Approximating the Held-Karp Bound for Metric TSP in Nearly Linear Work and Polylogarithmic Depth

Sorrachai Yingchareonthawornchai (ETH Zürich)

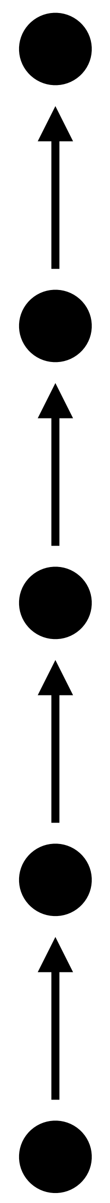
Joint work with
Zhuan Khye Koh (Boston University)
Omri Weinstein (Hebrew University)

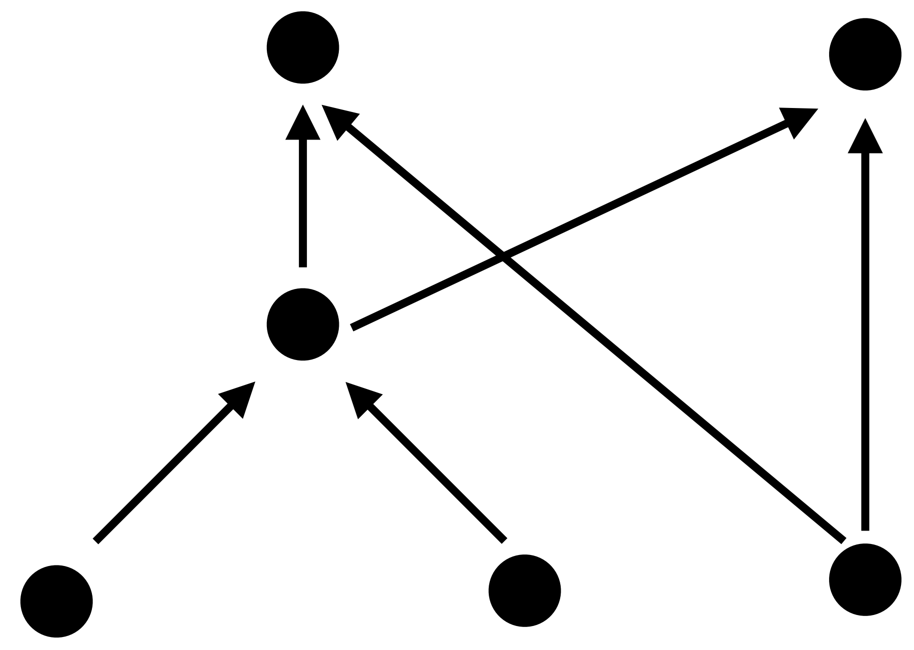
Data Structure and Optimization Reunion Workshop

20 March 2025

Work-Depth Model

- **Work** = total number of operations
- **Depth** = the length of a longest chain of dependent operations
- Fast parallel algorithm = nearly linear work and polylog depth





- **Metric TSP**
  - **Input**: a complete graph graph G = (V,E,c) where $\forall u, v, w \in V, c_{uv} \leq c_{uw} + c_{wv}$
  - **Output**: a min-cost Hamiltonian cycle
  - **Implicit input**: the instance is implicitly defined as the metric completion of the underlying graph G = (V,E,c)
  - **Output:** a min-cost Eulerian multigraph of G
- **APX-hard** [Lampis'12]
- 1.5-approximation algorithm by [Christofide'76]
- 1.5 - 10^{-36} approximation by [Karlin, Klein, Gharan'22]

Subtour Elimination LP

$$
\begin{aligned}
\mathrm{SE}(\hat{G}, \hat{c}) = \min \sum_{u,v} \hat{c}_{\{u,v\}} y_{\{u,v\}} & \\
\text{s.t.} \sum_{u} y_{\{u,v\}} = 2 & \qquad \forall v \in V \\
\sum_{u \in S, v \notin S} y_{\{u,v\}} \geq 2 & \qquad \forall \emptyset \subsetneq S \subsetneq V \\
0 \leq y_{\{u,v\}} \leq 1 & \qquad \forall u, v \in V.
\end{aligned}
$$

The optimal value of the SE coincides with the Held-Karp bound

The Held-Karp bound is defined based on the notion of 1-trees [Held and Karp'70]

The integrality gap of SE is conjectured to be 4/3 [Goemans'95]

k-ECSM: Given an undirected graph $G = (V, E)$ with n nodes, m edges and edge costs $c \in \mathbb{R}^m_{>0}$, find a minimum cost k-edge-connected spanning multi-subgraph.

