A note on the uniform edge-partition of a tree [1]

Bang Ye Wu [a] Hung-Lung Wang [b] Shih Ta Kuan [a]
Kun-Mao Chao [b,c]

[a] *Dept. of Computer Science and Information Engineering, Shu-Te University, YenChau, KaoShiung, Taiwan 824, R.O.C.*

[b] *Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan 106, R.O.C.*

[c] *Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan 106, R.O.C.*

Abstract

We study the problem of uniformly partitioning the edge set of a tree with n edges into k connected components, where $k \leq n$. The objective is to minimize the ratio of the maximum to the minimum number of edges of the subgraphs in the partition. We show that, for any tree and $k \leq 4$, there exists a k-split with ratio at most two. (Proofs for $k = 3$ and $k = 4$ are omitted here.) For general k, we propose a simple algorithm that finds a k-split with ratio at most three in $O(n \log k)$ time. Experimental results on random trees are also shown.

Key words: tree, partition, optimization problem, algorithm.

Email addresses: `bangye@mail.stu.edu.tw` (Bang Ye Wu), `r92085@csie.ntu.edu.tw` (Hung-Lung Wang), `s91113111@mail.student.stu.edu.tw` (Shih Ta Kuan), `kmchao@csie.ntu.edu.tw` (Kun-Mao Chao).

[1] This is an excerpt from the paper "on the uniform edge-partition of a tree" by Wu, Wang, Kuan, and Chao. A full version can be found in *Discrete Applied Mathematics*.

1 Introduction

Graph partition is an important problem in computer science. It finds applications in parallel computing, data storage and segmentation, and operation research. Most of the previous research was devoted to the *vertex partition*, and many variants of the problem have been defined and investigated with different objectives and constraints. To measure how uniform a partition is, three natural objectives are usually used.

- To minimize the maximum (min-max).
- To maximize the minimum (max-min).
- To minimize the ratio of the maximum to the minimum (min-ratio).

Many problems in this line of investigation have been shown to be NP-hard [1,10]. For the vertex partition of a tree, polynomial time algorithms for both the min-max and the max-min objectives were developed [5,7,15,17]. Becker and Perl [6] summarized their previous results with some other co-authors and showed that the tree vertex partition problem of several other objective functions can also be solved by using the shifting algorithm. An open problem in that paper is the *most uniform vertex partitioning* problem for trees, in which the objective is to minimize the difference between the maximum and the minimum weights of the vertex set in the partition. For a special case that the tree is a path, a solution was given in [16]. One can image that the problem is more difficult than the min-max or max-min problem since both the smallest and the largest parts are concerned.

In this paper, we study the problem of splitting a tree into k parts with approximately equal number of *edges* in each part subject to that the edges in each part are connected. How well can one do it?

More formally, we define a k-split of a tree T as follows. Let T be a tree and $1 \leq k \leq e(T)$. A k-tuple $(T_1, T_2, \ldots, T_k)$ is a k-split of T if (1) each T_i is a connected subgraph of T; and (2) T_i and T_j are edge disjoint for $i \neq j$; and (3) the union of all the subgraphs forms the whole tree T.

......

2 Notations and Preliminaries

Let $E(T)$ denote the edge set of a tree T and $e(T)$ denote the number of edges of tree T. Throughout this paper, $n = e(T)$. An edge with endpoints u and v is denoted by (u, v). Let T be a rooted tree and v be a vertex of T. We use T_v to denote the subtree rooted at v, i.e. the subgraph induced on v and all its descendants. Let u be a child of v. The subgraph $T_u \cup (u, v)$ is called a *branch* of v.

Definition 1: Let T be a tree and $1 \leq k \leq e(T)$. The ratio of a k-split $(T_1, T_2, \ldots, T_k)$ of T is defined by $\frac{\max_i\{e(T_i)\}}{\min_i\{e(T_i)\}}$.

By $T = A \uplus B$, we denote that T is split into A and B, i.e., the edge sets of the two subgraphs form a partition of $E(T)$. It is also noted that A and B share a common vertex if $T = A \uplus B$. By $T = A \uplus B \uplus C$, we understand a 3-split (A, B, C) of T, in which B intersects with both A and C. It includes the case that the three subgraphs share a common vertex.

PROBLEM: Minimum Ratio k-Split

INSTANCE: A tree T and an integer $1 < k < e(T)$.

GOAL: Find a k-split of T with minimum ratio.

Theorem 1: The Minimum Ratio k-Split problem is NP-hard.

