Algorithmica (2018) 80:398–409
https://doi.org/10.1007/s00453-016-0265-1

Simpler and Better Approximation Algorithms for the Unweighted Minimum Label *s*-*t* Cut Problem

Peng Zhang[1] · **Bin Fu**[2] · **Linqing Tang**[3]

Received: 5 June 2015 / Accepted: 2 December 2016 / Published online: 12 December 2016

Abstract Given a graph $G = (V, E)$ with a label set $L = \{\ell_1, \ell_2, \ldots, \ell_q\}$, in which each edge has a label from L, and a source $s \in V$ together with a sink $t \in V$, the Minimum Label s-t Cut problem asks to pick a set $L' \subseteq L$ of labels with minimized cardinality, such that the removal of all edges with labels in L' from G disconnects s and t. Let $n = |V|$ and $m = |E|$. The previous best known approximation ratio for this problem in literature is $O(m^{1/2})$. We present two simple and purely combinatorial approximation algorithms for the problem with ratios $O(n^{2/3}/\mathrm{OPT}^{1/3})$ and $O(m^{1/2}/\mathrm{OPT}^{1/2})$, where OPT is the optimal value of the input instance. The former result gives the first approximation ratio which is sublinear in n for the problem, and in particular applies to the instances with dense graphs (e.g., $m = \Theta(n^2)$). The latter

Peng Zhang is supported by the National Natural Science Foundation of China (61672323). Natural Science Foundation of Shandong Province ZR2015FM008 and ZR2016AM28, and the Fundamental Research Funds of Shandong University (2015JC006). Bin Fu is supported by NSF Career Award 0845376. Linqing Tang is partially supported by Hundred Talent Program of Chinese academy of Sciences under Prof. Angsheng Li.

✉ Peng Zhang
algzhang@sdu.edu.cn

Bin Fu
bfu@utpa.edu

Linqing Tang
linqing@ios.ac.cn

[1] School of Computer Science and Technology, Shandong University, Jinan 250101, China

[2] Department of Computer Science, University of Texas-Pan American, Edinburg, TX 78539, USA

[3] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

result improves the previous ratio $O(m^{1/2})$, as we can always assume that OPT is a super-constant.

Keywords Label cut · Minimum cut · Shortest path · Approximation algorithms

Mathematics Subject Classification 68W25 · 90C27

1 Introduction

In this paper, we study the Minimum Label s-t Cut problem (Label Cut for short), as shown in Definition 1.

Definition 1 The Minimum Label s-t Cut problem.

Instance: We are given a (directed or undirected) graph $G = (V, E)$, a source $s \in V$, a sink $t \in V$, and a label set L containing q labels $\ell_1, \ell_2, \dots, \ell_q$. Each edge in graph G has a label from L.

Goal: A label subset $L' \subseteq L$ is called a *label s-t cut*, if the removal of all edges with labels in L' from G disconnects s and t (that is, disconnects all s-t paths). The goal of the problem is to find a minimum size label s-t cut.

The problem defined here is actually the *unweighted* Label Cut problem, since the cost of a solution $L' \subseteq L$ is just its size $|L'|$, and this is equivalent to that each label in the problem has a unit weight.

Origins of the problem. The Label Cut problem is quite natural that it can appear in many contexts. We came across this problem from the work of Jha et al. [12], Sheyner et al. [19], and Sheyner et al. [20] in computer security, in particular on intrusion detection. In this application, an intruder wants to intrude some system, while the system protector wants to prevent the intrusion. The intruder is equipped with some attack methods, called "atomic attacks". At the beginning the intruder lies in the initial state. The intruder may change himself/herself from one state to another by applying the corresponding atomic attack. When the intruder reaches the success state, it means that he/she successfully intrudes the system. On the other hand, the protector pay some cost to disable an atomic attack of the intruder. The goal of the protector is to prevent the intruder from reaching the success state at the minimum cost. We can use "attack graph" to describe formally the above intrusion and protection. An attack graph G has nodes representing various states, and directed edges representing state transitions and 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. Then the computational task is to find a subset of atomic attacks of the minimum total cost, such that the removal of all edges labeled by these atomic attacks disconnects s and t. This is precisely the Label Cut problem.

