

# Approximation Algorithm for Vertex Cover with Multiple Covering Constraints

**Eunpyeong Hong**

Department of Computer Science and Information Engineering,  
National Taiwan University, Taipei, Taiwan  
ephong93@gmail.com

**Mong-Jen Kao**

Department of Computer Science and Information Engineering,  
National Chung-Cheng University, Chiayi, Taiwan  
mjkao@cs.ccu.edu.tw

---

## Abstract

We consider the vertex cover problem with multiple coverage constraints in hypergraphs. In this problem, we are given a hypergraph  $G = (V, E)$  with a maximum edge size  $f$ , a cost function  $w : V \rightarrow \mathbb{Z}^+$ , and edge subsets  $P_1, P_2, \dots, P_r$  of  $E$  along with covering requirements  $k_1, k_2, \dots, k_r$  for each subset. The objective is to find a minimum cost subset  $S$  of  $V$  such that, for each edge subset  $P_i$ , at least  $k_i$  edges of it are covered by  $S$ . This problem is a basic yet general form of classical vertex cover problem and a generalization of the edge-partitioned vertex cover problem considered by Bera et al.

We present a primal-dual algorithm yielding an  $(f \cdot H_r + H_r)$ -approximation for this problem, where  $H_r$  is the  $r^{\text{th}}$  harmonic number. This improves over the previous ratio of  $(3cf \log r)$ , where  $c$  is a large constant used to ensure a low failure probability for Monte-Carlo randomized algorithms. Compared to previous result, our algorithm is deterministic and pure combinatorial, meaning that no Ellipsoid solver is required for this basic problem. Our result can be seen as a novel reinterpretation of a few classical tight results using the language of LP primal-duality.

**2012 ACM Subject Classification** Mathematics of computing  $\rightarrow$  Approximation algorithms

**Keywords and phrases** Vertex cover, multiple cover constraints, Approximation algorithm

**Digital Object Identifier** 10.4230/LIPIcs.ISAAC.2018.43

**Funding** This work is supported in part by Ministry of Science and Technology (MOST), Taiwan, under Grants MOST107-2218-E-194-015-MY3 and MOST106-2221-E-001-006-MY3.

## 1 Introduction

The vertex cover problem is one of the most well-known and fundamental problem in graph theory and approximation algorithms. Given an undirected hypergraph  $G = (V, E)$  and a cost function  $w : V \rightarrow \mathbb{Z}^+$ , the objective is to find a minimum cost subset  $S \subseteq V$  such that any edge in  $E$  is incident to some vertex in  $S$ .

This problem is known to be NP-hard, and  $f$ -approximation algorithms based on simple LP rounding and LP primal-duality are known for this problem [10], where  $f$  is the maximum size of the hyperedges. Assuming the unique game conjecture, approximating this problem to a ratio better than  $(f - \epsilon)$  is NP-hard for any  $\epsilon > 0$  [8].

The partial vertex cover problem is a natural generalization of the vertex cover problem. In this problem, we are given an additional parameter  $k$  which is called the covering requirement. The objective of this problem is to find a minimum cost subset of  $V$  which covers at least  $k$  edges in  $E$ , i.e., at least  $k$  edges of  $E$  are incident to at least one vertex in  $S$ .



© Eunpyeong Hong and Mong-Jen Kao;  
licensed under Creative Commons License CC-BY

29th International Symposium on Algorithms and Computation (ISAAC 2018).

Editors: Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao; Article No. 43; pp. 43:1–43:11

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Various methods have been developed to obtain tight approximations for this problem. Bshouty and Burroughs [3], who first proposed this problem, provided a 2-approximation algorithm for graphs, i.e., the case for which  $f = 2$ , using LP-rounding. Their algorithm generates  $|V|$  candidate covers each of which is constructed by guessing the most expensive vertex used in the optimal solution. Gandhi et al. [6] used the same technique to develop a primal-dual method which yields an  $f$ -approximation for hypergraphs. Mestre [9] used a more clever way to guess the most expensive vertex and improved the time complexity of the above algorithms.

Fujito [5] developed an  $f$ -approximation for hypergraphs, based on a primal-dual method exploiting the property of minimal solutions. Bar-Yehuda [1] used the same property to obtain the same result using a local-ratio method. Hochbaum [7] adopted Lagrangian relaxation to get a 2-approximation on graphs.

Bera et al. [2] considered a generalization of the partial vertex cover problem for which they called the partition vertex cover problem. In this problem, we are given a partition  $E_1, E_2, \dots, E_r$  of the edges along with covering requirements  $k_1, k_2, \dots, k_r$ . The objective is to find a minimum cost vertex subset that covers at least  $k_i$  edges of  $E_i$  for each  $1 \leq i \leq r$ .