- LP relaxation:

$$\begin{aligned} \min \; & c^\top x \\ \text{s.t.} \; & \sum_{e \in \delta(S)} x_e \geq k & \forall \emptyset \subsetneq S \subsetneq V \\ & x_e \geq 0 & \forall e \in E \end{aligned}$$

Fact: Held-Karp Bound = LP value of SE = LP value of 2ECSM

Cunningham [via Monma, Munson, and Pulleyblank, 1990] and Goemans and Bertsimas [GB93]:

k-ECSM: Given an undirected graph $G = (V, E)$ with n nodes, m edges and edge costs $c \in \mathbb{R}^m_{>0}$, find a minimum cost k-edge-connected spanning multi-subgraph.

- LP relaxation:

$$\begin{aligned} \min\ & c^\top x \\ \text{s.t.} \sum_{e \in \delta(S)} & x_e \geq k \qquad \forall \emptyset \subsetneq S \subsetneq V \\ & x_e \geq 0 \qquad \forall e \in E \end{aligned}$$

- Covering LP with $\Omega(2^n)$ constraints.

Goal: Compute a $(1 + \varepsilon)$-approximate LP solution in nearly linear work and polylog depth.

*Slides from Zhuan Khye Koh

Multiplicative Weights Update (MWU)

- Many MWU variants [Luby Nisan '93] [Plotkin Shmoys Tardos '95] [Garg Könemann '07] [Fleischer '00] [Young '01] [Young '14] [Allen-Zhu Orrechia '15] [Mahoney Rao Wang Zhang '16] ...

MWU "compiles" kECSM into a sequence of mincut problems

*Slides from Zhuan Khye Koh

Multiplicative Weights Update (MWU)

- Many MWU variants [Luby Nisan '93] [Plotkin Shmoys Tardos '95] [Garg Könemann '07] [Fleischer '00] [Young '01] [Young '14] [Allen-Zhu Orrechia '15] [Mahoney Rao Wang Zhang '16] ...

- We consider epoch-based MWU.

- In iteration t, let $w^{(t)} \in \mathbb{R}^m_{\geq 0}$ be the edge weights. Given a lower bound λ on the mincut and $\varepsilon > 0$, define

$$\mathcal{C}^{(t)} := \{C \text{ cut } : w^{(t)}(C) < (1+\varepsilon)\lambda\}.$$

*Slides from Zhuan Khye Koh

Multiplicative Weights Update (MWU)

- Many MWU variants [Luby Nisan '93] [Plotkin Shmoys Tardos '95] [Garg Könemann '07] [Fleischer '00] [Young '01] [Young '14] [Allen-Zhu Orrechia '15] [Mahoney Rao Wang Zhang '16] ...

- We consider epoch-based MWU.

- In iteration t, let $w^{(t)} \in \mathbb{R}^m_{\geq 0}$ be the edge weights. Given a lower bound λ on the mincut and $\varepsilon > 0$, define

$$\mathcal{C}^{(t)} := \{C \text{ cut } : w^{(t)}(C) < (1+\varepsilon)\lambda\}.$$

$$\left.\begin{array}{l} \textbf{While } \mathcal{C}^{(t)} \neq \emptyset\text{:} \\ \quad \text{1. Select cut(s) from } \mathcal{C}^{(t)}. \\ \quad \text{2. Multiplicatively increase } w^{(t)} \text{ along these cuts.} \end{array}\right\} \text{an epoch}$$

- $\lambda \leftarrow \lambda(1+\varepsilon)$ and a new epoch begins.

