Parameterized Algorithms for Partial Vertex Covers in Bipartite Graphs

Vahan Mkrtchyan¹, Garik Petrosyan², K. Subramani³(✉), and Piotr Wojciechowski³

¹ School of Advanced Studies, Gran Sasso Science Institute, L’Aquila, Italy
vahan.mkrtchyan@gssi.it
² Department of Informatics and Applied Mathematics, Yerevan State University, Yerevan, Armenia
garik.petrosyan.1@gmail.com
³ West Virginia University, Morgantown, WV, USA
ksmani@csee.wvu.edu, pwojciec@mix.wvu.edu

Abstract. In this paper, we discuss parameterized algorithms for variants of the partial vertex cover problem. Recall that in the classical vertex cover problem (VC), we are given a graph \( G = (V, E) \) and a number \( K \) and asked if we can cover all of the edges in \( E \), using at most \( K \) vertices from \( V \). In the partial vertex cover problem (PVC), in addition to the parameter \( K \), we are given a second parameter \( K' \) and the question is whether we can cover at least \( K' \) of the edges in \( E \) using at most \( K \) vertices from \( V \). The weighted generalizations of the VC and PVC problems are called the weighted vertex cover (WVC) and the partial weighted vertex cover problem (WPVC) respectively. In the WPVC problem, we are given two parameters \( R \) and \( L \), associated respectively with the vertex set \( V \) and edge set \( E \) of the graph \( G \). Additionally, we are given non-negative integral weight functions for the vertices and the edges. The goal then is to cover edges of total weight at least \( L \), using vertices of total weight at most \( R \). (In the WVC problem, the goal is to cover all the edges with vertices whose total weight is at most \( R \)). This paper studies several variants of the PVC problem and establishes new results from the perspective of fixed-parameter tractability and \( \text{W[1]-hardness} \). We also introduce a new problem called the partial vertex cover with matching constraint and show that it is fixed-parameter tractable for a certain class of graphs.

1 Introduction

In this paper, we study several variants of the vertex cover problem (VC), from the perspectives of parameterized algorithm design and complexity. In particular,

The work of the first author has been partially supported by the Italian MIUR PRIN 2017 Project ALGADIMAR “Algorithms, Games, and Digital Markets.” The research of the third author has been supported in part by the Air-Force Office of Scientific Research through Grant FA9550-19-1-017 and in part by the Air-Force Research Laboratory, Rome through Contract FA8750-17-S-7007.

© Springer Nature Switzerland AG 2020
L. Gasieniec et al. (Eds.): IWOCA 2020, LNCS 12126, pp. 395–408, 2020.
https://doi.org/10.1007/978-3-030-48966-3_30
we consider the partial vertex cover problem (PVC), wherein the goal is to cover a certain threshold of edges (as opposed to all of the edges) using the fewest number of vertices. We also look into weighted variants of this problem. Our primary focus is on bipartite graphs. In this case, the problem is called the partial vertex cover problem on bipartite graphs (PVCB). The PVCB problem is an important problem with a number of applications in computer security [9] and risk assessment [6]. In the weighted partial vertex cover problem (WPVC), we are given a graph \( G = (V, E) \), cost function \( c: V \to \mathbb{N} \), profit function \( p: E \to \mathbb{N} \), and positive integers \( R \) and \( L \). The goal is to check whether there is a subset \( V' \subseteq V \) of cost at most \( R \), such that the total profit of edges covered by \( V' \) is at least \( L \). In this paper we study the fixed-parameter tractability of WPVC in bipartite graphs (WPVCB). By extending the methods of Amini et al. [1], we show that WPVCB is Fixed-Parameter Tractable (FPT) with respect to \( R \) if \( c \equiv 1 \). On the negative side, it is \( \text{W[1]} \)-hard for arbitrary \( c \), even when \( p \equiv 1 \). In particular, WPVCB is \( \text{W[1]} \)-hard parameterized by \( R \). We complement this negative result by proving that for bounded-degree graphs WPVC is FPT with respect to \( R \). Additionally, we show that WPVC is FPT with respect to \( L \). Finally, we discuss a variant of PVCB in which the edges covered are constrained to include a large matching and derive a parameterized algorithm for this version of the problem.

The rest of this paper is organized as follows: The problems studied in this paper are formally described in Sect. 2. In Sect. 3, we discuss the motivation for our work and mention related approaches in the literature. Our main results are described in Sect. 4. A variant of the partial vertex cover problem with applications to computational social choice is detailed in Sect. 5. In Sect. 6, we examine the parameterized complexity of the WPVCB problem when there is a separate budget for each partition of the vertices. We conclude in Sect. 7, by summarizing our results and outlining avenues for future research.

2 Statement of Problems

We focus on finite, undirected graphs that have no loops or parallel edges. As usual, the degree of a vertex is the number of edges of the graph that are incident with it. The maximum degree of the graph \( G \) is just the maximum of all degrees of vertices of \( G \). A graph \( G = (V, E) \) is bipartite, if its vertex set can be partitioned into two sets \( V_1 \) and \( V_2 \), such that each edge of \( G \) connects a vertex from \( V_1 \) to one from \( V_2 \). Given a graph \( G = (V, E) \), and a set \( S \subset V \) of vertices, an edge \( (i, j) \in E \) is covered by \( S \) if \( i \in S \) or \( j \in S \). Let \( E(S) \) be the set of edges of \( G \) that are covered by \( S \). The classical vertex cover problem (VC) is defined as finding the smallest set \( S \) of vertices of the input graph \( G \), so that \( E(S) = E \). The vertex cover problem is a well-known \textbf{NP-complete} problem [16]. In this paper, we study the following variants of VC:

1. The partial vertex cover problem (PVC) -
Definition 1. Given an undirected graph $G = (V, E)$, vertex-cardinality parameter $k_1$, and edge-cardinality parameter $k_2$, is there a subset $V'$ of $V$, such that $|V'| \leq k_1$ and $V'$ covers at least $k_2$ edges?

