On Partial Vertex Cover and Budgeted Maximum Coverage Problems in Bipartite Graphs

Bugra Caskurlu[1,⋆,⋆⋆], Vahan Mkrtchyan[2,⋆], Ojas Parekh[3,⋆⋆⋆], and K. Subramani[2,⋆,†]

[1] Computer Engineering Department, TOBB University of Economics and Technology,
Ankara, 06560, Turkey
caskurlu@gmail.com

[2] LDCSEE, West Virginia University, Morgantown, WV, USA
vahan.mkrtchyan@mail.wvu.edu, ksmani@csee.wvu.edu

[3] Sandia National Laboratories, Albuquerque, NM, USA
odparek@sandia.gov

Abstract. Graphs are often used to model risk management in various systems. Particularly, Caskurlu et al. in [6] have considered a system which essentially represents a tripartite graph. The goal in this model is to reduce the risk in the system below a predefined risk threshold level. It can be shown that the main goal in this risk management system can be formulated as a Partial Vertex Cover problem on bipartite graphs. It is well-known that the vertex cover problem is in **P** on bipartite graphs; however, the computational complexity of the partial vertex cover problem on bipartite graphs is open. In this paper, we show that the partial vertex cover problem is **NP-hard** on bipartite graphs. Then, we show that the budgeted maximum coverage problem (a problem related to partial vertex cover problem) admits an $\frac{8}{9}$-approximation algorithm in the class of bipartite graphs, which matches the integrality gap of a natural LP relaxation.

Keywords: Bipartite graph, vertex cover problem, partial vertex cover problem, budgeted maximum coverage problem.

1 Introduction

Covering problems arise often in practice. A mobile phone service provider should ensure that its base stations cover the signals transmitted from the phones of its customers. A chain market such as Walmart should ensure that it has a store close to its customers. The applications of the covering problems are not limited to corporations to sell a service to customers. The Air Force on a no-fly zone mission or border patrol officers trying to secure borders are to solve some form of a covering problem.

[⋆] This research has been supported in part by the Air Force of Scientific Research through Award FA9550-12-1-0199.

[⋆⋆] This work is done when the author was at West Virginia University. The author was supported by the National Science Foundation through Awards CNS-0849735 and CCF-0827397.

[⋆⋆⋆] Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000.

[†] The author is supported by the National Science Foundation through Award CCF-1305054.

J. Diaz et al.(Eds.): TCS 2014, LNCS 8705, pp. 13–26, 2014.

In many real life situations the corporations or the government is constrained in the resources it can allocate for the covering mission. The constraints may be hard constraints such as a government agency to operate within its approved budget, or profit dictated soft constraints such as a mobile phone service provider may decide not to cover a rural area since the revenues will not match the covering costs. Therefore, the goal in many real life situation can be cast as covering the domain as much as possible for a given fixed amount of resources to be allocated.

There is merit in studying partial covering problems both due to their wide applicability in a large range of applications and their theoretical importance as being natural generalizations of classical covering problems. In this paper, we study the partial vertex cover (PVC) problem and the related budgeted maximum coverage problem on bipartite graphs. Though it is well-known that the vertex cover problem is polynomial-time solvable on bipartite graphs, the computational complexity of partial vertex cover when restricted to bipartite graphs (PVCB) remained open. In parallel to this, let us note that approximation algorithms for this problem are known, with the best approximation algorithm in the literature offering an approximation ratio of $(\frac{4}{3} + \epsilon)$ [19].

The contributions of this paper are as follows:

(i) The partial vertex cover and budgeted maximum coverage problems are **NP-hard** on bipartite graphs.
(ii) The (weighted) budgeted maximum coverage problem on bipartite graphs admits an $\frac{8}{9}$-approximation algorithm, matching the integrality gap of the natural linear programming relaxation.

The rest of this paper is organized as follows: Section 2 presents a formal definition of the main problems that are considered in the paper. We present a concrete application to motivate the PVCB problem in Section 3. The related work is presented in Section 4. The computational complexity of the PVCB problem is established in Section 5. In Section 6, we present an $\frac{8}{9}$-approximation algorithm for the budgeted maximum coverage problem on bipartite graphs; note that this problem is computationally identical to the PVCB problem. We conclude and point out several research directions in Section 7.

2 Statement of Problems

If $G = \langle V, E \rangle$ is a graph, and u is a vertex, e is an edge of G, respectively, then we will say that u covers e in G, if u is incident to e, that is, u is one of the end-vertices of e. For a set $V' \subseteq V$, let $E_{V'}$ denote the set of edges of G that are covered by vertices of V'.

In the classical vertex cover (VC) problem, we are given an undirected graph $G = \langle V, E \rangle$, where V is the vertex set with $|V| = n$, E is the edge set with $|E| = m$. The goal is to find a minimum cardinality subset $V' \subset V$, such that $E_{V'} = E$.

In this paper we will consider the partial vertex cover (PVC) problem restricted to bipartite graphs.

Definition 1. *Given an integer t, and an undirected bipartite graph $G = \langle V, E \rangle$. The PVCB problem is defined as finding a minimum cardinality subset $S \subset V$ such that $|E_S| \geq t$.*

It is trivial to observe that the PVC problem is a generalization of the VC problem. We will also consider the budgeted maximum coverage problem, and the following its extension.