*Slides from Zhuan Khye Koh

Multiplicative Weights Update (MWU)

While $\mathcal{C}^{(t)} \neq \emptyset$:

1. Select cut(s) from $\mathcal{C}^{(t)}$.
2. Multiplicatively increase $w^{(t)}$ along these cuts.

*Slides from Zhuan Khye Koh

Multiplicative Weights Update (MWU)

While $\mathcal{C}^{(t)} \neq \emptyset$:

1. Select cut(s) from $\mathcal{C}^{(t)}$.
2. Multiplicatively increase $w^{(t)}$ along these cuts.

Sequential MWU: Select one cut from $\mathcal{C}^{(t)}$

$\implies \tilde{O}(m/\varepsilon^2)$ iterations [Garg Könemann '07] [Fleischer '00].

Parallel MWU: Select all cuts from $\mathcal{C}^{(t)}$

$\implies \tilde{O}(\log(|\mathcal{C}^{(t)}|)/\varepsilon^4)$ iterations [Luby Nisan '93] [Young '01].

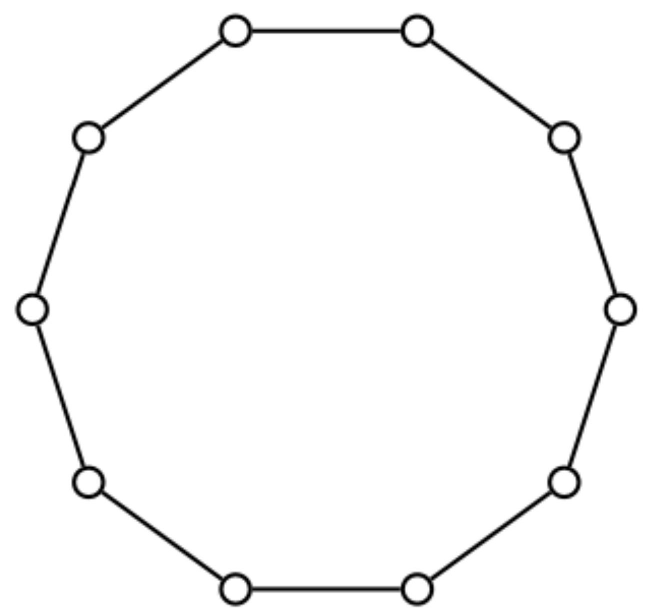
- $|\mathcal{C}^{(t)}| \leq \#\ (1+\varepsilon)$-mincuts $= O(n^2)$ [Nagamochi Nishimura Ibaraki '94] [Henzinger Williamson '96].

*Slides from Zhuan Khye Koh

Implementing MWU for k-ECSM LP

- Sequential MWU has a $\tilde{O}(m/\varepsilon^2)$-time implementation [Chekuri Quanrud '17].

- Parallel MWU incurs $\Omega(n^2)$ work, because $|\mathcal{C}^{(t)}| = \Omega(n^2)$ for some graphs.

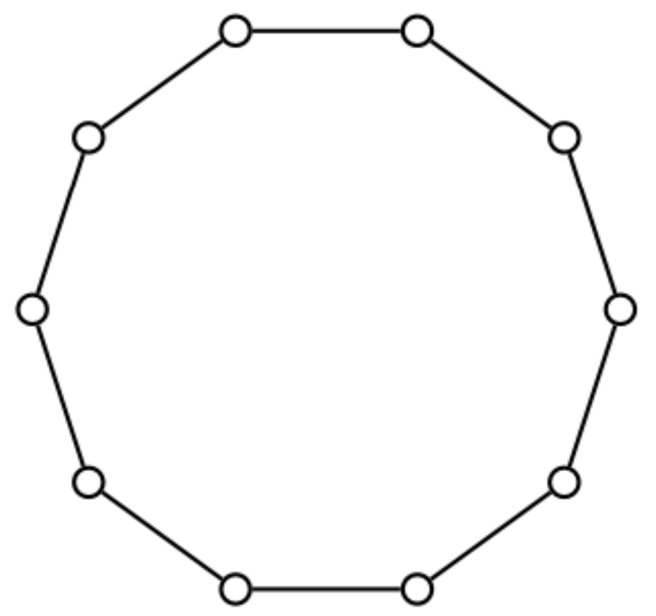


*Slides from Zhuan Khye Koh

Implementing MWU for k-ECSM LP

- Sequential MWU has a $\tilde{O}(m/\varepsilon^2)$-time implementation [Chekuri Quanrud '17].