Very interestingly, the Label Cut problem independently arose in the research about network survivability by Coudert et al. [6]. In the area of telecommunications, a virtual network (e.g., IP/WDM and MPLS networks, as pointed out in [6]) is a multi-layer network whose links correspond to paths in the underlying physical network. More importantly, multiple links in such a virtual network may share a common physical link in the underlying network. The set of such edges is thus called a "shared risk link group", as once the underlying physical link fails, all these edges fail. Each shared risk link group is associated with a distinct label, and each virtual link in the multi-layer network has one or more labels.[1] Then, the Label Cut problem is related to the vulnerability of a virtual multi-layer network, since the minimum label s-t cut gives a tight lower bound on the number of failures that can disconnect the distinguished node pair s and t.

From the viewpoint of combinatorial optimization, the Label Cut problem is a natural generalization of the classic Minimum s-t Cut problem, since Minimum s-t Cut can be viewed as a special case of Label Cut in which each edge has a unique label. It is well-known that Minimum s-t Cut can be solved in polynomial time (see, e.g., [1, Chapter 7]). By contrast, Label Cut is NP-hard and has even high approximation hardness (see the related work in Sect. 1.1).

Notations. Throughout the paper, for an input graph G, we use n to denote its vertex number, and m its edge number. Given an instance of an optimization problem such as Label Cut, we use OPT to denote the optimal value of the instance.

In the Label Cut problem, given an edge set $E' \subseteq E$, we use $L(E')$ to denote the set of labels appearing in E'. Note that L also denotes the label set in the Label Cut problem. We do not introduce more symbols to distinguish these two cases and just want to keep them simple and easily understandable. Similarly, given an edge e, we use $\ell(e)$ to denote the label of e. Note that we also write $\ell \in L$ and in this case ℓ denotes some label in L.

1.1 Related Work

Jha et al. [12] proved that Label Cut is NP-hard by reducing the Hitting Set problem to it. They also expressed the Label Cut problem as a Hitting Set problem (the collection of subsets to hit is $U = \{S \mid S$ is the set of labels appearing on an s-t path$\}$ and the element set is just the label set L). The authors [12] then translated the well-known greedy algorithm for Set Cover (see, e.g., [13] and [22, Chapter 2]) to get an approximation algorithm for Label Cut with approximation ratio $O(\log |U|)$. The issue here is that in general U may be of exponential size.

Coudert et al. [6] proved that the Label Cut problem (called the Minimum Color s-t Cut problem in [6]) is NP-hard and APX-hard by reducing the MAX 3SAT problem to it. They also presented approximation algorithms for the Label Cut problem in some special graphs and gave approximation hardness results for the Label Cut problem.

Zhang et al. [24] gave the first non-trivial approximation algorithm for the Label Cut problem in general graphs and improved approximation hardness result for the Label Cut problem. Specifically, they gave a polynomial time $O(m^{1/2})$-approximation algorithm for the Label Cut problem, where m is the number of edges in graph G. On the approximation hardness side, they showed that the Label Cut problem can not be

[1] The Label Cut problem in which each edge may have more than one labels can be easily reduced to the one in Definition 1.

approximated within $2^{(\log|\mathcal{I}|)^{1-1/(\log\log|\mathcal{I}|)^c}}$ for any constant $c < 1/2$ unless P = NP, where $|\mathcal{I}|$ is the input length of the problem.

Fellows et al. [8] considered the parameterized complexity of the Label Cut problem. They showed that the Label Cut problem is $W[2]$-hard when parameterized by the maximum number of labels allowed in a solution, even in graphs whose path-width is bounded from above by a small constant.

Jegelka et al. [10,11] studied a more general cut problem called Cooperative s-t Cut, which finds an s-t cut such that an objective function is minimized, where the objective function can be arbitrary submodular function defined on 2^E. It is not difficult to see that Cooperative s-t Cut is a generalization of Label Cut (i.e., Label s-t Cut). Jegelka et al. [10,11] gave some approximation algorithms for the Cooperative s-t Cut problem, with much effort put in how to deal with the submodularity of the cost function in the problem. However, when translated to Label Cut, these results do not improve the previous best known approximation ratio $O(m^{1/2})$ for Label Cut.

Besides the Label Cut problem, there are still many other classic optimization problems that have been studied under the edge-labeled model, such as the Minimum Label Spanning Tree problem [2,4,9,16], the Minimum Label s-t Path problem [3,9], the Minimum Label Traveling Salesman problem [7,23], the Minimum Label Perfect Matching problem [18], and the Minimum Label Steiner Tree problem [5], etc. The readers are advised to refer to the corresponding references for these results.

