

Approximation and Hardness Results for Label Cut and Related Problems

Peng Zhang ^{*} Linqing Tang ^{*} Wenbo Zhao [†] Jin-Yi Cai [‡] Angsheng Li ^{*}

Abstract

We investigate a natural combinatorial optimization problem called the *Label Cut* problem. Given an input graph G with a source s and a sink t , the edges of G are classified into different categories, represented by a set of *labels*. The labels may also have weights. We want to pick a subset of labels of minimum cardinality (or minimum total weight), such that the removal of all edges with these labels disconnects s and t . We give the first non-trivial approximation and hardness results for the Label Cut problem. Firstly, we present an $O(\sqrt{m})$ -approximation algorithm for the Label Cut problem, where m is the number of edges in the input graph. Secondly, we show that it is **NP**-hard to approximate Label Cut within $2^{\log^{1-1/\log \log^c n} n}$ for any constant $c < 1/2$, where n is the input length of the problem. Thirdly, our techniques can be applied to other previously considered optimization problems. In particular we show that the Minimum Label Path problem has the same approximation hardness as that of Label Cut, simultaneously improving and unifying two known hardness results for this problem which were previously the best (but incomparable due to different complexity assumptions).

1 Introduction

In many graph optimization problems, it is natural to associate edges with *labels* (or *colors*) which partition the set of edges into categories. There are several classical optimization problems considered under this model, such as the Minimum Label Spanning Tree problem [3, 6, 14, 19] and the Minimum Label s - t Path problem (Label Path, for short) [4, 14]. In this paper we consider the Minimum Label s - t Cut problem (Label Cut, for short), in which we are asked to pick labels at minimum total cost to disconnect a source s and a sink t by removing edges from the graph whose labels are picked. The Label Cut problem is a natural generalization of the classical Minimum s - t Cut problem, in the sense that every edge in the latter has a unique label. We give the first non-trivial approximation algorithm as well as prove hardness results for the Label Cut problem. This problem is also a generalization of the Minimum Set Cover or Minimum Hitting Set problems. In fact it can be considered as a version of the Minimum Hitting Set problem where the collection of sets to be “hit” is implicitly represented (and could be exponentially many).

This problem is sufficiently natural that it can appear in many contexts. We came across this problem from the work of Jha, Sheyner and Wing [15], and of Sheyner, Haines, Jha, Lippmann,

^{*}State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, P.O. Box 8718, Beijing, 100080, China. Emails: algzhang@gmail.com, linqing@ios.ac.cn, angsheng@ios.ac.cn. Supported by NSFC grants No. 60325206 and No. 60310213. Corresponding authors: Linqing Tang, Peng Zhang.

[†]Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093, USA. Email: w3zhao@ucsd.edu.

[‡]Computer Sciences Department, University of Wisconsin, Madison, WI 53706. USA. Email: jyc@cs.wisc.edu. Supported by NSF CCF-0511679.

and Wing [23, 22] in computer security, in particular on intrusion detection and on generation and analysis of *Attack Graphs*[¶]. In this application, an attack graph G has nodes representing various states, and directed edges representing state transitions and are labeled by possible “atomic attacks”. A pair of special nodes s and t are also given representing the initial state and the success state (for the intruder). To disable an “atomic attack” incurs some cost (a unit or a weighted cost). Then the computational task is to find a subset of “atomic attacks” of minimum cardinality (or of minimum total weight), such that the removal of all edges labeled by these “atomic attacks” disconnects s and t . This is precisely the Minimum Label Cut problem.

Formally, an instance of the Minimum Label Cut problem consists of a graph $G = (V, E)$ (directed or undirected) with one source node $s \in V$ and one sink node $t \in V$, and a label (or color) set $L = \{1, 2, \dots, q\}$. Each edge $e \in E$ has a unique label $l(e) \in L$, but different edges may have the same label. Each label $l \in L$ has a nonnegative weight $w(l)$. A subset $L' \subseteq L$ of labels is called a *label cut* if the removal of all edges of labels in L' disconnects s and t . The problem asks to find a label cut L' such that the total weight of L' is minimized.

1.1 Related Works

In [15] it was observed that Label Cut is **NP**-hard by reducing Minimum Hitting Set to it. There is a simple duality between Minimum Hitting Set and Minimum Set Cover. It is well known that Minimum Set Cover has a greedy polynomial time $(1 + \log_e |U|)$ -approximation algorithm, where $|U|$ is the size of the underlying set. The same algorithm can be translated to Hitting Set by duality. In [15] the authors express the Label Cut problem as an *implicit* Minimum Hitting Set problem, and then translate the Set Cover algorithm to get an approximation algorithm. This implicit Minimum Hitting Set problem is as follows: Let $\mathcal{D} = \{L' \mid L' \text{ is the set of labels appearing on a path from } s \text{ and } t\}$. This is the set to hit. A set of labels is a label cut iff it intersects every $L' \in \mathcal{D}$. The approximation algorithm for Set Cover translates to an approximation algorithm for Label Cut with approximation guarantee of $1 + \ln |\mathcal{D}|$.

However, as observed by Jha, Sheyner and Wing [15], $|\mathcal{D}|$ is typically exponentially larger than the input size (number of edges in G). Since the removal of *all* edges is certainly a feasible solution to Label Cut, the worst-case approximation ratio $1 + \ln |\mathcal{D}|$ is useless. (There is an additional problem. This concerns the need to compute in polynomial time the next “greedy” step in the approximation algorithm for Set Cover. For implicit Minimum Hitting Set, we would need to find the next label which “hits” the most remaining paths from s to t , and it is not known how to do this in general. The authors in [15] consider special cases where this is feasible.)

We mention another problem, dual to Label Cut, called the Label Path problem. Given an edge-labeled graph, the Label Path problem finds an s - t path with minimum total number of labels on the path. Label Path is **NP**-hard by a reduction from 3SAT. Broersma et. al. [4] devised two exact exponential-time algorithms for the Label Path problem, with respective running times of $O(n \cdot \min\{q^d, 2^q\})$ and $O(n^2 q!)$, where d is the s - t distance. Through an approximation preserving reduction from the Red-Blue Set Cover problem [5], Wirth [24, Theorem 2.16] proved that unless **NP** \subseteq **DTIME** $(n^{\text{polylog}(n)})$ Label Path can not be approximated within ratio $O(2^{\log^{1-\epsilon} n})$. Very recently, Hassin, Monnot and Segev [14] gave an $O(\sqrt{n})$ -approximation algorithm for the weighted Label Path problem, and an approximation hardness result $O(\log^k n)$ (for any fixed $k \geq 1$) for the problem assuming **P** \neq **NP**. Notice that the two hardness results for Label Path are incomparable.