2. The weighted partial vertex cover problem (WPVC) -

Definition 2. Given an undirected graph $G = (V, E)$, weight-functions $c : V \to \mathbb{N}$ and $p : E \to \mathbb{N}$, vertex-weight parameter $R$, and edge-weight parameter $L$, is there a subset $S$ of $V$, such that $\sum_{v \in S} c(v) \leq R$ and $\sum_{e \in E(S)} p(e) \geq L$?

3. The partial vertex cover problem on bipartite graphs (PVCB) - This is the restriction of the partial vertex cover problem (PVC) to bipartite graphs.
4. The weighted partial vertex cover problem on bipartite graphs (WPVCB) - This is the restriction of the weighted partial vertex cover problem (WPVC) to bipartite graphs.
5. The 2-budget partial vertex cover problem on bipartite graphs (2-PVCB) -

Definition 3. Given an undirected bipartite graph $G = (V, U, E)$, vertex-cardinality parameters $R_V$ and $R_U$, and an edge-cardinality parameter $L$, is there a subset $S_V$ of $V$ and a subset $S_U$ of $U$, such that $|S_V| \leq R_V$, $|S_U| \leq R_U$, and the number of edges covered by $S_V \cup S_U$ is at least $L$?

6. The partial vertex cover problem with matching constraint (PVCBM) - This is a variant of the PVCB problem, in which we are given a third parameter $k_3$ and the goal is to find a vertex subset of size at most $k_1$, covering at least $k_2$ edges, such that the edges covered include a matching of size at least $k_3$.

The principal contributions of this paper are as follows:

1. Fixed-parameter tractability of the WPVCB problem with respect to $R$ when vertex weights are identically 1.
2. $\textbf{W}[1]$-hardness of the WPVCB problem with respect to $R$.
3. Fixed-parameter tractability of the weighted partial vertex cover problem in bounded degree graphs (not necessarily bipartite) with respect to $R$.
4. $\textbf{W}[1]$-hardness of the 2-PVCB problem with respect to $R_U$ and $R_V$.
5. Fixed-parameter tractability of the WPVC problem with respect to $L$.
6. A parameterized algorithm for the matching variant of the PVCB problem with respect to $k_1$.

3 Motivation and Related Work

When the weight functions $c$ and $p$ are identically one (i.e. $c \equiv p \equiv 1$), we get the well-known partial vertex cover problem (PVC). PVC represents a natural theoretical generalization of VC and is motivated by practical applications. Flow-based risk-assessment models in computational systems, for example, can be viewed as instances of PVC [6]. In particular PVC has applications to computer security even when restricted to bipartite graphs [9].
VC is polynomial-time solvable in bipartite graphs. However, the computational complexity of PVC in bipartite graphs has remained open until it was recently shown to be NP-hard [3, 9, 10, 14].

VC has also been extensively studied from the perspective of approximation algorithms. Many 2-approximation algorithms for VC are known [31]. There is an approximation algorithm for the VC problem which has an approximation factor of \(2 - \theta(\sqrt{\log n})\) [15]. This is the best known algorithm. The VC problem is also known to be APX-complete [27]. Moreover, it cannot be approximated within a factor of 1.3606 unless P = NP [12]. This lower bound was recently improved to \((\sqrt{2} - \epsilon)\) for any \(\epsilon > 0\) in [18]. If the unique games conjecture is true, then VC cannot be approximated within any constant factor smaller than 2 [17]. In [28], a \((\frac{2}{3} + \epsilon)\)-approximation algorithm is designed for WPVC for each \(\epsilon > 0\) when the input graph is bipartite.

All hardness results for the VC problem directly apply to the PVC problem because the PVC problem is an extension of the VC problem. Since the 1990’s the PVC problem and the partial-cover variants of similar graph problems have been extensively studied [7, 8, 20, 21, 24, 29]. In particular, there is an \(O(n \cdot \log n + m)\)-time 2-approximation algorithm for PVC based on the primal-dual method [21], as well as a combinatorial 2-approximation algorithm [22]. Both of these algorithms are for a more general soft-capacitated version of PVC. There are several older 2-approximations resulting from different approaches [5, 8, 13, 19]. Let us also note that the WPVC problem for trees is studied in [23], which provides an FPTAS for the problem. Additionally, the paper provides a polynomial time algorithm in the case when the vertices are unweighted.

