



Labeled cuts in graphs



Tridib Dutta^a, Lenwood S. Heath^b, V.S. Anil Kumar^{a,*}, Madhav V. Marathe^a

^a Department of Computer Science, and Biocomplexity Institute, Virginia Tech, Blacksburg, VA, United States

^b Department of Computer Science, Virginia Tech, Blacksburg, VA, United States

ARTICLE INFO

Article history:

Received 22 December 2010

Received in revised form 25 July 2016

Accepted 29 July 2016

Available online 9 August 2016

Communicated by T. Erlebach

Keywords:

Labeled graphs

Minimum cut

Regular languages

ABSTRACT

In this paper, we study labeled extensions of the classical s, t -mincut problem, in which we are given a graph $G = (V, E)$, two specific vertices $s, t \in V$, a set L of labels, and a labeling $\ell : E \rightarrow L$ of the edges. The goal is to choose a subset $L' \subseteq L$ of labels, so that s and t become disconnected when deleting the edges with labels in L' . We give an algorithm with an $O(n^{2/3})$ approximation factor guarantee, which improves the $O(\sqrt{m})$ approximation guarantee of Zhang et al. (2009) [16]. We also consider variants in which selected subsets of paths between s and t have to be removed (instead of all paths). These labeled cut problems are much harder than the classical mincut problem.

© 2016 Published by Elsevier B.V.

1. Introduction

Labeled graphs arise in a number of applications, e.g., edges in social contact graphs can be labeled by the time of contact [12]. The focus of this paper is on two classes of cut problems in labeled graphs that generalize the classical s, t -mincut problem in a natural manner. Let $G = (V, E)$ denote a directed graph, and let L denote a set of labels. Let $\ell : E \rightarrow L$ denote a labeling function. The first problem we consider is the s, t -Labeled Mincut Problem (LCP), which involves choosing the smallest subset $L' \subset L$ such that two given nodes s and t are disconnected in the graph $G' = (V, E \setminus \{e : \ell(e) \in L'\})$, obtained by deleting all edges with labels in the set L' . The LCP problem is used in the context of “attack graphs” in intrusion detection by Jha et al. [11]; in this case, the edge labels represent possible attacks, and a labeled mincut is the smallest set of attacks that disconnect two selected terminals. The complexity of LCP was first studied by Zhang et al. [16], who show that, unlike the classical mincut problem, this problem is NP-complete even to approximate within a factor of $O(2^{\log^{1-o(1)} n})$, where $n = |V|$. They also present a greedy algorithm that achieves an approximation factor of $O(\sqrt{m})$ for edge weighted graphs. In this paper, we present an algorithm with an approximation factor of $O(n^{2/3})$ for unweighted graphs, improving the result of [16] when $m = \Omega(n^{4/3+\epsilon})$ for $\epsilon > 0$.

The second class of labeled cut problems we consider is called the s, t -Language Constrained Labeled Cut (LCLC) problem. Here, we are given a formal language $\mathcal{L}$ defined on the set L . For a path P in G , let $\ell(P)$ denote the word formed by concatenating the labels on the edges in P . The LCLC problem involves choosing the smallest subset $L' \subset L$ of labels, such that $E \setminus \{e : \ell(e) \in L'\}$ contains no paths P from s to t with $\ell(P) \in \mathcal{L}$. In this paper, we assume that $\mathcal{L}$ is a regular language, which is a language that can be accepted by a finite automaton (see, e.g., [8]).

The above versions of the labeled cut problems are all quite hard in general, even for special classes of graphs, e.g., bounded treewidth graphs. Natural duals of these problems can be defined, which are variants of the classical flow problem.

* Corresponding author.

E-mail addresses: dutta.tridib@gmail.com (T. Dutta), heath@vt.edu (L.S. Heath), akumar@vbi.vt.edu (V.S.A. Kumar), mmarathe@vbi.vt.edu (M.V. Marathe).

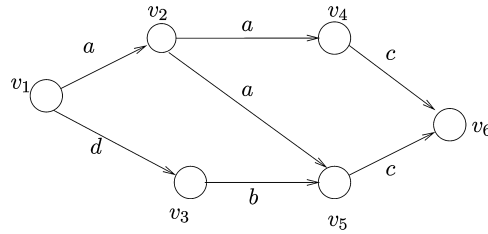


Fig. 1. An example illustrating the LCP and LCLC problems. Let $s = v_1$ and $t = v_6$. $L = \{a, b, c, d\}$ denotes the set of labels. The optimum LCP solution is the set $\{c\}$, while the minimum unlabeled cut has cost 2. Let $\mathcal{L}$ be the language of strings a^*c . The set $\mathcal{P}(G, \mathcal{L}, s, t)$ of $\mathcal{L}$ -respecting paths consists of only two paths, v_1, v_2, v_4, v_6 and v_1, v_2, v_5, v_6 , and either $\{a\}$ or $\{c\}$ is an optimal solution to the LCLC problem. Observe that the solution $\{a\}$ keeps s and t still connected.