Minimum s - t Cut is solvable in polynomial time. On the other hand, a generalization called the Multicut problem, asks for edge removals to disconnect a given set of source-sink pairs at

[¶]We are indebted to Jeannette Wing who gave an overview talk on their interesting work at Tsinghua University.

minimum cost, is **NP**-hard even for trees [13]. The relationship between the Multicut problem and the Shortest Path problem is very similar to that between Label Cut and Label Path. Multicut can be approximated within a factor of $O(\sqrt{n})$ in directed graphs [11], a factor of $O(\log k)$ in undirected graphs [12] (where k is the number of source-sink pairs), and a factor of 2 in trees [13]. It seems that to prove approximation hardness of Multicut is difficult, and the only known hardness results of Multicut are logarithmic in directed graphs [8] and a constant factor in undirected graphs [7, 17].

1.2 Our Results

In this paper, we give the first non-trivial approximation and hardness results for Label Cut. Firstly, we give a polynomial time $O(\sqrt{m})$ -approximation algorithm, where m is the number of edges in the input graph. The idea of the algorithm is that we first make the graph sparse by picking labels at *appropriately inexpensive* cost, and when the graph is sufficiently sparse we directly compute a minimum s - t cut in the resulting graph. The solution to the problem is then the set of labels picked in the first stage and the labels in the minimum s - t cut computed in the second stage. We use ideas and results of [14, 18]. We also extend the algorithm to deal with related cut problems in edge-classified graphs, such as Label Multicut, Label Multiway Cut, and Label k -Cut.

Secondly, on approximation hardness, we show that Label Cut can not be approximated within $2^{\log^{1-1/\log \log^c n} n}$ for any constant $c < 1/2$ unless **P** = **NP**, where n is the input length of the problem. We prove this by a gap-preserving reduction from the Label Cover problem, for which Dinur and Safra [10] proved such a hardness result. The proof is ultimately based on a reduction from the PCP theorem. Our method of the reduction also gives the same approximation hardness for the Minimum Label Path problem as that of Label Cut, thus simultaneously improving and unifying two known hardness results [14, 24] for this problem which were previously the best (but incomparable due to different complexity assumptions as discussed in Section 1.1).

The proof of hardness for Label Cut and Label Path relies on two ingredients. Firstly we show that the approximation hardness $2^{\log^{1-1/\log \log^c n} n}$ (under complexity assumption **P** \neq **NP**) is preserved even for a special type of instances of Label Cover, in which the two parts of an input bipartite graph have the same number of vertices. Secondly, we show that the mapping relation between labels in the Label Cover problem can be represented by some sub-structures in the Label Cut (resp. Label Path) problems, and thus the cut requirement in Label Cut (resp. the connectivity requirement in Label Path) is equivalent to the edge-covering requirement in Label Cover.

2 An $O(\sqrt{m})$ -Approximation Algorithm for Label Cut

Minimum s - t Cut can be solved in P. If a graph G does not have many edges, it also has an s - t cut without many edges. Consequently it also has a small label cut. Our plan is to successively remove edge sets which are relatively large, yet defined by small label sets. When the graph becomes sufficiently sparse we can directly remove the minimum s - t cut and its corresponding label set.

To do this we consider the Budgeted Maximum Coverage problem (BMCP, for short). We are given a ground set U , a collection \mathcal{S} of subsets of U , and a budget B . Each subset $S_i \in \mathcal{S}$ has a nonnegative weight. The problem is to find a sub-collection $\mathcal{S}' \subseteq \mathcal{S}$ such that the total weight of \mathcal{S}' is at most B and the number of elements in U covered by \mathcal{S}' is maximized. This problem can be approximated in polynomial time within a factor of $1 - \frac{1}{e}$ [18].

Now we give a detailed outline of our approximation algorithm for Label Cut. Denote by $w(L') = \sum_{l \in L'} w(l)$ the total weight of labels from a label subset $L' \subseteq L$. Let $m = |E|$ and $L^* \subseteq L$ be an optimal solution to Label Cut. We aim for an $m^{1/2}$ -approximation of the optimal solution,

i.e., we want to output a label cut $L' \subseteq L$ such that $w(L') \leq m^{1/2}w(L^*)$.

We start with an estimate Δ on the optimum $\text{OPT} = w(L^*)$ of the Label Cut instance, such that $\text{OPT} \leq \Delta < 2 \cdot \text{OPT}$. We try a sequence of Δ as guesses (see Theorem 2.1). Suppose we have a correct Δ . Starting with $H_0 = G$, we will perform a sequence of edge removal operations. This creates graphs $H_0, H_1, \dots, H_{m'}$, where H_{i+1} is obtained from H_i by the removal of some subset of edges. Each step i also produces a subset of labels L_i . The final output is the union of all these L_i .

At step i , we are given H_i . Keeping only edges in H_i with label weight $> \Delta$ (and delete others) defines a subgraph H'_i . Suppose that G is undirected, then we find connected components of H'_i . For directed G we find strongly connected components of H'_i . Then we merge the nodes from H_i within each (strongly) connected component defined above to obtain H_i^* . Edges within a component disappear. Other edges remain; in particular between the merged nodes there may be multiple edges in H_i^* . All edges retain their labels, and all label weights in H_i^* are $\leq \Delta$.

If $\text{OPT} \leq \Delta$, then vertices s and t remain distinct in H_i^* ; otherwise an s - t path using only edges of weight $> \Delta$ exists which has no intersection with the cut of total weight $\text{OPT} \leq \Delta$.

Now we use a Maxflow-Mincut algorithm to find the minimum cut size $c_i^* = |\text{mincut}_{H_i^*}(s, t)|$ of H_i^* . Here each edge in H_i^* counts as one, and multiple edges count with multiplicity. If $c_i^* \leq m^{1/2}$, then we find a minimum cut of H_i^* , and use all labels on this cut. This is our L_i , and with this we terminate with $m' = i$. Note that all edges of H_i of labels in L_i also form a cut in H_i . Now suppose $c_i^* > m^{1/2}$. We consider the following instance of BMCP where the ground set is the set of all edges $E(H_i)$ of the graph H_i , and the collection of subsets \mathcal{S} in BMCP consists of all $S_l = \{e \in E(H_i) \mid l(e) = l\}$ for $l \in L$, and where the weight of S_l is $w(l)$. We apply the polynomial time approximation algorithm [18] on this instance for budget Δ to get a sub-collection $\mathcal{S}' \subseteq \mathcal{S}$. Then we use all labels coming from \mathcal{S}' as our L_i in this case. We claim that the edge set $\bigcup_{l \in L_i} S_l$ corresponding to L_i has cardinality at least $\frac{1}{2}m^{1/2}$ (see Lemma 2.1).

This gives the approximation algorithm for Label Cut, as presented by Algorithm \mathcal{A} below. For any subset $E' \subseteq E$, let $L(E')$ denote the set of all labels in E' .