Another problem with a close relationship to WPVC is the budgeted maximum coverage problem (BMC). In this problem one tries to find a min-cost subset of vertices, such that the profit of covered edges is maximized. It can be shown that both problems are equivalent from the perspective of exact solvability. The BMC problem for sets (not necessarily graphs) admits a \((1 - \frac{1}{e})\)-approximation algorithm [25]. However, special cases that beat this bound are rare. The pipage rounding technique gives a \(\frac{3}{4}\)-approximation algorithm for the BMC problem on graphs [2]. This is improved to \(\frac{4}{5}\) for bipartite graphs [4]. Finally, in [9, 10], an \(\frac{8}{5}\)-approximation algorithm for the problem is presented when the input graph is bipartite and the vertices are unweighted. The result is based on the linear-programming formulation of the problem, and the constant \(\frac{8}{5}\) matches the integrality gap of the linear program used in the formulation. More recently, Vangelis Paschos has described a polynomial time approximation scheme for the edge-weighted maximum coverage problem on bipartite graphs [30].

In this paper, we address these problems from the perspective of fixed-parameter tractability. Recall that a combinatorial problem \(\Pi\) is said to be fixed-parameter tractable with respect to a parameter \(k\), if there is an algorithm for solving \(\Pi\), whose running time is bounded by \(f(k) \cdot \text{size}^{O(1)}\). Here \(f\) is a computable function of \(k\), and \text{size} is the length of the input. From the perspective of FPT, PVC is in some sense more difficult than VC. For instance, PVC
is \(\textbf{W}[1]\)-complete with respect to \(R\), the number of vertices in the cover \([11]\), while VC is FPT \([11, 26]\).

In \([1]\), the decision version of WPVCB is considered. The authors show that this problem is FPT with respect to the vertex budget \(R\), when the vertices and edges of the bipartite graph are unweighted \([1]\). In this paper, by extending the result of Amini et al. \([1]\), we show that the decision version of WPVCB is FPT with respect to \(R\), if the vertices have cost one, while the edges remain arbitrarily weighted. On the negative side, the problem is \(\textbf{W}[1]\)-hard for arbitrary vertex weights, even when edges have profit one. We complement this negative result by proving that for bounded-degree graphs WPVC is FPT with respect to \(R\). The same result holds for WPVC with respect to \(L\). We finish the paper by obtaining an FPT result for an extension of PVCB. Terms and concepts that we do not define can be found in \([11]\).

## 4 Main Results

In this section, we present our results. Our goal is to investigate the fixed-parameter tractability of the WPVCB problem.

When \(c\) and \(p\) are identically one (i.e. \(c \equiv p \equiv 1\)), we get the PVCB problem. When \(c\) is identically one, we get EPVCB. Finally, when \(p\) is identically one, we get the VPVCB problem. We will also use the same scheme of notations when the input graph need not be bipartite. In \([1]\), PVCB is considered and it is shown that the problem is FPT with respect to \(R\). We strengthen this result.

**Theorem 1.** EPVCB is FPT with respect to \(R\).

*Proof.* Roughly speaking, we obtain the result with the approach of \([1]\) by considering the weighted degree instead of the usual degree. Below we present the technical details.

Assume that we have an instance \(I\) of EPVCB. For a vertex \(v\) of \(B\), let \(\delta(v)\) be the set of edges of \(B\) incident with \(v\). Define the set \(S\) of vertices of \(B\) as follows:

\[
S = \left\{ v \in V(B) : p(\delta(v)) \geq \frac{L}{R} \right\}.
\]

Case 1: \(|S| \geq 2 \cdot R\). Consider the subgraph \(H\) of \(B\) induced by \(S\). Since \(B\) is bipartite, \(H\) is bipartite, too. Let \((X, Y)\) be the bipartition of \(H\), and assume that \(|X| \geq |Y|\). Since \(|X| + |Y| = |S| \geq 2 \cdot R\), we have \(|X| \geq R\). Take any \(R\) vertices of \(X\). Observe that \(X\) is an independent set in \(B\), hence these \(R\) vertices will cover at least \(L\) edges. Thus, the total profit is at least \(L\). This means that \(I\) is a yes-instance.

Case 2: \(|S| < 2 \cdot R\). Observe that any feasible solution to \(I\) must intersect \(S\). Hence, we do recursive guessing, that is, we try each vertex of \(S\) one by one as a possible vertex of the feasible solution.

In the Case 1, the algorithm will run in polynomial time, so the most expensive case is Case 2. Since the number of vertices in a feasible solution is at most \(R\), we have that the depth of the recursion is at most \(R\). Hence the total running time of our algorithm is \(O((2 \cdot R)^R \cdot \text{size}^{O(1)})\).

\[\square\]
Our next result shows that WPVCB and VPVCB are \textbf{W[1]-hard}. Our reduction is from the multi-colored clique problem \cite{11}. It is formulated as follows:

**Multi-colored Clique:** Given a graph $G$, positive integer $k$, and a partition $(V_1, \ldots, V_k)$ of the vertices of $G$, the goal is to check whether $G$ contains a $k$-clique $Q$, such that $Q$ contains exactly one vertex from each $V_j$ for $j = 1, \ldots, k$.

Multi-colored clique is a well-studied problem which is known to be \textbf{W[1]-hard} with respect to the number of partitions $k$. Observe that an edge $e$ connecting two vertices from $V_i$ $1 \leq i \leq k$ does not lie in a feasible clique. Thus, without loss of generality, we can assume that for $i = 1, \ldots, k$, $V_i$ is an independent set of vertices.

