Asia-Pacific Journal of Operational Research
Vol. 36, No. 2 (2019) 1940006 (9 pages)

DOI: 10.1142/S0217595919400062

Edge-Cuts of Optimal Average Weights

Scott Payne
Department of Mathematics, West Virginia University
Morgantown, WV 28506, USA
spayne7@mix.wvu.edu

Edgar Fuller
Department of Mathematics, Florida International University
Miami, FL 33199, USA
efuller@fiu.edu

Cun-Quan Zhang*
Department of Mathematics, West Virginia University
Morgantown, WV 28506, USA
cqzhang@mail.wvu.edu

Received 10 October 2018
Revised 25 March 2019
Accepted 25 March 2019
Published 3 May 2019

Let G be a directed graph associated with a weight $w : E(G) \to R^+$. For an edge-cut Q of G, the average weight of Q is denoted and defined as $w_{\text{ave}}(Q) = \frac{\sum_{e \in Q} w(e)}{|Q|}$. An optimal edge-cut with average weight is an edge-cut Q such that $w_{\text{ave}}(Q)$ is maximum among all edge-cuts (or minimum, symmetrically). In this paper, a polynomial algorithm for this problem is proposed for finding an optimal edge-cut in a rooted tree separating the root and the set of all leafs. This algorithm enables us to develop an automatic clustering method with more accurate detection of community output.

Keywords: Optimal edge-cut; average weight; algorithm; clustering; community selection.

1. Introduction

Max-Flow-Min-Cut is one of the oldest optimization problems in network theory (Ford and Fulkerson, 1956). It is also known that solving the max-cut problem is NP complete (Garey and Johnson, 1979; Karp, 1972). In this paper, we define a new optimization problem for finding edge-cuts in a special class of weighted directed graphs, specifically rooted weighted trees. These types of trees are typically used to represent relationships in hierarchical data structures. The minimum cut problem

*Corresponding author.

optimizes the sum of weights of edges in a given cut, over all cuts separating source and sink. Our new problem instead optimizes the average of the weights of edges in a given cut, over all root-separating cuts. Importantly, we propose a polynomial time algorithm for solving this optimization problem. Furthermore, the inequalities in the algorithm and proof may be adapted to solve either the minimization problem or the maximization problem. Here, we present the version for maximization, the version for minimization is similar and left to the reader.

1.1. *Applications for data mining*

In the development of clustering methods, one of the most challenging problems is identifying the level of association among data points, such as the vertices of a graph, that provides the proper output of communities. In the case of hierarchical clustering, this question becomes the determination of which collection of cuts along the tree in the hierarchical dendrogram will be selected to form the set of communities as the final output. It has been observed that "*There are no completely satisfactory algorithms that can be used for determining the number of population clusters for many types of cluster analysis*" (SAS Institute Inc., 2008).

In Qi *et al.* (2014), the minimum-cut approach was introduced for the automatic selection of the final output of communities. More recently, the authors have proposed AQCM, a parameter-free clustering method (Payne *et al.*, 2019) in which the community selection subprogram is further revised and improved by the application of the *optimal average cut method* which produces fine grained outputs and significantly improves the quality of earlier approaches. For the purpose of scientific completeness, this paper provides some mathematical supports of this newly developed method in Payne *et al.* (2019).

1.2. *Notation and definitions*

A rooted tree T is a directed graph whose underlying graph is a tree and, there is a given vertex v_0, called the root, such that, for every vertex $x \in V(T)$, the unique path of the tree from v_0 to x is a directed path.

Let T be a rooted, weighted tree with edge weight function $w : E(T) \to \mathbb{R}^+$ assigning a weight $w(e)$ to each edge $e \in T$. Let v_0 be the root of T. For any edge set $X \subseteq E(T)$, we set $w(X) := \sum_{e \in X} w(e)$ to be the sum of the weights of the edges in this subset. We denote by $E^+(v)$ the out edges of vertex v and $E^+(e)$ to refer to the out edges of the vertex that is the head of edge e when this notation is convenient. Set $\alpha_0 := \frac{w(E^+(v_0))}{|E^+(v_0)|}$. When we refer to edge cuts, we will usually use the symbol Q. *Furthermore*, all the edge cuts discussed are cuts separating the root v_0 from the set L of *leaf vertices* of T, we may refer to such cuts as *root-separating*. The algorithm presented here relies on the graph operation *edge contraction* which we denote using the standard notation T/e when we contract the edge e in the digraph T.

