Information Processing Letters 129 (2018) 31–34

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On maximum leaf trees and connections to connected maximum cut problems

Rajiv Gandhi [a,1], Mohammad Taghi Hajiaghayi [b,2], Guy Kortsarz [a,3], Manish Purohit [c], Kanthi Sarpatwar [d]

[a] *Rutgers University, Camden, NJ, United States*
[b] *University of Maryland, College Park, MD, United States*
[c] *Google, Mountain View, CA, United States*
[d] *IBM Research, Yorktown Heights, NY, United States*

ARTICLE INFO

Article history:
Received 1 December 2016
Received in revised form 11 April 2017
Accepted 5 June 2017
Available online 13 June 2017
Communicated by Marcin Pilipczuk

Keywords:
Approximation algorithms
Connected maximum cut
Maximum leaf trees

ABSTRACT

In an instance of the (directed) Max Leaf Tree (MLT) problem we are given a vertex-weighted (di)graph $G(V, E, w)$ and the goal is to compute a subtree with maximum weight on the leaves. The weighted Connected Max Cut (CMC) problem takes in an undirected edge-weighted graph $G(V, E, w)$ and seeks a subset $S \subseteq V$ such that the induced graph $G[S]$ is connected and $\sum_{e\in\delta(S)} w(e)$ is maximized.

We obtain a constant approximation algorithm for MLT when the weights are chosen from $\{0, 1\}$, which in turn implies a $\Omega(1/\log n)$ approximation for the general case. We show that the MLT and CMC problems are related and use the algorithm for MLT to improve the factor for CMC from $\Omega(1/\log^2 n)$ (Hajiaghayi et al., ESA 2015) to $\Omega(1/\log n)$.

1. Introduction

Given a vertex-weighted graph $G = (V, E, w)$, the Max Leaf Tree problem is to find a subtree T such that the total weight of the leaves of T is maximized. The closely related *Maximum Leaf Spanning Tree* problem seeks to find a *spanning* tree T of G that maximizes the sum of weights on the leaves of T. The Maximum Leaf Spanning Tree problem on unweighted graphs has been very well studied and constant factor approximation algorithms are known both for undirected [5,8,10,11,13] as well as directed graphs [3,4].

In the unweighted case, it is easy to observe that any subtree T can be augmented to a spanning tree T' without decreasing the number of leaves. Consequently, on unweighted graphs – both undirected and directed (in this case, we assume that all the nodes have a directed path from the root), the Max Leaf Tree problem is equivalent to the Maximum Leaf Spanning Tree problem. However, the two problems differ significantly in their weighted versions. In this paper, we study the approximability of the weighted Max Leaf Tree problem on both directed and undirected graphs.

Hajiaghayi et al. [6] initiated the study of the Connected Submodular Maximization problem: Given a graph $G = (V, E)$ and a non-negative submodular set function $f : 2^V \to \mathbb{R}^+ \cup \{0\}$, find a subset S of vertices that maximizes $f(S)$ such that the induced graph $G[S]$ is connected. It can be readily observed that the weighted Max Leaf Tree

E-mail addresses: rajivg@camden.rutgers.edu (R. Gandhi), hajiagha@cs.umd.edu (M.T. Hajiaghayi), guyk@camden.rutgers.edu (G. Kortsarz), mpurohit@google.com (M. Purohit), sarpatwa@us.ibm.com (K. Sarpatwar).

[1] Supported in Part by NSF Grant number CCF-1218620 and CCF-1433220.

[2] Supported in part by NSF CAREER award CCF-1053605, NSF BIG-DATA grant IIS-1546108, NSF AF:Medium grant CCF-1161365, DARPA GRAPHS/AFOSR grant FA9550-12-1-0423, and another DARPA SIMPLEX grant.

[3] Supported in Part by NSF Grant Number 1218620 and NSF Grant Number 1540547.

http://dx.doi.org/10.1016/j.ipl.2017.06.002
0020-0190/© 2017 Elsevier B.V. All rights reserved.

problem is a special case of the Connected Submodular Maximization problem where the submodular function is $f(S)=\sum_{u\in N(S)} w(u)$ where $N(S)$ refers to the set of all vertices that are not in S but have neighbors in S. Another important special case of the Connected Submodular Maximization problem is the Connected Maximum Cut (CMC) problem: Given an edge-weighted undirected graph $G=(V,E,w)$ find a set $S\subseteq V$ that maximizes the total weight of edges in the cut $\delta(S,V\setminus S)$ such that the induced graph $G[S]$ is connected. Hajiaghayi et al. [6] obtained the first $\Omega(\frac{1}{\log^2 n})$ approximation algorithm for the Connected Maximum Cut problem on weighted graphs.

1.1. Contribution and techniques

Our key results can be summarized as follows.

1. We obtain the first $\Omega(\frac{1}{\log n})$ approximation algorithm for the Max Leaf Tree problem with general weights on directed and undirected graphs.
2. We show that an α-approximation algorithm for the weighted Max Leaf Tree problem leads to an $\Omega(\alpha)$-approximation algorithm for the Connected Maximum Cut problem on general weighted graphs. Combined with the previous result, we obtain an $\Omega(\frac{1}{\log n})$-approximation, thus improving upon the $\Omega(\frac{1}{\log^2 n})$-approximation obtained by Hajiaghayi et al. [6].