They obtained a  $(6c \log r)$ -approximation for normal graphs, where  $c$  is a large constant used for Monte-Carlo randomized algorithms to ensure low error probability. They used randomized iterative rounding on a strong LP which is derived by knapsack inequalities on the natural LP. This approach generalizes to hypergraphs with an approximation guarantee of  $(3cf \log r)$ . They also showed that, even for normal graph for which  $f = 2$ , it is NP-hard to approximate this problem to a ratio better than  $H_r$ , which means  $O(f)$ -approximation for this problem is unlikely to exist.

Wolsey [11] proposed the submodular set cover problem, which is a general formulation to the above covering problems, and presented an  $H(\max_{S \in \mathcal{S}} g(\{S\}))$ -approximation, where  $g$  is the input submodular function and  $S$  is the ground set. Chuzhoy et al. [4] presented a simpler analysis to obtain a similar result. Fujito [5] presented a primal-dual algorithm for this problem which is useful for some special cases such as the partial vertex cover problem.

## Our Focus and Contributions

In this paper, we consider the vertex cover problem with multiple covering constraints (VC-MCC) in hypergraphs. In this problem, we are given a hypergraph  $G = (V, E)$ , a cost function  $w : V \rightarrow \mathbb{Z}^+$ , and a number of covering constraints  $(P_1, k_1), (P_2, k_2), \dots, (P_r, k_r)$ , where each  $P_i \subseteq E$  is a subset of  $E$  and  $k_i \in \mathbb{Z}^+$  is the covering requirement for  $P_i$ . The objective is to find a minimum cost subset  $S \subseteq V$  such that, for each  $1 \leq i \leq r$ , at least  $k_i$  edges of  $P_i$  are covered by  $S$ .

This problem is a basic yet general form of classical vertex cover and a further generalization of the edge-partitioned vertex cover problem considered in [2].

In this paper, we present a primal-dual algorithm that yields an  $(f \cdot H_r + H_r)$ -approximation for this problem, improving over the previous ratio of  $(cf \log r)$  due to [2]. Our main contribution is the following theorem.

► **Theorem 1.** *There is a deterministic  $(f \cdot H_r + H_r)$ -approximation algorithm for VC-MCC which runs in polynomial time, where  $H_r$  is the  $r^{\text{th}}$  harmonic number.*

Compared to the previous result of  $(cf \log r)$ , our algorithm is deterministic and pure combinatorial, which means that our algorithm does not rely on heavy Ellipsoid LP solvers for this basic problem. Considering the lower-bound of  $H_r$  on the approximation ratio due to [2] and the well-known lower-bound of  $f$  for vertex cover, our result is much closer to the tight extent possible.

The novelty of this work lies in the way how we handle the dual variables. In contrast to previously known primal-dual approaches for covering problems, in which the dual variables can be handled freely, our approach manages the dual solutions carefully so that the following two criteria are met.

1. During the process, the cost of any vertex to be opened in the future must only be paid by the dual values possessed by the current unfulfilled covering constraints.
2. The overall dual value possessed by any unfulfilled covering constraint remains the same all the time.

This makes the approximation guarantee of  $\log r$  possible.

Our result can be seen as a novel combination of the classical tight approximations with guarantees  $f$  and  $H_r$  for the covering problem, using the language of LP primal-duality.

Our ingredient includes the strong LP relaxation due to [2], which is derived by applying Knapsack-cover inequalities to the natural LP. We would like to remark, however, that the usage of strong LP relaxation in our result is not a necessity but rather a better and more intuitive exposition of our ideas on how the dual variables can be managed, and obtaining the same result using natural LP is possible.

### Organization of this paper

The rest of this paper is organized as follows. In Section 2, we define the notations we will be using throughout this paper and introduce the strong LP formulations. We present our approximation algorithm in Section 3 and conclude with future directions in Section 4.

## 2 Preliminary

We use  $G = (V, E)$  to denote a hypergraph  $G$  with a vertex set  $V$  and an edge set  $E \subseteq 2^V$ . Note that, under this notion, any edge  $e \in E$  is a subset of  $V$  that consists the incident vertices of the edge  $e$ . We use  $f_G$  to denote the maximum cardinality of the edges in  $G$ , i.e.,  $f_G = \max_{e \in E} |e|$ . The subscript  $G$  is omitted when no ambiguity is there in the context.

For any edge subset  $M \subseteq E$  and any vertex  $v \in V$ , we use  $M(v)$  to denote the set of edges in  $M$  that are incident to  $v$ , i.e.,  $M(v) := \{e \in M : v \in e\}$ . For any subset  $A \subseteq V$ , we use  $M(A)$  to denote the set of edges in  $M$  that are incident to the vertices in  $A$ , i.e.,  $M(A) = \bigcup_{v \in A} M(v)$ .

