Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 26, no. 1, pp. 91–110 (2022)
DOI: 10.7155/jgaa.00584

On the fixed-parameter tractability of the partial vertex cover problem with a matching constraint in edge-weighted bipartite graphs

Vahan Mkrtchyan[1] *Garik Petrosyan*[2]

[1]Gran Sasso Science Institute,
School of Advanced Studies, L'Aquila, Italy
[2]Department of Informatics and Applied Mathematics,
Yerevan State University, Yerevan, Armenia

Submitted: April 2021	Reviewed: October 2021	Revised: November 2021
Accepted: March 2022	Final: March 2022	Published: April 2022
Article type: Regular paper		Communicated by: G. Liotta

Abstract. In the classical partial vertex cover problem, we are given a graph G and two positive integers k_1 and k_2. The goal is to check whether there is a subset V' of V of size at most k_1, such that V' covers at least k_2 edges of G. The problem is NP-hard as it includes the Vertex Cover problem. Previous research has addressed the extension of this problem where one has weight-functions defined on sets of vertices and edges of G. In this paper, we consider the following version of the problem where as the input we are given an edge-weighted bipartite graph G with weights from $\mathbb{N}$, and three positive integers k_1, k_2 and k_3. The goal is to check whether G has a subset V' of vertices of G of size at most k_1, such that the edges of G covered by V' have weight at least k_2 and they include a matching of weight at least k_3. In the paper, we address this problem from the perspective of fixed-parameter tractability and algorithms. We present some W[1]-hardness, paraNP-hardness results for our problem. On the positive side, we show that the problem is fixed-parameter tractable with respect to certain parameters. One of our W[1]-hardness results is obtained via a reduction from the bi-objective knapsack problem, which we show to be W[1]-hard with respect to one of the parameters. We believe that this problem might be useful in obtaining similar results in other situations.

Keywords: partial vertex cover; bipartite graph; fixed-parameter tractability; W[1]-hardness

The work of the first author has been partially supported by the Italian MIUR PRIN 2017 Project ALGADIMAR "Algorithms, Games, and Digital Markets."

E-mail addresses: vahan.mkrtchyan@gssi.it (Vahan Mkrtchyan) garik.petrosyan.1@gmail.com (Garik Petrosyan)

This work is licensed under the terms of the CC-BY license.

1 Introduction

In the present paper, we study an extension of the classical vertex cover problem (VC). We address this problem from the perspective of parameterized complexity theory and algorithms. It deals with the partial vertex cover problem (PVC). In PVC the goal is to cover a certain number of edges (not necessarily all the edges of the input graph as in VC) using the minimum number of vertices. In our problem, which we call matching version of PVC (see [35]), we have one more constraint where we require a certain lower bound for the size of the maximum matching in the covered edges. We also consider the weighted variants of this problem. Our main goal is to investigate this problem in bipartite graphs. In this case, we refer to it as the matching version of the partial vertex cover problem in bipartite graphs (M-PVCB). Some applications of the partial vertex cover problem in bipartite graphs and related problems are given in [9, 11].

In the matching version of edge-weighted partial vertex cover problem (M-EPVC), we are given a graph $G = (V, E)$, a weight-function $p : E \to \mathbb{N}$, and three positive integers k_1, k_2 and k_3. The goal is to check whether there is a subset $V' \subseteq V$ of cardinality at most k_1, such that the total weight of edges covered by V' is at least k_2, and the covered edges include a matching of weight at least k_3.

In the present paper, we study the fixed-parameter tractability of M-EPVC in bipartite graphs (M-EPVCB). By obtaining a W[1]-hardness result for so called multi-objective knapsack problem, we show that M-EPVCB is W[1]-hard with respect to k_1. We also obtain similar results for other interesting parameters. In case of some parameters we are able to show that M-EPVCB is Fixed-Parameter Tractable (FPT) with respect to the parameter under consideration.

The paper is organized as follows: Main notations and definitions are given in Section 2. The formal statements of the problems studied in the paper are given in Section 3. Section 4 presents the related approaches in the literature. In Section 5, we obtain hardness results for the so-called multi-objective knapsack problem and some of its restrictions. These results are used later in Section 6 where the main results of the paper are obtained. We conclude the paper in Section 7, by summarizing our results and presenting some open problems that we feel deserve further investigation.

2 Main Notations and Definitions

Let $X = \{x_1, ..., x_r\}$ be a set and $w : X \to Z$ be a function. For any $X_0 \subseteq X$, define $w(X_0) = \sum_{x \in X_0} w(x)$.

In the paper, we consider finite, undirected graphs that do not contain loops or parallel edges. The degree of a vertex is the number of edges of the graph that are incident to it. The maximum degree of the graph G, denoted by $\Delta(G)$, is the maximum of all degrees of vertices of G. For a positive integer k, we let V_k and $V_{\geq k}$ be the sets of vertices of G that have degree k and at least k, respectively. Let $rad(G)$ and $diam(G)$ be the radius and diameter of G (see [16]). If P is a path of length k in G, then we will say that P is a k-path.

If I is a subset of vertices of a graph G, then I is called an independent set if any two vertices of I are not adjacent in G. Let $\alpha(G)$ be the cardinality of the largest independent set of G. A subset M of edges of G is called a matching, if no two edges of M are incident to the same vertex of G. A matching M of G is called an induced matching if G contains no path of length three, such that its first and third edge belong to M. Let $\nu(G)$ be the size of a largest matching of G, and let $\nu_{ind}(G)$ be the size of a largest induced matching of G. Clearly, in any graph G we have $\nu_{ind}(G) \leq \nu(G) \leq \frac{|V|}{2}$. If $w : E(G) \to \mathbb{N}$ is a weight-function defined on edges of G and $X \subseteq E(G)$, then let $\nu_w(X)$ be the maximum weight (with respect to w) of a matching M, such that $M \subseteq X$. In particular, when $X = E(G)$, instead of writing $\nu_w(E(G))$, we will write $\nu_w(G)$.

A graph $G = (V, E)$ is bipartite, if its vertex set V can be partitioned into two independent sets V_1 and V_2. Usually, V_1 and V_2 are called the bipartition of G. If G is a bipartite graph with a bipartition V_1 and V_2, such that any vertex of V_1 is adjacent to any vertex of V_2, then G is called a complete bipartite graph. We will denote such a bipartite graph as $K_{|V_1|,|V_2|}$.

Given a graph $G = (V, E)$, and a set $V_0 \subseteq V$ of vertices, an edge $(u, v) \in E$ is covered by V_0 if $u \in V_0$ or $v \in V_0$. Let $E(V_0)$ be the set of edges of G that are covered by V_0. The classical vertex cover problem (VC) is defined as finding a smallest set V_0 of vertices of the input graph $G = (V, E)$, such that $E(V_0) = E$. We will denote the cardinality of such set V_0 as $\tau(G)$. The vertex cover problem is a well-known NP-complete problem [25].

If Π is an algorithmic problem and t is a parameter, then the pair (Π, t) is called a parameterized problem. The parameterized problem (Π, t) is fixed-parameter tractable (or Π is fixed-parameter tractable with respect to the parameter t) if there is an algorithm A that solves Π exactly, whose running-time is $g(t) \cdot poly(size)$. Here g is some (computable) function of t, $size$ is the length of the input and $poly$ is a polynomial function. Usually, such an algorithm A is called an FPT algorithm for (Π, t). Sometimes we will say that A runs in FPT(t) time. If I is an instance of Π, then let $t(I)$ denote the value of the parameter t in I.