Algorithm \mathcal{A}

1. **let** $i \leftarrow 0$, $H_0 \leftarrow G$. Define H'_0 and H_0^* as described above.
2. **while** $|\text{mincut}_{H_i^*}(s, t)| > m^{1/2}$
 - a. Construct a BMCP instance I' as follows: the ground set $U = E(H_i)$, the collection \mathcal{S} consists of $S_l = \{e \in E(H_i) \mid l(e) = l\}$ with weight $w(l)$ for every $l \in L$, and the budget $B = \Delta$.
 - b. Approximately solve I' and let E_i be the set of covered edges.
 - c. **let** $L_i \leftarrow L(E_i)$, $H_{i+1} \leftarrow H_i \setminus E_i$. Define H'_{i+1} and H_{i+1}^* as described above.
 - d. **let** $i \leftarrow i + 1$.
3. **endwhile**
4. **let** $L_i \leftarrow L(\text{mincut}_{H_i^*}(s, t))$.
5. **return** $L' = \bigcup_i L_i$.

Let m' be the number of iterations of step 2, i.e., m' equals the value of i when step 2 terminates.

Lemma 2.1 *For Algorithm \mathcal{A} , if $\text{OPT} \leq \Delta$, then $m' < 2m^{1/2}$.*

Proof: Suppose that an optimal solution to the Label Cut instance I is $L^* \subseteq L$ with total weight $\text{OPT} = w(L^*) \leq \Delta$, and that $E^* \subseteq E(G)$ is the set of edges corresponding to the label subset L^* . In each iteration of step 2, $E^* \cap E(H_i^*)$ is a feasible solution to the BMCP instance I' , since the total weight of labels in $E^* \cap E(H_i^*)$ is at most $w(L^*)$,

$$w(L(E^* \cap E(H_i^*))) \leq w(L^*) \leq \Delta.$$

A crucial observation is that $E^* \cap E(H_i^*) \subseteq E^* \cap E(H_i)$ is also an s - t cut in H_i^* . This is because any s - t path in H_i^* can be extended to an s - t path in H_i by appending only edges of label weight $> \Delta$. The path in H_i must intersect $E^* \cap E(H_i)$, but all edges in E^* have label weight $\leq \Delta$, therefore the s - t path in H_i^* must intersect $E^* \cap E(H_i^*)$.

So, we know that

$$|E^* \cap E(H_i^*)| \geq |\text{mincut}_{H_i^*}(s, t)| > m^{1/2}.$$

By the work of [18], we know that E_i from 2.b. covers at least $(1 - \frac{1}{e})\text{OPT}_{\text{BMCP}}(I')$ edges, where $\text{OPT}_{\text{BMCP}}(I')$ is the optimum of instance I' . So in each iteration of step 2 at least

$$(1 - \frac{1}{e})\text{OPT}_{\text{BMCP}}(I') \geq (1 - \frac{1}{e})|E^* \cap E(H_i^*)| > \frac{1}{2}m^{1/2}$$

edges are removed from H_i . Since there are at most m edges in total in graph H_0 , it follows that the number of iterations $m' < 2m^{1/2}$. ■

Theorem 2.1 *The Label Cut problem can be approximated within a factor of $O(m^{1/2})$ in polynomial time, where m is the number of edges in the input graph of the problem.*

Proof: Let w_{\min} denote the minimum (nonzero) label weight. Then we know $w_{\min} \leq \text{OPT} \leq w(L)$. We try Algorithm \mathcal{A} with $\Delta = 2^k \cdot w_{\min}$ for every $k = 0, \dots, \lceil \log_2 \frac{w(L)}{w_{\min}} \rceil$, where Δ is used as an estimate of OPT . Let k be such that

$$2^{k-1} \cdot w_{\min} < \text{OPT} \leq 2^k \cdot w_{\min}.$$

Then for the guess $\Delta = 2^k \cdot w_{\min}$, we have

$$\text{OPT} \leq \Delta < 2 \cdot \text{OPT}.$$

With this Δ , we know that the number of iterations $m' < 2m^{1/2}$ by Lemma 2.1. Moreover, when step 2 terminates, $|\text{mincut}_{H_{m'}^*}(s, t)|$ is at most $m^{1/2}$, implying that

$$w(L_{m'}) \leq |\text{mincut}_{H_{m'}^*}(s, t)| \cdot \max\{w(l) : l \in L(E(H_{m'}^*))\} \leq m^{1/2} \Delta.$$

Thus the total weight of the output L' is at most

$$m' \Delta + w(L_{m'}) \leq 2m^{1/2} \cdot \Delta + m^{1/2} \cdot \Delta = O(m^{1/2}) \cdot \text{OPT}.$$

We note that, from the label set $L_{m'}$ derived from $\text{mincut}_{H_{m'}^*}(s, t)$, all edges of labels from $L_{m'}$ in $H_{m'}$ do constitute an s - t cut in $H_{m'}$, and therefore the output L' is an s - t cut in G .

Finally, for every k , we run Algorithm \mathcal{A} . (We never run beyond $m' \geq 2m^{1/2}$, and verify each time s and t remain distinct in the merging operations; if these are unsatisfied then we go to the next Δ .) Among all tries with different Δ we pick the label subset with minimum total weight as our final solution. ■

Our method here can be extended to deal with other cut problems in the edge-classified graph model, such as Label Multicut, Label Multiway Cut and Label k -Cut e.t.c. (we call them generally Label γ -Cut problems below). Algorithm \mathcal{A} can be generalized to deal with each of them by replacing the minimum s - t cut computation in step 2 with a known approximation algorithm for the corresponding (**NP**-hard) unlabeled problem. If the approximation ratio of the algorithm for the unlabeled problem is ρ , we can show that the approximation ratio of the modified algorithm \mathcal{A} is $O((m\rho)^{1/2})$. For the problems Multicut, Multiway Cut and k -Cut in undirected graphs, we know that ρ are $O(\log k)$ [12], 1.3438 [16], and 2 [21], respectively. The proof of Theorem 2.2 is omitted here due to space limit and is given in the Appendix.

Theorem 2.2 *The Label γ -Cut problem can be approximated within factor of $O((m\rho)^{1/2})$ in polynomial time, where m is the number of edges in the input graph and ρ is the approximation ratio of the existing approximation algorithm for γ -Cut.*

3 Approximation Hardness for Label Cut

In this section we prove that it is **NP**-hard to approximate the Minimum Label Cut problem to within $2^{\log^{1-o(1)} n}$. We do this by reducing the Minimum Label Cover problem to Label Cut. The Minimum Label Cover problem (Label Cover, for short) was implicit in [20] and first formally defined in [1]. Label Cover is one of the six canonical intractable problems in terms of approximation introduced by Arora and Lund [2]. The current best known approximation hardness result for Label Cover is due to Dinur and Safra [10], which states that the problem is **NP**-hard to approximate within $2^{\log^{1-1/\log \log^c n} n}$ for any constant $c < 1/2$.

