we get

$$\widetilde{w}(V_{1/2}) \leq \widetilde{w}(U_{1/2}) + \widetilde{w}(U_0, U_{1/2}) \leq 2 \cdot \text{cost}_{G[V_{1/2}]}(S) \ ,$$

and the claim is proven. □

It is now easy to show that the partition V_1, V_0 and $V_{1/2}$, along with the weight function $\widetilde{w}$, satisfy both conditions of Theorem 2. Combining Claim 5 with Claim 6 proves the first condition, while the second condition follows directly from Claim 6.

4 Applications

As mentioned in Section 1, the Nemhauser&Trotter Theorem has several application in designing approximation algorithms for Vertex Cover. In the following we show that several of these extend also to Generalized Vertex Cover, by using Theorem 2. The section is divided into four parts, with each part giving a different corollary of Theorem 2. We start with a $(2 - 2/d)$-approximation algorithm for graphs of bounded degree d, then continue to show a PTAS for planar graphs, and a $(2 - \lg\lg n / 2\lg n)$-approximation algorithm for general graphs. Finally, we will show that Theorem 2 gives a linear kernel for the parameterized Generalized Vertex Cover.

4.1 Bounded Degree Graphs

Our first application for Theorem 2 is a $(2 - 2/d)$-approximation algorithm for Generalized Vertex Cover in graphs of bounded degree d. This is an analogous result to an algorithm of Hochbaum [12] that applies the original Nemhauser&Trotter Theorem to obtain the same approximation ratio for Vertex Cover in graphs of bounded degree d. In her algorithm, Hochbaum uses a classical graph-theoretic result by Brooks [5] which states that any graph of bounded degree d which is not complete nor an odd cycle can be properly colored in d colors. (That is, its vertex set can be partitioned into d classes, with no edges between any pair of vertices in the same class.) Together with the Nemhauser&Trotter Theorem, this is basically all that is necessary for Hochbaum's algorithm. Indeed, in our case it is also all that is necessary, due to the following lemma:

Lemma 4. Generalized Vertex Cover *is polynomial-time solvable in cycles.*

Corollary 1. d-Generalized Vertex Cover *is approximable within* $2 - 2/d$*, for any* $d > 1$*.*

4.2 Planar Graphs

We next use the technique of Baker [1] together with Theorem 2 to obtain a PTAS for Generalized Vertex Cover in planar graphs. The main idea is to first use the algorithm of Theorem 2, and then to break the planar subgraph $G[V_{1/2}]$ into a set of k-outerplanar graphs, by removing a set of vertices from G whose weight is at most $\widetilde{w}(V_{1/2})/k$. Since k-outerplanar graphs have treewidth depending only on k, we can compute the optimum solution for each graph in the set of remaining outerplanar graphs. Furthermore, since $\widetilde{w}(V_{1/2})$ is at most twice the cost of the optimum solution in $(G[V_{1/2}], \widetilde{w})$ by Theorem 2, this removed set of vertices together with optimal solutions of the k-outerplanar graphs constitute a $(1 + 2/k)$-approximate generalized vertex cover. We begin with a formal definition of tree-decompositions and treewidth:

Definition 2 (Tree Decomposition [20]). *A* tree decomposition *of a graph* $G = (V, E)$ *is a pair* $(\mathcal{T}, \mathcal{X})$*, where* $\mathcal{X} \subseteq 2^V$ *is a family of subsets of* V*, and* $\mathcal{T}$ *a tree over* $\mathcal{X}$*, satisfying the following conditions: (i)* $\bigcup_{X \in \mathcal{X}} G[X] = G$*, and (ii)* $\mathcal{X}_v = \{X \in \mathcal{X} \,|\, v \in X\}$ *is connected in* $\mathcal{T}$ *for each* $v \in V$*. The* width *of* $\mathcal{T}$ *is* $\max_{X \in \mathcal{X}} |X| - 1$*. The* treewidth *of* G *is the minimum width over all tree decompositions of* G*.*

We solve Generalized Vertex Cover in graphs with bounded treewidth using a standard bottom-up dynamic programming approach.