A parameterized problem is called paraNP-hard, if it remains NP-hard even when the parameter under consideration is a constant. In the classical complexity theory, there is the notion of NP-hardness that indicates that a certain problem is less likely to be polynomial-time solvable. It relies on the assumption $P \neq NP$. The classical Satisfiability problem is an NP-hard problem and any problem such that Satisfiability can be reduced to it is NP-hard, too. Similarly, in parameterized complexity theory there is the notion of W[1]-hardness, which indicates that a certain parameterized problem is less likely to be fixed-parameter tractable. It relies on the assumption $FPT \neq W[1]$, which says that not all problems from $W[1]$ are fixed-parameter tractable. The Maximum Clique problem where the parameter under consideration is k - the size of the clique that we are looking for, is an example of a W[1]-hard problem, and any problem such that the maximum clique with respect to k can be FPT-reduced to it, is also W[1]-hard. Recall that an FPT reduction between two parameterized problems (Π_1, t_1) and (Π_2, t_2) is an algorithm R that maps instances of Π_1 to those of Π_2, such that

(i) for any instance $I_1 \in \Pi_1$, we have I_1 is a "yes"-instance of Π_1 if and only if $R(I_1)$ is a "yes"-instance of Π_2,

(ii) there is a computable function h, such that for any instance $I_1 \in \Pi_1$ $t_2(R(I_1)) \leq h(t_1(I_1))$,

(iii) there is a computable function g, such that R runs in time $g(t_1) \cdot poly(size)$.

The reader can learn more about this topic from [15], that can be a good guide for algorithmic concepts that are not defined in this paper.

3 Formal Statement of Main Problems

In this paper, we study the following variants of the VC problem:

(a) The partial vertex cover problem (PVC)

Definition 1 *Given a graph $G = (V, E)$ and two positive integers k_1, and k_2, check whether there is a subset V_0 of V, such that $|V_0| \leq k_1$ and $|E(V_0)| \geq k_2$.*

(b) The weighted partial vertex cover problem (WPVC)

Definition 2 *Given a graph $G = (V, E)$, weight-functions $c : V \to \mathbb{N}$ and $p : E \to \mathbb{N}$, two positive integers k_1 and k_2, check whether there is a subset V_0 of V, such that $\sum_{v \in V_0} c(v) \leq k_1$ and $\sum_{e \in E(V_0)} p(e) \geq k_2$?*

(c) The partial vertex cover problem on bipartite graphs (PVCB) – This is the restriction of the partial vertex cover problem (PVC) to bipartite graphs.

(d) 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.

(e) The VPVCB problem, a special case of the WPVCB problem, where all the edge weights are set to 1,

(f) The EPVCB problem, a special case of the WPVCB problem, where all the vertex weights are set to 1.

(g) The PVCB problem, a special case of the WPVCB problem, where all the vertex and edge weights are set to 1.

(h) The partial vertex cover problem with a matching constraint (M-PVCB) – 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 cardinality at most k_1, covering at least k_2 edges, such that the covered edges include a matching of size at least k_3.

(i) The edge-weighted partial vertex cover problem with a matching constraint (M-EPVCB) – This is a variant of the EPVCB problem, in which we are given a third parameter k_3 and the goal is to find a vertex subset of cardinality at most k_1, such that the covered edges have weight at least k_2 and they include a matching of weight at least k_3.

The main contributions of this paper are the following:

a. W[1]-hardness of the bi-objective knapsack problem and its two restrictions with respect to the budget B.

b. Reduction of the parameterized problem (M-EPVCB, k_1) to instances in which $k_1 < k_3 < k_2 < k_3 \cdot \Delta(G)$.

c. W[1]-hardness of the M-EPVCB problem with respect to k_1.

d. NP-hardness of M-EPVCB in complete bipartite graphs $K_{t,t}$ and the paraNP-hardness of this problem with respect to some parameters.

e. Hardness of M-EPVCB with respect to $|V| - 2\nu_{ind}(G)$ under the assumption FPT $\neq$ W[1].

f. Hardness of M-EPVCB in paths and cycles under the assumption FPT $\neq$ W[1].

g. Fixed parameter tractability of M-EPVCB with respect to $|V_{\geq 2}|$ and some other parameters.

Note that all our FPT algorithms depend on a single parameter.

4 Related Work

PVC represents a natural theoretical generalization of VC. It has some practical applications. Flow-based risk-assessment models in computational systems can be viewed as instances of PVC [9]. In particular, PVC has applications to computer security when the input is a bipartite graph [11].

VC is polynomial-time solvable in bipartite graphs. However, the computational complexity of PVC in bipartite graphs remained open until it was recently shown to be NP-hard by several authors [4, 11, 10, 23].

VC has been intensively studied from the perspective of approximation algorithms. There are many 2-approximation algorithms for VC (see, for example, [39]). Karakostas [24] provides an approximation algorithm for the VC problem which has a factor $(2 - \theta(\frac{1}{\sqrt{\log n}}))$. This is the best-known result for now. The VC problem is shown to be APX-complete in [36]. Moreover, it cannot be approximated within a factor of 1.3606 under the assumption $P \neq NP$ [17]. Recently, in [26], this lower bound was improved to $(\sqrt{2}-\epsilon)$ for any $\epsilon > 0$. If Khot's unique games conjecture is true, then VC cannot be approximated within any constant factor smaller than 2 [27]. Könnemann et al. [32] provide a $(\frac{4}{3} + \epsilon)$-approximation algorithm for WPVC when the input graph is bipartite. Here $\epsilon > 0$ is any constant.

All hardness results for the VC problem directly apply to the PVC problem because the PVC problem extends the VC problem. The PVC problem and the partial-cover variants of related graph problems have been extensively studied [7, 8, 29, 33, 31, 30]. For example, there is an $O(n \cdot \log n + m)$-time 2-approximation algorithm for PVC based on the primal-dual method [33]. Moreover, there is a combinatorial 2-approximation algorithm [6]. Both of the two algorithms are for a more general soft-capacitated version of PVC. There are several 2-approximations resulting from other approaches [5, 8, 22, 20]. Finally, note that the WPVC problem for trees is studied in [34]. The paper provides an FPTAS for the problem. Additionally, the paper provides a polynomial-time algorithm for the case of unweighted vertices (edges may have weights).

Another problem with a tight relationship to WPVC is the so-called budgeted maximum coverage problem (BMC). In the BMC problem one tries to find a min-cost subset of vertices, such that the profit of covered edges is maximized. It can be easily shown that the two 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 as shown in [28]. 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 [1]. This is improved to $\frac{4}{5}$ for bipartite graphs in [3]. In [11, 10], an $\frac{8}{9}$-approximation algorithm for the problem is presented when the input graph is bipartite and the vertices are unweighted (edges may have weights). The result is based on the natural linear-programming formulation of the problem. The constant $\frac{8}{9}$ matches the integrality gap of the linear program used in the formulation. Recently, in [37], V. Paschos presented a polynomial-time approximation scheme for the edge-weighted maximum coverage problem on bipartite graphs.

Another problem with a close relation to the BMC and PVC problems is the profit cover problem (PC). Like in the BMC and PVC problems, the PC problem does not require a solution that covers all the edges of a graph. However, instead of minimizing the number of vertices that cover a given number of edges or maximizing the number of edges covered by a fixed number of vertices, the goal in the PC problem is to maximize the profit. It is defined as the difference between the number of covered edges and the number of vertices in the cover. The PC problem has been considered in [38]. There, it is shown that there exists a $O(p \cdot n + 1.151^p)$ algorithm for the PC problem. Here p is the desired profit.

The 2-PVCB problem studied in [35] is closely related to the constrained minimum vertex cover problem on bipartite graphs (MIN-CVCB). In the MIN-CVCB problem, we are given two parameters k_A and k_B, and the goal is to find a cover of the bipartite graph $G = (A, B, E)$ (with a bipartition A and B) using at most k_A vertices from A and at most k_B vertices from B. The MIN-CVCB problem is NP-complete and can be solved by a fixed parameter tractable algorithm that runs in time $O(1.26^{k_A+k_B} + (k_A + k_B) \cdot |G|)$ as demonstrated in [14]. The 2-PVCB problem is a generalization of the MIN-CVCB problem in which one does not need to cover all the edges of the input graph G. In [35], it is shown that this generalization makes the problem no longer fixed-parameter tractable in k_A and k_B under the assumption FPT $\neq$ W[1].