In the Minimum Label Cover problem, we are given a bipartite graph $G = (U, V, E)$ where $E \subseteq U \times V$, two sets of possible labels B_1 and B_2 for vertices in U and V respectively, and a relation $\Pi \subseteq E \times B_1 \times B_2$ that consists of admissible pairs of labels for each edge $e \in E$. A *labeling* is a pair of functions $(\mathcal{P}_1, \mathcal{P}_2)$ that assigns a non-empty set of labels to every vertex in $U \cup V$, where $\mathcal{P}_1: U \rightarrow 2^{B_1}$ and $\mathcal{P}_2: V \rightarrow 2^{B_2}$. It is said to *cover* an edge $e = (u, v)$ (where $u \in U$ and $v \in V$) if for every label $b \in \mathcal{P}_2(v)$ there is some label $a \in \mathcal{P}_1(u)$ such that $(e, a, b) \in \Pi$. The l_1 -cost of a labeling is the l_1 norm of the vector $(|\mathcal{P}_1(u_1)|, \dots, |\mathcal{P}_1(u_{|U|})|)$, that is, the total number $\sum_{u \in U} |\mathcal{P}_1(u)|$ of labels assigned to vertices in U by function \mathcal{P}_1 counted with multiplicities. A *total-cover* (a solution) of G is a labeling that covers all the edges in G . The problem is to find a total-cover with minimal l_1 -cost. We always implicitly assume that the only instances we shall consider for Minimum Label Cover are such that a total-cover exists (see [2], p403, line –8).

Dinur and Safra [10] proved the approximation hardness result $2^{\log^{1-o(1)} n}$ for Label Cover via a reduction from the following version of the PCP Theorem.

Theorem 3.1 ([9]) *Let $\Psi = \{\psi_1, \dots, \psi_n\}$ be a system of local-tests over variables $X = \{x_1, \dots, x_{n'}\}$ such that each local-test depends on $D = \log \log^c n$ variables (for any constant $c < 1/2$), and each variable ranges over a field \mathcal{F} where $|\mathcal{F}| = O(2^{\log^{1-1/O(D)} n})$. It is **NP**-hard to distinguish between the following two cases:*

Yes: There is an assignment to the variables such that all ψ_1, \dots, ψ_n are satisfied.

No: No assignment can satisfy more than $\frac{2}{|\mathcal{F}|}$ fraction of the ψ_i 's.

Define $g_c(n) = 2^{\log^{1-1/\log \log^c n} n}$. Dinur and Safra proved that

Theorem 3.2 ([10]) *Let Ψ be a local-test system defined in Theorem 3.1. Then an instance $I = ((U, V, E), B_1, B_2, \Pi)$ of Label Cover can be constructed from Ψ in polynomial time such that for any constant $c < 1/2$,*

(Soundness) Ψ is satisfied $\implies \text{OPT}(I) = |U|$, and

(Completeness) No assignment can satisfy more than $\frac{2}{|\mathcal{F}|}$ fraction of $\Psi \implies \text{OPT}(I) > g|U|$,

where $g = g_c(|I|)$ and $\text{OPT}(I)$ denotes the optimum of Label Cover on the instance I .

We shall prove the approximation hardness of Label Cut via a gap-preserving reduction from Label Cover. For succinctness, denote by $E(v) \subseteq E$ all edges that are incident to v , and by

$N(v) \subseteq U$ all endpoints of edges in $E(v)$ that lie in U , i.e., $N(v)$ is the neighborhood of vertex v . Without loss of generality we assume $|N(v)| \geq 1$ for all $v \in V$. Similarly every u is in some neighborhood $N(v)$, i.e., $U = \bigcup_{v \in V} N(v)$. Let us begin with a simple lemma about labels in B_2 .

Lemma 3.1 *For any vertex $v \in V$ and label $b \in B_2$, if there is a vertex $u \in N(v)$ such that for all $a \in B_1$, $((u, v), a, b) \notin \Pi$, then vertex v can not be assigned with label b by any solution.*

Proof: The proof is straightforward. If vertex v is associated with label b by some solution $(\mathcal{P}_1, \mathcal{P}_2)$, since $(\mathcal{P}_1, \mathcal{P}_2)$ must be a total-cover, we know that every vertex $u \in N(v)$ has a mapping $((u, v), a, b) \in \Pi$ for some $a \in B_1$ by the cover condition, contradicting the condition in the lemma.

■

If a label $b \in B_2$ satisfies the condition in Lemma 3.1, we say this label b is *excluded* (by the vertex v). A label b is *non-excluded* (by the vertex v) iff for all $u \in N(v)$ there exists some label $a \in B_1$, such that $((u, v), a, b) \in \Pi$. Since we only consider instances where a total-cover exists, we must have for every v , it has a non-empty set of non-excluded labels.

The following lemma gives a transformation on the instances of Label Cover, which is useful for the gap-preserving reduction from Label Cover to Label Cut.

Lemma 3.2 *Without loss of generality, we may assume that instances of Label Cover satisfy $|U| = |V|$.*

Proof: Take any instance $I = ((U, V, E), B_1, B_2, \Pi)$ of Label Cover. We apply a transformation on I , which was originally introduced in [2, p. 413] (for the maximum version of Label Cover), to obtain an instance $I' = ((U', V', E'), B_1, B_2, \Pi')$ of Label Cover. U' consists of $|V|$ copies of U , and V' consists of $|U|$ copies of V . It is obvious that $|U'| = |V'|$ for the new instance I' . The edge set E' in I' consists of copies of E between each new pair of copies of U and V . The relation Π' is constructed by duplicating Π accordingly.

If I has a labeling $(\mathcal{P}_1, \mathcal{P}_2)$ with l_1 -cost equal to $|U|$, then I' has a labeling with l_1 -cost $|U'|$, since we can get a labeling for I' by duplicating $(\mathcal{P}_1, \mathcal{P}_2)$. Conversely, if every labeling for I has l_1 -cost $> g|U|$, then every labeling for I' has l_1 -cost $> g|U'|$. Suppose not. If I' has a labeling $(\mathcal{P}'_1, \mathcal{P}'_2)$ with l_1 -cost $\leq g|U'|$, then we can get a labeling for each copy of U and V by restricting $(\mathcal{P}'_1, \mathcal{P}'_2)$ on that copy. Picking the copy of U with the minimum l_1 -cost, together with any copy of V gives us a solution to I with l_1 -cost $\leq g|U|$. That is, for instances I and I' we have

$$\begin{aligned} \text{OPT}(I) = |U| &\implies \text{OPT}(I') = |U'|, \\ \text{OPT}(I) > g|U| &\implies \text{OPT}(I') > g|U'|. \end{aligned}$$

By adjusting the arbitrary constant $c < 1/2$, $g = g_c(n)$ can absorb any polynomial increase in n . The Lemma follows. ■

Lemma 3.3 *There is a polynomial-time gap-preserving reduction τ from Label Cover to Label Cut that transforms any instance $I = ((U, V, E), B_1, B_2, \Pi)$ of Label Cover to an instance $I' = ((V', E'), s, t, L)$ of Label Cut, such that,*

$$\begin{aligned} \text{OPT}_{\text{COVER}}(I) = |U| &\implies \text{OPT}_{\text{CUT}}(I') \leq |U| + |V|, \\ \text{OPT}_{\text{COVER}}(I) > g|U| &\implies \text{OPT}_{\text{CUT}}(I') > g|U| + |V|, \end{aligned}$$

where $\text{OPT}_{\text{COVER}}()$ (resp. $\text{OPT}_{\text{CUT}}()$) returns the optimum of any instance of Label Cover (resp. Label Cut).