1.2 Our Results

In this paper, we give improved and new approximation algorithms for the unweighted Label Cut problem. Our main result is an $O(n^{2/3}/\mathrm{OPT}^{1/3})$-approximation algorithm for the unweigthed Label Cut problem (Theorem 3), where $n = |V(G)|$ and OPT is the optimal solution value. This is the first approximation algorithm for Label Cut whose ratio is sublinear in n. Note that the previous best known approximation ratio for Label Cut is $O(m^{1/2})$ [24], where $m = |E(G)|$. In contrast, the ratio $O(n^{2/3}/\mathrm{OPT}^{1/3})$ (which is better than $O(n^{2/3})$) has its own significance, since in dense graphs m may be as large as $\Theta(n^2)$ and in this case $O(m^{1/2})$ degenerates into $O(n)$. Actually, when $m > n^{4/3}$, the ratio $O(n^{2/3})$ is better than $O(m^{1/2})$.

Our other result is an $O(m^{1/2}/\mathrm{OPT}^{1/2})$-approximation algorithm for the unweighted Label Cut problem (Theorem 2). Since we can always assume that OPT is a super-constant (i.e., larger than any constant),[2] our approximation ratio $O(m^{1/2}/\mathrm{OPT}^{1/2})$ improves the previous best known ratio $O(m^{1/2})$ [24] for the problem. In other words, the ratio $O(m^{1/2}/\mathrm{OPT}^{1/2})$ is better than $O(m^{1/2})$ when OPT is large. Please also note that the previous ratio $O(m^{1/2})$ has nothing to do with OPT.

These approximation results are achieved by a simple purely combinatorial two-stage strategy. The overall idea is that when a minimum size s-t cut (an s-t cut is a set of edges whose removal disconnects s and t) is not too large, we can use this minimum size s-t cut as a reasonable approximation for the minimum label s-t cut. To achieve

[2] If OPT is bounded from above by a constant C, we can find in polynomial time the optimal solution by enumerating all possible label cuts of size at most C.

this goal, in the first stage, we repeatedly remove shortest s-t paths from the graph, making the shortest s-t path of the current graph longer and longer. When the length of the shortest s-t path exceeds a specific threshold, the first stage terminates and we start the second stage. In the second stage, since every s-t path is relatively long, we can prove that in such graph the minimum size s-t cut is not too large. So, we just find a minimum size s-t cut of the current graph to separate s and t eventually. The final solution is the union of the labels obtained from these two stages, that is, the labels coming from the edges of shortest s-t paths and the labels coming from the edges of the minimum size s-t cut.

The two-stage idea for labeled graph problems is natural and has been applied for example, in [9] to solve the Minimum Label s-t Path problem and in [24] to solve the Label Cut problem. In [24], the greedy algorithm for Label Cut in its first stage removes as many as possible edges at a bounded cost by repeatedly solving (that is, finding an approximate solution of) the Budgeted Maximum Coverage problem [14]. When the graph is sparse enough that the size of the minimum s-t cut is relatively small, the greedy algorithm enters its second stage and computes a minimum size s-t cut. The Budgeted Maximum Coverage problem is NP-hard and can be approximated within $1 - \frac{1}{e}$ [14]. By contrast, our algorithms in this paper use the more fundamental Shortest s-t Path problem as a subroutine and this problem can be optimally solved very fast [2, Chapter 7].

The key of the analysis of our algorithms is two structural lemmas that relate the size of the minimum s-t cut in a graph to the number of edges in the graph (Lemma 1) or to the number of vertices in the graph (Lemma 2), conditioned on that the length of the shortest s-t path in the graph is relatively long. While Lemma 1 is proved using Menger's theorem, the proof of lemma 2 is based on a layered technique and is more interesting. We believe the two structural lemmas are of independent interest and may find more applications beyond Label Cut.

The hardness of Label Cut. Although improved upon the previous results, the approximation ratios $O\left(\frac{m^{1/2}}{\mathrm{OPT}^{1/2}}\right)$ and $O\left(\frac{n^{2/3}}{\mathrm{OPT}^{1/3}}\right)$ we obtained for Label Cut are still large. Basically, this is because Label Cut is a problem hard to approximate, which we explain from two aspects.