**Theorem 2.** WPVCB is \textbf{W[1]-hard} parameterized by $R$.\footnote{We are grateful to Magnus Wahlström for providing us with a proof of this theorem.}

**Proof.** We construct an FPT-reduction from \textsc{multi-colored clique}. Let $G = (V, E)$ be an instance of this problem with vertices partitioned as $V = V_1 \cup \ldots \cup V_k$. We create a bipartite graph $B = (U' \cup V' \cup Z, E')$ as follows. Let $U'$ and $V'$ be two copies of $V$, and let $V = V' = \{v_1, \ldots, v_n\}$, $U' = \{u_1, \ldots, u_n\}$, where for each $i \in [n]$, $u_i$ is a copy of $v_i$. Here as usual $[n] = \{1, \ldots, n\}$. For a vertex $v \in V$ let $\chi(v)$ be its color, i.e., $\chi(v) = i$ if $v \in V_i$, and extend this to $U' \cup V'$ so that $\chi(u_i) = \chi(v_i) + k$ ($V'$ inherits the original $\chi$ values on $v_1, \ldots, v_n$). For a vertex $x \in U' \cup V'$, let the cost of $x$ be $c(x) = 2^{\chi(x)}$. Add an edge $u_iv_j$ to $B$ if either $\chi(u_i) = \chi(v_j) + k$ and $i \neq j$, or $\chi(u_i) \neq \chi(v_j) + k$ and $u_iv_j \notin E(G)$. Give all these edges profit 1. Observe that a selection of one vertex from every color class of $U' \cup V'$ forms an independent set in $B$, if and only if it corresponds to two copies of a $k$-clique in $G$. Add two additional vertices $z_1$ and $z_2$, let $Z = \{z_1, z_2\}$ and give both the cost $2^{2-k+1}$. Finally, join every vertex $x \in U'$ to $z_2$, every vertex $x \in V'$ to $z_1$ and give these edges a profit value so that the total profit of all edges incident with $x$ equals $(2^{\chi(x)} \cdot (n + 1) + 5^{\chi(x)})$. (This is clearly possible, since the total profit of all previously created edges incident with $x$ is bounded by $n$.)

Set the budgets of the instance as vertex budget $R = \sum_{i=1}^{2-k} 2^i = 2^{2-k+1} - 2$ and profit threshold $L = \sum_{i=1}^{2-k} (2^i \cdot (n + 1) + 5^i) = (n + 1) \cdot R + \left(\frac{5}{4}\right) \cdot (2^{2-k} - 1)$. This finishes the instance description. It is clear that the construction can be performed in polynomial time, and the budget $R$ is a function of $k$.

In Fig. 1, a graph $G$ and the bipartite graph $B$ obtained after the reduction are given. In this example, $k = 2$, $V_1 = \{v_1, v_3\}$, $V_2 = \{v_2, v_4\}$ and $\chi(v_1) = \chi(v_3) = 1, \chi(v_2) = \chi(v_4) = 2, \chi(u_1) = \chi(u_3) = 3, \chi(u_2) = \chi(u_4) = 4$, and the profits of edges not incident with $z_1$ or $z_2$ are 1, finally, the edges incident with $z_1$ or $z_2$ are chosen so that the total profit of edges incident with any $x \in V(B) \setminus \{z_1, z_2\}$ is $(5 \cdot 2^{\chi(x)} + 5^{\chi(x)})$. For example, consider the edge $z_1v_1$. Recall that $\chi(v_1) = 1$, thus the total profit of all edges incident with $v_1$ should be $5 \cdot 2^1 + 5^1 = 15$. Since the profit of $v_1u_3$ is 1, the profit of the edge $z_1v_1$ is 14.
It remains to show that \((B,R,L)\) is a positive instance of WPVCB if and only if \(G\) has a multi-colored clique.

From the multi-colored clique problem to the partial vertex cover problem. Let \(X \subseteq V(G)\) be a multi-colored \(k\)-clique, and let \(S = \{u_i \mid u_i \in X\} \cup \{v_i \mid v_i \in X\}\). Since \(S\) contains one vertex for every color class of \(B\) its total cost equals \(R\), and since it induces an independent set in \(B\) the total profit of the edges covered equals \(L\).

From the Partial Vertex Cover Problem to the Multi-colored Clique Problem. Now for the more challenging part of the argument. We need to argue that the costs and profits balance out so that the only way to select vertices to a total profit of \(L\) is to select one vertex from every color class of \(B\). For this, first observe that for a vertex of class \(i \in [2 \cdot k]\), the ratio of the total profit of its incident edges to its cost is

\[
\frac{2^i \cdot (n+1) + 5^i}{2^i} = (n+1) + \left(\frac{5}{2}\right)^i.
\]

(1)

Let \(S\) be a partial vertex cover of profit at least \(L\) and cost at most \(R\), and for each \(i \in [2 \cdot k]\) let \(n_i\) be the number of vertices of \(S\) with color class \(i\). For each vertex, we can divide the contributions to the total profit into two parts. The first part corresponds to the \((n+1)\) term in (1), while the second corresponds to the \((\frac{5}{2})^i\) term. From the first part, each vertex \(v\) contributes a profit of at least \((n+1) \cdot c(v)\). It is clear that every selection of cost at most \(R\) contributes a profit of at least \((n+1) \cdot R\), regardless of the distribution \(n_i\). Therefore we focus on the contribution of the second part of the formula, with target profit \(L - (n+1) \cdot R\).