### Vertex Cover with Multiple Covering Constraints

In this problem, we are given a hypergraph  $G = (V, E)$ , a cost function  $w : V \rightarrow \mathbb{Z}^+$ , and a number of covering constraints  $(P_1, k_1), (P_2, k_2), \dots, (P_r, k_r)$ , where for each  $1 \leq i \leq r$ ,  $P_i \subseteq E$  is a subset of  $E$  and  $k_i \in \mathbb{Z}^+$  is the covering requirement for  $P_i$  to be fulfilled.<sup>1</sup> The objective of this problem is to find a vertex subset  $S \subseteq V$  of minimum cost such that  $|P_i(S)| \geq k_i$  for each  $1 \leq i \leq r$ .

Intuitively, this problem asks for a minimum cost subset such that in each  $P_i$ , at least  $k_i$  edges are covered. A natural LP relaxation for this problem is given in Figure 1.

We have two sets of indicator variables in this LP formulation: For each  $v \in V$ ,  $x_v$  denotes the inclusion of  $v$  into the cover and  $y_e$  for each  $e \in E$  indicates the coverage of  $e$  by the

<sup>1</sup> Without loss of generality, we assume that  $k_i \leq |P_i|$  for all  $1 \leq i \leq r$ .

$$\begin{array}{ll}
 \min & \sum_{v \in V} w_v x_v & \text{LP-(N)} \\
 \text{s.t.} & \sum_{v \in e} x_v \geq y_e, & \forall e \in E \\
 & \sum_{e \in P_i} y_e \geq k_i, & \forall 1 \leq i \leq r \\
 & x_v \geq 0, & \forall v \in V \\
 & 1 \geq y_e \geq 0, & \forall e \in E.
 \end{array}$$

■ **Figure 1** A natural LP relaxation for VC-MCC.

vertices chosen in the cover. The first inequality models the coverage of each edge  $e \in E$  and the second inequality models the covering requirement for each  $(P_i, k_i)$ ,  $1 \leq i \leq r$ .

However, the integrality gap of the natural LP can be arbitrarily large. This is illustrated by the following simple example. Consider a star with  $d + 1$  vertices. Suppose that the cost of every vertex is 1 and we only have one constraint consisting of all edges with covering requirement 1. The optimal integral cost for this example is 1 while its optimal fractional cost is  $1/d$ , resulting a gap of  $d$  which can be arbitrarily large.

### A Strong LP Relaxation

Instead of using the natural LP relaxation, we use a strong LP relaxation due to [2], which is derived by applying Knapsack-cover inequalities to the natural LP given above.

For any vertex subset  $A \subseteq V$  and any  $1 \leq i \leq r$ , define

$$k_i(A) := \max \left\{ k_i - |P_i(A)|, 0 \right\}.$$

Intuitively,  $k_i(A)$  denotes the residue covering requirement to be fulfilled for  $P_i$ , if the vertex set  $A$  were already chosen as part of the cover.

For any vertex  $v \in V \setminus A$ , define

$$\beta_i(v, A) := \min \left\{ |P_i(v) \setminus P_i(A)|, k_i(A) \right\}.$$

Intuitively,  $\beta_i(v, A)$  is the amount of covering requirement  $v$  can be fulfilled for  $P_i$  if  $A$  is already chosen as part of the cover. Clearly,  $\beta_i(v, A)$  will be either  $k_i(A)$  or the number of incident edges of  $v$  in  $P_i \setminus P_i(A)$ , which is  $|P_i(v) \setminus P_i(A)|$ .

The strong LP relaxation we consider is as follows:

$$\begin{array}{ll}
 \min & \sum_{v \in V} w_v \cdot x_v & \text{LP-(S)} \\
 \text{s.t.} & \sum_{v \in V \setminus A} \beta_i(v, A) \cdot x_v \geq k_i(A), & \forall 1 \leq i \leq r, \forall A \subseteq V \\
 & x_v \geq 0, & \forall v \in V.
 \end{array}$$

To see that LP-(S) gives a valid relaxation for VC-MCC, consider any feasible integral solution  $\hat{x}$ . It suffices to show that  $\hat{x}$  is also contained in the feasible region of LP-(S).

Consider an arbitrary subset  $A$  of  $V$  and any constraint  $1 \leq i \leq r$ . Clearly,  $\hat{x}$  must remain feasible even if the vertices of  $A$  were already chosen as part of the cover in advance for free. Hence, the number of edges  $\hat{x}$  covers for  $P_i$  is at least  $k_i(A)$ , and the inequality

$$\sum_{v \in V \setminus A} \beta_i(v, A) \cdot x_v \geq k_i(A)$$

must hold.