First, as mentioned in Sect. 1.1, the known approximation hardness factor of Label Cut is $2^{(\log |\mathcal{I}|)^{1-1/(\log\log |\mathcal{I}|)^c}}$ ($c < 1/2$ is any constant) [24], which is already high. One can verify that $2^{(\log n)^{1-1/(\log\log n)^c}}$ is higher than any poly-logarithmic factor, but lower than any polynomial factor, both when n is sufficiently large. To the best of our knowledge, there is (almost) no any problem whose approximation ratio is proved to be $2^{\log^{1-o(1)} |\mathcal{I}|}$. This means that the best approximation ratio we can expect for Label Cut should be a polynomial factor.

Second, we can show that the integrality gap of a natural LP-relaxation for Label Cut is very large. See the following linear program (LP-C). To see that it is an LP-relaxation for Label Cut, consider its 0–1 integer version. Given an instance of Label Cut, we define a variable $x_\ell \in \{0, 1\}$ for each label $\ell \in L$. The value of x_ℓ being 1 means that label ℓ is chosen and its value being 0 means not. Constraint (1) is to make sure that for every s-t path P in G, at least one label from the edges of P is chosen. In constraint (1), $\mathcal{P}_{st}$ denotes the set of all simple s-t paths in G, where an s-t path

P is viewed as a set of edges in that path. Then the set of labels with $x_\ell = 1$ forms a solution to the problem.

$$\begin{aligned} \min \quad & \sum_{\ell \in L} x_\ell && \text{(LP-C)} \\ \text{s.t.} \quad & \sum_{e \in P} x_{\ell(e)} \geq 1, \quad \forall P \in \mathcal{P}_{st} && (1) \\ & x_\ell \geq 0, \qquad\quad \forall \ell \in L \end{aligned}$$

It is easy to prove that (LP-C) has integrality gap $\Omega(m)$. This means that any approximation algorithm for Label Cut that only uses the fractional optimum of (LP-C) as the lower bound of OPT, cannot have better approximation ratio than the integrality gap. Consider the following instance. The graph G (can be either directed or undirected) is just an s-t path of length $n-1$. The label set L contains only one label ℓ. Each edge on the path is labeled with this unique label. Then it is easy to verify that $x_\ell = \frac{1}{m}$ is a feasible solution to (LP-C) with objective value $\frac{1}{m}$, while the optimal solution to the instance has value 1. Since $m = \Theta(n)$ in graph G, the integrality gap is also $\Omega(n)$.

A preliminary version of the results in this paper appeared in the Proceedings of the 10th Latin American Theoretical Informatics Symposium (LATIN 2012) [21]. In [21], two approximation algorithms with the same ratios as in this paper are provided for Label Cut. However, the first stage of the algorithms in [21] is to round an optimal fractional solution[3] to the linear program (LP-C).[4] Consequently, the algorithms in [21] are not purely combinatorial. In this paper, we further simplify the algorithms in [21] by using a shortest path algorithm to implement the first stage, getting two purely combinatorial approximation algorithms for Label Cut.

Organization of the paper. The remainder of the paper is organized as follows. We first give the $O\left(\frac{m^{1/2}}{\text{OPT}^{1/2}}\right)$-approximation algorithm for Label Cut in Sect. 2, to illustrate the basic idea of the algorithms in this paper. Then, we give the main result in Sect. 3, the $O\left(\frac{n^{2/3}}{\text{OPT}^{1/3}}\right)$-approximation algorithm for Label Cut.

2 Approximation Algorithm with Ratio $O\left(\frac{m^{1/2}}{\text{OPT}^{1/2}}\right)$

2.1 A Structural Lemma

We give a structure lemma (Lemma 1 below) which relates the size of the minimum s-t cut to the number of edges in a directed or undirected graph. This lemma, while its proof is not difficult, is crucial for the analysis of Algorithm 1.

[3] Using a shortest path algorithm (e.g., the Dijkstra's algorithm) as the separation oracle (LP-C) can be optimally solved in polynomial time.

[4] The algorithms presented in [21] are also two-stage algorithms; they do not depend only on the linear programming technique. So, their approximation ratios can be better than the integrality gap of (LP-C).

First recall the (edge version of) Menger's theorem [17] (See also [2, Theorem 8.9]). Two paths are called *edge-disjoint* if they do not share any edge.