1.2. Related work

Max Leaf Spanning Tree is a classical problem that has been very well-studied from the perspective of approximation algorithms [5,8,10,11,13] as well as fixed parameterized tractability [1,3]. As observed by Drescher and Vetta [4], the weighted Max Leaf Spanning Tree problem is as hard as the Independent Set problem and hence cannot be approximated within a factor of $\Omega(\frac{1}{n^{1-\epsilon}})$ unless P=NP. For unweighted directed graphs, Drescher and Vetta [4] obtained an $\Omega(\frac{1}{\sqrt{OPT}})$ approximation for the Max Leaf Spanning Tree problem. Daligault and Thomassé [3] discovered the first constant approximation algorithm for the same.

Hajiaghayi et al. [6] introduced the Connected Maximum Cut problem and gave an $\Omega(\frac{1}{\log n})$ approximation algorithm on unweighted graphs as well as an $\Omega(\frac{1}{\log^2 n})$ approximation algorithm on general weighted graphs. Lee, Nagarajan and Shen [9] studied a generalization of the connected maximum cut where connectivity and cut are defined by different graphs. Optimization of submodular functions over graphs subject to a connectivity constraint has been well studied in different contexts especially for monotone submodular functions [2,7].

2. Preliminaries

Problem definition (Directed) Weighted Maximum Leaf Tree : Given a graph $G=(V,E)$ and a weight function $w:V\to\mathbb{R}^+$, find a subtree T that maximizes $\sum_{v\in L(T)} w(v)$ where $L(T)$ denotes the set of leaves of the tree T. In the directed version, we are also given a root vertex r and the goal is to find an out-tree T rooted at r that maximizes $\sum_{v\in L(T)} w(v)$.

Problem definition Connected Max Cut : Given a graph $G=(V,E)$ and a weight function $w:E\to\mathbb{R}^+$, find a subset $S\subset V$ that maximizes $\sum_{e\in\delta(S,V\setminus S)} w(e)$ such that the subgraph induced by S is connected.

3. Approximation algorithms for general graphs

In this section, we consider the Weighted Maximum Leaf Tree on directed graphs and provide the first $\Omega(\frac{1}{\log n})$ approximation algorithm. Naturally all our results are applicable for undirected graphs as well. Our approach is to first reduce the problem to an instance of the *unweighted* maximum leaf tree problem in two stages and then use a constant factor approximation algorithm for the same.

3.1. Stage 1: reduction to binary weighted max leaf tree

We partition the vertices in V into $\log_2 n$ classes based on their weight as follows. Let OPT denote the total weight of leaves of an optimal weighted Max Leaf Tree and let $w_{max}=\max_{v\in V} w(v)$, $w_{min}=\min_{v\in V|w(v)\neq 0} w(v)$ denote the weights of the heaviest and lightest (non-zero) vertices in V respectively. Without loss of generality, we can assume that $OPT\geq w_{max}$ and hence we reset the weights of edges whose weights are less than $\frac{w_{max}}{2n}$ to 0. This alteration modifies OPT by at most an 1/2 fraction and we can now assume that $w_{min}>\frac{w_{max}}{2n}$. We define thresholds $\tau_0=\frac{w_{max}}{2n}$ and $\tau_i=\tau_0 2^i$ for $i\in[\log_2 2n]$. The vertices in V are grouped into $O(\log n)$ classes using these thresholds as follows - let $V_i=\{v\in V|\tau_{i-1}<w(v)\leq\tau_i\}$.

Let O denote an optimal solution for the Max Leaf Tree problem and let $OPT_i=\sum_{v\in V_i\cap L(O)} w(v)$ denote the contribution of vertices from V_i to the optimal solution. Let $i^*=\mathbf{arg\,max}_{i\in[\log_2 2n]} OPT_i$ denote the vertex class that has the highest contribution. We now set $w'(v)=0$ for all $v\notin V_{i^*}$ and set $w'(v)=1$ for all $v\in V_{i^*}$ to obtain a new instance $\mathcal{I}'$ of the Max Leaf Tree problem with only $\{0,1\}$ weights.

Lemma 3.1. *If there exists an α-approximation algorithm for the Max Leaf Tree problem with $\{0,1\}$ weights, then there exists an $\Omega(\frac{\alpha}{\log n})$-approximation algorithm for the Max Leaf Tree problem with general weight functions.*

Proof. Given an instance $\mathcal{I}=<G,w>$ of the general weighted Max Leaf Tree problem, we obtain an instance $\mathcal{I}'=<G,w'>$ of the Max Leaf Tree problem with $\{0,1\}$ weights using the procedure described above. By construction, we have that $OPT(\mathcal{I}')\geq\frac{OPT_{i^*}}{\tau_{i^*}}\geq\frac{OPT}{\tau_{i^*}\log_2 2n}$.

Let T be an α-approximate solution for the instance $\mathcal{I}'$. We now observe that the tree T is an $\Omega(\frac{\alpha}{\log n})$-approximate solution for the original instance $\mathcal{I}$ as well.

$$w(L(T))\geq\tau_{i^*-1}w'(L(T))\geq\tau_{i^*-1}\alpha OPT(\mathcal{I}')$$
$$\geq\frac{\tau_{i^*-1}\alpha OPT}{\tau_{i^*}\log_2 2n}\geq\frac{\alpha OPT}{2\log_2 2n}\qquad\square$$