- Parallel MWU incurs $\Omega(n^2)$ work, because $|\mathcal{C}^{(t)}| = \Omega(n^2)$ for some graphs.



New Selection Rule

Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a sequence of sets of cuts. In iteration t, select

$$S_{i(t)} \cap \mathcal{C}^{(t)}$$

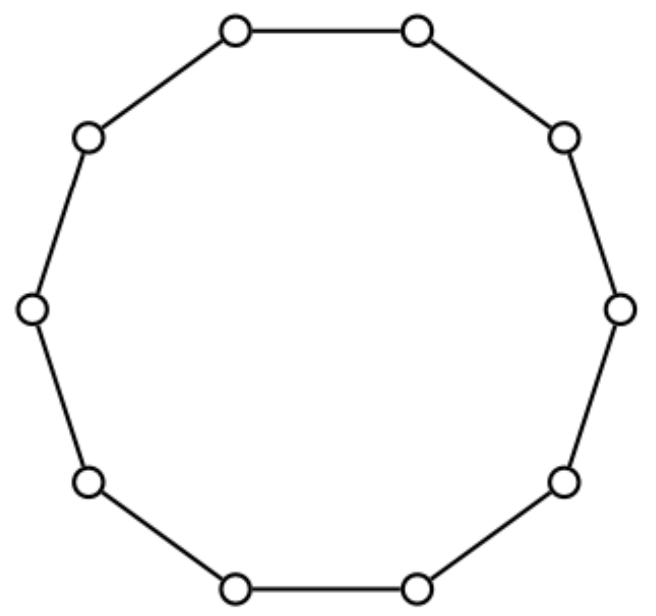
where $i(t)$ is the smallest index such that $S_{i(t)} \cap \mathcal{C}^{(t)} \neq \emptyset$.

*Slides from Zhuan Khye Koh

Implementing MWU for k-ECSM LP

- Sequential MWU has a $\tilde{O}(m/\varepsilon^2)$-time implementation [Chekuri Quanrud '17].

- Parallel MWU incurs $\Omega(n^2)$ work, because $|\mathcal{C}^{(t)}| = \Omega(n^2)$ for some graphs.



New Selection Rule

Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a sequence of sets of cuts. In iteration t, select

$$S_{i(t)} \cap \mathcal{C}^{(t)}$$

where $i(t)$ is the smallest index such that $S_{i(t)} \cap \mathcal{C}^{(t)} \neq \emptyset$.

- If $\cup_{i=1}^{\ell} S_i \cap \mathcal{C}^{(t)} = \emptyset \implies \mathcal{C}^{(t)} = \emptyset$, then $\mathcal{S}$ is a core-sequence of the epoch.

*Slides from Zhuan Khye Koh

Core-Sequence MWU

- Special cases:
  - $\mathcal{S} = (S_1, \ldots, S_\ell)$ where $|S_i| = 1$ for all $i \in [\ell] \implies$ sequential MWU.
  - $\mathcal{S} = (\mathcal{C}^{(t)}) \implies$ parallel MWU.

Core-Sequence MWU

- Special cases:
  - $\mathcal{S} = (S_1, \ldots, S_\ell)$ where $|S_i| = 1$ for all $i \in [\ell] \implies$ sequential MWU.
  - $\mathcal{S} = (\mathcal{C}^{(t)}) \implies$ parallel MWU.

Theorem [KW**Y**'25]

If every epoch has a core-sequence of length $\leq \ell$, in which every set has size $\leq k$, then core-sequence MWU returns a $(1+\varepsilon)$-appproximate solution in

$$\tilde{O}\left(\frac{\ell \log(k)}{\varepsilon^4}\right)$$

iterations.

- Ideally, we want a short core-sequence consisting of small sets.

Core-Sequence for the k-ECSM LP

Theorem [KW**Y**'25]

For the k-ECSM LP, every epoch has a core-sequence of length $O(\log n)$, in which every set has size $O(n)$.