Considering this second part, define \(R_t = \sum_{i=1}^t 2^i\) and \(L_t = \sum_{i=1}^t 5^i\). We show by induction that for every \(t \in [2 \cdot k]\), the largest possible contribution of a selection of cost at most \(R_t\) is \(L_t\) (which of course is achieved by making one selection per color class). For \(t = 1\) this is trivial. Therefore, by induction, let \(t > 1\) and assume that the claim holds for every value \(t' < t\). Let \(n'_i, i \in [t],\)
denote the number of vertices selected in color class $i$, for a selection of total cost at most $R_t$. Then if $n'_t = 0$, the maximum possible profit is $R_t \cdot \left(\frac{5}{2}\right)^{t-1} < 2^{t+1} \cdot \left(\frac{5}{2}\right)^{t-1} = 4 \cdot 5^{t-1} < 5^t$. Therefore, the total profit from the selection $n'_t$ is less than that from a single vertex of class $t$. Therefore $n'_t \geq 1$. But then the remaining budget is $R_t - 2^t = R_{t-1}$, and by induction the optimal selection has $n'_t = 1$ for every $i \in [t]$, completing the induction step. Therefore we may assume $n_i = 1$ for every $i \in [2 \cdot k]$ for our selection $S$.

But then, finally, we observe that a total profit of $L$ is possible only if $S$ is an independent set, since otherwise the profit of some edge will have been double-counted in the above calculations. Therefore, $S$ contains precisely one vertex of each color class of $B$ forming an independent set. In particular, if $v_i \in S$ is a selection in color class $j \in [k]$ for some $i \in [n]$, then the selection in color class $j + k$ must be $u_i$. Also, for every pair of color classes $i, i' \in [k]$ the selections in classes $i$ and $i' + k$ are independent in $B$ and therefore the selections in classes $i$ and $i'$ are neighbors in $G$. Thus $X = S \cap V'$ is a multi-colored clique in $G$, as required. 

\[\square\]

**Corollary 1.** The problem is \textbf{W[1]-hard} also in the variant where all edge profits are 1, i.e., the VPVCB problem is \textbf{W[1]-hard} with respect to $R$.

**Proof.** The only edges of weight more than 1 in the above reduction are the edges connecting to the special vertices $z_i$, and the largest edge weight used is bounded by a function $f(k) \cdot (n + 1)$. Therefore, instead of using edge weights we can in FPT time simply create the corresponding number of pendant vertices for each vertex. These pendants can be given the same weight as the vertices $z_i$. \[\square\]

A class of graphs is said to be bounded-degree, if there is a constant $C$, such that all graphs from the class have maximum degrees at most $C$. It turns out that when the input graphs have bounded-degree and need not be bipartite, we have the following result:

**Theorem 3.** WPVC is fixed-parameter tractable with respect to $R$ for bounded-degree graphs.

**Proof.** Let $I$ be an instance of the WPVC problem, where $G = (V, E)$ is a bounded-degree graph, $c : V \rightarrow \mathbb{N}$, $p : E \rightarrow \mathbb{N}$ are cost and profit functions, and $L, R$ are constants. Assume that for every $v \in V$, we have $d(v) \leq d$. For $i = 1, 2, \ldots, R$, let $M_i(G) = \{ v : c(v) = i \}$ (we disregard the vertices of cost greater than $R$). Choose a vertex $v_i \in M_i(G)$ which has the largest coverage, and let $M$ be the set comprised of chosen vertices $v_i$. The set of vertices of $G$ which have a neighbor from $M$, we denote by $N$. Consider the set $M \cup N$. It is obvious that $|M \cup N| \leq (d + 1) \cdot R$, where $d$ is the bound for the degree.

Let us show that if $I$ is a yes-instance, then we can construct a feasible set, which intersects $M \cup N$. Indeed, let $S$ be a feasible set, which does not have a vertex from $M \cup N$. Then we can replace any vertex $v \in S$ with $v_{c(v)}$ to get a new set $S'$. Since there is no vertex of $S$, which is a neighbor of $v_{c(v)}$, we have
that the profit of \( S' \) does not decrease. Moreover, as \( c(v) = c(v_{c(v)}) \), \( S' \) is also feasible. Observe that \( S' \) intersects \( M \cup N \).

Now we complete the proof by recursively guessing on \( M \cup N \). Since the number of vertices in a feasible set is at most \( R \), we have that the depth of the recursion is at most \( R \), hence the total running time of the algorithm is \( ((d + 1) \cdot R)^R \cdot \text{size}^{O(1)} \).

Exercise 5.11 from [11] implies that PVC is fixed-parameter tractable with respect to \( L \). Below, we strengthen the statement of this exercise by showing that the WPVC problem can be parameterized with respect to \( L \).

**Theorem 4.** WPVC is fixed-parameter tractable with respect to \( L \).

**Proof.** Let \( I \) be an instance of the WPVC problem, where \( G = (V, E) \) is a graph, \( c : V \to \mathbb{N} \), \( p : E \to \mathbb{N} \) are cost and profit functions, and \( L \) and \( R \) are constants. We can assume that no vertex of \( G \) is isolated. Moreover, without loss of generality, we can assume that for any vertex \( v \), we have \( c(v) \leq R \).

For every vertex \( v \) we denote \( p(v) = p(\delta(v)) = \sum_{e \in \delta(v)} p(e) \) the total profit of edges incident with \( v \). We can assume that \( p(v) \leq L - 1 \) for all vertices, as otherwise we will have a feasible solution comprised of one vertex and, as a result, \( I \) is a yes-instance. This, in particular, means that \( d(v) \leq L - 1 \). For \( i = 1, \ldots, L - 1 \) let \( v_i \) be a vertex which has profit \( i \) and for any other vertex \( u \), which has profit \( i \), we have \( c(v_i) \leq c(u) \). Let \( M \) be the set of those vertices \( v_i \). The set of vertices of \( G \) which have a neighbor from \( M \) is denoted by \( N \). It is obvious that \( |M \cup N| \leq (L - 1) + (L - 1)^2 < L^2 \).