Proof: Omitted. $\square$

Lemma 2: Let T be a rooted tree. For any $1 \leq \gamma \leq e(T)$, we can split T into (T_1, T_2) at a vertex v in linear time such that $\gamma \leq e(T_1) \leq 2\gamma$, in which v is a vertex satisfying $e(T_v) \geq \gamma$ and $e(T_u) < \gamma$ for any child u of v.

Proof: In linear time, we can traverse the tree in the post order and compute the number of edges for the subtree rooted at each vertex. Such a vertex v can be easily found while traversing the tree. Assume that $B_1, B_2, \ldots, B_k$ are the branches at v. If $e(T_v) = \gamma$, we are done. Otherwise, we can find $j \leq k$ such that $\sum_{i=1}^{j-1} e(B_i) < \gamma$ and $\sum_{i=1}^{j} e(B_i) \geq \gamma$. Since $e(B_j) \leq \gamma$, we have that $\sum_{i=1}^{j} e(B_i) \leq 2\gamma$. The union $\bigcup_{i=1}^{j} B_i$ is the desired subgraph. $\square$

Taking $\gamma = n/3$ in Lemma 2, we have the following result.

Corollary 3: For any tree T, there is a 2-split of T with ratio at most two. The numbers of the two subgraphs are at most $2n/3$ and at least $n/3$. Furthermore, such a 2-split can be found in $O(n)$ time.

One can further shows that the bounds are tight.

The following simple result shows an upper and a lower bounds for the sizes of the subgraphs in a k-split with a limited ratio.

Lemma 4: If $(T_1, T_2, \ldots, T_k)$ is a k-split of T with ratio r, then, for each subgraph T_i, $\frac{n}{r(k-1)+1} \leq e(T_i) \leq \frac{rn}{k+r-1}$.

Proof: Let x be the number of edges of the maximum component. Since the number of edges of the minimum component is no more than the mean of the remainder, i.e., $\frac{n-x}{k-1}$,

$$x \leq \frac{r(n-x)}{k-1}.$$

Solving the inequality, we have $x \leq \frac{rn}{k+r-1}$. Similarly, let y denote the minimum number of edges. The maximum is no less than the mean of the remainder, $\frac{n-y}{k-1}$, and we have $y \geq \frac{(n-y)}{r(k-1)}$, which implies $y \geq \frac{n}{r(k-1)+1}$. $\square$

3 On general k

3.1 A simple algorithm

We now propose a simple algorithm which finds a k-split of a tree with ratio at most three. Given a tree T and an integer k, the algorithm starts at the 1-split (T) and repeatedly computes a $(i+1)$-split from the i-split by 2-splitting the maximum subgraph. The time complexity of this algorithm is $O(n \log k)$.

Algorithm Simple-Split

Input: A tree T and an integer $k \leq e(T)$.

Output: A k-split of T.

1: Initiate an empty queue Q of trees, and insert T into Q.

2: For $i \leftarrow 1$ to $k-1$ do

2.1: Choose a tree Y in Q with maximum number of edges.

2.2: Find a 2-split (Y_1, Y_2) of Y with ratio at most two.

2.3: Remove Y from Q.

2.4: Insert Y_1 and Y_2 into Q.

3: Output the k trees in Q as the k-split of T.

In the next theorem, we show the performance of the algorithm.

Theorem 5: Given a tree T with n edges and an integer $k \leq n$, the algorithm **Simple-Split** finds a k-split of T with ratio at most 3 in $O(n \log k)$ time.

Proof: Let M_i and m_i be respectively the maximum and minimum numbers of edges of trees in the queue Q at i-th iteration. We first claim that the ratio M_i/m_i is at most 3 for each i. Initially Q contains only the input tree T, and $M_1/m_1 = 1$. Suppose that $M_i/m_i \leq 3$ for some i. We shall show that $M_{i+1}/m_{i+1} \leq 3$, and then the above claim is consequently true by induction. At $(i+1)$-th iteration, the maximum tree Y is chosen and split into Y_1 and Y_2 with ratio at most 2. Therefore, $M_{i+1} \leq M_i$, and $m_{i+1} = \min\{m_i, e(Y_1), e(Y_2)\}$. Since $\min\{e(Y_1), e(Y_2)\} \geq e(Y)/3 = M_i/3$ and $M_i/m_i \leq 3$, we have