To see that LP-(S) is indeed a stronger relaxation than LP-(N), let us consider the simple star example and the inequality with respect to  $A = \emptyset$ . Clearly,  $\beta_1(v, A) = 1$  for all vertices  $v$  in this star. As a result, we have a constraint  $\sum_{v \in V} x_v \geq 1$ , and the optimal fractional solution will also be 1.

### The Dual LP for LP-(S)

In this paper we will be working around the dual LP of LP-(S), which is given as follows.

$$\begin{array}{ll} \max & \sum_{1 \leq i \leq r, A \subseteq V} k_i(A) \cdot z_{i,A} & \text{LP-Dual-(S)} \\ \text{s.t.} & \sum_{1 \leq i \leq r, A \subseteq V \setminus \{v\}} \beta_i(v, A) \cdot z_{i,A} \leq w_v, & \forall v \in V, \quad (*) \\ & z_{i,A} \geq 0, & \forall 1 \leq i \leq r, \forall A \subseteq V \end{array}$$

## 3 Our Approximation Algorithm for VC-MCC

In this section, we present our approximation algorithm for VC-MCC. Given an instance  $\Pi = (G = (V, E), w, (P_i, k_i)_{1 \leq i \leq r})$  of VC-MCC, the algorithm will compute a series of feasible LP solutions of  $\Pi$  to LP-Dual-(S). During this process, a feasible cover for  $\Pi$  will gradually be formed. The approximation guarantee is then established by comparing the cost of the cover to the values of the dual solutions the algorithm computes.

In the following section we describe the algorithm in details. In §3.2 we establish the approximation guarantee.

### 3.1 The Algorithm

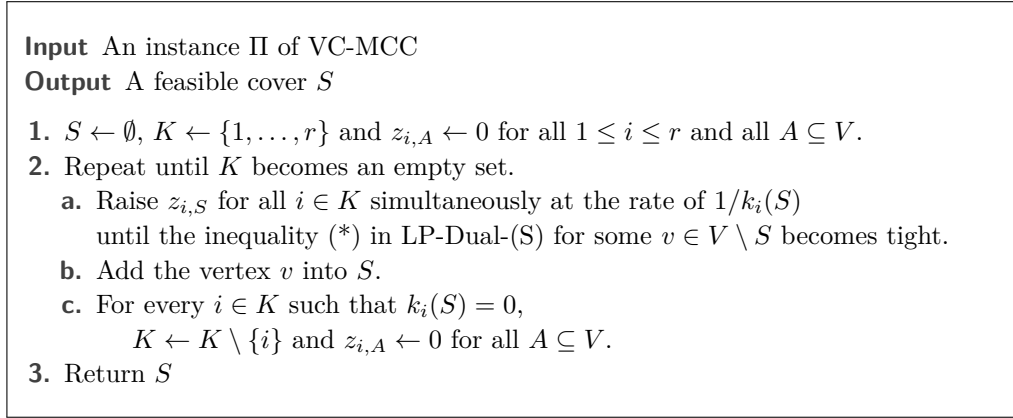
The algorithm takes as input an instance  $\Pi$  of VC-MCC and outputs a feasible cover  $S$  for  $\Pi$ .

Initially,  $S$  is set to be an empty set. In addition, the algorithm will maintain a set  $K$  which contains the set of covering constraints that have not been satisfied yet. The set  $K$  is initialized to be  $\{1, 2, \dots, r\}$ .

In the following, we first describe our primal-dual process. Then we describe how this primal-dual process can be transformed into a polynomial-time algorithm.

Our primal-dual process, denoted PD-VC-MCC, starts with a trivial dual solution for which  $z_{i,A} = 0$  for all  $1 \leq i \leq r$  and all  $A \subseteq V$ . In each iteration, it proceeds as follows:

1. It raises  $z_{i,S}$  at the rate of  $1/k_i(S)$  for all  $i \in K$  until the Inequality (\*) of some vertex in LP-Dual-(S), say,  $v \in V \setminus S$ , becomes tight. Then it adds the vertex  $v$  to the set  $S$ .



■ **Figure 2** A formal description of our primal-dual process PD-VC-MCC.

2. For each constraint  $i \in K$  with  $k_i(S)$  becoming zero after  $v$  is added to  $S$ , our primal-dual process:
  - a. sets  $z_{i,A}$  to be zero for all  $A \subseteq V$  and
  - b. removes  $i$  from  $K$ .

This process repeats until the set  $K$  becomes empty. Then  $S$  is returned as the approximate solution. A high-level pseudo-code of this primal-dual process is given in Figure 2 for further reference.