Definition 2. *Given an integer k, and undirected bipartite graph $G = \langle V, E, w \rangle$, and $w : E \to Z$ denoting a weighting function from the set of edges to the set of positive integers. The EBMCB problem is defined as finding a subset $S \subset V$ with $|S| \leq k$ maximizing $\sum_{e:e \in E_S} w(e)$.*

Definition 3. *Given an integer k, and undirected bipartite graph $G = \langle V, E \rangle$ with V denoting the set of vertices, E denoting the set of edges. The BMCB problem is defined as finding a subset $S \subset V$ with $|S| \leq k$ maximizing the number of covered edges.*

The PVC and the BMC problems are equivalent from the perspective of exact solution. This follows from the observation that one can simply swap the budget constraint with objective and reverse the optimization direction. On the other hand, let us observe that the two problems are likely to be different from the perspective of approximate solution. The BMC problem for sets (not necessarily for graphs) admits a $(1 - \frac{1}{e})$-approximation algorithm [15], where $e = 2.718....$, and it is known that there exists a constant $\gamma > 0$, such that Set Cover problem admits no $\gamma \cdot \log n$ approximation [8], under the assumption that **NP** has no quasi-polynomial time algorithms.

3 Motivation

The number of attacks faced by an ordinary device that has an access to Internet remains quite high, which means that it is unreasonable to expect a manual response to an attack. A good example of a system for which an automated response is needed are devices that have an operating system. One of the approaches to handle the situation, is to dynamically control the device's vulnerability to threats and limit undesirable consequences as much as possible.

In order to protect a system, one needs to minimize the risk it faces. It is reasonable to assume that the risk depends on three factors, which are threats and their probabilities of occurrence, the existing weaknesses of the system, and the consequences that the system experiences after a successful attack.

While threats are not under the control of the device, the other two can be handled by it. Here it is implicitly assumed that the device can be protected by decreasing its functionality. This implies that our main goal is becoming to minimize the tension between functionality and security. In great contrast with the approach of statically configuring a system, one can dynamically monitor the risk level, and allow users to have maximum functionality where the level of risk is assumed to be on a predefined level.

In [6], the risk of a computational system is modeled as a flow between the first and last partitions in a tripartite graph, where the vertices of the three partitions represent threats to the system, vulnerabilities of the system, and the assets of the system as shown in Figure 1.

In the risk management model given in [6], the goal is to reduce the risk in the system (flow between the first and last partitions) below a predefined risk threshold level by

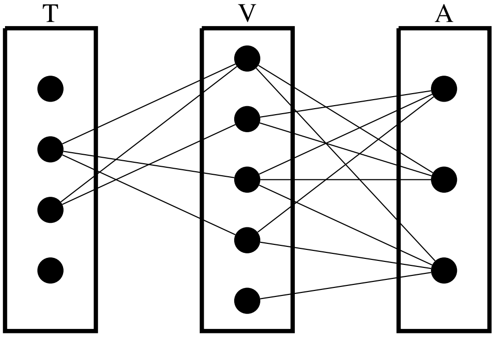

Fig. 1. Risk in a computational system can be modeled in terms of its constituent components. The threats, weaknesses (corresponding to specific vulnerabilities), and assets form three disjoint sets, named as T, V, and A respectively. An edge between vertices represents a contribution to the system risk. The system's risk is the total flow between the first and third sets.

either restricting the permissions of the users, or encapsulating the system assets. These two strategies correspond to deleting minimum number of vertices from the second and the third partitions of the tripartite graph so that the flow between the first and the third partitions are reduced below the predefined threshold level. The equivalence of this risk management system and the partial vertex cover problem on bipartite graphs is established in [6].

From the point of view of risk management the important instances of the problem correspond to the cases when the graph modeling the system has a bounded degree, or has a relatively predictable structure (such as being a tree). This paper focuses on the problem when the graphs are arbitrary bipartite graphs.

4 Related Work

The vertex cover (VC) problem is one of the classical **NP-complete** problems listed by Karp [13]. There are several polynomial-time approximation algorithms for the VC problem within a factor of 2, and the best-known approximation algorithm for the VC problem has an approximation factor of $\left(2 - \theta\left(\frac{1}{\sqrt{\log n}}\right)\right)$ [12]. The VC problem is known to be **APX-complete** [21]. Moreover, it cannot be approximated to within a factor of 1.3606 unless **P = NP** [7], and not within any constant factor smaller than 2, unless the *unique games conjecture* is false [14].

Since the PVC problem subsumes the VC problem for $t = m$, all the hardness results given above for the VC problem directly apply to the PVC problem. The PVC problem and the partial variants of similar graph problems have been extensively studied for more than a decade [4], [18], [16], [17], [3]. Particularly, there are an $O(n \cdot \log n + m)$-time 2-approximation algorithm [20], that is based on the method of primal-dual, a combinatorial 2-approximation algorithm [3], and some $(2-o(1))$-approximation algorithms [2], [5], [9], [11]. Let us also note that in [19] for each $\epsilon > 0$ an approximation algorithm with performance ratio $(\frac{4}{3} + \epsilon)$ for WPVCB is presented. Moreover, a $\frac{3}{4}$-approximation algorithm for the weighted budgeted maximum coverage problem is general graphs given in [1].

Although the VC problem and the PVC problem have almost matching approximation ratios and inapproximability results, the PVC problem is in some sense more difficult than the VC problem. For instance, the PVC problem is **W[1]-complete** while the VC problem is fixed parameter tractable [10].

5 Computational Complexity of the PVCB Problem

In this section we prove Theorem 1, which shows that the PVCB problem is **NP-hard**. This also establishes the hardness of the weighted versions of the PVCB and the BMCB problems.

Theorem 1. *The PVCB problem is* **NP-hard**.