Definition 1. For $e \in E(T-L)$ we define the **contractibility of** e, denoted by $\lambda(e)$, with the following formula:

$$\lambda(e) = \frac{w(E^+(e)) - w(e)}{|E^+(e)| - 1}.$$

Definition 2. We say an edge e is *contractible* if $\lambda(e) > \alpha_0$.

Definition 3. The edge set $E(T-L)$ may be ordered $\lambda(e_1) \geq \lambda(e_2) \geq \cdots \geq \lambda(e_m)$. We may refer to this ordering as the *contractibility ordering*.

2. Optimization Problem and Algorithm

Input. A rooted weighted tree T with edge weight $w : E(T) \to \mathbb{R}^+$ and the root v_0 and set of leaves L.

Output. An edge cut Q of T separating the root v_0 and the set L of leaves such that $\frac{\sum_{e\in Q} w(e)}{|Q|}$ is maximum among all such edge cuts.

The algorithm is as follows:

Step 1. Determine

$$\alpha_0 = \frac{w(E^+(v_0))}{|E^+(v_0)|}.$$

Step 2. Sort the edges e_i of the $E(T-L)$ so that

$$\lambda(e_1) \geq \lambda(e_2) \geq \cdots \geq \lambda(e_m),$$

where $\lambda(e_i)$ is the **contractibility of the edge** e_i as in Definition 1.

Step 3. If $\lambda(e_1) > \alpha_0$ then

(i) Denote the *in edge* to e_1 by e^*. Contract $T \leftarrow T/e_1$, and
(ii) update λ value for e^*, or update α_0 if e_1 had no *in edge* (it was in $E^+(v_0)$), and
(iii) repeat Step 2.

If $\lambda(e_1) \leq \alpha_0$ then go to the END STEP.

END STEP. Output: $Q = E^+(v_0)$.

Remark. The output Q above is an edge set of the contracted graph resulting from the running of the algorithm, however Q is also a subset of the original set of edges input to the algorithm. It is in the context of the input graph T that the set Q is the solution to the optimization problem presented here.

Figures 1 and 2 illustrate an example of the output of this algorithm.

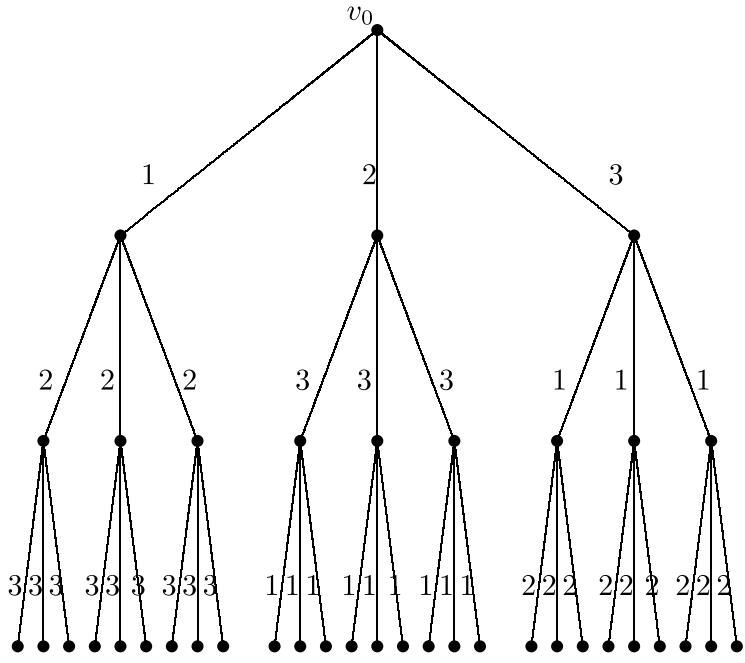

Fig. 1. The input: a weighted tree with the root v_0.