Let us show that if \( I \) is a yes-instance and \( S \) is a feasible set in \( I \), then we can construct a feasible set that intersects \( M \cup N \). Indeed, assume that \( S \) does not contain any vertex from \( M \cup N \). Then we can replace any vertex \( v \in S \) by the vertex \( v_{p(v)} \in M \). Since there is no vertex in \( S \) which is a neighbor of \( v_{p(v)} \), it follows that the total vertex cost has not increased, and, as \( p(v) = p(v_{p(v)}) \), \( S' \) is also feasible. Now we complete the proof by recursively guessing on \( M \cup N \). We remove any isolated vertex that may arise in each branch of recursion. Let us show that the depth of the recursion is less than \( 2 \cdot L \). For the sake of contradiction, assume that during the recursive guessing, the algorithm has considered the vertices \( z_1, \ldots, z_{2L} \). Let \( Z = \{ z_1, \ldots, z_{2L} \} \). Since the algorithm has considered these vertices, we have that \( c(Z) \leq R \). Then for the profit of edges covered by \( Z \), we will have the following bound:

\[
p(E(Z)) \geq |E(Z)| \geq \frac{d(z_1) + \ldots + d(z_{2L})}{2} \geq \frac{2 \cdot L}{2} = L.
\]

Thus, \( Z \) is a feasible set, hence \( I \) is a yes-instance. This means that there is no need to consider \( 2 \cdot L \) or more vertices during the recursive guessing. Hence the depth of the recursion is less than \( 2 \cdot L \). Since \( |M \cup N| \leq L^2 \), we have that the running time of our algorithm is bounded by \((L^2)^2 L \cdot \text{size}^{O(1)} = L^{4L} \cdot \text{size}^{O(1)} \). \( \square \)
5 The Matching Problem

We now consider a variant of PVCB. In this variant, we are given a bipartite graph $G$ and three integers $k_1$, $k_2$, and $k_3$. The goal is to check whether there is a subset of at most $k_1$ vertices, that covers at least $k_2$ edges, such that the covered edges contain a matching of size at least $k_3$. This variant is called PVCBM. Clearly, this problem is \textbf{NP-hard}, since when $k_3 = 0$ it results in PVCB. Since PVCB is FPT with respect to $k_1$, it would be interesting to parameterize this new version of the problem with respect to $k_1$. Observe that we can assume that $k_3 \leq k_1$ otherwise the problem is a trivial no-instance.

**Theorem 5.** PVCBM is fixed-parameter tractable with respect to the parameter $k_1$.

**Proof.** Let $PVCB(A,B)$ be an FPT algorithm for PVCB that checks whether there is a subset of $A$ vertices that covers at least $B$ edges of the input bipartite graph. Now, assume that the graph $G$ and the parameters $k_1$, $k_2$, and $k_3$ are given in the matching problem. First, we run $PVCB(k_1,k_2)$. If there is no such subgraph, then the answer to the matching problem is also negative. So we can assume that $PVCB(k_1,k_2)$ returns such a subgraph. Next, by trying $R = 0, 1, \ldots, k_1$ we can find the smallest $R$ for which $PVCB(R,k_2)$ is a yes-instance.

Let $H$ be the edge-induced subgraph on these $\geq k_2$ edges. As usual, let $\nu(G)$ be the size of the largest matching in $G$, and let $\tau(G)$ be the size of the smallest vertex cover in $G$. By the classical König theorem we have $\nu(G) = \tau(G)$ for any bipartite graph $G$.

Observe that we can assume that $R < k_3 \leq k_1$. To see this, observe that $R$ represents the number of vertices required to cover all the edges in $H$. In other words, it is a vertex cover of $H$. Thus, $R = \nu(H)$ and since $H$ is a bipartite subgraph of $G$, $R$ is also the size of a maximum matching in $H$. Thus, if $R \geq k_3$, then the edges in $H$, which number at least $k_2$ can be covered by $R \leq k_1$ vertices and a matching of size $R \geq k_3$ is contained in $H$. Also, observe that if $\tau(G) < k_3$, then we have trivial no-instance, as $G$ contains no matching of size $k_3$. Thus, we can assume that $\nu(G) = \tau(G) \geq k_3$. Since $\tau(H) = R < k_3 \leq \tau(G)$, we have that $E(H) \neq E(G)$. Thus, there is as an edge $e$ lying outside $H$. Add $e$ to $H$. If $\tau(H)$ has increased by adding $e$, define $R := R + 1$, otherwise let $R$ be the same. Repeat this process of adding edges outside $H$. Since $\tau(H) = R < k_3 \leq \tau(G)$, at some point we will arrive into $H$ such that $R = \tau(H) = k_3 \leq k_1$. Observe that $H$ can be covered with at most $R \leq k_1$ vertices, it has at least $k_2$ edges and it contains a matching of size $k_3$. Thus, the problem is a yes-instance.