$$\frac{M_{i+1}}{m_{i+1}} \leq \frac{M_i}{M_i/3} = 3.$$

Next, we turn to the time complexity. Let $f_n(i)$ be the total time complexity of executing Step 2.2 in the first i iterations. By Corollary 3, splitting a tree of M_i edges at i-th iteration takes $O(M_i)$ time. Since the ratio M_i/m_i is at most three, by Lemma 4, we have

$$M_i \leq \frac{3n}{i+2}.$$

Therefore, for some constant c, $f_n(1) \leq cn$, and

$$f_n(i) \leq f_n(i-1) + c\frac{3n}{i+2}$$

for $i > 1$. Solving the recurrence relation, we have

$$\begin{aligned} f_n(k) &\leq c\sum_{i=1}^{k} \frac{3n}{i+2} \\ &< 3cn\sum_{i=1}^{k} \frac{1}{i} = 3cnH_k, \end{aligned}$$

in which H_k is the well-known k-th *harmonic number*. Since $H_k = O(\log n)$, we obtain $f_n(k) = O(n \log k)$.

For Step 2.1, 2.3, and 2.4, by simply using a data structure like *heap* to store the numbers of edges of the trees in the queue, all the operations can be done in totally $O(k \log k)$ time. Therefore the total time complexity is $O(n \log k)$. $\square$

4 Concluding Remarks

One of the most important open problems in this line of investigation is that whether there exists a k-split with ratio at most two for general k. Our future work includes exact and approximation algorithms for finding the min-ratio k-split for general or fixed k.

Acknowledgments

......

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela and M. Protasi, *Complexity and Approximation — Combinatorial optimization problems and their approximability properties*, Springer Verlag, 1999.

[2] I. Averbakh and O. Berman, A heuristic with worst-case analysis for minimax routing of two traveling salesmen on a tree, *Discrete Appl. Math.*, 68:17–32, 1996.

[3] I. Averbakh and O. Berman, $(p-1)/(p+1)$-approximate algorithms for p-traveling salesmen problem on a tree with minmax objective, *Discrete Appl. Math.*, 75:201–216, 1997.

[4] V. Bafna, E. L. Lawler and P. Pevzner, Approximation algorithms for multiple sequence alignment, *Theoretical Computer Science*, 182:233–244, 1997.

[5] R. Becker and Y. Perl, Shifting algorithms for tree partitioning with general weighting functions, *J. Algorithms*, 4:101–120, 1983.

[6] R. Becker and Y. Perl, The shifting algorithm technique for the partitioning of trees, *Discrete Appl. Math.*, 62:15–34, 1995.

[7] R. Becker, S.R. Schach and Y. Perl, A shifting algorithm for min-max tree partitioning, *J. ACM*, 29:58–67, 1982.

[8] R. Becker, B. Simeone and Y.-I Chiang, A shifting algorithm for continuous tree partitioning, *Theor. Comput. Sci.*, 282:353–380, 2002.

[9] G.J. Chang and F.K. Hwang, Optimality of consecutive and nested tree partitions, *Networks*, 30:75–80, 1997.

[10] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, San Francisco, 1979.

[11] D. Gusfield, Efficient methods for multiple sequence alignment with guaranteed error bounds, *Bulletin of Mathematical Biology*, 55:141–154, 1993.

[12] D. Gusfield, *Algorithms on Strings, Trees, and Sequences – Computer Science and Computational Biology*, Cambridge University Press, 1997.

[13] C.M. Huang, B.Y. Wu and C.B. Yang, Tree edge decomposition with an application to minimum ultrametric tree approximation, unpublished manuscript.

[14] I. Krasikov, On a tree cutting problem of P. Ash, *Discrete Math.*, 93:51–61, 1991.

[15] S. Kundu and J. Misra, A linear tree partitioning algorithm, *SIAM J. Comput*, 6:151–154, 1977.

[16] M. Lucertini, Y. Perl and B. Simeone, Most uniform path partitioning and its use in image processing, *Discrete Appl. Math.* 42:227-256, 1993.

[17] Y. Perl and S.R. Schach, Max-min tree partitioning, *J. ACM*, 28:5–15, 1981.

[18] P. Pevzner, Multiple alignment, communication cost, and graph matching, *SIAM J. Appl. Math.*, 52:1763–1779, 1992.

[19] B.Y. Wu and K.-M. Chao, *Spanning Trees and Optimization Problems*, Chapman & Hall / CRC Press, 2004.

[20] B.Y. Wu, G. Lancia, V. Bafna, K.-M. Chao, R. Ravi and C.Y. Tang, A polynomial time approximation scheme for minimum routing cost spanning trees, *SIAM J. Comput.*, 29:761–778, 2000.