Proof. We will prove Theorem 1 by giving a Karp reduction from the CLIQUE problem. Recall that in the CLIQUE problem, we are given an undirected graph $G' = \langle V', E' \rangle$, and an integer k, and the goal is to find whether there exists a complete subgraph of G' with k vertices. Assume we are given an arbitrary undirected graph $G' = \langle V', E' \rangle$, where n' and m' denote $|V'|$ and $|E'|$ respectively, and an integer k. We construct a corresponding bipartite graph $G = \langle V_1 \cup V_2, E \rangle$ as explained below.

For every vertex $v_i' \in V'$, G has a corresponding vertex $v_i \in V_1$. For every edge $e' \in E'$, there is a corresponding *edge block* in G. The term *edge block* refers to two vertices and an edge in between. So, for each edge $e' \in E'$, G has two corresponding vertices $e_1 \in V_2$ and $e_2 \in V_1$ and the edge (e_1, e_2). In order to capture the incidence matrix of G', for each edge $e' = (v_i', v_j')$ of G', G has 2 additional edges (v_i, e_1) and (v_j, e_1). Let n and m denote the number of vertices and edges of G, respectively. Notice that the bipartite graph G has $n' + 2 \cdot m'$ vertices and $3 \cdot m$ edges. More precisely, we have $n = n' + 2 \cdot m'$, and $m = 3 \cdot m'$. We use the term *left vertex* of an edge block for the vertex of the edge block that belongs to V_2. The other vertex of the edge block that belongs to V_1, is referred to as the *right vertex* of the edge block throughout the paper.

In Figure 2, we are given a simple undirected graph G' on the left that consists of 2 vertices v_1' and v_2' and an edge $e' = (v_1', v_2')$ in between. The figure has the corresponding bipartite graph G on the right. The vertices v_1 and v_2 of G correspond to 2 vertices of G'. The 2 vertices e_1 and e_2 of G and the edge e in between is the corresponding edge block of the edge e' of G'. The 2 edges (v_1, e_1) and (v_2, e_1) of G capture the incidence matrix of G'.

We will prove Theorem 1 by showing that solving the CLIQUE problem on G' reduces to solving the PVCB problem on G with $t = m - \frac{k \cdot (k-1)}{2}$. In the rest of the paper, without loss of generality, we will assume that $m' > \frac{k \cdot (k-1)}{2}$, and $k \geq 5$. Notice that these assumptions do not violate the soundness of the proof since the CLIQUE problem is still **NP-hard** after these assumptions on the input. We precisely will show the following: there is a complete subgraph of k vertices on G' if and only if there is a subset of $k + m' - \frac{k \cdot (k-1)}{2}$ vertices V'' of G such that the number of edges that are covered by V'' is at least $m - \frac{k \cdot (k-1)}{2}$.

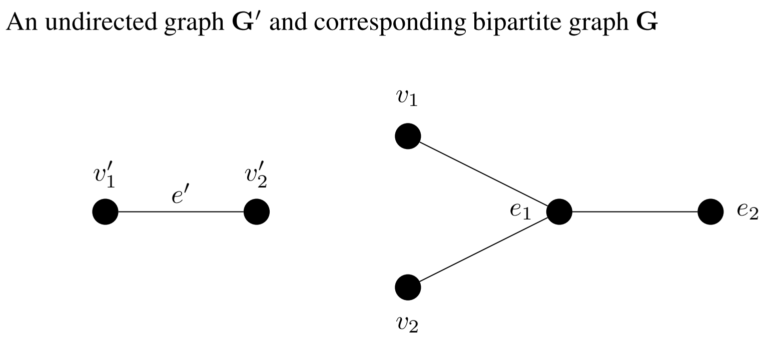
An undirected graph $\mathbf{G}'$ and corresponding bipartite graph $\mathbf{G}$
v_1'
e'
v_2'
v_1
e_1
e_2
v_2

Fig. 2. Construction of the corresponding bipartite graph $\mathbf{G}$ for a given undirected graph $\mathbf{G}'$

Without loss of generality we can assume that V'' does not contain the right vertex of any of the edge blocks of G, since the right vertex of an edge block is incident to only one edge and that edge can be covered by the left vertex of the edge block as well. Therefore, without loss of generality, we can assume that all the vertices of V'' are the vertices that correspond to the vertices of G', or the left vertices of the edge blocks. Since the number of edges that are to be covered by V'' is at least $m - \frac{k \cdot (k-1)}{2}$, the number of edges that are not covered by V'' is at most $\frac{k \cdot (k-1)}{2}$. Therefore, V'' has to contain the left vertices of at least $m' - \frac{k \cdot (k-1)}{2}$ edge blocks. We will complete the proof of Theorem 1 by proving Lemma 1, which maps the yes instances of the CLIQUE problem to the yes instances of the PVCB problem, and Lemma 2, which maps the no instances of the CLIQUE problem to the no instances of the PVCB problem.

Lemma 1. *If there exists a complete subgraph of* k *vertices on* G'*, then there exists a subset* V'' *of vertices of* G *such that* $|V''| = k + m' - \frac{k \cdot (k-1)}{2}$*, and* V'' *covers at least* $m - \frac{k \cdot (k-1)}{2}$ *edges of* G*.*

Proof. Assume that G' has a complete subgraph of k vertices and let V'' be composed of the following $k + m' - \frac{k \cdot (k-1)}{2}$ vertices of G. For every vertex of the complete subgraph of G', let the corresponding vertex of G be in V''. Notice that there are exactly k such vertices. The complete subgraph of this k vertices has $\frac{k \cdot (k-1)}{2}$ edges in G'. Therefore, there are $m' - \frac{k \cdot (k-1)}{2}$ edges of G' that are *not* in the complete subgraph of k vertices in G'. For each of these $m' - \frac{k \cdot (k-1)}{2}$ edges of G', let the left vertex of the corresponding edge block in G be contained in V''. Notice that there are exactly $m' - \frac{k \cdot (k-1)}{2}$ such vertices in V''. So, $|V''| = k + m' - \frac{k \cdot (k-1)}{2}$ as desired.