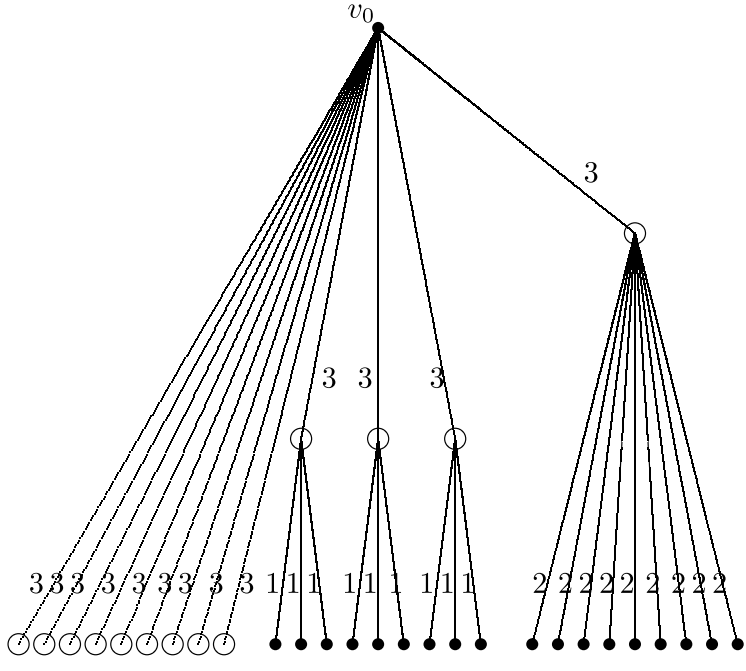

Fig. 2. The output (after contractions): $E^+(v_0)$ is the optimal average edge-cut.

In the next section, we will prove that the algorithm provides an optimal solution. That is, we prove the following theorem as our main result.

Theorem 1. *The output Q of the algorithm is an edge-cut of the input rooted tree T with the average weight*

$$\frac{\sum_{e\in Q} w(e)}{|Q|},$$

maximum among all edge-cuts separating the root v_0 and the set L of leaves.

3. Proof of Optimality

3.1. *Lemmas*

Before the proof of Theorem 1, we need the following lemmas.

Lemma 1. *Let Q be a root-separating cut of T and let $e \in Q$ but e is not a leaf-edge of T. If $\lambda(e) > \frac{w(Q)}{|Q|}$, then $\exists Q' \neq Q$ with $\frac{w(Q')}{|Q'|} > \frac{w(Q)}{|Q|}$. Specifically, $Q' = (Q\backslash\{e\}) \cup E^+(e)$.*

Proof. By the given conditions that $\lambda(e) > \frac{w(Q)}{|Q|}$, we have the following.

$$\frac{w(E^+(e)) - w(e)}{|E^+(e)| - 1} = \lambda(e) > \frac{w(Q)}{|Q|} = \frac{w(Q\backslash\{e\}) + w(e)}{|Q\backslash\{e\}| + 1}. \tag{1}$$

We will use the following classical inequality

$$\frac{a}{b} > \frac{c}{d} \Rightarrow \frac{a+c}{b+d} > \frac{c}{d}, \tag{2}$$

and define a, b, c, d as follows.

$$a = w(E^+(e)) - w(e),$$

$$b = |E^+(e)| - 1,$$

$$c = w(Q\backslash\{e\}) + w(e),$$

$$d = |Q\backslash\{e\}| + 1.$$

Then, with the above definitions, Inequality (1) is the LHS of the implication (2). The RHS of implication (2) is as follows.

$$\frac{w(Q\backslash\{e\}) + w(E^+(e))}{|Q\backslash\{e\}| + |E^+(e)|} > \frac{w(Q\backslash\{e\}) + w(e)}{|Q\backslash\{e\}| + 1} = \frac{w(Q)}{|Q|}. \tag{3}$$

Inequality (3) says that the cut $Q' = (Q\backslash\{e\}) \cup E^+(e)$ has the average weight *greater than* the average weight of Q. □

Definition 4. Let Q be a root-separating cut of T with $Q \neq E^+(v_0)$. Let H be the component of $T - Q$ such that the root $v_0 \in V(H)$. For terminology, we will refer

to H as *the subtree (of T) internal to Q*, or we may say *H is the internal subtree of Q*.

Lemma 2. *Let $Q \neq E^+(v_0)$ be a root-separating cut of T and let H be the subtree internal to Q. Let $e \in H$ be a leaf edge of H. Then at least one of the following holds*:

(i) $\lambda(e) > \frac{w(Q)}{|Q|}$

(ii) $\exists\, Q' \neq Q$ *with internal subtree H' such that* $|E(H')| < |E(H)|$ *and* $\frac{w(Q')}{|Q'|} \geq \frac{w(Q)}{|Q|}$. *Specifically,* $Q' = (Q \backslash E^+(e)) \cup \{e\}$.