Lemma 5. Generalized Vertex Cover *can be solved in* $2^{O(w)} \cdot n$ *time in graphs of treewidth at most* w*.*

Corollary 2. Generalized Vertex Cover *in planar graphs has a PTAS.*

4.3 General Graphs

We now show that the $(2 - \lg\lg n/2\lg n)$-approximation algorithm of [2] can be extended to Generalized Vertex Cover, due to Theorem 2. The central component in the algorithm of [2] is given in the following lemma:

Lemma 6 ([2]). *There is a polynomial-time algorithm that given a graph* $G = (V, E)$*, a weight function* $w : V \to \mathbb{Q}^{\geq 0}$*, and an integer* k*, such that (i)* $|V| \geq (2k-1)^k$*, and (ii)* G *does not contain an odd cycle of length at most* $2k - 1$*, computes a vertex cover* C *of* G *with* $w(C) \leq (1 - \frac{1}{2k})w(V)$*.*

Another component we will use is given due to the local-ratio technique: We can remove odd cycles in a given instance (G, w) of Generalized Vertex Cover at a relatively small cost to our approximation guarantee. We have the following lemma:

Lemma 7. *Let* (G, w) *be an instance of* Generalized Vertex Cover*, and let* C *be an odd cycle in* G *of size* $2t - 1$*, where* $t \leq k$*. Then, the weight function* w_ϵ *which assigns* ϵ *to all vertices and edges in* C*, and 0 to all other vertices and edges* $(2 - \frac{1}{k})$*-effective.*

Corollary 3. Generalized Vertex Cover *is approximable within* $2 - \lg\lg n/2\lg n$*.*

4.4 Fixed-Parameter Tractability

The Nemhauser&Trotter Theorem has applications outside the world of approximation algorithms, most notably in the world of parameterized complexity. Chen *et al.* [7] observed that this theorem gives a $2k$ kernel for unweighted parameterized Vertex Cover problem, when the parameter is the total weight of the required vertex cover. We next note that, using Theorem 2, this straightforwardly extends to Generalized Vertex Cover parameterized by the cost of the optimal solution.

Parameterized complexity deals with parameterized problems, problems whose instances are given together with a numeric *parameter* k that encodes various structural properties of the input, *e.g.* solution size, maximum degree, and so forth. This allows a refined definition of tractable problems, where a tractable problem is now one with an algorithm running in $f(k)poly(n)$ time, where n is the instance size and $f()$ is any computable function. The class FPT is the class of all parameterized problems with an $f(k)poly(n)$ algorithm. A *kernelization algorithm*, or simply a *kernel*, is a commonly used technique for showing that a parameterized problem is in FPT. Formally, a kernel is a polynomial-time algorithm that transforms an instance (I, k) to an instance (I', k'), with $|I'| + k' \leq f(k)$ for some computable function $f()$, and such that (I, k) is a "yes"-instance iff (I', k') is a "yes"-instance. It is easy to see that Theorem 2 gives exactly this, when the parameter is taken as the cost of the solution.

Corollary 4. Generalized Vertex Cover *parameterized by the cost k of the optimal solution has a $2k$ kernel.*