Theorem 1 (Menger's theorem) *Let G be a (directed or undirected) graph and s and t be two distinct vertices. Then the size of the minimum s-t cut is equal to the maximum number of edge-disjoint s-t paths.*

Note that for the undirected graph, an s-t cut (S, T) $(s \in S \subseteq V, t \in T = V \setminus S)$ is the set of edges running between S and T, while for the directed graph, a (directed) s-t cut (S, T) is the set of edges *only* from S to T.

Lemma 1 (Structural Lemma) *In a (directed or undirected) graph G, if every s-t path has length (number of edges) at least $m^{\frac{1}{2}}/a^{\frac{1}{2}}$, where $m = |E(G)|$ and a is a positive number, then the size of the minimum s-t cut is at most $m^{\frac{1}{2}}a^{\frac{1}{2}}$.*

Proof Since there are m edges in total, and every s-t path has length $\geq m^{\frac{1}{2}}/a^{\frac{1}{2}}$, there are at most $m^{\frac{1}{2}}a^{\frac{1}{2}}$ edge-disjoint s-t paths. By Theorem 1, the size of the minimum s-t cut is $\leq m^{\frac{1}{2}}a^{\frac{1}{2}}$. □

2.2 The Algorithm

The overall idea for solving Label Cut is simple: If in the problem the size of the minimum s-t cut is not so large, then it can be used as a good approximation for the minimum label s-t cut. So, we need to remove as many as possible edges at a bounded cost to make the minimum s-t cut sufficiently small. Consequently, the algorithm solving Label Cut consists of two stages: Remove as many as possible edges in its first stage, and compute a minimum s-t cut in its second stage. As usual, to obtain a good approximation ratio, a carefully selected balance between the costs of these two stages is needed. Last but not least, we find a very simple way to implement the first stage. The whole algorithm is shown as Algorithm 1.

Algorithm 1

```
1 Guess a value $\Delta$ such that OPT $\leq \Delta < 2 \cdot$ OPT.
2 while the length (number of edges) of the shortest $s$-$t$ path is $\leq m^{\frac{1}{2}}/\Delta^{\frac{1}{2}}$ do
3     Find a shortest $s$-$t$ path $P$.
4     Pick all labels appearing in $P$. Remove all edges from $G$ with these labels.
5 end while
6 Denote by $G_R$ the remaining graph. Find a minimum $s$-$t$ cut $E'$ of $G_R$. Pick all labels appearing in $E'$.
7 Let $L_1$ be the set of labels picked from steps 2 to 5, and $L_2 = L(E')$ be the set of labels picked in step
  6. Output $L_1 \cup L_2$ as the final solution.
```

Suppose we have the correct value of Δ in step 1. First we give the analysis of Algorithm 1. The guess skill in step 1 is a standard technique in the design of approximation algorithms, whose details will be briefly explained after Theorem 2. In Algorithm 1, steps 2 to 5 form the first stage, and step 6 forms the second stage.

Theorem 2 *Algorithm 1 is an* $O\left(m^{\frac{1}{2}}/OPT^{\frac{1}{2}}\right)$*-approximation algorithm for the unweighted* Minimum Label s-t Cut *problem in both directed and undirected graphs.*

Proof First note that $L_1 \cup L_2$ must be a feasible solution to the Label Cut problem.

Let h be the number of paths selected in the first stage. Note that these paths are disjoint, and they do not share any label. So, the optimal solution must select at least one label from each of these h paths, meaning that OPT $\geq h$. Consequently, we have

$$|L_1| \leq \frac{m^{1/2}}{\Delta^{1/2}} h \leq \frac{m^{1/2}}{\Delta^{1/2}} \mathrm{OPT} = O\left(\frac{m^{1/2}}{\mathrm{OPT}^{1/2}}\right) \mathrm{OPT}.$$

When the algorithm reaches stage 2, every s-t path has length at least $\frac{m^{1/2}}{\Delta^{1/2}} \geq \frac{|E(G_R)|^{1/2}}{\Delta^{1/2}}$. Applying Lemma 1 on graph G_R (just letting $a = \Delta$), the minimum s-t cut E' of G_R satisfies $|E'| \leq |E(G_R)|^{1/2}\Delta^{1/2} \leq (m\Delta)^{1/2}$. Then we have

$$|L_2| \leq |E'| \leq (m\Delta)^{1/2} = O\left(\frac{m^{1/2}}{\mathrm{OPT}^{1/2}}\right) \mathrm{OPT}.$$