We remark that, the step 2.(a) above of resetting  $z_{i,A}$  to zero for all  $A \subseteq V$  when  $i$  is to be removed from  $K$  is very important and is the key to obtain a guarantee of  $H_r$ . The reason is that it allows the overall contribution of the remaining covering constraints to remain balanced.

Our approximation algorithm, denoted Approx-VC-MCC, mimics the operations of the above primal-dual process. Instead of maintaining the dual variables, it keeps track of the slack of the Inequality (\*) in LP-Dual-(S) for each vertex, i.e., the amount before it becomes tight.

For each  $v \in V$ , let  $\hat{w}_v$  denote the slack of the vertex constraint  $v$  before it becomes tight. Initially,  $\hat{w}_v$  is set to be  $w_v$ . We need a notion that reflects the raising process of the dual variables. For each  $v \in V$  and  $A \subseteq V \setminus \{v\}$ , define

$$s(v, A) := \sum_{i \in K} \frac{\beta_i(v, A)}{k_i(A)}.$$

Intuitively,  $s(v, A)$  denotes the speed for which  $\hat{w}_v$  will decrease if we raised the dual variables  $z_{i,A}$  at the speed of  $1/k_i(A)$  for all  $i \in K$ .

Furthermore, in order for the update of  $\hat{w}_v$  for each  $v \in V$  to proceed, we use  $\Phi_{v,i}$  to denote the contribution of dual variables  $z_{i,A}$  for all possible  $A$  towards  $\hat{w}_v$ .  $\Phi_{v,i}$  is initialized to be zero for all  $v \in V$  and  $1 \leq i \leq r$ .

Now we formally describe our approximation algorithm. In each iteration, the algorithm finds the among the vertices in  $V \setminus S$  the one with the smallest ratio of  $\hat{w}_v/s(v, A)$ . Formally speaking, it computes

$$v = \arg \min_{u \in V \setminus S} \frac{\hat{w}_u}{s(u, A)} \quad \text{and} \quad t_v = \frac{\hat{w}_v}{s(v, A)}.$$

Intuitively,  $v$  is the first vertex constraint to become tight in this iteration in the primal-dual process and  $t_v$  is the corresponding amount of time it takes.

Then the algorithm proceeds as follows:

1. For each  $u \in V \setminus S$ , the algorithm:
  - a. updates  $\hat{w}_u$  by setting  $\hat{w}_u \leftarrow \hat{w}_u - s(u, S) \cdot t_v$ .
  - b. update the contribution  $\Phi_{u,i}$  for each  $i \in K$  by setting  $\Phi_{u,i} \leftarrow \Phi_{u,i} + \frac{\beta_i(u, S)}{k_i(S)} \cdot t_v$ .
2. Add  $v$  to the set  $S$ .
3. For each  $i \in K$  such that  $k_i(S)$  is zero, the algorithm does the following:
  - a. Update  $\hat{w}_u$  for all  $u \in V \setminus S$  by setting  $\hat{w}_u \leftarrow \hat{w}_u + \Phi_{u,i}$ .
  - b. Remove  $i$  from  $K$ .

The algorithm repeats until the set  $K$  becomes empty. Then  $S$  is returned as the approximate solution.

### 3.2 Analysis

In this section, we provide the analysis of our approximation algorithm Approx-VC-MCC and prove Theorem 1. First we show that our algorithm always terminates and returns a feasible cover. Then we establish the approximation guarantee.

#### Feasibility of algorithm Approx-VC-MCC

We first establish the feasibility of our primal-dual process. Then we argue that algorithm Approx-VC-MCC does mimic the execution of this process and runs in polynomial time.

► **Lemma 2.** *The primal-dual process PD-VC-MCC always terminates and returns a feasible cover.*

**Proof.** Since PD-VC-MCC only terminates when the set  $K$  becomes empty and it finds a feasible cover, it suffices to argue that PD-VC-MCC always terminates, provided that there is a feasible cover for the input instance.

Assume for contradiction that the input instance has a feasible solution but PD-VC-MCC does not terminate. Consider the set  $S$  the process currently has. The process runs eternally since no vertex  $v \in V \setminus S$  becomes tight as  $z_{i,S}$  is constantly raising for all  $i \in K$ . This implies that  $\beta_i(v, S) = |P_i(v) \setminus P_i(S)| = 0$  for all  $v \in V \setminus S$  and all  $i \in K$ .