Theorem [KW**Y**'25]

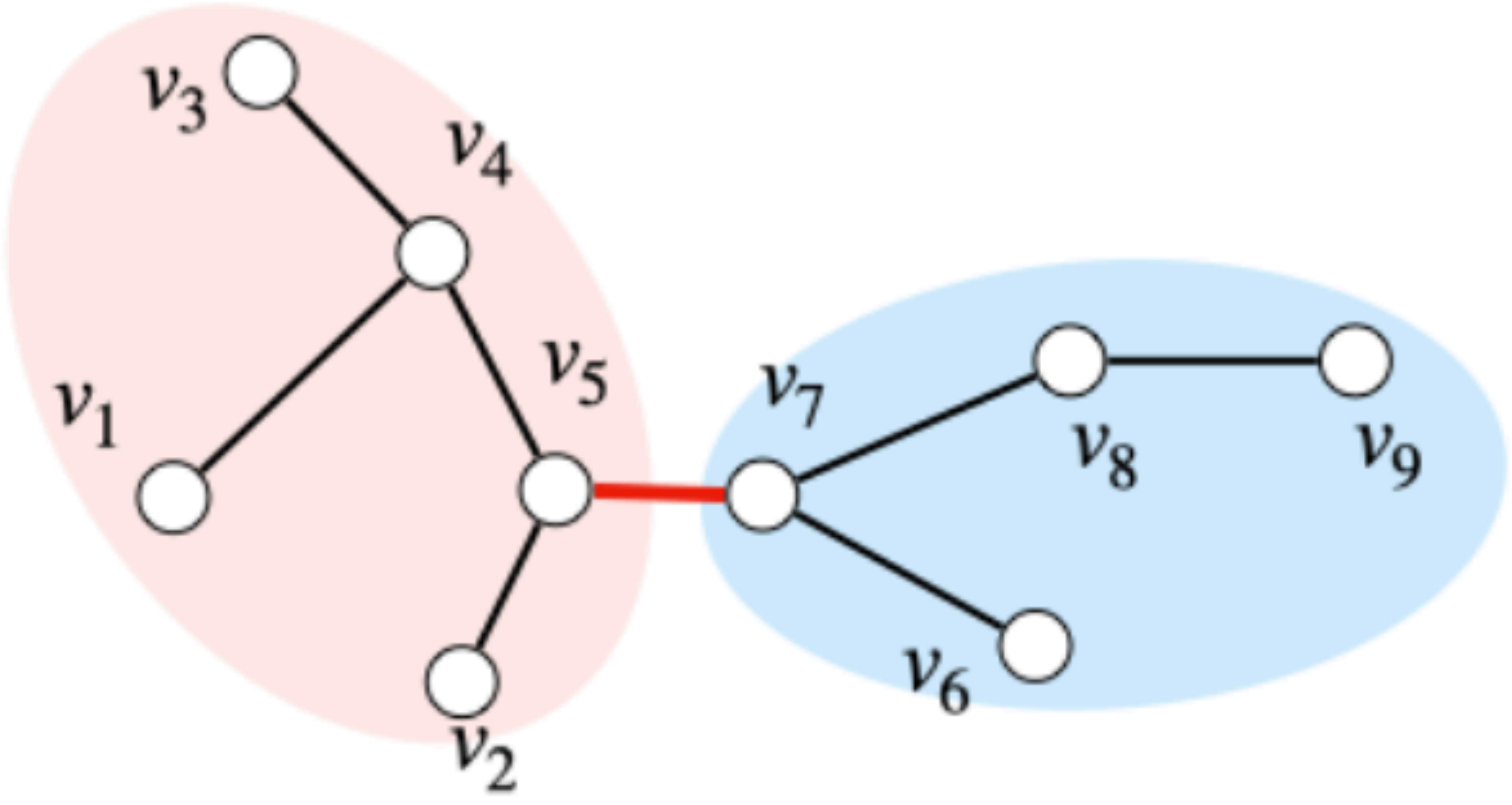
There is a parallel PTAS for the k-ECSM LP using $\tilde{O}(m/\varepsilon^4)$ work and $\tilde{O}(1/\varepsilon^4)$ depth.

- 2-ECSM LP optimum $=$ Held–Karp bound for metric TSP.
- Extends to the k-edge-connected spanning subgraph (k-ECSS) LP.