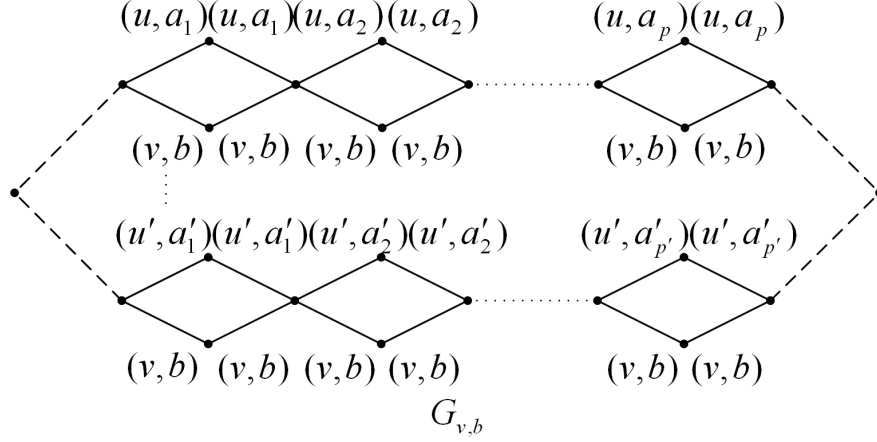


Figure 1: Gadget $G_{v,b}$ for vertex $v \in V$ and label $b \in B_2$

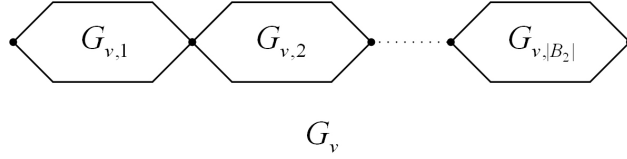


Figure 2: Gadget G_v for vertex $v \in V$

Proof: Given an instance $I = ((U, V, E), B_1, B_2, \Pi)$ of Label Cover, the instance $I' = (G', s, t, L)$ of Label Cut is constructed as follows. For every vertex v and non-excluded label b , there is a gadget $G_{v,b}$, as shown in Figure 1. $G_{v,b}$ consists of a series of parallel chains, each of which corresponds to an edge $(u, v) \in E(v)$. For such an edge (u, v) , suppose that the set of labels mapped to $b \in B_2$ by Π is $\{a_1, a_2, \dots, a_p\} \subseteq B_1$, i.e., this is the subset of labels $a \in B_1$ such that $((u, v), a, b) \in \Pi$. (Since b is a non-excluded label by v , for every neighbor $u \in N(v)$, this number $p \geq 1$. This p depends on v, b and u . Note also that $|N(v)| \geq 1$ as noted earlier.) Then in $G_{v,b}$ the chain corresponding to edge (u, v) contains p consecutive diamonds, with the i th diamond ($1 \leq i \leq p$) having label (u, a_i) for its two top edges, and label (v, b) for its two bottom edges (see Figure 1). All other chains are constructed in the same manner, one chain for each $u \in N(v)$. The dashed lines (not belonging to the graph) at the two ends of $G_{v,b}$ mean that the ends of each chain are merged.

Next, a gadget G_v for every vertex $v \in V$ is constructed by linking several gadgets $G_{v,b}$ together, where each gadget $G_{v,b}$ corresponds to a non-excluded label b by v (see Figure 2.) Note that only $G_{v,b}$ for non-excluded labels b for the vertex v appear in G_v . Its number is at least one but $\leq |B_2|$. (In Figure 2 for notational simplicity we assume that for the given example of G_v , all $|B_2|$ labels are non-excluded labels for v and are denoted as $1, 2, \dots, |B_2|$.) Finally, graph $G' = (V', E')$ consists of a series of parallel gadgets G_v for every vertex $v \in V$. All the heads of gadgets G_v are merged into one vertex, that is, the source vertex s , meanwhile all the tails of gadgets G_v are merged into one vertex, that is, the sink vertex t . The label set L contains all labels of the forms (u, a) and (v, b) used in the construction. Every label in L has unit weight. Since each diamond of G' corresponds to a tuple of Π , we know that I' can be constructed in polynomial time.

Suppose $(\mathcal{P}_1^*, \mathcal{P}_2^*)$ is an optimal solution to the Label Cover instance I whose l_1 -cost is $|U|$.

For any $v \in V$, since $\mathcal{P}_2^*(v)$ is non-empty, there exists some $b \in \mathcal{P}_2^*(v)$. This b is obviously a non-excluded label, by the definition of a solution in Label Cover. So (v, b) is a label in L . Now we pick the label (v, b) . Next, for all $u \in U$, pick all labels (u, a) for $a \in \mathcal{P}_1^*(u)$. Since the l_1 -cost of $(\mathcal{P}_1^*, \mathcal{P}_2^*)$ is $|U|$, this counts $|\mathcal{P}_1^*(u)|$ for different $u \in U$ with multiplicity, it follows that we have picked a total of $|U| + |V|$ labels from L . We claim that removing all edges with the picked labels disconnects s and t in G' . This is equivalent to disconnecting every G_v . Fix any $v \in V$. We consider the picked label (v, b) , and show that the gadget $G_{v,b}$ is disconnected. This in turn is equivalent to disconnecting every chain in $G_{v,b}$, one chain for each $u \in N(v)$. Consider such a chain defined by a particular $u \in N(v)$. Since all edges labeled (v, b) have been removed, we just need to prove that at least one diamond has been completely removed. Suppose there are p diamonds, where the upper path is labeled by $(u, a_1), \dots, (u, a_p)$, and these a_i are those satisfying $((u, v), a_i, b) \in \Pi$. (see Figure 1). Since $(\mathcal{P}_1^*, \mathcal{P}_2^*)$ is a total-cover, for some $1 \leq i \leq p$, $a_i \in \mathcal{P}_1^*(u)$. But we did pick labels (u, a) , for all $a \in \mathcal{P}_1^*(u)$. It follows that $\text{OPT}_{\text{CUT}}(I') \leq |U| + |V|$.