References

1. B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.
2. R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 25:27–46, 1985.
3. R. Bar-Yehuda and D. Rawitz. On the equivalence between the primal-dual schema and the local ratio technique. *SIAM Journal on Discrete Mathematics*, 19(3):762–797, 2005.
4. H. L. Bodlaender. Classes of graphs with bounded tree-width. Technical Report RUU-CS-86-22, Utrecht University, 1986.
5. R. L. Brooks. On colouring the nodes of a network. *Mathematical Proceedings of the Cambridge Philosophical Society*, 37:194–197, 1941.
6. M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 642–651, 2001.
7. J. Chen, I. A. Kanj, and W. Jia. Vertex cover: Further observations and further improvements. *Journal of Algorithms*, 41(2):280–301, 2001.
8. M. Chlebík and J. Chlebíková. Improvement of nemhauser-trotter theorem and its applications in parametrized complexity. In *9th Scandinavian Workshop on Algorithm Theory*, pages 174–186, 2004.
9. I. Dinur and S. Safra. The importance of being biased. In *34th ACM Symposium on the Theory of Computing*, pages 33–42, 2002.
10. M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.
11. R. Hassin and A. Tamir. Improved complexity bounds for location problems on the real line. *Operations Research Letters*, 10:395–402, 1991.
12. D. S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6:243–254, 1983.
13. D. S. Hochbaum. Solving integer programs over monotone inequalities in three variables: a framework of half integrality and good approximations. *European Journal of Operational Research*, 140(2):291–321, 2002.
14. K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual-fitting with factor-revealing LP. *Journal of the ACM*, 50(6):795–824, 2003.
15. R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.
16. J. Könemann, O. Parekh, and D. Segev. A unified approach to approximating partial covering problems. In *14th European Symposium on Algorithms*, pages 468–479, 2006.
17. D. König. Graphok és matrixok [Hungarian; Graphs and matrices]. *Matematikai és Fizikai Lapok*, 38:116–119, 1931.
18. R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.
19. G. Nemhauser and L. T. Jr. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8(2):232–248, 1975.
20. N. Robertson and P. D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

A Omitted Proofs

Proof (Proof of Lemma 4). We first show how to solve the problem in paths using dynamic programming. Let G be a path $v_1, \ldots, v_n$, and let G_i denote the path $v_1, \ldots, v_i$ for all $i \in \{1, \ldots, n\}$. We compute two tables Π and Π', where $\Pi(i)$ is the cost of the optimal generalized vertex cover of G_i that contains v_i, and $\Pi'(i)$ is the cost of the optimal generalized vertex cover of G_i that does not contain v_i. We have $\Pi(1) = w(v_1)$ and $\Pi'(1) = 0$, and we may compute $\Pi(i)$ for $i > 1$ using the following recurrence:

$$\Pi(i) = \min\{\Pi(i-1) + w(v_i), \Pi'(i-1) + w(v_i)\}$$
$$\Pi'(i) = \min\{\Pi(i-1), \Pi'(i-1) + w(v_{i-1}, v_i)\}$$

The cost of the optimal solution for the path G is $\min\{\Pi(n), \Pi'(n)\}$. Also, note that the computation of $\Pi(i)$ can be easily modified to output a corresponding optimal generalized vertex cover.

Next, we solve the problem in cycles using a reduction to the problem in paths. Given a weighted cycle G on the vertices $v_1, \ldots, v_n$, we define G_1 to be the path $v_0, v_1, \ldots, v_n$, where $w(v_0) = \infty$ and $w(v_0, v_1) = w(v_1, v_n)$, and G_2 to be the path $v_1, \ldots, v_n, v_{n+1}$, where $w(v_{n+1}) = \infty$ and $w(v_n, v_{n+1}) = w(v_1, v_n)$. It is not hard to verify that $\mathrm{opt}(G) = \min\{\mathrm{opt}(G_1), \mathrm{opt}(G_2)\}$. ☐

Proof (Proof of Corollary 1). Given an instance (G, w) of Generalized Vertex Cover, with G having no degree greater than d, apply Theorem 2 to obtain V_1, V_0, $V_{1/2}$, and $\widetilde{w}$. If $G[V_{1/2}]$ is a cycle, then an optimal generalized vertex cover in $(G[V_{1/2}], \widetilde{w})$ can be found in polynomial-time by Lemma 4. If $G[V_{1/2}]$ is the complete graph, then $G[V_{1/2}]$ contains $O(d^2) = O(1)$ vertices, and we can trivially compute in polynomial-time an optimal generalized vertex cover.