Finally, let us observe that the running-time of this algorithm is FPT in $k_1$. We need at most $k_1$ calls of $PVCB(k_1,k_2)$. Since the latter is FPT with respect to $k_1$, we have the result. \hfill $\square$

6 The 2-PVCB Problem

In this section, we consider the problem of finding a partial vertex cover on a bipartite graph when we have a separate budget for each partition. In Theorem 6
we show that this problem is $\mathsf{W[1]}$-hard with respect to both vertex budgets even when both the vertices and edges are unweighted.

**Theorem 6.** $2$-PVCB is $\mathsf{W[1]}$-hard with respect to $R_V$ and $R_U$.

**Proof.** We will show this by a reduction from clique on regular graphs. This problem is known to be $\mathsf{W[1]}$-complete (Theorem 13.25 of [11]).

Let $G = (V, E)$ be an undirected graph where each vertex has degree $r$. From $G$ we construct the bipartite graph $G' = (V', U', E')$ as follows:

1. For each vertex $v_i \in V$, add the vertex $v'_i$ to $V'$.
2. For each edge $e_l = (v_i, v_j) \in E$ add the vertex $u'_l$ to $U'$. Additionally add the edge $(v'_i, u'_l)$ to $E'$ for each $v'_i \in V' \setminus \{v'_i, v'_j\}$. This connects the vertex $u'_l$ to every vertex $v'_i$ that does not correspond to an endpoint of $e_l$.

Figure 2 shows the original regular graph $G$ and the corresponding bipartite graph $G'$ constructed as in the reduction.

![Fig. 2. A graph $G$ and the bipartite graph $G'$ obtained from it in the reduction.](image)

Note that $G$ has a clique of size $k = 2$ using vertices $v_1$ and $v_2$. This corresponds to the vertex cover using vertices $v'_1$ and $v'_2$ from $V'$ and vertex $u'_1$ from $U'$ which covers $(4 - 2) \cdot \binom{2}{2} + (4 - 2) \cdot 2 = 6$ edges of $G'$.

We now show that $G$ has a clique of size $k$ if and only if we can cover at least $L = (n - k) \cdot \binom{k}{2} + k \cdot (m - r)$ edges of $G'$ using at most $R_{V'} = k$ vertices from $V'$ and at most $R_{U'} = \binom{k}{2}$ vertices from $U'$. Since checking if $G$ has a clique of size $n$ can be done in polynomial time, we assume without loss of generality that $k < n$.

First assume that $G$ has a clique of size $k$. Let $S \subseteq V$ denote the vertices in the clique. We construct the desired partial vertex cover of $G'$ as follows:

1. For each $v_i \in S$, add $v'_i$ to $S_{V'}$.
2. For each edge $e_l = (v_i, v_j)$ such that $v_i, v_j \in S$, add the vertex $u'_l$ to $S_{U'}$. Note that there are exactly $\binom{k}{2}$ such edges.
Observe the following:

1. Each vertex $v_i \in S$ is the endpoint of exactly $r$ edges in $E$. Thus, there are $(m - r)$ edges in $E$ which do not have $v_i$ as an endpoint. This means that, by construction, $v'_i \in S_{V'}$ covers $(m - r)$ edges. Thus, the vertices in $S_{V'}$ cover a total of $k \cdot (m - r)$ edges.

2. Each vertex $u'_i$ in $S_{U'}$, corresponds to an edge of $G$ with both endpoints in $S$. Thus, there is an edge between $u'_i$ and every vertex in $V' \setminus S_{V'}$. This means that $u'_i$ covers an additional $(n - k)$ edges. Thus, the vertices in $S_{U'}$ cover a total of $(n - k) \cdot {k \choose 2}$ additional edges.

Thus, the sets $S_{V'}$ and $S_{U'}$ cover a total of $L = (n - k) \cdot {k \choose 2} + k \cdot (m - r)$ edges.

Now suppose that there exist sets $S_{V'} \subseteq V'$ and $S_{U'} \subseteq U'$ such that $|S_{V'}| \leq k$, $|S_{U'}| \leq {k \choose 2}$, and the sets $S_{V'}$ and $S_{U'}$ cover at least $L = (n - k) \cdot {k \choose 2} + k \cdot (m - r)$ edges of $G'$.

We can assume without loss of generality that $|S_{V'}| = k$ and $|S_{U'}| = {k \choose 2}$ since adding additional vertices will not reduce the number of edges covered. Let $S = \{v_i : v'_i \in S_{V'}\}$.

Observe the following:

1. As before, each vertex in $S_{V'}$ has degree $(m - r)$. Thus, the vertices in $S_{V'}$ cover a total of $k \cdot (m - r)$ edges. Thus, the vertices in $S_{U'}$ must cover an additional $(n - k) \cdot {k \choose 2}$ edges.

2. There are $(n - k) > 0$ vertices in $V' \setminus S_{V'}$. Thus, there must be an edge between each of the ${k \choose 2}$ vertices in $S_{U'}$ and each of the $(n - k)$ vertices in $V' \setminus S_{V'}$. Otherwise, fewer than $(n - k) \cdot {k \choose 2}$ additional edges will be covered by the vertices in $S_{U'}$.

3. Let $u'_i$ be a vertex in $S_{U'}$ and let $e_i = (v_i, v'_j)$ be the corresponding edge in $G$. Note there is no edge in $E'$ between $u'_i$ and $v'_j$. Thus, both $v'_i$ and $v'_j$ are in $S_{V'}$. This means that every vertex $u'_i \in S_{U'}$ corresponds to an edge $e_i \in E$ with both endpoints in $S$.