This means that all the edges have already been covered by  $S$ , a contradiction. ◀

To see that algorithm Approx-VC-MCC simulates the execution of PD-VC-MCC, it suffices to observe that

- $\hat{w}_v$  records the slack  $w_v - \sum_{1 \leq i \leq r, A \subseteq V \setminus \{v\}} \beta_i(v, A) \cdot z_{i,A}$  of the constraint (\*) for all  $v \in V \setminus S$  during all iterations,
- $\Phi_{v,i}$  keeps track of the value  $\sum_{A \subseteq V \setminus \{v\}} \beta_i(v, A) \cdot z_{i,A}$  so that it can be used to reflect the operation of resetting  $z_{i,A}$  to zero for  $i$  that is about to be removed from  $K$ .

We have the following lemma.

► **Lemma 3.** *Algorithm Approx-VC-MCC mimics the execution of primal-dual process PD-VC-MCC and runs in polynomial time.*

Lemma 2 and Lemma 3 establish the feasibility of algorithm Approx-VC-MCC.

**Approximation Guarantee**

To establish the approximation guarantee, we compare the cost of the solution our algorithm returns to the values of the dual solutions our primal-dual process maintains, which will be valid lower-bounds for the cost of optimal solutions by the weak LP duality.

Let  $S = \{v_1, v_2, \dots, v_m\}$  denote the cover returned by the algorithm, where the indices of the vertices denote the order for which they are added to the set  $S$ . For any  $0 \leq j \leq m$ , we use  $A_j$  to denote the set of the first  $j$  vertices that are added to  $S$ , i.e.,  $A_j := \{v_1, v_2, \dots, v_j\}$ .

Without loss of generality, we also assume that the covering constraints  $P_1, P_2, \dots, P_r$  are fulfilled by the algorithm in this order.

For any  $1 \leq i \leq r$ , let  $\pi(i)$  denote the index for which the inclusion of  $v_{\pi(i)}$  into  $S$  fulfills  $P_i$ . Consider the moment when  $P_i$  is just fulfilled, i.e., when  $i$  was removed from  $K$  by the algorithm, and  $z_{i,A}$  has not yet been reset. Let  $\hat{z}^{(i)}$  denote the dual solution the algorithm maintains at this moment, and  $\text{Val}(\hat{z}^{(i)})$  denote the objective value of  $\hat{z}^{(i)}$ . It follows that

$$\text{Val}(\hat{z}^{(i)}) := \sum_{1 \leq t \leq r, A \subseteq V} k_t(A) \cdot \hat{z}_{t,A}^{(i)} = \sum_{i \leq t \leq r} \sum_{0 \leq j < \pi(i)} k_t(A_j) \cdot \hat{z}_{t,A_j}^{(i)}, \quad (1)$$

where the second equality holds since our algorithm resets  $\hat{z}_{t,A}^{(i)}$  to zero for all  $1 \leq t < i$  and all  $A \subseteq V$ .

In the above equality we write  $\text{Val}(\hat{z}^{(i)})$  as the sum of dual values each unfulfilled covering requirement possesses. The following lemma says that the dual value possessed by each unfulfilled constraint is the same.

► **Lemma 4.** *For any  $1 \leq i \leq r$  and any  $t_1, t_2$  with  $i \leq t_1 \neq t_2 \leq r$ , we have*

$$\sum_{0 \leq j < \pi(i)} k_{t_1}(A_j) \cdot \hat{z}_{t_1, A_j}^{(i)} = \sum_{0 \leq j < \pi(i)} k_{t_2}(A_j) \cdot \hat{z}_{t_2, A_j}^{(i)}.$$

**Proof.** This lemma follows directly from the way our primal-dual approach handles the dual variables. Since  $k_i(S)$  only changes when a new vertex becomes tight and since we always raise  $z_{i,S}$  for each  $i \in K$  at the rate of  $1/k_i(S)$ , the total dual value possessed by any  $P_i$  with  $i \in K$  will be the same. ◀

In the following we analyze the cost of  $S$  and relate it to the dual values of  $\hat{z}^{(i)}$  the algorithm maintains for all  $1 \leq i \leq r$ .

Consider a vertex  $v \in S$  and the moment when  $v$  just becomes tight. Suppose that at that time, the algorithm has already fulfilled  $t$  covering constraints. Then, from the Inequality (\*) of LP-Dual-(S), it follows that

$$w_v = \sum_{1 \leq i \leq r, A \subseteq V \setminus \{v\}} \beta_i(v, A) \cdot z_{i,A} = \sum_{t < i \leq r, A \subseteq V \setminus \{v\}} \beta_i(v, A) \cdot z_{i,A},$$

where the second equality holds since, by design, our algorithm has already reset  $z_{i,A}$  to zero for all  $1 \leq i \leq t$  and all  $A \subseteq V$  before  $v$  becomes tight.