All we need to prove is that V'' covers at least $m - \frac{k \cdot (k-1)}{2}$ edges of G. Let us first consider the edges of G that capture the incidence relation of the edges of G'. Recall that for every edge $e' = (v_i', v_j')$ of G', there are 2 edges in G to capture the incidence relation of e', namely (v_i, e_1) and (v_j, e_2). So, in total there are $2 \cdot m'$ such edges in G. The $k \cdot (k-1)$ edges of G that capture the incidence relation of the $\frac{k \cdot (k-1)}{2}$ edges of the complete subgraph of G' are covered by the k vertices of V'' that correspond to the k

vertices of G'. The remaining $2 \cdot m' - k \cdot (k-1)$ edges of G, that capture the incidence relation of the $m' - \frac{k\cdot(k-1)}{2}$ edges of G' that are not part of the complete subgraph, are covered by the left vertices of the $m' - \frac{k\cdot(k-1)}{2}$ edge blocks in V''. Therefore, all $2 \cdot m'$ edges of G that capture the incidence relation of the m' edges of G' are covered by V''.

Recall that there are m' additional edges in G. These edges are the edges of the m' edge blocks. The left vertices of the $m' - \frac{k\cdot(k-1)}{2}$ edge blocks that are contained in V'' cover $m' - \frac{k\cdot(k-1)}{2}$ of those edges. Therefore, there are only $\frac{k\cdot(k-1)}{2}$ edges of G that are not covered by V''. So, V'' covers $m - \frac{k\cdot(k-1)}{2}$ edges as stated by Lemma 1. □

Lemma 2. *If G' does not have a complete subgraph of k vertices, then no subset V'' of vertices of G such that $|V''| = k + m' - \frac{k\cdot(k-1)}{2}$ covers at least $m - \frac{k\cdot(k-1)}{2}$ edges of G.*

Proof. Assume G' does not have a complete subgraph of k vertices. For the purpose of contradiction, assume that there is a subset V'' of vertices of G such that $|V''| = k + m' - \frac{k\cdot(k-1)}{2}$, and V'' covers at least $m - \frac{k\cdot(k-1)}{2}$ edges of G.

Since V'' covers at least $m - \frac{k\cdot(k-1)}{2}$ edges of G and $m' > \frac{k\cdot(k-1)}{2}$, V'' covers at least $m' - \frac{k\cdot(k-1)}{2}$ edges of the edge blocks. Therefore, V'' contains the left vertices of at least $m' - \frac{k\cdot(k-1)}{2}$ edge blocks. Since $|V''| = k + m' - \frac{k\cdot(k-1)}{2}$, V'' contains at most k vertices of G that correspond to the vertices of G'.

First consider the case where V'' contains exactly k vertices of G that correspond to the vertices of G', and exactly $m' - \frac{k\cdot(k-1)}{2}$ left vertices of edge blocks. Since there are only $m' - \frac{k\cdot(k-1)}{2}$ left vertices of edge blocks, V'' does not cover $\frac{k\cdot(k-1)}{2}$ edges of the edge blocks. Since V'' covers at least $m - \frac{k\cdot(k-1)}{2}$ in total, V'' covers all the edges of G that capture the incidence relation of all the edges of G'. Since G' does not have a complete subgraph of k vertices, the k vertices of G that correspond to the vertices of G' cover both of the edges that capture the incidence relation of $\frac{k\cdot(k-1)}{2} - \alpha$ edges of G' for some $1 \leq \alpha < \frac{k\cdot(k-1)}{2}$. Since V'' covers all $2 \cdot m'$ edges of G that capture the incidence relation of the edges of G', V'' contains the left vertices of all the edge blocks of G that correspond to the $m' - \frac{k\cdot(k-1)}{2} + \alpha$ edges of G'. This is a contradiction since we assumed that V'' contains exactly $m - \frac{k\cdot(k-1)}{2}$ left vertices of edge blocks.

Therefore, V'' contains exactly $k - l$ vertices of G that correspond to some $k - l$ vertices of G', and exactly $m' - \frac{k\cdot(k-1)}{2} + l$ left vertices of edge blocks for some $0 < l < k$. Recall that the incidence relation of each edge of G' is captured by 2 edges in G. Notice that the subgraph formed by this $k - l$ vertices of G' contains at most $\frac{(k-l)\cdot(k-l-1)}{2}$ edges of G'. Therefore, the corresponding $k - l$ vertices of V'' cover both of the incidence edges of at most $\frac{(k-l)\cdot(k-l-1)}{2}$ edges of G'. In other words, at least one incidence edge of $m' - \frac{(k-l)\cdot(k-l-1)}{2}$ edges of G' is not covered the $k - l$ vertices of G that correspond to some $k - l$ vertices of V''. Since we already have $\frac{k\cdot(k-1)}{2} - l$ edges of edge blocks left uncovered, the left vertices of the edge blocks in V'' has to cover an incidence edge for at least $m' - \frac{(k-l)\cdot(k-l-1)}{2} - l$ edges of G'. This is not possible since the left vertex of each edge block in G covers the corresponding incidence edges of exactly one edge of G', and $m' - \frac{(k-l)\cdot(k-l-1)}{2} - l > m' - \frac{k\cdot(k-1)}{2} + l$ for $k \geq 5$. □

6 An $\frac{8}{9}$-Approximation Algorithm for the EBMCB Problem

The edge-weighted budgeted maximum coverage problem in general graphs is known to be NP-hard, and the result from the previous section shows that this the case for bipartite graphs as well. Thus a natural question is the approximability of the problem. Ageev and Sviridenko [1] give a $\frac{3}{4}$-approximation for general graphs via the pipage rounding technique. We present an $\frac{8}{9}$-approximation for bipartite graphs and show that this matches the integrality gap of the LP-relaxation. We note that our techniques can also be extended to general graphs to give an alternate LP-based $\frac{3}{4}$-approximation.