Assume therefore that $G[V_{1/2}]$ in neither complete nor a cycle, and let $S^*_{1/2}$ denote the optimal solution for $(G[V_{1/2}], \widetilde{w})$. By Theorem 2, we know that $\mathrm{cost}_{G[V_{1/2}]}(S^*_{1/2}) \geq \frac{1}{2} \cdot \widetilde{w}(V_{1/2})$. Since $G[V_{1/2}]$ has degree bounded by d, and it is nor an odd cycle neither a complete graph, we can color it using d colors [5]. Now, let $S_{1/2}$ be the subset of $V_{1/2}$ excluding the heaviest color class with respect to $\widetilde{w}$. We have,

$$\mathrm{cost}_{G[V_{1/2}]}(S^*_{1/2}) \geq \frac{1}{2} \cdot \widetilde{w}(V_{1/2}) \geq \frac{d}{2(d-1)} \cdot \widetilde{w}(S_{1/2}) = \frac{d}{2(d-1)} \cdot \mathrm{cost}_{G[V_{1/2}]}(S_{1/2}) \ .$$

and therefore $V_1 \cup S_{1/2}$ is $(2 - \frac{2}{d})$-approximate. ☐

Proof (Proof of Lemma 5). We start with some notation. Let R be an arbitrary root of $\mathcal{T}$, and let $\mathcal{T}(X)$ denote the subtree of $\mathcal{T}$ whose root is X. (In this notation $\mathcal{T}(R) = \mathcal{T}$.) Also, let G_X be the subgraph of G that is induced by the vertex set $\bigcup_{Y \in \mathcal{T}(X)} Y$.

Now, for any $X \in \mathcal{X}$, and any $S \subseteq X$, let $\Pi(X, S)$ denote the cost of an optimal generalized vertex cover S^* in (G_X, w) such that $S^* \cap X = S$. We compute $\Pi(X, S)$ in a bottom-up fashion. First, if X is a leaf in $\mathcal{T}$ then $\Pi(X, S)$ can be computed directly. Otherwise, let X_0 be an internal node of $\mathcal{T}$ and let $X_1, X_2, \ldots, X_d$ be its children. Since $\Pi(X_0, S_0)$ is obtained by some vertex set $S^* \subseteq \bigcup_{Y \in \mathcal{T}(X)} Y$ such that $S_0 = X_0 \cap S^*$ we can compute $\Pi(X_0, S_0)$ using the following recurrence:

$$\Pi(X_0, S_0) = \min_{S : S_0 \subseteq S \subseteq \bigcup_{i=0}^{d} X_i} \left(\mathrm{cost}_{G[X_0]}(S_0) + \sum_{i=1}^{d} (\Pi(X_i, S \cap X_i) - \mathrm{cost}_{G[X_0 \cap X_i]}(S_0 \cap X_i)) \right) \ .$$

The idea is that we add the cost of S^* in G_{X_i}, for any $i = 1, \dots, d$, to the cost of S^* in $G[X]$, and then the subtract weights that were counted more than once. For example, when $d = 1$, we add $\Pi(X_1, S \cap X_1)$ to the cost of S_0 in $G[X_0]$ and subtract the cost of $S_0 \cap X_1$ in $G[X_0 \cap X_1]$.

Note that the computation of Π can be easily modified to computing a corresponding generalized vertex cover. Since one may assume w.l.o.g. that $\mathcal{T}$ is binary, the running time for computing an entry $\Pi(X_0, S_0)$ is $2^{O(w)}$. Hence, the total running time is $2^{O(w)} \cdot n$. □

Proof (Proof of Corollary 2). Given a planar embedding of a planar graph, the vertices on the exterior face are said to be at level 0. After the removal of level 1 vertices, the vertices of the resulting exterior face are said to be in level 1. Level i vertices are the vertices that are contained in exterior face after the removal of vertices from level $0, \dots, i-1$. We denote the set of level i vertices by L_i. In these terms, a k-outerplanar graph has a planar embedding with k layers.