Let SOL be the solution value of Algorithm 1. Therefore,

$$SOL = |L_1 \cup L_2| = O\left(\frac{m^{1/2}}{\mathrm{OPT}^{1/2}}\right) \mathrm{OPT}.$$

It is well known that the Shortest s-t Path problem and the Minimum s-t Cut problem in both directed and undirected graphs can be solved in polynomial time. In the first stage, at most polynomial number of paths are computed since edges are removed each time. So, Algorithm 1 runs in polynomial time, concluding the theorem. □

Guess the value of Δ in Algorithm 1. Guessing the value of Δ in step 1 is a standard technique used in the design of approximation algorithms. While it can be used directly as in Algorithm 1, for completeness, we show the details of the guess procedure below. By the definition of Label Cut, we know that $1 \leq$ OPT $\leq q$. (Recall that q is the number of labels in the input instance. See Definition 1.) Let $b = \lceil \log q \rceil + 1$. Define

$$\Delta_i = \begin{cases} 2^{i-1}, & 1 \leq i \leq b-1 \\ \min\{2^{i-1}, q\}, & i = b \end{cases}$$

For each value $\Delta_i \in \{1, 2, 4, \ldots, \Delta_b\}$, we run Algorithm 1 (except step 1). So, we find b solutions in total and the actual final solution returned by the algorithm is the best one (the solution with the minimum size) among them.

Suppose $\Delta_{k-1} <$ OPT $\leq \Delta_k$ for some $k \in \{1, \ldots, b\}$. (Let $\Delta_0 = \frac{1}{2}$. This Δ_0 is not a guess of Δ; it is only used here for the analysis.) Therefore, we have $\Delta_k = 2\Delta_{k-1} <$ $2 \cdot$ OPT and hence OPT $\leq \Delta_k < 2 \cdot$ OPT. This is the correct value of Δ we used in step 1. For this particular Δ_k, Theorem 2 obviously holds. Since the actual final solution is the minimum size one among the b solutions found above, which include the solution corresponding to Δ_k, we know that the approximation ratio of the whole algorithm is really $O(m^{\frac{1}{2}}/\mathrm{OPT}^{\frac{1}{2}})$.

The guess procedure described above contains only $b = O(\log q)$ iterations. So, the running time of the whole algorithm is still polynomial in the input length. Note that by this technique, knowing OPT lies in $[1, q]$ suffices to guess Δ; we need not to know the exact value of OPT.

Since the guess technique is a standard technique, it is described as only one step in Algorithm 1. In this sense, we say that Algorithm 1 is a two-stage algorithm. Of course, the actual meaning is that each iteration of the whole algorithm consists of two stages.

3 Approximation Algorithm with Ratio $O\left(\frac{n^{2/3}}{\mathrm{OPT}^{1/3}}\right)$

In this section, we give our main result in this paper, that is, an $O(n^{\frac{2}{3}}/\mathrm{OPT}^{\frac{1}{3}})$-approximation algorithm for the unweighted Label Cut problem. First we give some intuition to design such an algorithm. From Sect. 2, we know that the structural Lemma 1, which relates the size of the minimum s-t cut to the number of edges in the graph, plays an important role in the analysis of Algorithm 1. So, for an algorithm whose approximation ratio is sublinear in n, the number of vertices in the graph, a similar lemma relating the size of the minimum s-t cut to n is needed.

3.1 A Structural Lemma

In the case of the structural Lemma 1, which is in terms of m, Menger's Theorem is helpful to prove this lemma. However, Menger's Theorem (even its vertex version) cannot again help to prove similar structural lemma in terms of n (Lemma 2 below). Instead, we directly prove such a lemma by an interesting layered technique.

Lemma 2 (Structural Lemma) *In a (directed or undirected) graph G, if every s-t path has length (number of edges) at least $n^{\frac{2}{3}}/a^{\frac{1}{3}}$, where $n = |V(G)|$ and a is a positive number, then the size of the minimum s-t cut is $O(n^{\frac{2}{3}}a^{\frac{2}{3}})$.*

Proof For two vertices u and v, define the distance $d(u, v)$ between them as the length of the shortest u-v path. We partition all vertices in G into several disjoint sets (layers) $V_0, V_1, V_2, \ldots$ according to their distance from vertex s, that is, $V_i = \{v \in V : d(s, v) = i\}$. Since $d(s,t) \geq n^{\frac{2}{3}}/a^{\frac{1}{3}}$, there are at least $\frac{n^{2/3}}{a^{1/3}} + 1$ different i's such that $V_i \neq \emptyset$. We omit the vertices in G that are not reachable from s.