We employ the following linear programming relaxation for the problem on a bipartite graph $G = (V = V_L \cup V_R, E)$, where V_L and V_R are the independent sets. For a set $S \subseteq V$, we let $S_L = S \cap V_L$ and $S_R = S \cap V_R$.

$$\max \sum_{e \in E} w(e) \cdot z_e$$

$$x_u + x_v \geq z_{uv}, \text{ for all } uv \in E \tag{1}$$

$$\sum_{u \in V} x_u \leq k \tag{2}$$

$$0 \leq x_u \leq 1, \text{ for all } u \in V \tag{3}$$

$$z_e \leq 1, \text{ for all } e \in E \ , \tag{4}$$

where we assume that $w(e) \geq 0$ for all $e \in E$. This assumption allows us to omit the usual non negativity constraint on each z_e. This omission will slightly simplify our analysis and presentation.

Our results exploit the structure of extreme points of the above polyhedron. Consider a feasible solution (x, z) for the above linear program. We say an edge uv is *slack* (with respect to (x, z)) if,

(i) $x_u + x_v > z_{uv}$, or
(ii) $z_{uv} < 1$.

If an edge is not slack, we call it *tight*. The following lemma shows that extreme points are well-structured. For an extreme point $(\hat{x}, \hat{z})$, let $\hat{V} = \{u \in V \mid 0 < \hat{x}_u < 1\}$ be the set of vertices with fractional $\hat{x}$ value, and let $\hat{G} = (\hat{V}, \hat{E})$ be the graph induced by $\hat{V}$.

Lemma 3. *The tight edges induce a single connected component in $\hat{G}$, spanning all its vertices.*

Proof. Suppose, for the moment, that there is a vertex set $J \subseteq \hat{V}$ such that all edges $e \in \hat{E}$ with exactly one endpoint in J are slack. Note that every edge $uv \in E$ with exactly one endpoint in $\hat{V}$ is slack. To see this, suppose $\hat{x}_u \in \{0, 1\}$ and $\hat{x}_v \in (0, 1)$; if $\hat{x}_u = 0$, condition (ii) is satisfied, and otherwise condition (i) is satisfied since $z_{uv} \leq 1$. Thus we actually have that any edge in E with exactly one endpoint in J is slack.

Now we will show that we may perturb the $\hat{x}$ values for vertices in J such that all constraints, except possibly (2), remain satisfied. We will add a value $\varepsilon \neq 0$ to $\hat{x}_u$ for each vertex $u \in J_L$, and we will subtract ε from each $\hat{x}$ value for vertices in J_R. This modified solution continues to satisfy (1) for each edge with both endpoints in J. We observed above that edges in E with exactly one endpoint in J are slack.

Suppose we have such a slack edge uv with $u \in J$ and $v \notin J$. Let $\delta \in \{\varepsilon, -\varepsilon\}$ represent the perturbation added to $\hat{x}_u$. If uv satisfies (i), then (1) is satisfied for uv by selecting ε of small enough magnitude so that $\delta \leq \hat{x}_u + \hat{x}_v - \hat{z}_{uv}$, where the right-hand side is positive as desired. If uv satisfies (ii), then we add δ to z_{uv} as well. This ensures that (1) is satisfied for uv, and we may select ε of small enough magnitude to satisfy (4); this does not affect the feasibility of any other constraint, even those for other edges.

In addition to the above constraints on the selection of ε, we may select ε of small enough magnitude to satisfy (3), since vertices in J have fractional $\hat{x}$ value. Let the components of the vector $\boldsymbol{\varepsilon} \in \mathcal{R}^{V \times E}$ correspond to the perturbation described above. We have indeed shown that $(\hat{x}, \hat{z}) + \boldsymbol{\varepsilon}$ is feasible for all constraints except possibly (2); however, we have also shown that we may select ε so that $(\hat{x}, \hat{z}) - \boldsymbol{\varepsilon}$ is feasible for these constraints as well. These two solutions average to $(\hat{x}, \hat{z})$, and since $\varepsilon \neq 0$ they are distinct. In fact if we have $|J_L| = |J_R|$, then observe that both of these solutions satisfy (2) as well and are feasible, contradicting the assumption that $(\hat{x}, \hat{z})$ is an extreme point.

To finish the proof, suppose we have two disjoint vertex sets $S, T \subseteq \hat{V}$ akin to J from above; specifically, any edge in E with exactly one endpoint in some S or T is slack. We may independently select $\varepsilon_S \neq 0$ and $\varepsilon_T \neq 0$ for S and T, respectively, so that the only potentially violated constraint is (2). From above, we may assume $|S_L| \neq |S_R|$ and $|T_L| \neq |T_R|$. Thus, in order to satisfy (2), we need only choose ε_S and ε_T so that:

$$\varepsilon_S \cdot (|S_L| - |S_R|) + \varepsilon_T \cdot (|T_L| - |T_R|) = 0 \ ,$$

which can be done so that ε_S and ε_T have arbitrarily small magnitude as required. Thus $(\hat{x}, \hat{z})$ is a convex combination of the feasible solutions $(\hat{x}, \hat{z}) \pm (\boldsymbol{\varepsilon_S} + \boldsymbol{\varepsilon_T})$ — a contradiction.