For each  $t < i \leq r$ , define  $\Phi_{v,i} := \sum_{A \subseteq V \setminus \{v\}} \beta_i(v, A) \cdot z_{i,A}$ . Then we have

$$w_v = \sum_{t < i \leq r} \Phi_{v,i}.$$

Intuitively,  $\Phi_{v,i}$  is the share for which the covering constraint  $i$  contributes towards the cost of vertex  $v$ . We will charge the cost of  $v$  to the covering constraints  $P_i$  for all  $t < i \leq r$ , each of which gets a charge of  $\Phi_{v,i}$ .



For each  $1 \leq i \leq r$ , let  $\text{cost}(P_i)$  denote the total charge  $P_i$  receives from the vertices in  $S$ . We have the following lemma, which bounds  $\text{cost}(P_i)$  using the dual value it possesses in  $\hat{z}^{(i)}$ .

► **Lemma 5.** *For any  $1 \leq i \leq r$ , we have*

$$\text{cost}(P_i) \leq (f+1) \cdot \sum_{0 \leq j < \pi(i)} k_i(A_j) \cdot \hat{z}_{i,A_j}^{(i)}.$$

**Proof.** From the definition of  $\text{cost}(P_i)$ , only  $v_1, v_2, \dots, v_{\pi(i)}$  will charge  $P_i$ , and it follows that

$$\begin{aligned} \text{cost}(P_i) &= \sum_{1 \leq t \leq \pi(i)} \Phi_{v_t, i} = \sum_{1 \leq t \leq \pi(i)} \sum_{A \subseteq V \setminus \{v_t\}} \beta_i(v_t, A) \cdot \hat{z}_{i,A}^{(i)} \\ &= \sum_{1 \leq t \leq \pi(i)} \sum_{0 \leq j < t} \beta_i(v_t, A_j) \cdot \hat{z}_{i,A_j}^{(i)} \\ &= \sum_{0 \leq j < \pi(i)} \sum_{j < t \leq \pi(i)} \beta_i(v_t, A_j) \cdot \hat{z}_{i,A_j}^{(i)}. \end{aligned}$$

Hence, to prove this lemma, it suffices to show that

$$\sum_{0 \leq j < \pi(i)} \sum_{j < t \leq \pi(i)} \beta_i(v_t, A_j) \cdot \hat{z}_{i,A_j}^{(i)} \leq (f+1) \cdot \sum_{0 \leq j < \pi(i)} k_i(A_j) \cdot \hat{z}_{i,A_j}^{(i)}. \quad (2)$$

■ To prove Ineq. (2), we first prove the following inequality:

$$\sum_{0 \leq j < \pi(i)} \sum_{j < t < \pi(i)} \beta_i(v_t, A_j) \cdot \hat{z}_{i,A_j}^{(i)} \leq f \cdot \sum_{0 \leq j < \pi(i)} k_i(A_j) \cdot \hat{z}_{i,A_j}^{(i)} \quad (3)$$

Compare the l.h.s. and the r.h.s. of (3), it suffices to argue that

$$\sum_{j < t < \pi(i)} \beta_i(v_t, A_j) \leq f \cdot k_i(A_j) \quad \text{for all } 0 \leq j < \pi(i).$$

Consider any fixed  $j$  with  $0 \leq j < \pi(i)$  and any  $t$  with  $j < t < \pi(i)$ . Since  $v_t$  is not the vertex whose inclusion into  $S$  fulfills  $P_i$ , it follows that  $|P_i(v_t) \setminus P_i(A_j)| < k_i(A_j)$ , and hence  $\beta_i(v_t, A_j) = |P_i(v_t) \setminus P_i(A_j)|$ .

Furthermore, under the condition that  $A_j$  has already been chosen, the inclusion of  $\{v_{j+1}, v_{j+2}, \dots, v_{\pi(i)-1}\}$  into  $S$  does not fulfill  $P_i$ .

This implies that  $\left| \bigcup_{j < t < \pi(i)} (P_i(v_t) \setminus P_i(A_j)) \right| < k_i(A_j)$ . Therefore, we have

$$\begin{aligned} \sum_{j < t < \pi(i)} \beta_i(v_t, A_j) &= \sum_{j < t < \pi(i)} |P_i(v_t) \setminus P_i(A_j)| \\ &\leq f \cdot \left| \bigcup_{j < t < \pi(i)} (P_i(v_t) \setminus P_i(A_j)) \right| < f \cdot k_i(A_j), \end{aligned}$$

where the second last inequality holds since the size of each hyperedge is at most  $f$ . This proves Ineq. (3).