Thus, $G$ has ${k \choose 2}$ edges with both endpoints in $S$. Since $|S| = k$, $S$ must be a clique of size $k$. \qed

7 Conclusion

In this paper, we studied the partial vertex cover problem from the perspective of parameterized tractability and $\textbf{W[1]-hardness}$. Although our primary focus was on bipartite graphs, we obtained new results for the general case as well. Our main contributions include showing that a restricted version of the WPVCB problem is fixed-parameter tractable and that this problem is $\textbf{W[1]-hard}$, with respect to the vertex-weight parameter. We also showed that the WPVC problem is fixed-parameter tractable in bounded degree graphs. Finally, we introduced a new variant of the partial vertex cover problem called PVCBM and showed that it is fixed-parameter tractable.
References

1. Amini, O., Fomin, F.V., Saurabh, S.: Implicit branching and parameterized partial cover problems. J. Comput. Syst. Sci. 77, 1159–1171 (2011)
2. 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). https://doi.org/10.1007/3-540-48777-8_2
3. Apollonio, N., Simeone, B.: The maximum vertex coverage problem on bipartite graphs. Discrete Appl. Math. 165, 37–48 (2014)
4. Apollonio, N., Simeone, B.: Improved approximation of maximum vertex coverage problem on bipartite graphs. SIAM J. Discrete Math. 28(3), 1137–1151 (2014)
5. Bar-Yehuda, R.: Using homogeneous weights for approximating the partial cover problem. J. Algorithms 39(2), 137–144 (2001)
6. Bilgin, C.C., Caskurlu, B., Gehani, A., Subramani, K.: Analytical models for risk-based intrusion response. Comput. Netw. (Special issue on Security/Identity Architecture) 57(10), 2181–2192 (2013)
7. Bläser, M.: Computing small partial coverings. Inf. Process. Lett. 85(6), 327–331 (2003)
8. 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). https://doi.org/10.1007/BFb0028569
9. Caskurlu, B., Mkrtchyan, V., Parekh, O., Subramani, K.: Partial vertex cover and budgeted maximum coverage in bipartite graphs. SIAM J. Discrete Math. 31(3), 2172–2184 (2017)
10. Caskurlu, B., Mkrtchyan, V., Parekh, O., Subramani, K.: On partial vertex cover and budgeted maximum coverage problems in bipartite graphs. In: Diaz, J., Lanese, I., Sangiorgi, D. (eds.) TCS 2014. LNCS, vol. 8705, pp. 13–26. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44602-7_2
11. Cygan, M., et al.: Parameterized Algorithms, pp. 3–555. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21275-3. ISBN 978-3-319-21274-6
12. Dinur, I., Safra, S.: On the hardness of approximating minimum vertex cover. Ann. Math. 162(1), 439–485 (2005)
13. Hochbaum, D.S.: The t-vertex cover problem: extending the half integrality framework with budget constraints. In: Jansen, K., Rolim, J. (eds.) APPROX 1998. LNCS, vol. 1444, pp. 111–122. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0053968
14. Joret, G., Vetta, A.: Reducing the rank of a matroid. Discrete Math. Theor. Comput. Sci. 17(2), 143–156 (2015)
15. Karakostas, G.: A better approximation ratio for the vertex cover problem. ACM Trans. Algorithms 5(4), 1–8 (2009)
16. Karp, R.: Reducibility among combinatorial problems. In: Miller, R., Thatcher, J. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press, New York (1972)
17. Khot, S., Regev, O.: Vertex cover might be hard to approximate to within 2 − ε. J. Comput. Syst. Sci. 74, 335–349 (2008)
18. Khot, S., Minzer, D., Safra, M.: Pseudorandom sets in Grassmann graph have near-perfect expansion. Electronic Colloquium on Computational Complexity, Report No. 6 (2018)
19. Khuller, S., Gandhi, R., Srinivasan, A.: Approximation algorithms for partial covering problems. J. Algorithms 53(1), 55–84 (2004)
20. 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). https://doi.org/10.1007/978-3-540-92248-3_22
21. Mestre, J.: A primal-dual approximation algorithm for partial vertex cover: making educated guesses. Algorithmica 55(1), 227–239 (2009)
22. 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). https://doi.org/10.1007/978-3-540-75520-3_31
23. Mkrtchyan, V., Parekh, O., Segev, D., Subramani, K.: The approximability of partial vertex covers in trees. In: Steffen, B., Baier, C., van den Brand, M., Eder, J., Hinchey, M., Margaria, T. (eds.) SOFSEM 2017. LNCS, vol. 10139, pp. 350–360. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51963-0_27
24. 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). https://doi.org/10.1007/978-3-540-69507-3_31
25. Moss, A., Khuller, S., (Seffi) Naor, J.: The budgeted maximum coverage problem. Inf. Process. Lett. 70(1), 39–45 (1999)
26. 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). https://doi.org/10.1007/11534273_5
27. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. Syst. Sci. 43(3), 425–440 (1991)
28. Parekh, O., Könemann, J., Segev, D.: A unified approach to approximating partial covering problems. Algorithmica 59(4), 489–509 (2011)
29. 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). https://doi.org/10.1007/11940128_60
30. Paschos, V.Th.: A polynomial time approximation schema for max $k$–vertex cover in bipartite graphs (2019). https://arxiv.org/abs/1909.08435v1
31. Vazirani, V.V.: Approximation Algorithms. Springer, New York (2001)