If the tight edges induced more than one connected component in $\hat{G}$ (possibly one with just an isolated vertex), then two of these components could play the role of S and T above, which cannot exist. □

We will use this lemma to show the structure of fractional entries in an extreme point solution $(\hat{x}, \hat{z})$. As above let $\hat{V}$ correspond to the vertices with fractional $\hat{x}$ value. Let $\hat{V}_0$ and $\hat{V}_1$ correspond to vertices with $\hat{x}$ value 0 and 1, respectively.

Theorem 2. *The fractional entries of $\hat{x}$ take one of two values, α or $1 - \alpha$, for some $\alpha \in (0, 1)$. Moreover,*

$$\min\{|\hat{V}_L|, |\hat{V}_R|\} + |\hat{V}_1| \leq k \ .$$

Proof. By the above lemma, there is a tree consisting of tight edges spanning the vertices in $\hat{V}$. By the definition of a tight edge uv, we have that $\hat{x}_u + \hat{x}_v = \hat{z}_{uv}$ and $\hat{z}_{uv} = 1$. Select a root r for this tree and suppose $\hat{x}_r = \alpha$. Now the first claim of the theorem follows, since it is apparent that vertices with odd distance to r must have $\hat{x}$ value $1 - \alpha$ and those with even distance to r have value α.

Suppose that the vertices in $\hat{V}_L$ have value α. Then the budget constraint, (2) becomes:

$$\alpha \cdot |\hat{V}_L| + (1 - \alpha) \cdot |\hat{V}_R| + |\hat{V}_1| \leq k \ ,$$

from which the second claim follows. □

One consequence of Theorem 2 is that $\hat{V}_1$ together with the smaller of $\hat{V}_L$ and $\hat{V}_R$ is a feasible solution. Now, let us compose two feasible solutions that will help us in designing the approximation algorithm. Let A be the set of edges between $\hat{V}_L$ and $\hat{V}_R$. Let B_L be the set of edges with one endpoint in $\hat{V}_L$ and one endpoint in $\hat{V}_0$, and let B_R be defined analogously. Note that we will always select the vertices in $\hat{V}_1$ and ignore the edges with at least one endpoint in $\hat{V}_1$ below. For the sake of exposition, suppose $|\hat{V}_L| \leq |\hat{V}_R|$ and that each vertex in $\hat{V}_L$ has $\hat{x}$ value α (hence those in $\hat{V}_R$ have value $1-\alpha$).

1. Select $\hat{V}_L$ and augment this by greedily selecting an additional $k - |\hat{V}_L|$ vertices from $\hat{V}_R$ with largest coverage. This solution has coverage at least $w(A)+w(B_L)+\frac{k-|\hat{V}_L|}{|\hat{V}_R|} \cdot w(B_R)$.
2. Greedily select k vertices from $\hat{V}_R$ with largest coverage. This solution has coverage at least $\frac{k}{|\hat{V}_R|} \cdot (w(A) + w(B_R))$.

The above are derived by applying the following lemma. For a vertex $v \in V$ let $w(v) = \sum_{e \in \delta(v)} w(e)$, where $\delta(v)$ denotes the set of edges of G, that are incident to v. As with edge sets, for a set $S \subseteq V$, let $w(S) = \sum_{u \in S} w(u)$.

Lemma 4. *Let $S \subseteq V$ be an independent set of vertices. The t vertices in S with largest coverage have a collective coverage of at least $t \cdot \frac{w(S)}{|S|}$.*

Proof. Let $T \subseteq S$ be the t vertices in S with largest coverage. Suppose, for the sake of contradiction, that $w(T) < t \cdot \frac{w(S)}{|S|}$. Then some vertex in T has coverage less than $\frac{w(S)}{|S|}$. Therefore by choice of T, every vertex in $S - T$ has coverage less than $\frac{w(S)}{|S|}$. This implies that $w(S) = w(T) + w(S-T) < t \cdot \frac{w(S)}{|S|} + (|S|-t) \cdot \frac{w(S)}{|S|} = w(S)$. □

Applying the lemma with $S = \hat{V}_R$ and $t = k - |\hat{V}_L|$ and $t = k$ yield the bounds from (1) and (2), respectively. As in the proof of Theorem 2, the budget constraint (2) from the LP (ignoring $\hat{V}_1$) yields $k \geq \alpha \cdot |\hat{V}_L| + (1-\alpha) \cdot |\hat{V}_R|$. Substituting this bound for k into the above coverage bounds yield the following lower bounds on the respective coverage obtained by our two solutions:

1. $W_1 = w(A) + w(B_L) + (1-\alpha) \cdot \beta \cdot w(B_R)$, and
2. $W_2 = (1 - \alpha \cdot \beta) \cdot (w(A) + w(B_R))$,

where $\beta = 1 - \frac{|\hat{V}_L|}{|\hat{V}_R|}$. Note that we may assume that $\beta \in (0,1)$. If $\beta = 0$ then $|\hat{V}_L| = |\hat{V}_R| \leq k$, and the set with better coverage is optimal. If $\beta = 1$ then $\hat{V}_L = \emptyset$, hence $B_L = \emptyset$. Since our extreme point solution $\hat{x}$ is not integral, we have $\alpha \in (0,1)$, hence the edge set $A = \emptyset$ as well. The coverage of Solution (2) in this case is $(1-\alpha) \cdot (w(A) + w(B_R)) = w(A) + \alpha w(B_L) + (1-\alpha) \cdot w(B_R) \geq OPT$.

