Uniform edge-partition of a tree

Kun-Mao Chao
@2019
@2021

T: a tree

$E(T)$: the edgeset of T

$e(T)$: the number of edges of T

$\overset{\shortparallel}{n}$ $\quad 1 \le k \le e(T)$

A k-tuple $(T_1, T_2, \cdots, T_k)$ is a k-split of T if

(1) each T_i is a connected subgraph of T;

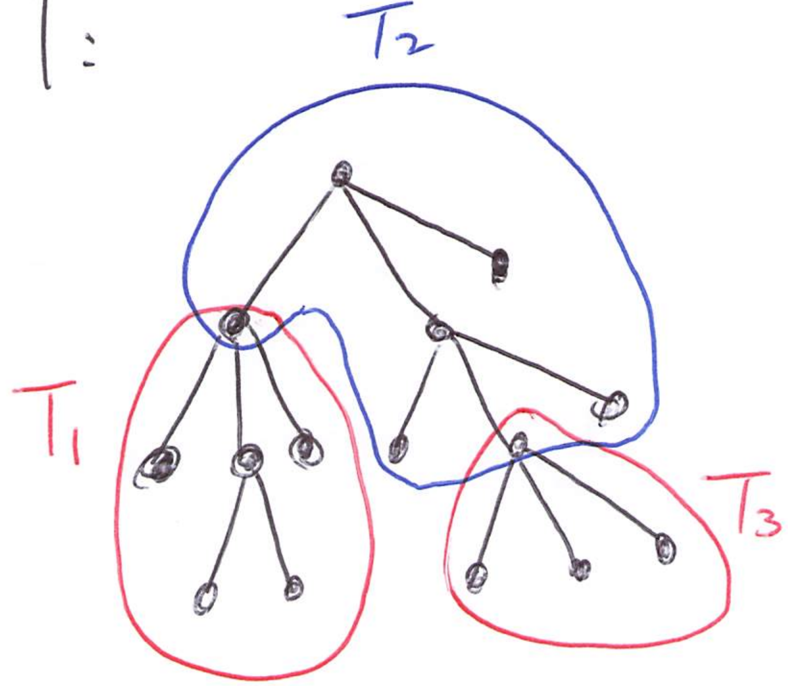
(2) T_i and T_j are edge disjoint for $T_i \neq T_j$;

(3) the union $T_1 \cup T_2 \cup \cdots \cup T_k = T$

The ratio of a k-split $(T_1, T_2, \cdots, T_k)$ of T is defined as $\dfrac{\max_i \{e(T_i)\}}{\min_i \{e(T_i)\}}$.

Kun-Mao Chao

$T:$



3-split

$T_1 \cup T_2 \cup T_3 = T$

$e(T_1) = 5$

$e(T_2) = 6$ ← max

$e(T_3) = 3$ ← min

The ratio is $\frac{e(T_2)}{e(T_3)} = 2$.

Kun-Mao Chao
@2019
2021

$1 \le r \le e(T)$

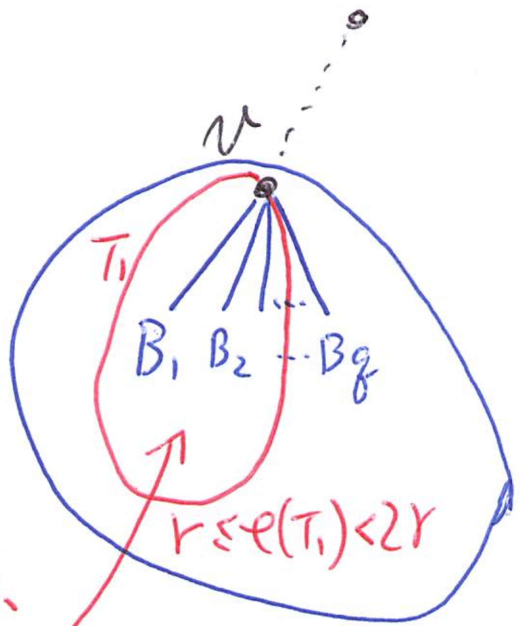
v: $e(T_v) \ge r$ & $e(T_u) < r$ for any child u.