In this paper, we address our problems from the perspective of fixed-parameter tractability. From this point of view, PVC is in some sense more difficult than VC. For example, PVC is W[1]-hard with respect to k_1 (that is, the number of vertices in the cover) [15]. On the other hand, VC is FPT [15, 21].

In [2] the decision version of WPVCB is considered. There, the authors show that this problem is FPT with respect to the vertex budget k_1, when the vertices and edges of the bipartite graph are unweighted. In [35], by extending the result of Amini et al. [2], it is shown that the decision version of WPVCB is FPT with respect to k_1, if the vertices have cost one, while the edges may have arbitrary weights. On the other hand, the problem is W[1]-hard for arbitrary vertex weights, even when edges have profit one [35]. Mkrtchyan et al. [35] prove that for bounded-degree graphs WPVC is FPT with respect to k_1. Similar conclusion holds for WPVC with respect to k_2. Finally, Mkrtchyan et al. [35] show that M-PVCB is FPT with respect to the budget k_1. Terms and concepts that we do not define in the paper can be found in [15].

5 The multi-objective knapsack problem

In this section, we consider the multi-objective version of the classical knapsack problem. We present some hardness results for this version.

Recall that in the ordinary version of the problem, as the input we are given n items $A = \{a_1, ..., a_n\}$, a cost function $c : A \to N$, a profit function $p : A \to N$ and two constants S and T. The goal is to check whether there is a subset A_0 of A, such that $c(A_0) \leq S$ and $p(A_0) \geq T$. This problem is NP-complete. It is natural to consider the following extension of this problem where as the input we have many cost functions and many profit functions. The goal in this new version is to check whether there is a subset of items whose cost with respect to any of the cost functions is at most some given bound, and its profit with respect to any of the profit functions is at least some other bound. This version of the problem is called the multi-objective knapsack problem.

In this paper, we will need only the case of this problem where as the input we have one cost function that is identically one and two profit functions. Let us formulate this version precisely:

Problem: We are given a set of items $A = \{a_1, ..., a_n\}$, a constant B (that we will call a budget), two constants P_1 and P_2, and two profit functions $pr_1 : A \to \mathbb{N}$ and $pr_2 : A \to \mathbb{N}$. The goal is to check whether there is a subset $S \subseteq A$ with $|S| \leq B$, such that $pr_1(S) \geq P_1$ and $pr_2(S) \geq P_2$.

In this paper, we call this problem bi-objective knapsack problem or BKP for short. Below we are going to obtain some W[1]-hardness results for BKP and its restrictions.

In the "compendium of parameterized problems" (page 92 of [12]), SubSet Sum problem is defined, which is the following: we are given a set of integers $X = \{x_1, ..., x_n\}$, an integer s and a positive integer k. The goal is to check whether X has a subset X' of cardinality k such that the sum of numbers in X' is exactly s. In [12] it is stated that this problem is W[1]-hard with respect to k. The reference given there is [18] where the authors proved the W[1]-hardness of Sized SubSet Sum (see page 123 of the paper) which is the same problem except that all numbers involved are positive integers. Clearly the hardness of Sized SubSet Sum implies the hardness of SubSet Sum as the latter is just an extension of the former.

Now, let us use this in order to obtain a reduction for the multi-objective knapsack problem where the weight functions can take negative values too. Assume that we have an instance of the SubSet Sum problem. Consider elements $A = \{a_1, ..., a_n\}$. Let $pr_1(a_j) = x_j$ and $pr_2(a_j) = -x_j$. Define $B = k$, $P_1 = s$ and $P_2 = -s$. Then we have that there is X', $|X'| = k$ such that $\sum_{x \in X'} x = s$ if and only if $\sum_{x \in X'} x \geq s$ and $\sum_{x \in X'}(-x) \geq -s$. The latter is true if and only if there is $Y \subseteq A$ with $|Y| = B$ such that $pr_1(Y) \geq P_1$ and $pr_2(Y) \geq P_2$.

Thus, this version of the problem is W[1]-hard with respect to B. This proof has two drawbacks. First, we required that $|Y| = B$. This is not a problem if the weights are non-negative, however, when they can be negative, then conditions $|Y| = B$ and $|Y| \leq B$ are not the same. Second, the weights can take negative values which we did not assume in our initial formulation of BKP.

We can fix the above two drawbacks as follows. Let us FPT-reduce the above mentioned variant of the knapsack problem with negative weights to the case when everything is positive. If this is achieved, as a side effect, we will also solve the issue over $|Y| = B$ and $|Y| \leq B$.

Theorem 1 *BKP is W[1]-hard with respect to B.*

Proof: Assume that $A = \{a_1, ..., a_n\}$, $1 \leq B \leq n$, $pr_1, pr_2 : A \to Z$ and $P_1, P_2 \in Z$ are given. Let us reduce this to the case when everything is positive. For $i = 1, 2$ define:

$$Q_i = 1 + \sum_{x \in A} |pr_i(x)|,$$

and for any $x \in A$ let

$$pr'_i(x) = pr_i(x) + Q_i,$$

$$P'_i = P_i + B \cdot Q_i.$$

Observe that $Q_i > 0$ and $pr'_i(x) > 0$. We can assume that $-Q_i < P_i < Q_i$, as if $P_i \geq Q_i$, then we have a trivial "no"-instance, and if $P_i \leq -Q_i$ we have a trivial "yes"-instance. Thus, $P'_i > 0$.

Let us show that there is $Y \subseteq A$, with $|Y| = B$ such that $pr_i(Y) \geq P_i$ if and only if $pr'_i(Y) \geq P'_i$. By our definitions, we have

$$pr'_i(Y) = pr_i(Y) + |Y| \cdot Q_i = pr_i(Y) + B \cdot Q_i.$$

Hence, $pr'_i(Y) \geq P'_i = P_i + B \cdot Q_i$ if and only if $pr_i(Y) \geq P_i$.

Thus, this is a reduction. Moreover, observe that it is a polynomial-time reduction. Since the value of B is unchanged, we have an FPT reduction. The proof is complete. □

Now, we are going to show that BKP remains W[1]-hard even if we have some additional restrictions on the profit functions. We will need these results in order to obtain some of our main results in the next section.

Theorem 2 *BKP remains W[1]-hard with respect to B even if $pr_1(x) - pr_2(x) \leq pr_2(x) < pr_1(x)$ for any $x \in A$.*

Proof: We reduce BKP to itself when this additional constraint is satisfied. Assume that $A = \{a_1, ..., a_n\}$, $1 \leq B \leq n$, $pr_1, pr_2 : A \to N$ and $P_1, P_2 \in N$ are given. First let us show that we can assume that for any $x \in A$ we have $pr_2(x) < pr_1(x)$. For this purpose, define

$$Q = 1 + \max_{x \in A} \left\lceil \frac{pr_2(x)}{pr_1(x)} \right\rceil.$$

Let $pr'_1(x) = Q \cdot pr_1(x)$, $pr'_2(x) = pr_2(x)$, $P'_1 = Q \cdot P_1$ and $P'_2 = P_2$. Let the budget B remain unchanged. We have that there is $Y \subseteq A$ with $|Y| = B$ such that $pr'_i(Y) \geq P'_i$ if and only if $pr_i(Y) \geq P_i$ (since everything is unchanged or multiplied by the same constant). Clearly, the new instance can be obtained in polynomial time. Let us show that the new instance satisfies $pr'_2(x) < pr'_1(x)$ for any $x \in A$. The inequality $pr'_2(x) < pr'_1(x)$ is equivalent to $pr_2(x) < Q \cdot pr_1(x)$ or $\frac{pr_2(x)}{pr_1(x)} < Q$. The latter we always have by the definition of Q.