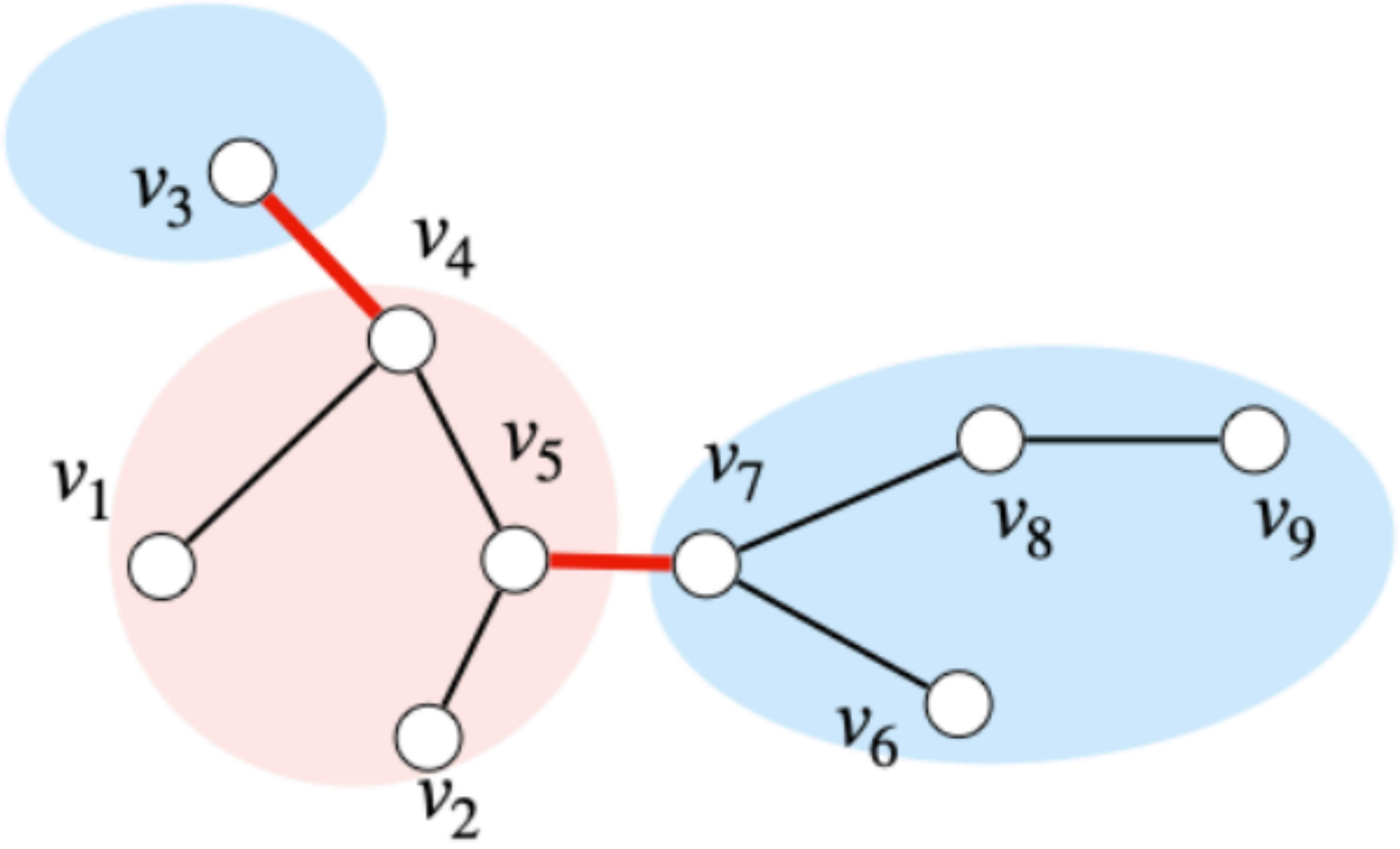
Using weight thresholding technique from [CHNSS**Y**'22]