We claim that there exists an $i \leq \frac{n^{2/3}}{a^{1/3}}$ such that $|V_i||V_{i+1}| \leq 100(na)^{2/3}$. Assume for contradiction for every $0 \leq i \leq \frac{n^{2/3}}{a^{1/3}}$, we have $|V_i||V_{i+1}| > 100(na)^{2/3}$. Then for each two consecutive layers V_i and V_{i+1}, at least one of them has size $> 10(na)^{1/3}$. Therefore, there are at least $\frac{1}{2} \cdot \frac{n^{2/3}}{a^{1/3}}$ different i's satisfying $|V_i| > 10(na)^{1/3}$ and thus there are at least $5n$ vertices in these V_i's. This cannot be true since G has only n vertices.

Since there exists an $i \leq \frac{n^{2/3}}{a^{1/3}}$ such that $|V_i||V_{i+1}| \leq 100(na)^{2/3}$, we know that the minimum s-t cut E' satisfies $|E'| \leq 100(na)^{2/3}$. □

3.2 The Algorithm

The approximation algorithm for unweighted Label Cut whose performance ratio is sublinear in n is shown as Algorithm 2 below, which is almost the same as Algorithm 1. The only difference is that the upper bound on the length of the shortest s-t path is $n^{\frac{2}{3}}/\Delta^{\frac{1}{3}}$ in (step 2 of) Algorithm 2, while this bound in Algorithm 1 is $m^{\frac{1}{2}}/\Delta^{\frac{1}{2}}$. For completeness we give the full description of Algorithm 2.

Algorithm 2

```
1 Guess a value $\Delta$ such that OPT $\leq \Delta < 2 \cdot$ OPT.
2 while the length (number of edges) of the shortest $s$-$t$ path is $\leq n^{\frac{2}{3}}/\Delta^{\frac{1}{3}}$ do
3     Find a shortest $s$-$t$ path $P$.
4     Pick all labels appearing in $P$. Remove all edges from $G$ with these labels.
5 end while
6 Denote by $G_R$ the remaining graph. Find a minimum $s$-$t$ cut $E'$ of $G_R$. Pick all labels appearing in $E'$.
7 Let $L_1$ be the set of labels picked from steps 2 to 5, and $L_2 = L(E')$ be the set of labels picked in step
  6. Output $L_1 \cup L_2$ as the final solution.
```

In Algorithm 2, steps 2 to 5 form the first stage, and step 6 forms the second stage.

Theorem 3 *Algorithm 2 is an $O\left(n^{\frac{2}{3}}/OPT^{\frac{1}{3}}\right)$-approximation algorithm for the unweighted* Minimum Label s-t Cut *problem in both directed and undirected graphs.*

Proof The proof given briefly below is similar to that of Theorem 2. First note that Algorithm 2 obviously gives a feasible solution and runs in polynomial time.

Let h be the number of paths selected in the first stage. Since these paths do not share any label, we know OPT $\geq h$. Consequently, we have

$$|L_1| \leq \frac{n^{2/3}}{\Delta^{1/3}}h \leq \frac{n^{2/3}}{\Delta^{1/3}}\text{OPT} = O\left(\frac{n^{2/3}}{\text{OPT}^{1/3}}\right)\text{OPT}.$$

When the algorithm reaches the second stage, every s-t path has length at least $\frac{n^{2/3}}{\Delta^{1/3}} \geq \frac{|V(G_R)|^{2/3}}{\Delta^{1/3}}$. Applying Lemma 2 on graph G_R (just letting $a = \Delta$), the minimum s-t cut E' of G_R satisfies $|E'| = O(|V(G_R)|^{2/3}\Delta^{2/3}) = O((n\Delta)^{2/3})$. Then we have

$$|L_2| \leq |E'| = O((n\Delta)^{2/3}) = O\left(\frac{n^{2/3}}{\text{OPT}^{1/3}}\right)\text{OPT}.$$