Now we prove the second part of the lemma by its contrapositive. Suppose the optimum of the Label Cut instance I' is $\leq g|U| + |V|$, and an optimal solution is $L^* \subseteq L$. For clarity let us call a label in L of the form (u, a) a u -label, and a label of the form (v, b) a v -label. Denote by n_{ul} (resp. n_{vl}) the number of u -labels (resp. v -labels) in L^* . Since G' consists of $|V|$ subgraphs G_v in parallel, the removal of edges with labels in L^* must disconnect each G_v . By construction of G_v as a chain of $G_{v,b}$, it must disconnect some $G_{v,b}$, and therefore must pick at least one v -label (v, b) for every $v \in V$. This defines our assignment \mathcal{P}_2 on V , and $n_{\text{vl}} \geq |V|$. Next we define \mathcal{P}_1 on U as follows: $\mathcal{P}_1(u) = \{a \mid (u, a) \text{ is a picked } u\text{-label in } L^*\}$. Since $n_{\text{vl}} \geq |V|$, it follows that $n_{\text{ul}} \leq g|U|$. Since L^* disconnects $G_{v,b}$, for every $u \in N(v)$, L^* must include at least one u -label (u, a) such that $((u, v), a, b) \in \Pi$. In particular $\mathcal{P}_1(u)$ is non-empty. Since $U = \bigcup_{v \in V} N(v)$, this is true for all $u \in U$. This also shows that $(\mathcal{P}_1, \mathcal{P}_2)$ is a solution to the Label Cover instance I . Since $n_{\text{ul}} \leq g|U|$, we know that the l_1 -cost of $(\mathcal{P}_1, \mathcal{P}_2)$ is $\leq g|U|$. This completes the proof of the contrapositive. ■

Theorem 3.3 *For any constant $c < 1/2$, the Label Cut problem can not be approximated within $2^{\log^{1-1/\log \log^c n} n}$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$, where n is the input length of the problem.*

Proof: Given any Label Cover instance I , first we apply the transformation presented in Lemma 3.2 to get an instance $\hat{I} = ((U, V, E), B_1, B_2, \Pi)$ of Label Cover such that $|U| = |V|$. Denote by \overline{m} the length of the Label Cover instance I , and by \overline{n} the length of the Label Cut instance $I' = (G', s, t, L)$ generated by the reduction τ in Lemma 3.3 on instance \hat{I} . By Lemma 3.3, the gap between the two cases in the lemma is at least $g/2$, where $g = g_c(\overline{m})$. By Theorems 3.1 and 3.2, it is \mathbf{NP} -hard to approximate Label Cover within ratio $g_c(\overline{m})$ for any constant $c < 1/2$. Since essentially the edges (together with their labels) in G' reflect the relation Π , we know that $\overline{n} \leq \overline{m}^k$ for some positive constant k , i.e., $\overline{m} \geq \overline{n}^{1/k}$. So we have that it is \mathbf{NP} -hard to approximate Label Cut (instances generated by the reduction τ of Lemma 3.3) within ratio $g_c(\overline{m})/2$ for any constant $c < 1/2$. This implies the approximation hardness $g_{c'}(\overline{n})$ of Label Cut for any constant $c' < 1/2$ under the complexity assumption $\mathbf{P} \neq \mathbf{NP}$ (the coefficient $1/2$ and exponential $1/k$ can be absorbed in the exponent of the hardness $g_{c'}(\overline{n})$). ■

4 Approximation Hardness for Label Path

The Minimum Label Path problem has been well studied before [24, 14]. The instance of the problem is the same as that of Label Cut. The goal of the problem is to find an s - t path P with minimum total weight of labels that appear on P . We show that the approximation hardness of Label Path is $2^{\log^{1-1/\log \log^c n} n}$ for any constant $c < 1/2$ under the complexity assumption $\mathbf{P} \neq \mathbf{NP}$.

This simultaneously improves and unifies two known hardness results for this problem which were previously the best (but incomparable due to different complexity assumptions). The previous results were (a) it is **NP**-hard to approximate Label Path within factor $\log^k n$ for any fixed $k \geq 1$ [14], and (b) Label Path can not be approximated within $O(2^{\log^{1-\epsilon} n})$ for any constant $\epsilon > 0$ unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\text{polylog}(n)})$ [24]. In both of these two hardness results, n denotes the number of vertices in the input graph.

We prove the approximation hardness for Label Path via a gap-preserving reduction from Label Cover that is similar to that from Label Cover to Label Cut given in Section 3. The reduction uses two types of gadgets $G'_{v,b}$ and G'_v , which are transformations of the two gadgets $G_{v,b}$ and G_v (see Figures 1 and 2) used for Label Cut. According to the construction of the gadgets $G'_{v,b}$ and G'_v and the designation method of labels to edges in the gadgets, the connectivity requirement in the Label Path instance generated by the reduction is equivalent to the edge-covering requirement of the given Label Cover instance. Again, the proof (including the reduction and the gadgets $G'_{v,b}$ and G'_v) of Theorem 4.1 is omitted here and is given in the Appendix.

Theorem 4.1 *For any constant $c < 1/2$, the Label Path problem can not be approximated within $2^{\log^{1-1/\log \log^c n} n}$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$, where n is the input length of the problem.*

5 Conclusions

We introduce and study the Minimum Label Cut problem in the edge-classified graph model, which is the labeled (or colored) version of the classical minimum s - t cut problem. We present a polynomial time approximation algorithm for Label Cut with performance ratio $O(\sqrt{m})$, where m is the number of edges in the input graph. We show that unless $\mathbf{P} = \mathbf{NP}$ Label Cut can not be approximated in polynomial time within a factor of $2^{\log^{1-1/\log \log^c n} n}$ for any constant $c < 1/2$, where n is the length of the instance of Label Cut, via a gap-preserving reduction from the Minimum Label Cover problem. Our method of the reduction also gives a stronger approximation hardness result for the Minimum Label Path problem, improving and unifying the best previously known hardness results [14, 24] for this problem. It remains an interesting problem to improve the approximation algorithm as well as to improve approximation hardness results (say, n^ϵ -hardness) for Label Cut and the related problems.

References

- [1] S. Arora, L. Babai, J. Stern, Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In Proceedings of the 34th IEEE Symposium on Foundations of Computer Science, pages 724–733, 1993. Journal version: Journal of Computer and System Sciences. **54**(2):317–331, 1997.
- [2] S. Arora, C. Lund. Hardness of Approximation. In: D. Hochbaum (editor), Approximation Algorithms for NP-hard Problems, PWS Publishing Company, 399–446, 1997.
- [3] H. Broersma, X. Li. Spanning trees with many or few colors in edge-colored graphs. *Discussiones Mathematicae Graph Theory*, **17**(2):259–269, 1997.
- [4] H. Broersma, X. Li, G. Woeginger, S. Zhang. Paths and cycles in colored graphs. *Australasian Journal on Combinatorics*, **31**:299–311, 2005.