## 43:10 On Vertex Cover with Multiple Covering Constraints

■ Second, observe that the following inequality holds

$$\sum_{0 \leq j < \pi(i)} \beta_i(v_{\pi(i)}, A_j) \cdot \hat{z}_{i, A_j}^{(i)} \leq \sum_{0 \leq j < \pi(i)} k_i(A_j) \cdot \hat{z}_{i, A_j}^{(i)}, \quad (4)$$

since  $\beta_i(v_{\pi(i)}, A_j) \leq k_i(A_j)$  by the definition of  $\beta_i(v_{\pi(i)}, A_j)$ .

From Ineq. (3) and Ineq. (4), the Inequality (2) is proved and this lemma holds. ◀

In the following we establish the approximation guarantee of our algorithm.

► **Lemma 6.**

$$\text{cost}(S) \leq (f + 1) \cdot H_r \cdot \text{OPT},$$

where  $H_r$  is the  $r^{\text{th}}$  harmonic number and  $\text{OPT}$  is the cost of any optimal solution.

**Proof.** By Lemma 5 and the definition of  $\text{cost}(P_i)$ , we have

$$\text{cost}(S) = \sum_{1 \leq i \leq r} \text{cost}(P_i) \leq (f + 1) \cdot \sum_{1 \leq i \leq r} \sum_{0 \leq j < \pi(i)} k_i(A_j) \cdot \hat{z}_{i, A_j}^{(i)}.$$

By Lemma 4, for each  $1 \leq i \leq r$ , we have

$$\sum_{0 \leq j < \pi(i)} k_i(A_j) \cdot \hat{z}_{i, A_j}^{(i)} = \frac{1}{r - i + 1} \cdot \text{Val}(\hat{z}^{(i)}).$$

Since each  $\hat{z}^{(i)}$  is a feasible dual solution for LP-Dual-(S), we have  $\text{Val}(\hat{z}^{(i)}) \leq \text{OPT}$  for all  $1 \leq i \leq r$ , and

$$\text{cost}(S) \leq (f + 1) \cdot \sum_{1 \leq i \leq r} \frac{1}{r - i + 1} \cdot \text{OPT} \leq (f + 1) \cdot H_r \cdot \text{OPT}$$

as claimed. ◀

## 4 Conclusion

We conclude with future directions and open problems. First, considering the lower-bounds of  $H_r$  and  $f$  for this problem, our  $(f \cdot H_r + H_r)$ -approximation ratio has an extra  $H_r$  factor in it. However, it seems unclear how this excess  $H_r$  factor can be dropped.

Although the approaches of [5, 6, 9] can be used to obtain tight  $f$ -approximation for the partial vertex cover problem, it seems difficult to adopt their techniques to our problem. The reason is that, when multiple covering constraints exist, it seems intricate how the key properties of their approaches can be ensured simultaneously for each covering constraint. We believe that this would be an interesting direction to explore.

Second, clarifying the exact lower bound of approximation ratio for this problem is also interesting. For now,  $\max(H_r, f)$  is what we only know.

---

## References

- 1 Reuven Bar-Yehuda. Using Homogeneous Weights for Approximating the Partial Cover Problem. *J. Algorithms*, 39(2):137–144, May 2001.
- 2 Suman K. Bera, Shalmoli Gupta, Amit Kumar, and Sambuddha Roy. Approximation algorithms for the partition vertex cover problem. *Theoretical Computer Science*, 555:2–8, 2014. Special Issue on Algorithms and Computation.

- 3 Nader H. Bshouty and Lynn Burroughs. Massaging a Linear Programming Solution to Give a 2-Approximation For a Generalization of the Vertex Cover Problem. In *In Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science*, pages 298–308. Springer, 1998.
- 4 Julia Chuzhoy. Covering Problems with Hard Capacities. *SIAM J. Comput.*, 36(2):498–515, August 2006. doi:10.1137/S0097539703422479.
- 5 Toshihiro Fujito. On approximation of the submodular set cover problem. *Oper. Res. Lett.*, 25 (4):169–174, November 1999.
- 6 Rajiv Gandhi, Samir Khuller, and Aravind Srinivasan. Approximation Algorithms for Partial Covering Problems. *J. Algorithms*, 53(1):55–84, October 2004.
- 7 Dorit S. Hochbaum. The t-vertex cover problem: Extending the half integrality framework with budget constraints. In *Approximation Algorithms for Combinatorial Optimization*, pages 111–122, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.
- 8 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within  $2 - \epsilon$ . *Journal of Computer and System Sciences*, 74(3):335–349, 2008. Computational Complexity 2003.
- 9 Julián Mestre. A Primal-Dual Approximation Algorithm for Partial Vertex Cover: Making Educated Guesses. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 182–191, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- 10 Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2001.
- 11 L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, December 1982.