The dual for the LCP problem is a flow with capacities associated with labels, instead of with individual edges. The dual of the LCLC problem is a maximum flow problem in which only flow paths whose labels form words in the language $\mathcal{L}$ can be used. Unlike classical flows, the maxflow-mincut duality does not hold, and there can be a large gap between these quantities—this follows from the gap between flows and cuts for length constrained paths [3] (we note that the LCLC problem generalizes the problem of cutting length constrained paths).

Labeled versions of a few other combinatorial problems have been studied. One of the earliest such problems is the minimum label spanning tree (MinLST) problem, in which the goal is to choose a spanning tree, whose cost, measured by the sum of the costs of the labels of the edges in the spanning tree, is minimized. Krumke et al. [13] give an $O(\log n)$ approximation bound for this problem. Carr et al. [5] show that MinLST is a special case of the Red-Blue Set Cover problem and study its complexity. Another labeled problem is the minimum label s, t path (MinLP) problem, in which the objective is to find a path from s to t , minimizing the combined cost of its labels. Hassin et al. [7] show that MinLP can be approximated within a factor of $O(\sqrt{n})$. The closest prior work related to the LCLC problem is by Barrett et al. [4], who study the problem of finding a shortest language constrained path in a labeled graph. For regular languages, they reduce this problem to shortest paths in an appropriately defined product graph. Applications of language constrained shortest paths include multi-modal routing, formulating database queries [1], and web searches [14]. LCP is a special case of the *Submodular Cut* problem, in which the cost of a cut is defined by a submodular function (for LCP, define the cost of a cut as the number of distinct labels on the edges in the cut). This was studied recently by Jegelka et al. [10], who show that this problem is difficult to approximate within a factor of $O(n^{1/3})$ in general.

Our Contributions. In this paper, we study the complexity of the LCP and LCLC problems. For the LCP problem, we present an algorithm, *ApproxLCP*, with an approximation guarantee of $O(n^{2/3})$, which improves upon the result of Zhang et al. [16] for unweighted graphs with $m \geq n^{4/3+\epsilon}$, for any $\epsilon > 0$. *ApproxLCP* is based on rounding a linear program for hitting all paths of length at most D , where D is a parameter (instead of all paths, as in the case of LCP). This LP has exponential size, and the separation oracle (which arises in the use of the Ellipsoid method [15]) corresponds to the Minimum Label s, t path (MinLP) problem [7], which is difficult to solve even approximately within a factor of $\Omega(2^{\log^{1-\epsilon} n})$ for some $\epsilon > 0$.

For the LCLC problem, we show that it cannot be approximated within a factor of $o(2^{\log^{1-1/\log \log^c n} n})$, for any $c < 1/2$, by extending the hardness result for LCP [16]. Using the product graph construction of Barrett et al. [4], we show that the LCLC problem can be reduced to LCP for regular languages. This leads to an $O(\min\{N\sqrt{m}, (nN)^{2/3}\})$ approximation algorithm for LCLC, where N denotes the number of states in a finite state machine for the language $\mathcal{L}$.

2. Preliminaries

Let $G = (V, E)$ denote a directed graph, and let $n = |V|$ and $m = |E|$. Let $s, t \in V$ be a given pair of nodes. Let L denote a set of edge labels. Let $\ell : E \rightarrow L$ denote an edge labeling function. For a path $P = e_1, \dots, e_k$ in G , $\ell(P) = \ell(e_1) \circ \dots \circ \ell(e_k)$ denotes the word formed by concatenating the labels on the edges on P ; here “ $\circ$ ” denotes the concatenation operator. Sometimes, we abuse the notation and also use $\ell(P)$ as the set of labels on P , whenever it is clear from the context. Given a regular language $\mathcal{L}$ defined on the alphabet L , we say that a path P is $\mathcal{L}$ -constrained (or $\mathcal{L}$ -respecting) if $\ell(P) \in \mathcal{L}$. For a pair of nodes s, t in G , let $\mathcal{P}(G, s, t)$ denote the set of all s - t paths in G , and let $\mathcal{P}(G, \mathcal{L}, s, t)$ denote the set of all $\mathcal{L}$ -respecting paths from s to t .

The Labeled Cut Problem (LCP) is to choose a smallest set $L' \subset L$ of labels so that no path in $\mathcal{P}(G, s, t)$ remains after deleting all the edges with labels in L' . The Language Constrained Minimum Cut (LCLC) problem is to choose the smallest subset $L' \subset L$ of labels so that the graph $G' = (V, E \setminus \{e : \ell(e) \in L'\})$, resulting from the deletion of edges with labels in L' , contains no $\mathcal{L}$ -constrained s - t paths (equivalently, $\mathcal{P}(G', \mathcal{L}, s, t) = \emptyset$).

Fig. 1 shows an example for both the LCP and LCLC problems. Note that the optimal solutions to these problems cost less than the solutions to the (unlabeled) minimum s, t -cut. In a graph with all edges having the same label, the LCP and LCLC solutions have cost 1, which can be arbitrarily smaller than the cost of the minimum unlabeled solution.

Download English Version:

<https://daneshyari.com/en/article/4952526>

Download Persian Version:

<https://daneshyari.com/article/4952526>

[Daneshyari.com](https://daneshyari.com)