Thus, in the very beginning we can assume that for any $x \in A$ we have $pr_2(x) < pr_1(x)$. Let us show that we can assume the other inequality as well. For a given instance define:

$$P_0 = \max\{1, \max_{x \in A}(pr_1(x) - 2pr_2(x))\}.$$

Observe that $P_0 \geq 1$ by definition. For any $x \in A$ and $i = 1, 2$ define:

$$pr'_i(x) = pr_i(x) + P_0,$$

$$P'_i = P_i + B \cdot P_0.$$

As we have added the same number to every profit, we have for any $x \in A$ $pr'_2(x) < pr'_1(x)$. Since in the problem we were looking for $|Y| = B$, we can prove similarly (see the proof of Theorem 1) that there is such $Y \subseteq A$ with $|Y| = B$, such that $pr_i(Y) \geq P_i$ if and only if $pr'_i(Y) \geq P'_i$. Thus we have a reduction. Moreover, it is a polynomial-time reduction and the value of the budget is not changed. Thus, all we are left is to show that for any $x \in A$ we have $pr'_1(x) \leq 2 \cdot pr'_2(x)$. By definition, the last inequality is the same that

$$pr_1(x) + P_0 \leq 2 \cdot (pr_2(x) + P_0),$$

or equivalently,

$$pr_1(x) - 2 \cdot pr_2(x) \leq P_0.$$

However, the last one is always true because of the definition of P_0. The proof is complete. □

Theorem 3 *BKP remains W[1]-hard with respect to B even if $pr_1(x) - pr_2(x) \leq pr_2(x) < pr_1(x)$ for any $x \in A$ and $\sum_{i=1}^{n}[pr_1(a_i) - pr_2(a_i)] < \min_{x \in A} pr_2(x)$.*

Proof: By the previous theorem, we know that BKP with the first condition satisfied remains W[1]-hard with respect to B. Let us reduce these instances to ones when the second condition in the theorem is satisfied, too.

Start with an instance of BKP with the first condition and let

$$T = \sum_{i=1}^{n}[pr_1(a_i) - pr_2(a_i)].$$

Now, let us define new profit functions and new lower bounds for our parameters in the following way:

$$pr'_i(x) = pr_i(x) + (T + 1),$$

and

$$P'_i = P_i + B(T+1).$$

As before, we can show that this established a reduction. Moreover,

$$0 < pr'_1(x) - pr'_2(x) \le pr'_2(x) < pr'_1(x),$$

All we are left is to show that our new constraint is true in the new instances as well. We have:

$$\sum_{i=1}^{n}[pr'_1(a_i) - pr'_2(a_i)] = \sum_{i=1}^{n}[pr_1(a_i) - pr_2(a_i)] < T+1 < pr'_2(z)$$

for any item z. Thus, our new condition is satisfied. The proof is complete. $\square$

6 Main results

In this section, we obtain our main results. Some of our proofs rely on the hardness results established in the previous section. We start with the following observation that will allow us to obtain some restrictions for the values of k_1.

Observation 1 *The instances of M-EPVCB, in which $k_1 \ge \tau(G) = \nu(G)$ can be solved in time $O(|V| \cdot |E|^2 + |V|^2 \log |V|)$.*

Proof: Since G is bipartite, we can find a smallest vertex cover in it with the classical Edmonds-Karp maximum flow algorithm in time $O(|V| \cdot |E|^2)$. Because of our assumption, G contains at most k_1 vertices in this vertex cover. Thus, in order to solve the instance, it suffices to check whether $w(E) \ge k_2$ and whether the weight of maximum weighted matching is at least k_3. Clearly, this can be done in time $O(|V| \cdot |E| + |V|^2 \log |V|)$ by the maximum weighted matching algorithm from [19]. The proof is complete. $\square$

Observation 2 *The instances of M-EPVCB, in which $k_2 \le k_3$ can be solved in time $O(|V| \cdot |E| + |V|^2 \log |V|)$.*

Proof: Assume that we have an instance with $k_3 > k_2$. We claim that (G, w, k_1, k_2, k_3) is a "yes"-instance if and only if (G, w, k_1, k_3, k_3) is a "yes"-instance. Let us assume that (G, w, k_1, k_2, k_3) is a "yes"-instance. Then since the covered edges include a matching of w-weight at least k_3, we have that the covered edges are of weight at least k_3. Thus, (G, w, k_1, k_3, k_3) is a "yes"-instance. On the other hand, if (G, w, k_1, k_3, k_3) is a "yes"-instance. Then for any $k_2 < k_3$, we have that (G, w, k_1, k_2, k_3) is a "yes"-instance. Thus, the instances with $k_3 \ge k_2$ can be reduced to those with $k_3 = k_2$.

Theorem 4.6 of [13] implies that one can find a maximum weighted k_1-matching (if it exists) in time $O(k_1 \cdot (|E| + |V| \log |V|))$. In the paper, a k_1-matching is defined as a matching of cardinality k_1. Thus, in order to solve the case $k_2 = k_3$ of our problem, we just need to find a maximum weighted k_1-matching and check whether its weight is at least k_2. Hence, combined with the previous paragraph and taking into account that $k_1 \le |V|$, we have that the instances with $k_2 \le k_3$ are solvable in time $O(|V| \cdot |E| + |V|^2 \log |V|)$. The proof is complete. $\square$

The next proposition allows us to reduce the solution of some instances of M-EPVCB to instances of EPVCB. Since EPVCB is FPT with respect to k_1 [35], these instances can be solved in FPT(k_1) time.

Proposition 1 *Let (G, w, k_1, k_2, k_3) be an instance of M-EPVCB with $\frac{k_2}{k_3} \geq \Delta(G)$. Then it is a "yes"-instance if and only if (G, w, k_1, k_2) is a "yes"-instance of EPVCB.*

Proof: One direction is trivial. Let us assume that (G, w, k_1, k_2) is a "yes"-instance of EPVCB and V' is the corresponding partial cover. Since G is a bipartite graph, we have that the graph $G[E_{V'}]$ is a bipartite subgraph of G with maximum degree at most $\Delta(G)$. By König's theorem [16], the edge-set of any bipartite graph of maximum degree Δ can be partitioned into Δ matchings. Thus, we can write

$$E(G[E(V')]) = M_1 \cup ... \cup M_{\Delta(G)}.$$

Here $M_1, ..., M_{\Delta(G)}$ are matchings of $G[E(V')]$. Then:

$$\max_{1 \leq j \leq \Delta(G)} w(M_j) \geq \frac{w(E(G[E(V')]))}{\Delta(G)} \geq \frac{k_2}{\Delta(G)} \geq k_3.$$

Thus, (G, w, k_1, k_2, k_3) is a "yes"-instance of the edge-weighted version of the matching problem. The proof is complete. □

Theorem 4 *The instances of M-EPVCB, in which $k_3 \leq k_1$ can be solved in time $O((2k_1)^{k_1} \cdot \log k_1 \cdot |V|^{O(1)})$.*

Proof: We follow the proof of Theorem 5 from [35]. Let EPVCB(A, B) be the FPT(A) algorithm that solves EPVCB (see Theorem 1 of [35]). Its running-time is $O((2A)^A \cdot |V|^{O(1)})$. Let R be the smallest integer for which $EPVCB(R, k_2)$ is feasible. Let J be the set of edges of the input graph that can be covered with its R vertices. We have $w(J) \geq k_2$.

We can assume that $R < k_1$. Let H be the spanning subgraph of our graph whose edge-set is J. By the classical König theorem we have $\nu(G) = \tau(G)$ for any bipartite graph G. By Observation 1, we can assume that $k_1 < \tau(G)$.

Observe that we can assume that $R < k_3 \leq k_1 < \tau(G)$. If $R \geq k_3$, then clearly H can be covered with at most k_1 vertices, it has weight at least k_2, and it has a matching of size $R \geq k_3$ hence of weight at least 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 $\tau(H) = R < k_3 \leq \tau(G)$ and at each step τ can increase by at most one, at some point we will arrive at H such that $R = \tau(H) = k_3 \leq k_1$. Observe that H can be covered with at most k_1 vertices, it has weight at least k_2 and it contains a matching of size k_3, hence of weight at least k_3. Thus, the problem is a "yes"-instance.