- [5] R. Carr, S. Doddi, G. Konjevod, M. Marathe. On the red-blue set cover problem. In Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 345–353, 2000.
- [6] R.-S. Chang, S.-J. Leu. The minimum labeling spanning trees. *Information Processing Letters*, **63**(5):277–282, 1997.
- [7] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani. D. Sivakumar. On the hardness of approximating Multicut and Sparsest-Cut. In Proceedings of IEEE Conference on Computational Complexity, pages 144–153, 2005.
- [8] J. Chuzhoy, S. Khanna. Hardness of cut problems in directed graphs. In Proceedings of the 38th Annual ACM Symposium on Theory of Computing, pages 527–536, 2006.
- [9] I. Dinur, E. Fischer, G. Kindler, R. Raz, S. Safra. PCP characterizations of NP: towards a polynomially-small error-probability. In Proceedings of the 31st Annual ACM Symposium on Theory of Computing, pages 29–40, 1999.
- [10] I. Dinur, S. Safra. On the hardness of approximating Label Cover. *Information Processing Letters* **89**(5):247–254, 2004.
- [11] A. Gupta. Improved results for directed multicut. In Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 454–455, 2003.
- [12] N. Garg, V. Vazirani, M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM Journal on Computing*, **25**:235–251, 1996.
- [13] N. Garg, V. Vazirani, M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, **18**:3–20, 1997.
- [14] R. Hassin, J. Monnot, D. Segev. Approximation algorithms and hardness results for labeled connectivity problems. *Journal of Combinatorial Optimization*, **14**(4):437–453, 2007.
- [15] S. Jha, O. Sheyner, J.M. Wing. Two formal analyses of attack graphs. In Proceedings of the 15th IEEE Computer Security Foundations Workshop, pages 49–63, Nova Scotia, Canada, June 2002.
- [16] D. Karger, P. Klein, C. Stein, M. Thorup, N. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. In Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pages 668–678, 1999.
- [17] S. Khot, N. Vishnoi. The unique games conjecture, integrality gap for cut problems and the embeddability of negative type metrics into l_1 . In Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, pages 53–62, 2005.
- [18] S. Khuller, A. Moss, J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, **70**(1):39–45, 1999.
- [19] S. Krumke, H. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, **66**(2):81–85, 1998.
- [20] C. Lund, M. Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, **41**(5):960–981, 1994.

- [21] H. Saran, V. Vazirani. Finding k -cuts within twice the optimal. SIAM Journal on Computing, **24**:101–108, 1995.
- [22] O. Sheyner and J.M. Wing, Tools for Generating and Analyzing Attack Graphs. Proceedings of Workshop on Formal Methods for Components and Objects, 2004, pp. 344–371.
- [23] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing, Automated Generation and Analysis of Attack Graphs, Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, May 2002.
- [24] H. Wirth. Multicriteria Approximation of Network Design and Network Upgrade Problems. PhD thesis, Department of Computer Science, Würzburg University, 2001.

Appendix

Proof of Theorem 2.2

Proof: We modify Algorithm \mathcal{A} such that it is suitable to deal with Label γ -Cut by replacing the minimum s - t cut computation in step 2 by an existing ρ -approximation algorithm (call it algorithm \mathcal{B}), for the corresponding unlabeled γ -Cut problem. Denote by c_i the capacity (i.e., the number of edges in our setting) of the γ -cut returned by algorithm \mathcal{B} on graph H_i^* . Step 2 of Algorithm \mathcal{A} is replaced by

2. **while** $c_i = |\mathcal{B}(H_i^*)| > (m\rho)^{1/2}$.

Denote by c_i^* the optimum of γ -Cut on the instance H_i^* . Since algorithm \mathcal{B} returns a ρ -approximation on H_i^* , we know that $\rho \cdot c_i^* \geq c_i > (m\rho)^{1/2}$, i.e., we have

$$c_i^* > (m/\rho)^{1/2}.$$

Recall that $E^* \subseteq E(G)$ is the set of edges corresponding to an optimum solution $L^* \subseteq L$ of Label γ -Cut. Since $E^* \cap E(H_i^*)$ is a γ -cut for H_i^* , we have $|E^* \cap E(H_i^*)| \geq c_i^* > (m/\rho)^{1/2}$. Moreover, since $E^* \cap E(H_i^*)$ is also a feasible solution to the BMCP instance I' , for the edge subset E_i we have

$$|E_i| \geq (1 - \frac{1}{e})\text{OPT}_{\text{BMCP}}(I') \geq (1 - \frac{1}{e})|E^* \cap E(H_i^*)| > \frac{1}{2}(m/\rho)^{1/2}.$$

So we know that the number of iterations of step 2 of the modified Algorithm \mathcal{A} is at most

$$\frac{m}{\frac{1}{2}(m/\rho)^{1/2}} = 2(m\rho)^{1/2}.$$

By the guess technique used in Theorem 2.1, there is a try of the modified Algorithm \mathcal{A} whose output satisfies that

$$w(\bigcup_i L_i) \leq 2(m\rho)^{1/2} \cdot \Delta + (m\rho)^{1/2} \cdot \Delta = O((m\rho)^{1/2}) \cdot \text{OPT},$$

completing the proof of the theorem. ■

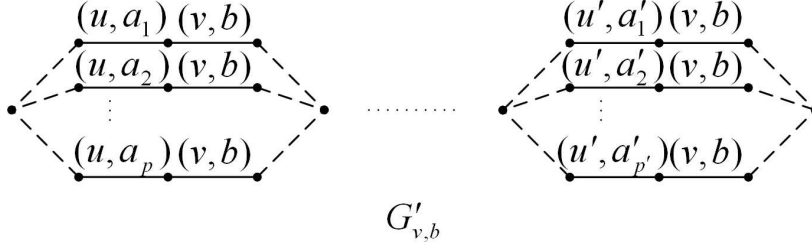


Figure 3: Gadget $G'_{v,b}$ for vertex $v \in V$ and label $b \in B_2$

Proof of Theorem 4.1

Proof: We prove the theorem via a gap-preserving reduction τ' which transforms any Label Cover instance $I = ((U, V, E), B_1, B_2, \Pi)$ to a Label Path instance $I' = ((V', E'), s, t, L)$ in polynomial time. First we give the reduction τ' .

Given an instance $I = ((U, V, E), B_1, B_2, \Pi)$ of Label Cover, the instance $I' = (G', s, t, L)$ of Label Path is constructed as follows. For every vertex $v \in V$ and non-excluded label b , there is a gadget $G'_{v,b}$ (see Figure 3). $G'_{v,b}$ consists of a series of consecutive subgraphs, with each of them corresponding to an edge $(u, v) \in E(v)$. For such an edge $(u, v) \in E(v)$, suppose that the set of labels mapped to $b \in B_2$ by Π is $\{a_1, a_2, \dots, a_p\} \subseteq B_1$. (Since b is a non-excluded label by v , we know that for every neighbor $u \in N(v)$, this number p (depends on v, b and u) is at least 1. Note also that $|N(v)| \geq 1$ as discussed in Section 3.) Then the subgraph corresponding to edge (u, v) contains p parallel paths of length 2 (for clarity, we call each such path a *rung*), with the i th rung ($1 \leq i \leq p$) having labels (u, a_i) and (v, b) for its two consecutive edges (see Figure 3). All other subgraphs are constructed in the same manner, one subgraph for each $u \in N(v)$. The dashed lines (not belonging to the graph) at the two ends of the paths mean that for each subgraph the ends of the paths are merged.