We have shown that we can achieve a coverage of $\max\{W_1, W_2\}$. All we need is to show that the left-side of the inequality given below, an upper bound on the performance of our algorithm, is at most $\frac{9}{8}$. In order to prove this, consider a nonlinear program that seeks to maximize the ratio,

$$\frac{w + \alpha \cdot w_L + (1-\alpha) \cdot w_R}{\max\{W_1, W_2\}} \geq \frac{OPT}{\max\{W_1, W_2\}} \ .$$

The program has variables w, w_L, and w_R which correspond to values for $w(A)$, $w(B_L)$, and $w(B_R)$ respectively. Similarly there is a variable m corresponding to $\max\{W_1, W_2\}$.

$$\begin{aligned} \rho = \max \frac{w + \alpha \cdot w_L + (1-\alpha) \cdot w_R}{m} & \\ w + w_L + (1-\alpha) \cdot \beta \cdot w_R &\leq m \\ (1 - \alpha \cdot \beta) \cdot (w + w_R) &\leq m \\ w, w_L, w_R &\geq 0 \\ \alpha, \beta &\in (0,1) \ . \end{aligned}$$

The program finds a worst-case setting of the parameters for our algorithm, and the optimal value ρ is an upper bound on the worst-case performance of our algorithm. We simplify the program a bit by letting $x = \frac{w}{m}$, $y = \frac{w_L}{m}$, and $z = \frac{w_R}{m}$.

$$\begin{aligned} \rho = \max(x + \alpha \cdot y + (1-\alpha) \cdot z) & \\ x + y + (1-\alpha) \cdot \beta \cdot z &\leq 1 \\ (1 - \alpha \cdot \beta) \cdot (x + z) &\leq 1 \\ x, y, z &\geq 0 \\ \alpha, \beta &\in (0,1) \ . \end{aligned}$$

For any fixed values of α and β, we consider the above program as a linear program, whose target function depends on α and β: $\rho = \rho(\alpha, \beta)$. Now, in order to simplify our analysis, we drop the non-negativity constraints on x and z, obtaining an upper bound on $\rho(\alpha, \beta)$.

$$\begin{aligned} \rho(\alpha, \beta) \leq \max(x + \alpha \cdot y + (1-\alpha) \cdot z) & \\ x + y + (1-\alpha) \cdot \beta \cdot z &\leq 1 \\ (1 - \alpha \cdot \beta) \cdot x + (1 - \alpha \cdot \beta) \cdot z &\leq 1 \\ y &\geq 0 \ . \end{aligned}$$

Note that the above LP is feasible and bounded for $\alpha, \beta \in (0,1)$, since the first constraint implies an upper bound of $\frac{1}{\beta}$ on the objective function. Hence, we may assume without loss of generality that we seek an extreme-point solution, in which case all three constraints must be tight since we have three variables. Thus $y = 0$, and we may solve the following linear system to obtain x and z in terms of α and β:

$$\begin{aligned} x + (1-\alpha) \cdot \beta \cdot z &= 1 \\ (1 - \alpha \cdot \beta) \cdot x + (1 - \alpha \cdot \beta) \cdot z &= 1 \ . \end{aligned}$$

Taking into account these values, the objective function becomes,

$$\begin{aligned} f(\alpha,\beta) &= x + \alpha \cdot y + (1-\alpha) \cdot z \\ &= (1-\alpha) \cdot (x+z) + \alpha \cdot x \\ &= \frac{(1-\alpha) \cdot [1-(1-\alpha)\cdot\beta] + \alpha \cdot (1-\beta)}{(1-\alpha\cdot\beta) \cdot [1-(1-\alpha)\cdot\beta]} \ . \end{aligned}$$

Thus we obtain an upper bound on our worst-case approximation ratio by maximizing this quantity with respect to α and β:

$$\rho = \max_{\alpha,\beta\in[0,1]} \rho(\alpha,\beta) \le \max_{\alpha,\beta\in[0,1]} f(\alpha,\beta) \ .$$

Maximizing $f(\alpha,\beta)$ is equivalent to maximizing

$$g(\alpha,\beta) = 1 - \frac{1}{f(\alpha,\beta)} = \frac{\alpha\cdot(1-\alpha)\cdot\beta\cdot(1-\beta)}{(1-\alpha)\cdot[1-(1-\alpha)\cdot\beta] + \alpha\cdot(1-\beta)} \ .$$

Maximizing $g(\alpha,\beta)$ is equivalent to minimizing

$$h(\alpha,\beta) = \frac{1}{g(\alpha,\beta)} = \frac{1}{1-\beta} + \frac{1}{\alpha\cdot(1-\alpha)\cdot\beta} \ .$$

In minimizing $h(\alpha,\beta)$ for any fixed value of $\beta \in (0,1)$, we see that $\alpha = \frac{1}{2}$, and some elementary calculus yields that $h(\frac{1}{2},\beta)$ is minimized over $\beta \in (0,1)$ at $\beta = \frac{2}{3}$. Evaluating f at these worst-case values of $\alpha = \frac{1}{2}$ and $\beta = \frac{2}{3}$ yields $\rho \le \frac{9}{8}$.

Remark 1. It is natural to wonder whether the $\frac{8}{9}$-approximation factor in the above algorithm can be improved. It turns out that the integrality gap of the linear program [22] that we considered is $\frac{8}{9}$. It is an indication that using this linear program alone, one cannot improve the approximation factor. In order to see this, consider the tree from figure 3, and let the vertex budget $k = 2$. Observe that in this tree with two vertices one can cover four edges. However, the linear program corresponding to this tree has a fractional solution $(\hat{x},\hat{z})$ whose cost is $\frac{9}{2}$. In order to construct such a solution, for each vertex u let $\hat{x}_u = \frac{1}{2}$, if u is not a leaf, and is zero, otherwise. For each edge e let $\hat{z}_e = \frac{1}{2}$, if e is incident to a leaf, and is one, otherwise. It is not hard to see that this fractional solution is a feasible solution, and its cost is $\frac{9}{2}$, which implies that the integrality gap of our linear program is $\frac{4}{\frac{9}{2}} = \frac{8}{9}$.