The running time of the above algorithm is $O((2k_1)^{k_1} \cdot \log k_1 \cdot |V|^{O(1)})$. In order to see this, just observe that with the standard binary search approach we can have at most $\log k_1$ calls of $EPVCB(R, k_2)$. The proof is complete. □

The four statements proved above imply the following

Corollary 1 *When parameterizing M-EPVCB with respect to k_1, one can focus on instances in which $k_1 < k_3 < k_2 < k_3 \cdot \Delta(G)$.*

In [35], it is shown that M-PVCB is FPT with respect to k_1. On the other hand, in the same paper it is proved that VPVCB and WPVCB are W[1]-hard with respect to k_1. Thus, their matching extensions are W[1]-hard with respect to k_1, too. It is interesting to wonder whether M-EPVCB is FPT with respect to k_1. Our next result addresses this question.

Theorem 5 *M-EPVCB is W[1]-hard with respect to* k_1*.*

Proof: By Theorem 2, BKP remains W[1]-hard even if for any item $x \in A$ we have

$$pr_1(x) - pr_2(x) \leq pr_2(x) < pr_1(x).$$

Let us reduce this problem to M-EPVCB. For each item $x \in A = \{a_1, ..., a_n\}$ put a 2-path (a path of length two) with edge-weights $w(e) = pr_1(x) - pr_2(x)$ and $w(e) = pr_2(x)$ for edges e of the 2-path (Figure 1). Observe that we have an edge-weighted bipartite graph. Let $k_1 = B$, $k_2 = P_1$ and $k_3 = P_2$.

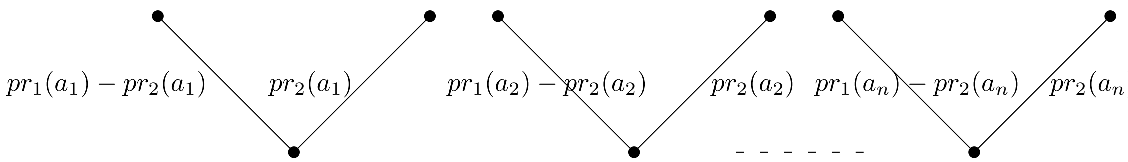

Figure 1: The bipartite graph obtained after the reduction from BKP.

Let us show that we have a reduction. Observe that we can always avoid taking the degree-one vertices in our cover as we can simply take the degree-two vertex instead of it. Thus, we have a bijection among subsets of items and subsets of degree-two vertices of our bipartite graph. Moreover, for any subset $S \subseteq A$, we have

$$w(E(S)) = \sum_{x \in S} [(pr_1(x) - pr_2(x)) + pr_2(x)] = pr_1(S),$$

and

$$\nu_w(E(S)) = \sum_{x \in S} \max\{pr_1(x) - pr_2(x), pr_2(x)\} = pr_2(S),$$

as we were considering the restriction of the multi-objective knapsack problem in which $pr_1(x) - pr_2(x) \leq pr_2(x)$ for any $x \in A$. Thus,

$$pr_1(S) \geq P_1 \text{ if and only if } w(E(S)) \geq k_2,$$

and

$$pr_2(S) \geq P_2 \text{ if and only if } \nu_w(E(S)) \geq k_3.$$

Thus, we have a polynomial-time reduction. Observe that $k_1 = B$, thus we have an FPT reduction. The proof is complete. □

In the reduction presented above, we have bounded maximum degree. Actually, it is two. Thus, we have

Corollary 2 *Under the assumption FPT* $\neq$ *W[1], M-EPVCB is not FPT with respect to* $\Delta(G)$*.*

Since $\Delta(G) = 2$ in the above reduction we have that M-EPVCB is W[1]-hard with respect to $k_1 + \Delta(G)$. On the other hand, because of Observation 1, we can always assume that $k_1 < \tau(G) = \nu(G)$. Thus, one may wonder whether M-EPVCB is FPT with respect to $\nu(G) + \Delta(G)$. Since by

König's theorem, $E(G)$ can be partitioned into $\Delta(G)$ matching, we have that $|E| \leq \Delta(G) \cdot \nu(G)$. Thus, M-EPVCB is FPT with respect to the mentioned parameter.

Though M-EPVCB is W[1]-hard with respect to k_1, it is easy to show that it is FPT with respect to the complementary parameter $|V| - k_1$. In order to see this, in the given instance we can check whether $|V| - k_1 \geq \frac{|V|}{2}$. If it holds, then $|V| \leq 2 \cdot (|V| - k_1)$. In this case, we can generate all subsets X of V, and for each X, we check that $|X| \leq k_1$, it has coverage at least k_2 and the includes a matching of weight at least k_3. These conditions can be checked in time $O(|V| \cdot |E| + |V|^2 \log |V|)$ [19]. Thus, this case we can overcome in time

$$2^{|V|} \cdot O(|V| \cdot |E| + |V|^2 \log |V|) \leq O(4^{|V|-k_1} \cdot (|V| \cdot |E| + |V|^2 \log |V|)).$$

On the other hand, if $|V| - k_1 \leq \frac{|V|}{2}$, then $k_1 \geq \frac{|V|}{2}$. Since in any bipartite graph G $\tau(G) \leq \frac{|V|}{2}$ (just take the smallest set in the bipartition of G), we have $\tau(G) \leq k_1$. Observation 1 implies that these instances can be solved in time $O(|V| \cdot |E|^2 + |V|^2 \log |V|)$. In conclusion, the total running-time of our algorithm is $O(4^{|V|-k_1} \cdot (|V| \cdot |E|^2 + |V|^2 \log |V|))$.

PVCB and its weighted extensions considered in this paper are NP-hard. Below we show that M-EPVCB remains hard for very restricted subclass of regular bipartite graphs.

Theorem 6 *M-EPVCB is NP-hard in complete bipartite graphs $K_{t,t}$.*

Proof: We reduce M-EPVCB to its restriction in complete bipartite graphs. For a given weighted bipartite graph, define $C = |V|^2$. First, let us embed our bipartite graph G into a bipartite $\Delta(G)$-regular graph G' in a standard way. That is, we add isolated vertices to the smallest set in the bipartition of G, so that two sets have equal size. Then we start adding edges so that the graph remains bipartite and becomes regular. The number of added edges is less than $(|V| - 1)^2 < |V|^2 = C$ as the size of larger part of G is at most $|V| - 1$ and minimum degree we can assume to be at least 1 and maximum degree is at most $|V| - 1$. Next, we define the weight function on G' as follows: $w'(e) = C \cdot w(e)$ for edges of G and $w'(e) = 1$ for edges from $E(G') \backslash E(G)$. Finally, for a given k_1, k_2, k_3 define $k'_1 = k_1$, $k'_2 = Ck_2$ and $k'_3 = Ck_3$. Since $\log C \leq poly(size)$, we have that we have increased the parameters polynomially. Observe that the reduction can be performed in polynomial time.