Let G be a planar graph, and let $G[V_{1/2}]$ be the planar subgraph obtained by Theorem 2. By the theorem, a PTAS on $(G[V_{1/2}], \widetilde{w})$ would imply a PTAS on (G, w). Let k be some positive integer, and let G_i, $i \in \{0, \dots, k\}$ be the graph obtained by removing layers $i + (k+1)j$, for $j = 0, 1, \dots,$ from $G[V_{1/2}]$. Since G_i is k-outerplanar for every $i \in \{0, \dots, k\}$ its tree width is $3k - 1$ [4], and therefore we can compute an optimal generalized vertex cover U_i^* in G_i in polynomial time using Lemma 5. Letting $U_i = U_i^* \cup \bigcup_j L_{i+(k+1)j}$, we have

$$\begin{aligned}\sum_i \text{cost}_G(U_i) &= \sum_i \text{cost}_{G_i}(U_i^*) + \sum_i w(L_i) \\ &\le \sum_i \text{opt}(G[V_{1/2}]) + 2\text{opt}(G[V_{1/2}]) \\ &= (k+2) \cdot \text{opt}(G[V_{1/2}]) \ ,\end{aligned}$$

where $\text{opt}(G)$ is the cost of an optimal generalized vertex cover for $(G[V_{1/2}], \widetilde{w})$. Hence, the solution with minimum weight among the U_is is $(1 + \frac{2}{k})$-approximate. Finally, picking $k = \frac{2}{\epsilon}$ we obtain a $(1+\epsilon)$-approximation for every ϵ. □

Proof (Proof of Lemma 7). First, observe that no solution for (G, w_ϵ) may weigh more than $(2t-1)\cdot\epsilon$, since $w_\epsilon(V) = (2t-1) \cdot \epsilon$. Furthermore, since the weights are uniform, paying a penalty for an edge is never better than taking one of its end-points, and so an optimal solution for (G, w_ϵ) must have cost at least $t \cdot \epsilon$. It follows that any solution for (G, w_ϵ) is $(2 - 1/t)$-approximate, and since $t \le k$, we get that any solution is $(2 - 1/k)$-approximate, and the lemma follows. □

Proof (Proof of Corollary 3). Let k be the smallest integer for which $(2k-1)^k \ge |V|$. Note that k can be computed in polynomial time. We first remove odd cycles of length at most $2k - 1$ from the input graph. This is done as follows: As long as there is an odd cycle C in G having positive weights on all its vertices and edges, we construct a weight function w_ϵ as in Lemma 6, which is $(2-1/k)$-effective according to the lemma. We then continue searching in $(G, w - w_\epsilon)$. Lemma 2 ensures that a $(2-1/k)$-approximate solution with respect to $(G, w - w_\epsilon)$ will also be $(2-1/k)$-approximate with respect to $(G, w - w_\epsilon)$. Let w' denote the weight function that we obtain when no odd cycles having positive weights on all their vertices and edges are left. Also, let G' be the graph obtained by removing all zero w'-weight vertices from G, along with all edges incident to them, and all zero w'-weight edges. Clearly, if U is a $(2-1/k)$-approximate solution for (G', w'), then $U \cup V_Z$ is $(2 - 1/k)$-approximate for (G, w'), where V_Z is the set of all zero w'-weight vertices in G.

We next use Theorem 2 on (G', w') to obtain a partition of the vertices of G' into three subsets, V'_1, V'_0, and $V'_{1/2}$, and to obtain the weight function $\widetilde{w}$. According to Theorem 2, an α-approximation for $(G[V'_{1/2}], \widetilde{w})$ implies an α-approximation on (G', w'). Since $G[V'_{1/2}]$ satisfies the conditions of Lemma 7, we can find a vertex cover C for $G[V'_{1/2}]$ such that $w(C) \leq (1 - 1/2k) \cdot w(V)$. Due to the second condition in Theorem 2, the subset C is a $(2-1/k)$-approximate solution for $(G[V'_{1/2}], \widetilde{w})$. $\square$