Proof. Suppose (**i**) is not true. That is, $\frac{w(Q)}{|Q|} \geq \lambda(e)$. Thus, we have the following:

$$\frac{w(Q\backslash E^+(e)) + w(E^+(e))}{|Q\backslash E^+(e)| + |E^+(e)|} = \frac{w(Q)}{|Q|} \geq \lambda(e) = \frac{w(E^+(e)) - w(e)}{|E^+(e)| - 1}. \tag{4}$$

We will use the following classical inequality.

$$\frac{a+c}{b+d} \geq \frac{c}{d} \Rightarrow \frac{a}{b} \geq \frac{a+c}{b+d}. \tag{5}$$

Define a, b, c, d as follows:

$$a = w(Q\backslash E^+(e)) + w(e),$$

$$b = |Q\backslash E^+(e)| + 1,$$

$$c = w(E^+(e)) - w(e),$$

$$d = |E^+(e)| - 1.$$

Then with the above definitions, inequality (4) is the LHS of the implication (5). The RHS of implication (5) is as follows:

$$\frac{w(Q\backslash E^+(e)) + w(e)}{|Q\backslash E^+(e)| + 1} \geq \frac{w(Q\backslash E^+(e)) + w(E^+(e))}{|Q\backslash E^+(e)| + |E^+(e)|} = \frac{w(Q)}{|Q|}. \tag{6}$$

Inequality (6) says that the cut $Q' = (Q\backslash E^+(e)) \cup \{e\}$ has the average weight *at least* the average weight of Q. Clearly if H' is the internal subtree of Q', then $|E(H')| < |E(H)|$. So condition **(ii)** holds as desired. □

Lemma 3. *Let e_1 be the maximum edge in the contractibility ordering of $E(T-L)$. Assume that $\lambda(e_1) > \alpha_0$. Then an optimal solution of $T' = T/e_1$ is also an optimal solution of T.*

Proof. It is sufficient to show that *there exists* an edge cut Q_0 of T which achieves the maximum average weight among all root-separating cuts of T and $e_1 \notin Q_0$.

Let Q_0 be an optimal average weight cut in T. Assume $e_1 \in Q_0$. Observe that if $\frac{w(Q_0)}{|Q_0|} = \alpha_0$, then $\lambda(e_1) > \frac{w(Q_0)}{|Q_0|}$ and by Lemma 1, we may define $Q' = (Q_0\backslash\{e_1\}) \cup$

$E^+(e_1)$ and we have $\frac{w(Q')}{|Q'|} > \frac{w(Q_0)}{|Q_0|}$, a contradiction to the optimality of Q_0. So $\frac{w(Q_0)}{|Q_0|} > \alpha_0$ must be true, and $Q_0 \neq E^+(v_0)$, and also

$$\frac{w(Q_0)}{|Q_0|} \geq \lambda(e_1). \tag{7}$$

Among all optimal average weight cuts Q_0 satisfying the above, we may assume Q_0 has the smallest possible internal subtree H_0. Note that $H_0 \neq \emptyset$. Then by Lemma 2, there is a leaf edge $e \in H_0$ with

$$\lambda(e) > \frac{w(Q_0)}{|Q_0|}. \tag{8}$$

But since $e \neq e_1$, we have $\lambda(e) > \lambda(e_1)$ (by Inequalities (7) and (8)), a contradiction to the definition of e_1 as the maximum edge in the contractibility ordering. So $e_1 \notin Q_0$ as desired. □

3.2. *Proof of Theorem* 1

By Lemma 3, it is sufficient to show that, after all possible contractions of edges, $E^+(v_0)$ is an optimal solution in the resulting tree.