Let us show (G, w, k_1, k_2, k_3) is a "yes"-instance if and only if $(G', w', k'_1, k'_2, k'_3)$ is a "yes"-instance. If in G we have a feasible solution, then clearly it is feasible in G' as everything is multiplied by C. Now, assume that V_0 is a feasible set in G'. Let $E_1 = E(G') \backslash E(G)$. We have $w'(E(V_0)) \geq k'_2 = Ck_2$ and $\nu_{w'}(E(V_0)) \geq k'_3 = Ck_3$. Hence

$$w'(E(V_0) \backslash E_1) \geq Ck_2 - |E_1| > Ck_2 - |V|^2 = Ck_2 - C = C(k_2 - 1)$$

and similarly

$$\nu_{w'}(E(V_0) \backslash E_1) \geq Ck_3 - |E_1| > Ck_3 - |V|^2 = Ck_3 - C = C(k_3 - 1).$$

Thus,

$$w(E(V_0) \cap E(G)) = w(E(V_0)) > k_2 - 1$$

and

$$\nu_w(E(V_0) \cap E(G)) = \nu_w(E(V_0)) > k_3 - 1.$$

Thus, $w(E(V_0)) \geq k_2$ and $\nu_w(E(V_0)) \geq k_3$. We have reduced the edge-weighted matching problem in bipartite graphs to its restriction in bipartite regular graphs in polynomial time. Thus, the problem is NP-hard in bipartite regular graphs. Now, in order to complete the proof of the theorem, observe that if we want to obtain a complete bipartite graph in the reduction, we only need to continue adding edges of weight 1. The proof is complete. □

Remark 1 *Observe that the vertices of $V(G')\backslash V(G)$ in the reduction do not play a role since they do not cover the old edges. Hence if we have a coverage greater than $C(k_2-1)$ in G', then all these edges will be covered with vertices from $V(G)$. Thus, they will give rise to a coverage larger than k_2-1 in G with vertices of G.*

The proved hardness result has some consequences. In order to state them, recall that a matching M of G is called an induced matching, if G contains no path of length three, such that its first and third edge belong to M. Let $\nu_{ind}(G)$ be the size of a largest induced matching of G. In complete bipartite graphs $K_{t,t}$, we have that $\nu_{ind}(G)=1$, the domination number is 2, $2\alpha(G)-|V|=0$ ($\alpha(G)$ is the size of the largest independent set in G) and $\Delta(G)\cdot\nu(G)-|E(G)|=0$ (this is true for any bipartite regular graph). Thus, M-EPVCB is paraNP-hard with respect to these parameters. Below we obtain a hardness result with respect to $|V|-2\nu_{ind}(G)$.

Theorem 7 *Under the assumption FPT $\neq$ W[1], M-EPVCB cannot be FPT with respect to $|V|-2\nu_{ind}(G)$.*

Proof: We reduce from the restriction of BKP from Theorem 3. For a given instance of BKP from this theorem, consider the disjoint 2-paths from the proof of Theorem 5. Now, from each 2-path take one vertex of degree 1, and identify these n vertices in order to get the tree G' (see Figure 2). The resulting vertex z in G' has degree n. Observe that $|V(G')|-2\nu_{ind}(G')=1$.

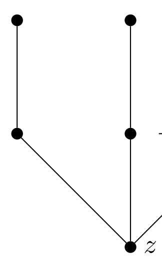

Figure 2: The tree G' obtained after the reduction from BKP.

Let us show that the instance of BKP is a "yes"-instance if and only if the instance of M-EPVCB is a "yes"-instance. Forward direction is trivial. Let us prove the converse statement. Assume that the instance of M-EPVCB is a "yes"-instance. It suffices to show that there is a feasible set that does not take z. Assume that we have a feasible set V_0 in M-EPVCB. We can assume that $z\in V_0$. If all neighbors of z are in V_0, then we can remove it without losing feasibility. Thus, we can assume that at least one neighbor of z does not belong to V_0. Replace z with this neighbor in order to obtain a set V_1. Observe that because of the condition

$$\sum_{i=1}^{n}[pr_1(a_i)-pr_2(a_i)]<\min_{x\in A}pr_2(x)$$

we have the same lower bounds for the coverage and the maximum weighted matching of covered edges. Thus, we have a feasible set that avoids z. The proof is complete. □

In the previous theorem we showed that M-EPVCB is hard already when $|V| - 2\nu_{ind}(G) = 1$. One may wonder what happens when $|V| - 2\nu_{ind}(G) = 0$. In this case, we have an induced perfect matching in G. Thus, G is 1-regular. Therefore, by taking k_1 edges of maximum weight, we can check whether these edges have coverage at least k_2 and k_3. If they do, we have a "yes"-instance. Otherwise, it is a "no"-instance. Clearly, this can be done in time $O(|V| \cdot \log |V|)$.

Theorem 8 *Under the assumption FPT $\neq$ W[1], M-EPVCB is not polynomial-time solvable in paths and cycles.*

Proof: In Theorem 5, we have shown that M-EPVCB is W[1]-hard with respect to k_1 in vertex-disjoint 2-paths. Now, we are going to reduce these instances to cycles and paths in polynomial-time. Clearly, this will prove the statement.

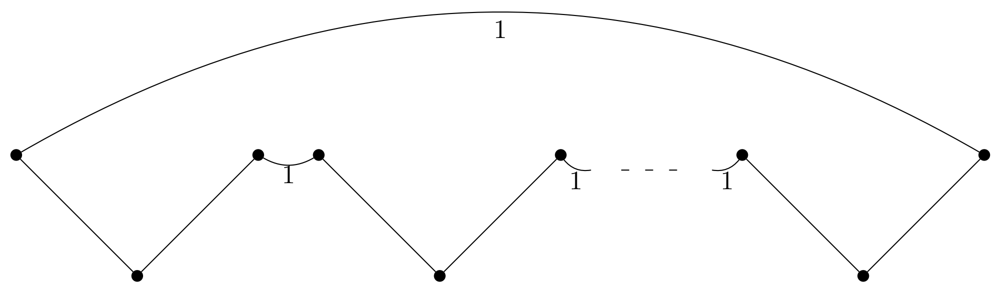

Figure 3: The path/cycle obtained from vertex disjoint 2-paths.

We follow the strategy of the proof of Theorem 6. Assume that G is a vertex union of 2-paths. Let us take a constant $C = |V|^2$ and define the new values of parameters as we did in the proof of Theorem 6. Now, in order to obtain cycles or paths, we add edges of weight 1 (see Figure 3). Since G is of maximum degree two, this is always possible. As in Theorem 6, one can prove that the original instance is a "yes"-instance if and only if the new instance is a "yes"-instance. The proof is complete. □

Recall that $rad(G)$ and $diam(G)$ denote the radius and diameter of G. For more information on them, the reader can refer to [16]. In Theorem 7, we proved that M-EPVCB remains hard in a class of trees of radius two and diameter four. This implies that under the assumption FPT $\neq$ W[1], M-EPVCB cannot be FPT with respect to $diam(G)$ and $rad(G)$. One may ask question about the parameters $|V| - diam(G)$ and $|V| - rad(G)$. Observe that in paths, we have that $|V| - diam(G)$ is constant. Thus, under the assumption FPT $\neq$ W[1], M-EPVCB cannot be FPT with respect to it. On the other hand, for any graph G, we have $rad(G) \leq \frac{|V|}{2}$. Thus, $|V| - rad(G) \geq \frac{|V|}{2}$. Thus, M-EPVCB is FPT with respect to $|V| - rad(G)$. Finally, let us note that in paths $|V| - 2 \cdot rad(G)$ is constant, too. Thus, under the assumption FPT $\neq$ W[1], M-EPVCB cannot be FPT with respect to it, too.

Observe that M-EPVCB is hard with respect to $|V_1|$ as cycles demonstrate. In these instances, we have $|V_1| = 0$. Also, observe that the problem is hard with respect to $|V_{\geq 3}|$ as paths demonstrate.

Theorem 9 *M-EPVCB is FPT with respect to $|V_{\geq 2}|$. It can be solved with an algorithm running in time $O\left(2^{2^{|V_{\geq 2}|}} \cdot \left(|V|^2 \cdot |E| + |V|^3 \cdot \log |V|\right)\right)$.*

Proof: For a given instance of M-EPVCB, we consider two cases. If $\log |V| \leq |V_{\geq 2}|$, then $|V|$ is bounded in terms of our parameter. Thus, we can solve these instances in FPT($|V_{\geq 2}|$) time. More precisely, there are

$$2^{|V|} \leq 2^{2^{|V_{\geq 2}|}}$$

subsets of vertices. We can generate all of them and check each of them for feasibility. It can be done in time $O(|V| \cdot |E| + |V|^2 \log |V|)$ [19]. Thus the running-time of this case is

$$O(2^{2^{|V_{\geq 2}|}} \cdot (|V| \cdot |E| + |V|^2 \cdot \log |V|)).$$

Now assume that $\log |V| \geq |V_{\geq 2}|$. Observe that our graph can be represented as the vertices of $V_{\geq 2}$ that may or may not be joined to some vertices of V_1, plus we may have isolated edges (we can ignore isolated vertices), see Figure 4.