Let SOL be the solution value of Algorithm 2. Therefore,

$$SOL = |L_1 \cup L_2| = O\left(\frac{n^{2/3}}{\text{OPT}^{1/3}}\right)\text{OPT},$$

finishing the proof of the theorem. □

Acknowledgements We are grateful to the anonymous reviewers for their warm suggestions, which help to improve the presentation of the paper.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs, NJ (1993)
2. Broersma, H., Li, X.: Spanning trees with many or few colors in edge-colored graphs. Discuss. Math. Graph Theory **17**(2), 259–269 (1997)
3. Broersma, H., Li, X., Woeginger, G., Zhang, S.: Paths and cycles in colored graphs. Australas. J. Comb. **31**, 299–311 (2005)
4. Chang, R.-S., Shing-Jiuan, L.: The minimum labeling spanning trees. Inf. Process. Lett. **63**(5), 277–282 (1997)
5. Consoli, S., Moreno-Pérez, J.A., Darby-Dowman, K., Mladenovic, N.: Discrete particle swarm optimization for the minimum labelling steiner tree problem. Nat. Comput. **9**(1), 29–46 (2010)
6. Coudert, D., Datta, P., Perennes, S., Rivano, H., Voge, M.-E.: Shared risk resource group: complexity and approximability issues. Parallel Process. Lett. **17**, 169–184 (2007)
7. Couëtoux, B., Gourvès, L., Monnot, J., Telelis, O.: Labeled traveling salesman problems: complexity and approximation. Discrete Optim. **7**(1–2), 74–85 (2010)
8. Fellows, M., Guo, J., Kanj, I.: The parameterized complexity of some minimum label problems. J. Comput. Syst. Sci. **76**(8), 727–740 (2010)
9. Hassin, R., Monnot, J., Segev, D.: Approximation algorithms and hardness results for labeled connectivity problems. J. Comb. Optim. **14**(4), 437–453 (2007)
10. Jegelka, S., Bilmes, J.: Cooperative Cuts: Graph Cuts with Submodular Edge Weights. Technical Report TR-189, Max Planck Institute for Biological Cybernetics (2010)
11. Jegelka, S., Bilmes, J.: Graph Cuts with Interacting Edge Costs-Examples, Approximations, and Algorithms. CoRR, arXiv:1402.0240 (2014)
12. Jha, S., Sheyner, O., Wing, J.M.: Two formal analyses of attack graphs. In: Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW), pp. 49–63. IEEE Computer Society (2002)
13. Johnson, D.: Approximation algorithms for combinatorial problems. J. Comput. Syst. Sci. **9**, 256–278 (1974)
14. Khuller, S., Moss, A., Naor, J.: The budgeted maximum coverage problem. Inf. Process. Lett. **70**(1), 39–45 (1999)
15. Korte, B., Vygen, J.: Combinatorial Optimization: Theory and Algorithms, 5th edn. Springer, Berlin (2012)
16. Krumke, S.O., Wirth, H.C.: On the minimum label spanning tree problem. Inf. Process. Lett. **66**(2), 81–85 (1998)
17. Menger, K.: Zur allgemeinen kurventheorie. Fundam. Math. **10**, 96–115 (1927)
18. Monnot, J.: The labeled perfect matching in bipartite graphs. Inf. Process. Lett. **96**, 81–88 (2005)
19. Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.: Automated generation and analysis of attack graphs. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 273–284. Oakland, CA (2002)
20. Sheyner, O., Wing, J.M.: Tools for generating and analyzing attack graphs. In: de Boer, F.S., Bonsangue, M.M., Graf, S., de Roever, W.P. (eds.) Proceedings of Formal Methods for Components and Objects, Second International Symposium (FMCO 2003), vol. 3188 of LNCS, pp. 344–371. Springer (2004)
21. Tang, L., Zhang, P.: Approximating minimum label s-t cut via linear programming. In: Fernández-Baca, D. (ed.) Proceedings of the 10th Latin American Theoretical Informatics Symposium (LATIN), vol. 7256 of LNCS, pp. 655–666. Springer (2012)

22. Vazirani, V.V.: Approximation Algorithms. Springer, Berlin (2001)
23. Xiong, Y., Golden, B., Wasil, E.: The colorful traveling salesman problem. In: Baker, E.K., Joseph, A., Mehrotra, A., Trick, M. (eds.) Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies, pp. 115–123. Springer, Berlin (2007)
24. Zhang, P., Cai, J.-Y., Tang, L., Zhao, W.: Approximation and hardness results for label cut and related problems. J. Comb. Optim. **21**(2), 192–208 (2011)