Suppose on the contrary that the output of the algorithm is not the proposed cut. Let Q be an optimal cut with an internal subtree H with $|H|$ as small as possible. Suppose that $Q \neq E^+(v_0)$. That is,

$$\frac{w(Q)}{|Q|} > \frac{w(E^+(v_0))}{|E^+(v_0)|} = \alpha_0. \tag{9}$$

Apply Lemma 2 here. The case (ii) of Lemma 2 does not occur since the cut Q is optimal. Hence the case (i) implies the existence of a leaf e in the internal subtree H with

$$\lambda(e) > \frac{w(Q)}{|Q|}. \tag{10}$$

By Inequalities (9) and (10), the leaf e is a contractible edge. Moreover, the maximum edge in the contractibility ordering is contractible, contradicting the assumption that the algorithm terminated.

4. Conclusions and Remarks

4.1. *Computational complexity of algorithm*

In addition, we find that the computational complexity of the algorithm is polynomial and so its implementation will be efficient and scalable to large datasets. Let $|V(T-L)| = n$. The cost of step 1 is a constant. Steps 2 and 3 are repeated at most n times. In the first loop of repeating, the cost of step 2 is $O(n(\log_2 n))$ or $O(n^2)$ since the sorting of λ's is involved, while the cost of step 2 in each loop after the first sorting is at most $O(n)$ since we only need to place one item $\lambda(e^*)$ to a proper

position in a well-sorted sequence. The cost of step 3 in every loop is at most $O(n)$ since only updating of the graph and the revising the weight λ are involved. Hence, the cost of the worst case has an upper bound of $O(n^2) + n(O(n) + O(n)) = O(n^2)$.

4.2. *Conjectures and open problems*

Problem 1. Let G be a network with the source s and the sink t and associated with a weight $w : E(G) \to R^+$. Find an edge-cut T separating t from s such that the average weight of T is maximum among all such edge-cuts.

The problems of finding cuts with total weights maximized or minimized have been well studied. One is known as an NP-complete problem, while another is polynomial (Bondy *et al.*, 1976; Cormen *et al.*, 2009; Goldberg and Tarjan, 1988). But few study have been done yet for finding an edge-cut with the average weight maximized (or minimized). We would like to propose the following conjecture.

Conjecture 1. Problem 1 is an NP-complete problem.

Acknowledgment

This research project has been partially supported by an National Science Foundation Grant No. DMS-1700218.

References

Bondy, JA, USR Murty *et al.* (1976). *Graph Theory with Applications*, Vol. 290. Citeseer.

Cormen, TH, CE Leiserson, RL Rivest and C Stein (2009). *Introduction to Algorithms*. MIT Press.

Ford, LR and DR Fulkerson (1956). Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3), 399–404.

Garey, MR and DS Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Vol. 29. New York: WH Freeman.

Goldberg, AV and RE Tarjan (1988). A new approach to the maximum-flow problem. *Journal of the ACM* (*JACM*), 35(4), 921–940.

Karp, RM (1972). Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pp. 85–103. Springer.

Payne, S, E Fuller and C-Q Zhang (2019). Automatic quasi clique merge (aqcm) algorithm — a parameter free clustering method. *Preprint*.

Qi, X, W Tang, Y Wu, G Guo, E Fuller and C-Q Zhang (2014). Optimal local community detection in social networks based on density drop of subgraphs. *Pattern Recognition Letters*, 36, 46–53.

SAS Institute Inc. (2008). SAS/STAT© 9.2 User's Guide.

Biography

Scott Payne is a research associate for Data Mining Team at West Virginia University. He received his BS (2011) and MSc (2014) from West Virginia University.

Edgar Fuller is Distinguished University Professor of Mathematics and Director of the Center for Transforming Teaching in Mathematics at Florida International University. He served as Chair of the Department of Mathematics at West Virginia University from 2008–2018. He currently works on applications of topology and geometry to the study of community detection in complex networks and machine learning algorithms. He was the co-recipient of the Tyco Brahe award from the NASA Office of Safety and Mission Assurance in 2005. His other interests include applications of computational geometry to knot theory and mathematics education. He currently is co-Principal Investigator for an NSF funded project developing active learning curricula in calculus. He received his Ph.D. from the University of Georgia in the area of differential geometry and topology.

Cun-Quan Zhang is the Eberly Family Distinguished Professor of Mathematics at West Virginia University. His major research areas are graph theory and its applications in data mining. He was the PI of 13 grants from National Science Foundation, National Security Agency, etc. and a co-PI of an NIH grant. He did not have undergraduate education due to Cultural Revolution in China. He received his MSc (1981) from Qufu Normal University, China; and his PhD (1986) from Simon Fraser University, Canada.