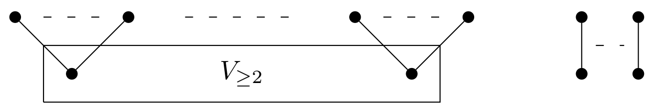

Figure 4: The way G looks from the perspective of $V_{\geq 2}$. We can ignore isolated vertices in G.

Observe that if our problem is a yes-instance, then there is a solution that takes some vertices from $V_{\geq 2}$ plus some independent vertices from the vertices of these isolated edges. Thus, we can consider the following simple algorithm: let us generate all subsets X of $V_{\geq 2}$ that have size at most k_1. We have $2^{|V_{\geq 2}|} \leq |V|$ possibilities. For each of these choices we add $k_1 - |X|$ independent vertices from isolated edges that have the largest coverage. We can do this in time $O(|V| \cdot \log |V|)$. We test the resulting set for feasibility in time $O(|V| \cdot |E| + |V|^2 \log |V|)$ [19]. Clearly, this algorithm solves our problem exactly. Moreover, in this case its running time is

$$2^{|V_{\geq 2}|} \cdot \left(O(|V| \cdot \log |V|) + O(|V| \cdot |E| + |V|^2 \cdot \log |V|)\right) = O(|V|^2 \cdot |E| + |V|^3 \cdot \log |V|).$$

Combining the two cases, we have that M-EPVCB is FPT with respect to $|V_{\geq 2}|$, and the running-time of the corresponding algorithm is

$$O\left(2^{2^{|V_{\geq 2}|}} \cdot \left(|V|^2 \cdot |E| + |V|^3 \cdot \log |V|\right)\right).$$

The proof is complete. □

7 Future Work

In this paper, we have shown that M-EPVCB is W[1]-hard or paraNP-hard with respect to many parameters. We also observed that in case of some parameters the problem is FPT. There are questions that deserve further investigation. Below we present some of them.

We have observed that M-EPVCB is FPT with respect to $|V| - k_1$. It would be interesting to investigate its hardness with respect to $|V| - 2 \cdot k_1$.

In Theorem 5, we have shown that M-EPVCB is W[1]-hard with respect to k_1. It would be interesting to strengthen this result and show

Conjecture 1 *M-EPVCB is W[1]-hard with respect to* $\min\{|X|, |Y|\}$.

Here X and Y denote the two sets in the bipartition of the input graph. In other words, the vertex-set of the input graph is $X \cup Y$ and any edge connects a vertex from X to a vertex from Y.

In Corollary 2, we have shown that M-EPVCB is less likely to be FPT with respect to $\Delta(G)$. We suspect that

Conjecture 2 *M-EPVCB is W[1]-hard with respect to* $|V| - \Delta(G)$.

We can show that Conjecture 1 implies Conjecture 2. It suffices to present an FPT reduction from M-EPVCB considered with respect to $\min\{|X|, |Y|\}$ to M-EPVCB considered with respect to $|V| - \Delta(G)$. Assume that we have G and let (X, Y) be the bipartition of G. Assume that $|Y| \leq |X|$. Add a new vertex z to Y and add $k = |V(G)|$ vertices to X. Finally join z to all vertices of X both old and new. Let H be the resulting bipartite graph. Observe that its size is polynomial in terms of G. Moreover, as we did in the proof of Theorem 6, we can multiply the weights of edges of G with a big constant C and define the new edges of H to have weight one. As in the proof of this theorem, we define the parameters k_1, k_2 and k_3 in the same way. One can show that originally we had a "yes"-instance if and only if the new instance is a "yes"-instance. Moreover, observe that this reduction is polynomial-time. It remains to bound the parameters. Observe that

$$|V(H)| = |V(G)| + 1 + k$$

and

$$\Delta(H) = |X| + k.$$

Thus,

$$|V(H)| - \Delta(H) = |Y| + 1 = \min\{|X|, |Y|\} + 1.$$

Thus, this reduction is an FPT-reduction. We finish the discussion with the following

Observation 3 *The hardness of M-EPVCB with respect to* $\min\{|X|, |Y|\}$ *is equivalent to that of M-EPVCB with respect to* $\nu(G)$.

Proof: Since $\nu(G) \leq \min\{|X|, |Y|\}$, one direction is trivial. Assume that the problem is hard with respect to $\min\{|X|, |Y|\}$. Let us show that it is hard with respect to $\nu(G)$. Let us embed $G = (X, Y, E)$ into a complete bipartite graph $K_{|X|,|Y|}$. As previously, we multiply edge-weights of old edges with a big, but polynomially bounded constant (see Theorem 6). The weights of new edges are one. As before, we have a polynomial-time reduction. Thus, the two conditions of FPT-reductions are satisfied. It suffices to show that $\nu(K_{|X|,|Y|})$ is bounded in terms of a function of $\min\{|X|, |Y|\}$. We have

$$\nu(K_{|X|,|Y|}) = \min\{|X|, |Y|\}.$$

Thus, the described reduction is an FPT reduction. The proof is complete. □

Acknowledgement

We thank our referees for their many comments on the previous version of the paper that helped us to improve the presentation of our results.

References

[1] A. A. Ageev and M. Sviridenko. Approximation algorithms for maximum coverage and max cut with given size of parts. In *IPCO*, pages 17–30, 1999. doi:10.1007/3-540-48777-8_2.

[2] O. Amini, F. Fomin, and S. Saurabh. Implicit branching and parameterized partial cover problems. *J. Comp. Sys. Sciences*, 77:1159–1171, 2011. doi:10.1016/j.jcss.2010.12.002.

[3] N. Apollonio and B. Simeone. Improved approximation of maximum vertex coverage problem on bipartite graphs. *SIAM J. Discret. Math.*, 28(3):1137–1151, 2014. doi:10.1137/130931059.

[4] N. Apollonio and B. Simeone. The maximum vertex coverage problem on bipartite graphs. *Discret. Appl. Math.*, 165:37–48, 2014. doi:10.1016/j.dam.2013.05.015.

[5] R. Bar-Yehuda. Using homogeneous weights for approximating the partial cover problem. *J. Algorithms*, 39(2):137–144, 2001. doi:10.1006/jagm.2000.1150.

[6] R. Bar-Yehuda, G. Flysher, J. Mestre, and D. Rawitz. Approximation of partial capacitated vertex cover. In L. Arge, M. Hoffmann, and E. Welzl, editors, *Algorithms - ESA 2007, 15th Annual European Symposium, Eilat, Israel, October 8-10, 2007, Proceedings*, volume 4698 of *Lecture Notes in Computer Science*, pages 335–346. Springer, 2007. doi:10.1007/978-3-540-75520-3_31.

[7] M. Bläser. Computing small partial coverings. *Inf. Process. Lett.*, 85(6):327–331, 2003. doi:10.1016/S0020-0190(02)00434-9.

[8] N. H. Bshouty and L. Burroughs. Massaging a linear programming solution to give a 2-approximation for a generalization of the vertex cover problem. In M. Morvan, C. Meinel, and D. Krob, editors, *STACS 98, 15th Annual Symposium on Theoretical Aspects of Computer Science, Paris, France, February 25-27, 1998, Proceedings*, volume 1373 of *Lecture Notes in Computer Science*, pages 298–308. Springer, 1998. doi:10.1007/BFb0028569.