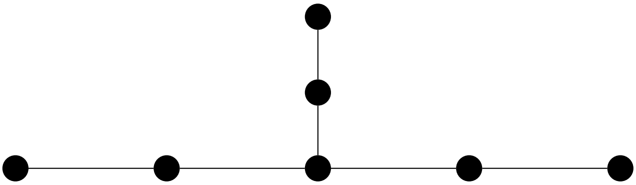

Fig. 3. An example showing that the integrality gap is $\frac{8}{9}$

7 Conclusion

In this paper, we studied the partial vertex cover, the budgeted maximum coverage problems and their weighted analogues on bipartite graphs. We proved that the PVCB problem is **NP-hard**. Then we presented an $\frac{8}{9}$-approximation algorithm for the EBMCB problem, which is computationally identical to the PVCB problem.

From our perspective, the following lines of research appear promising:

- Determining whether the PVCB problem is **APX-hard** or not.
- Obtaining an α-approximation algorithm for the PVCB problem where $\alpha \leq \frac{4}{3}$.
- Considering the main problems in the class of bounded-degree graphs.

References

1. Ageev, A.A., Sviridenko, M.I.: Approximation algorithms for maximum coverage and max cut with given sizes of parts. In: Cornuéjols, G., Burkard, R.E., Woeginger, G.J. (eds.) IPCO 1999. LNCS, vol. 1610, pp. 17–30. Springer, Heidelberg (1999)
2. Bar-Yehuda, R.: Using homogeneous weights for approximating the partial cover problem. J. Algorithms 39(2), 137–144 (2001)
3. Bar-Yehuda, R., Flysher, G., Mestre, J., Rawitz, D.: Approximation of Partial Capacitated Vertex Cover. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 335–346. Springer, Heidelberg (2007)
4. Bläser, M.: Computing small partial coverings. Inf. Process. Lett. 85(6), 327–331 (2003)
5. Bshouty, N.H., Burroughs, L.: Massaging a linear programming solution to give a 2-approximation for a generalization of the vertex cover problem. In: Morvan, M., Meinel, C., Krob, D. (eds.) STACS 1998. LNCS, vol. 1373, pp. 298–308. Springer, Heidelberg (1998)
6. Caskurlu, B., Gehani, A., Bilgin, C.C., Subramani, K.: Analytical models for risk-based intrusion response. Computer Networks (Special issue on Security/Identity Architecture) 57(10), 2181–2192 (2013)
7. Dinur, I., Safra, S.: On the hardness of approximating minimum vertex cover. Ann. of Math. (2) 162(1), 439–485 (2005)
8. Feige, U.: A threshhold of $ln(n)$ for approximating Set Cover. In: Proc. of the 28th Annual ACM Symposium on the theory of Computing, pp. 314–318 (1996)
9. Gandhi, R., Khuller, S., Srinivasan, A.: Approximation algorithms for partial covering problems. J. Algorithms 53(1), 55–84 (2004)
10. Guo, J., Niedermeier, R., Wernicke, S.: Parameterized Complexity of Generalized Vertex Cover Problems. In: Dehne, F., López-Ortiz, A., Sack, J.-R. (eds.) WADS 2005. LNCS, vol. 3608, pp. 36–48. Springer, Heidelberg (2005)
11. Hochbaum, D.S.: The t-Vertex Cover Problem: Extending the Half Integrality Framework with Budget Constraints. In: Jansen, K., Rolim, J.D.P. (eds.) APPROX 1998. LNCS, vol. 1444, pp. 111–122. Springer, Heidelberg (1998)
12. Karakostas, G.: A better approximation ratio for the vertex cover problem. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1043–1050. Springer, Heidelberg (2005)
13. Karp, R.: Reducibility among combinatorial problems. In: Miller, R., Thatcher, J. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press (1972)
14. Khot, S., Regev, O.: Vertex cover might be hard to approximate to within $2 - \epsilon$. J. Comput. Syst. Sci. 74, 335–349 (2008)

15. Khuler, S., Moss, A., Naor, J.(S.): The Budgeted Maximum Coverage problem. Inform. Process. Lett. 70(1), 39–45 (1999)
16. Kneis, J., Langer, A., Rossmanith, P.: Improved upper bounds for partial vertex cover. In: Broersma, H., Erlebach, T., Friedetzky, T., Paulusma, D. (eds.) WG 2008. LNCS, vol. 5344, pp. 240–251. Springer, Heidelberg (2008)
17. Kneis, J., Mölle, D., Richter, S., Rossmanith, P.: Intuitive algorithms and *t*-vertex cover. In: Asano, T. (ed.) ISAAC 2006. LNCS, vol. 4288, pp. 598–607. Springer, Heidelberg (2006)
18. Kneis, J., Mölle, D., Rossmanith, P.: Partial vs. Complete Domination: *t*-Dominating Set. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) SOFSEM 2007. LNCS, vol. 4362, pp. 367–376. Springer, Heidelberg (2007)
19. Könemann, J., Parekh, O., Segev, D.: A unified approach to approximating partial covering problems. Algorithmica 59(4), 489–509 (2011)
20. Mestre, J.: A primal-dual approximation algorithm for partial vertex cover: Making educated guesses. Algorithmica 55(1), 227–239 (2009)
21. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. System Sci. 43(3), 425–440 (1991)
22. Vazirani, V.V.: Approximation Algorithms. Springer-Verlag New York, Inc., New York (2001)