2-split (T_1, T_2)

$\Downarrow$

$r \le e(T_1) < 2r$

T:



$e(T_v) \ge r$

$e(B_1) \le r$

$e(B_2) \le r$

$\vdots$

$e(B_q) \le r$

Find $j \le q$ s.t.

$\sum_{i=1}^{j-1} e(B_i) < r$

$\sum_{i=1}^{j} e(B_i) \ge r$

$\parallel$

$\underbrace{\sum_{i=1}^{j-1} e(B_i)}_{<r} + \underbrace{e(B_j)}_{\le r} < 2r$

post-order traversal in linear time

3

Kun-Mao Chao@2019

Take $\gamma = \frac{n}{3}$

$$\frac{n}{3} \le e(T_1) < \frac{2n}{3}$$

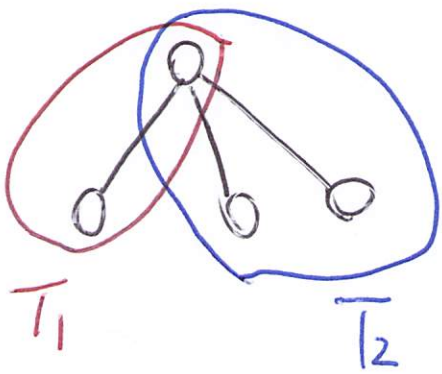
$$\Rightarrow \frac{n}{3} < e(T_2) \le \frac{2n}{3}$$

$\le \frac{2}{3}n$

$$\frac{\max\{e(T_1), e(T_2)\}}{\min\{e(T_1), e(T_2)\}} \le 2$$

$\ge \frac{1}{3}n$

Therefore, for any tree T, there is a 2-split of T with ratio at most 2.



Kun-Mao Chao @2019

Algorithm Simple-Split

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

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.

Q: (T^1)

(T_1^2, T_2^2)

(T_1^3, T_2^3, T_3^3)

$\vdots$ 2-splitting the maximum T_i

$(T_1^k, T_2^k, \cdots, T_k^k)$

Kun-Mao Chao

The simple-split algorithm @2019

starts at the 1-split(T) and

repeatedly computes the (i+1)-split

from the i-split by 2-splitting the

maximum subgraph.

The resulting k-split is with

ratio at most 3.

M_i : the maximum number of edges at the i-th iteration

m_i : minimum "

Initially

$$\boxed{\frac{M_1}{m_1} = 1}$$

Suppose that $\frac{M_i}{m_i} \le 3$ for some i.

$$M_{i+1} \le M_i \leftarrow Y <^{Y_1}_{Y_2}$$

$$m_{i+1} = \min\{\underset{\ge \frac{M_i}{3}}{m_i}, \underset{\ge \frac{M_i}{3}}{e(Y_1)}, \underset{\ge \frac{M_i}{3}}{e(Y_2)}\} \Rightarrow \frac{M_{i+1}}{m_{i+1}} \le \frac{M_i}{\frac{M_i}{3}} = 3$$

Kun-Mao Chao @2019

The ith iteration takes $O(M_i)$ time.

$$M_i \le 3\underset{\sim}{m_i} \le 3 \cdot \underbrace{\frac{(n-M_i)}{i-1}}$$

Average of the rest

$$(T_1^i, T_2^i, \cdots, T_i^i)$$

$$\Rightarrow M_i \le \frac{3n}{i+2}$$

$$c\left(\frac{3n}{1+2} + \frac{3n}{2+2} + \frac{3n}{3+2} + \cdots + \frac{3n}{k+2}\right)$$

$$= 3cn\left(\frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \cdots + \frac{1}{k+2}\right)$$

$$< 3cn \underbrace{\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}\right)}_{O(\log k)}$$

$O(n \log k)$ time.