[9] B. Caskurlu, A. Gehani, C. Ç. Bilgin, and K. Subramani. Analytical models for risk-based intrusion response. *Comput. Networks*, 57(10):2181–2192, 2013. doi:10.1016/j.comnet.2013.03.012.

[10] B. Caskurlu, V. Mkrtchyan, O. Parekh, and K. Subramani. On partial vertex cover and budgeted maximum coverage problems in bipartite graphs. In J. Díaz, I. Lanese, and D. Sangiorgi, editors, *Theoretical Computer Science - 8th IFIP TC 1/WG 2.2 International Conference, TCS 2014, Rome, Italy, September 1-3, 2014. Proceedings*, volume 8705 of *Lecture Notes in Computer Science*, pages 13–26. Springer, 2014. doi:10.1007/978-3-662-44602-7_2.

[11] B. Caskurlu, V. Mkrtchyan, O. Parekh, and K. Subramani. Partial vertex cover and budgeted maximum coverage in bipartite graphs. *SIAM J. Discret. Math.*, 31(3):2172–2184, 2017. doi:10.1137/15M1054328.

[12] M. Cesati. Compendium of parameterized problems. 2006. URL: http://cesati.sprg.uniroma2.it/research/compendium/compendium.pdf.

[13] J. Chen, Y. Guo, and Q. Huang. Linear-time parameterized algorithms with limited local resources. *CoRR*, abs/2003.02866, 2020. URL: https://arxiv.org/abs/2003.02866, arXiv:2003.02866.

[14] J. Chen and I. A. Kanj. Constrained minimum vertex cover in bipartite graphs: complexity and parameterized algorithms. *J. Comput. Syst. Sci.*, 67(4):833–847, 2003. doi:10.1016/j.jcss.2003.09.003.

[15] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

[16] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[17] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Ann. of Math.*, 162(1):439–485, 2005.

[18] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness II: on completeness for W[1]. *Theor. Comput. Sci.*, 141(1&2):109–131, 1995. doi:10.1016/0304-3975(94)00097-3.

[19] H. N. Gabow. Data structures for weighted matching and nearest common ancestors with linking. In D. S. Johnson, editor, *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California, USA*, pages 434–443. SIAM, 1990. URL: http://dl.acm.org/citation.cfm?id=320176.320229.

[20] R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. *J. Algorithms*, 53(1):55–84, 2004. doi:10.1016/j.jalgor.2004.04.002.

[21] J. Guo, R. Niedermeier, and S. Wernicke. Parameterized complexity of generalized vertex cover problems. In F. K. H. A. Dehne, A. López-Ortiz, and J. Sack, editors, *Algorithms and Data Structures, 9th International Workshop, WADS 2005, Waterloo, Canada, August 15-17, 2005, Proceedings*, volume 3608 of *Lecture Notes in Computer Science*, pages 36–48. Springer, 2005. doi:10.1007/11534273_5.

[22] D. S. Hochbaum. The t-vertex cover problem: Extending the half integrality framework with budget constraints. In K. Jansen and D. S. Hochbaum, editors, *Approximation Algorithms for Combinatorial Optimization, International Workshop APPROX'98, Aalborg, Denmark, July 18-19, 1998, Proceedings*, volume 1444 of *Lecture Notes in Computer Science*, pages 111–122. Springer, 1998. doi:10.1007/BFb0053968.

[23] G. Joret and A. Vetta. Reducing the rank of a matroid. *Discret. Math. Theor. Comput. Sci.*, 17(2):143–156, 2015. URL: http://dmtcs.episciences.org/2135.

[24] G. Karakostas. A better approximation ratio for the vertex cover problem. *ACM Trans. Algorithms*, 5(4):41:1–41:8, 2009. doi:10.1145/1597036.1597045.

[25] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

[26] S. Khot, D. Minzer, and M. Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. *Electron. Colloquium Comput. Complex.*, page 6, 2018. URL: https://eccc.weizmann.ac.il/report/2018/006.

[27] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. doi:10.1016/j.jcss.2007.06.019.

[28] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999. doi:10.1016/S0020-0190(99)00031-9.

[29] J. Kneis, A. Langer, and P. Rossmanith. Improved upper bounds for partial vertex cover. In H. Broersma, T. Erlebach, T. Friedetzky, and D. Paulusma, editors, *Graph-Theoretic Concepts in Computer Science, 34th International Workshop, WG 2008, Durham, UK, June 30 - July 2, 2008. Revised Papers*, volume 5344 of *Lecture Notes in Computer Science*, pages 240–251, 2008. doi:10.1007/978-3-540-92248-3_22.

[30] J. Kneis, D. Mölle, S. Richter, and P. Rossmanith. Intuitive algorithms and t-vertex cover. In T. Asano, editor, *Algorithms and Computation, 17th International Symposium, ISAAC 2006, Kolkata, India, December 18-20, 2006, Proceedings*, volume 4288 of *Lecture Notes in Computer Science*, pages 598–607. Springer, 2006. doi:10.1007/11940128_60.

[31] J. Kneis, D. Mölle, and P. Rossmanith. Partial vs. complete domination: t-dominating set. In J. van Leeuwen, G. F. Italiano, W. van der Hoek, C. Meinel, H. Sack, and F. Plasil, editors, *SOFSEM 2007: Theory and Practice of Computer Science, 33rd Conference on Current Trends in Theory and Practice of Computer Science, Harrachov, Czech Republic, January 20-26, 2007, Proceedings*, volume 4362 of *Lecture Notes in Computer Science*, pages 367–376. Springer, 2007. doi:10.1007/978-3-540-69507-3_31.

[32] J. Könemann, O. Parekh, and D. Segev. A unified approach to approximating partial covering problems. *Algorithmica*, 59(4):489–509, 2011. doi:10.1007/s00453-009-9317-0.

[33] J. Mestre. A primal-dual approximation algorithm for partial vertex cover: Making educated guesses. *Algorithmica*, 55(1):227–239, 2009. doi:10.1007/s00453-007-9003-z.

[34] V. Mkrtchyan, O. Parekh, D. Segev, and K. Subramani. The approximability of partial vertex covers in trees. In B. Steffen, C. Baier, M. van den Brand, J. Eder, M. Hinchey, and T. Margaria, editors, *SOFSEM 2017: Theory and Practice of Computer Science - 43rd International Conference on Current Trends in Theory and Practice of Computer Science, Limerick, Ireland, January 16-20, 2017, Proceedings*, volume 10139 of *Lecture Notes in Computer Science*, pages 350–360. Springer, 2017. doi:10.1007/978-3-319-51963-0_27.

[35] V. Mkrtchyan, G. Petrosyan, K. Subramani, and P. J. Wojciechowski. Parameterized algorithms for partial vertex covers in bipartite graphs. In L. Gasieniec, R. Klasing, and T. Radzik, editors, *Combinatorial Algorithms - 31st International Workshop, IWOCA 2020, Bordeaux, France, June 8-10, 2020, Proceedings*, volume 12126 of *Lecture Notes in Computer Science*, pages 395–408. Springer, 2020. doi:10.1007/978-3-030-48966-3_30.

[36] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991. doi:10.1016/0022-0000(91)90023-X.

[37] V. T. Paschos. A polynomial time approximation schema for maximum k-vertex cover in bipartite graphs. *CoRR*, abs/1909.08435, 2019. URL: http://arxiv.org/abs/1909.08435, arXiv:1909.08435.

[38] U. Stege, I. van Rooij, A. Hertel, and P. Hertel. An o(pn + 1.151^{p})-algorithm for p-profit cover and its practical implications for vertex cover. In P. Bose and P. Morin, editors, *Algorithms and Computation, 13th International Symposium, ISAAC 2002 Vancouver, BC, Canada, November 21-23, 2002, Proceedings*, volume 2518 of *Lecture Notes in Computer Science*, pages 249–261. Springer, 2002. doi:10.1007/3-540-36136-7_23.

[39] V. V. Vazirani. *Approximation algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001. URL: http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-65367-7.