Next, a gadget G'_v for every vertex $v \in V$ is constructed by stacking up several gadgets $G'_{v,b}$ with each of them corresponding to a non-excluded label b by v (see Figure 4). Note that only $G'_{v,b}$ for non-excluded labels b for the vertex v appear in G'_v . Its number is at least one but $\leq |B_2|$. (For notational simplicity we assume that for the given example of G'_v in Figure 4 all $|B_2|$ labels are non-excluded labels for v .) Finally, the graph $G' = (V', E')$ consists of a series of consecutive gadgets G'_v (in arbitrary order) for every vertex $v \in V$. The left end of the first gadget G'_v in G' denotes the source vertex s , meanwhile the right end of the last gadget G'_v in G' denotes the sink vertex t . The label set L contains all u -labels and v -labels used in the construction. Every label in L has unit weight. Since each rung of G' corresponds to a tuple of Π , I' can be constructed in polynomial time.

For the Label Cover instance I and the Label Path instance I' generated by the reduction τ' , we claim that

$$\begin{aligned} \text{OPT}_{\text{COVER}}(I) = |U| &\implies \text{OPT}_{\text{PATH}}(I') \leq |U| + |V|, \\ \text{OPT}_{\text{COVER}}(I) > g|U| &\implies \text{OPT}_{\text{PATH}}(I') > g|U| + |V|, \end{aligned}$$

where $\text{OPT}_{\text{PATH}}()$ returns the optimum of any instance of Label Path.

Suppose that $(\mathcal{P}_1^*, \mathcal{P}_2^*)$ is an optimal solution to the Label Cover instance I with l_1 -cost $|U|$. We define a label subset L' for the Label Path instance I' as follows. For every $v \in V$, since $\mathcal{P}_2^*(v) \neq \emptyset$, we pick some $b \in \mathcal{P}_2^*(v)$ to include (v, b) in L' . These are all the v -labels in L' . For u -labels we

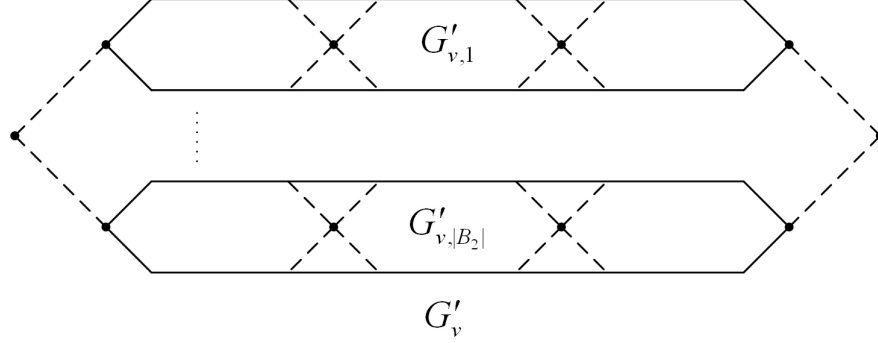


Figure 4: Gadget G'_v for vertex $v \in V$

include all (u, a) in L' for every $u \in U$ and $a \in \mathcal{P}_1^*(u)$. Since the l_1 -cost of labeling $(\mathcal{P}_1^*, \mathcal{P}_2^*)$ is $|U|$, \mathcal{P}_1^* assigns only one label to every vertex $u \in U$, (as every u is in some $N(v)$, \mathcal{P}_1^* assigns at least one, and hence, exactly one label to u), and we have picked also one label (v, b) in L' for every $v \in V$. Note also that for any $v \in V$, the label b we picked for v is obviously a non-excluded label, by the definition of a solution in Label Cover, and hence (v, b) is a label in L . Consider any vertex $v \in V$ and a non-excluded label b by v . For every $u \in N(v)$ and label $a \in B_1$ such that $((u, v), a, b) \in \Pi$, the label (u, a) is assigned to some edge in the gadget $G'_{v,b}$ by its construction. So we know that every u -label we put in L' is actually also in the label set L . That is, L' is a subset of L with size of $|U| + |V|$.

Then fix any vertex $v \in V$. We consider the picked label (v, b) , and show that L' contains a path passing through the gadget $G'_{v,b}$. (Note that we view a path as the set of edges that appear on it, and we say that L' contains an edge subset E' if L' contains $L(E')$.) Recall that $G'_{v,b}$ consists of a series of consecutive subgraphs, one subgraph for each $u \in N(v)$. Consider the subgraph for a particular $u \in N(v)$. Suppose that there are p rungs in the subgraph, where the first edges of all such p rungs are labeled with $(u, a_1), \dots, (u, a_p)$, and these a_i are those satisfying $((u, v), a_i, b) \in \Pi$ (see Figure 3). Since $(\mathcal{P}_1^*, \mathcal{P}_2^*)$ is a total-cover, for some $1 \leq i \leq p$, $a_i \in \mathcal{P}_1^*(u)$. But we did pick labels (u, a) for all $a \in \mathcal{P}_1^*(u)$. So L' must contain a whole rung for the considered subgraph corresponding to u for $G'_{v,b}$. Since this holds for every subgraph corresponding to $u \in N(v)$ in $G'_{v,b}$, we know that L' contains a path in $G'_{v,b}$ from its left end to its right end. This gives a path passing through G'_v (See Figure 4). These paths for every gadget G'_v constitute an s - t path of G' , that is, L' is a feasible solution to instance I' . Since $|L'| = |U| + |V|$, we know that $\text{OPT}_{\text{PATH}}(I') \leq |U| + |V|$.

For the second part of the claim we prove its contrapositive. Suppose that P^* is an optimal s - t path of G' with $L(P^*) = L^*$ and $|L^*| \leq g|U| + |V|$. Denote by n_{ul} (resp. n_{vl}) the number of u -labels (resp. v -labels) in L^* . Since G' consists of $|V|$ consecutive subgraphs G'_v , P^* must traverse each G'_v . By the construction of G'_v as a stack of $G'_{v,b}$, it must traverse at least one $G'_{v,b}$, and therefore L^* must pick at least one v -label (v, b) for every $v \in V$. This defines our assignment \mathcal{P}_2 on V , and $n_{\text{vl}} \geq |V|$. Next we define \mathcal{P}_1 on U as follows: $\mathcal{P}_1(u) = \{a \mid (u, a) \text{ is a picked } u\text{-label in } L^*\}$. Since $n_{\text{vl}} \geq |V|$, it follows that $n_{\text{ul}} \leq g|U|$. Since for every v , P^* traverses some $G'_{v,b}$ of G'_v , for every $u \in N(v)$, L^* must include at least one u -label (u, a) such that $((u, v), a, b) \in \Pi$. In particular $\mathcal{P}_1(u)$ is non-empty. Since $U = \bigcup_{v \in V} N(v)$, this is true for all $u \in U$. It follows that $(\mathcal{P}_1, \mathcal{P}_2)$ is a solution to the Label Cover instance I . Since $n_{\text{ul}} \leq g|U|$, we know that the l_1 -cost of $(\mathcal{P}_1, \mathcal{P}_2)$ is $\leq g|U|$. This completes the proof of the contrapositive.

we know that it By an argument similar to that of Theorem 3.3, we know that it is **NP**-hard to approximate Label Path within ratio $g_c(\bar{n})$ for any constant $c < 1/2$, where \bar{n} denotes the input

length of Label Path. The proof of the theorem is concluded. ■