Combinatorial Tools

Def: A cut C k-*respects* a tree T if $|E(T) \cap C| = k$



A 1-respecting cut in a tree



A 2-respecting cut in a tree

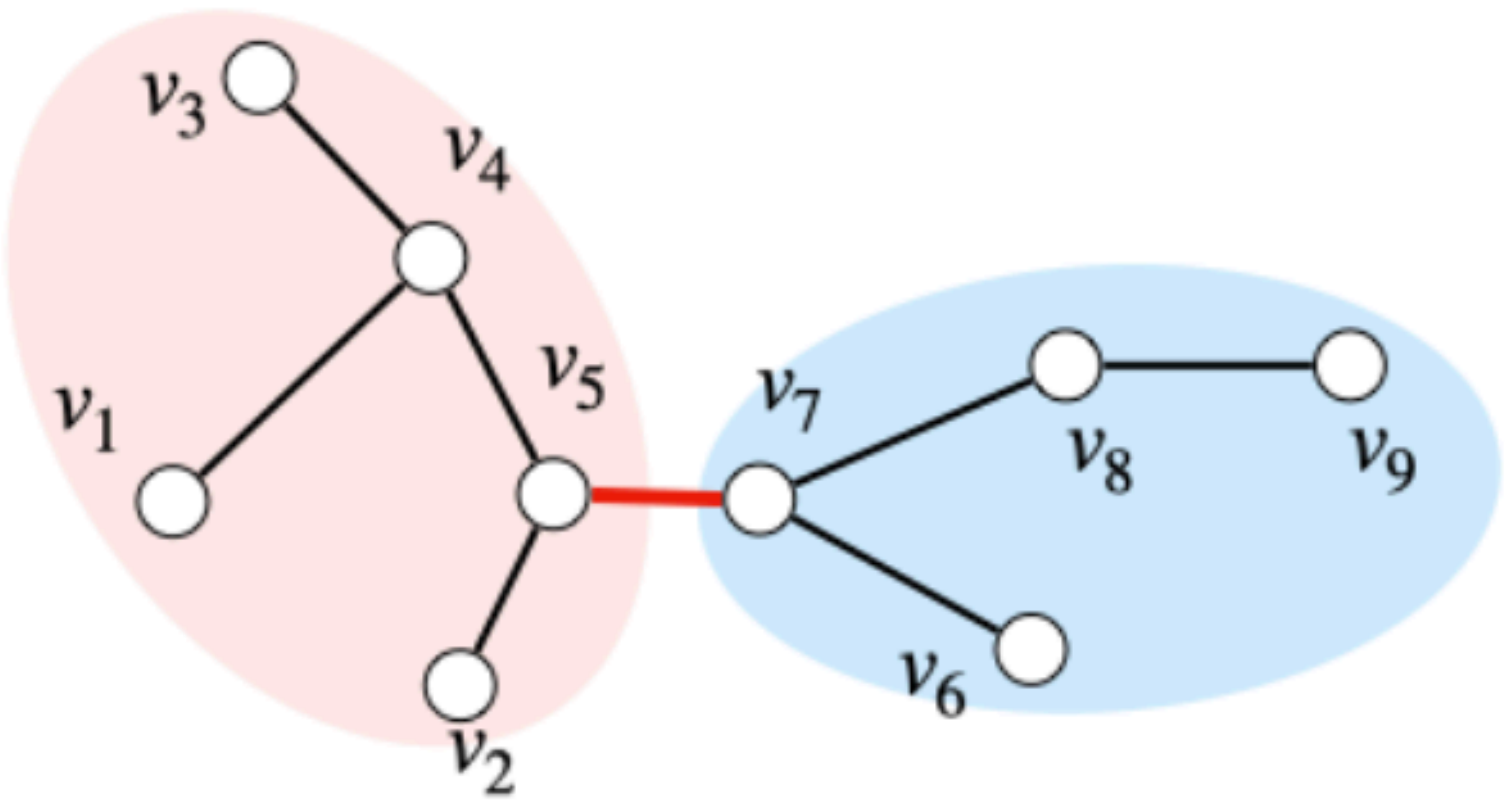
Combinatorial Tools

The Tree Packing Theorem: There is a set of O(log n) spanning trees such that every (1+eps)-mincut 1-or-2 respects some tree

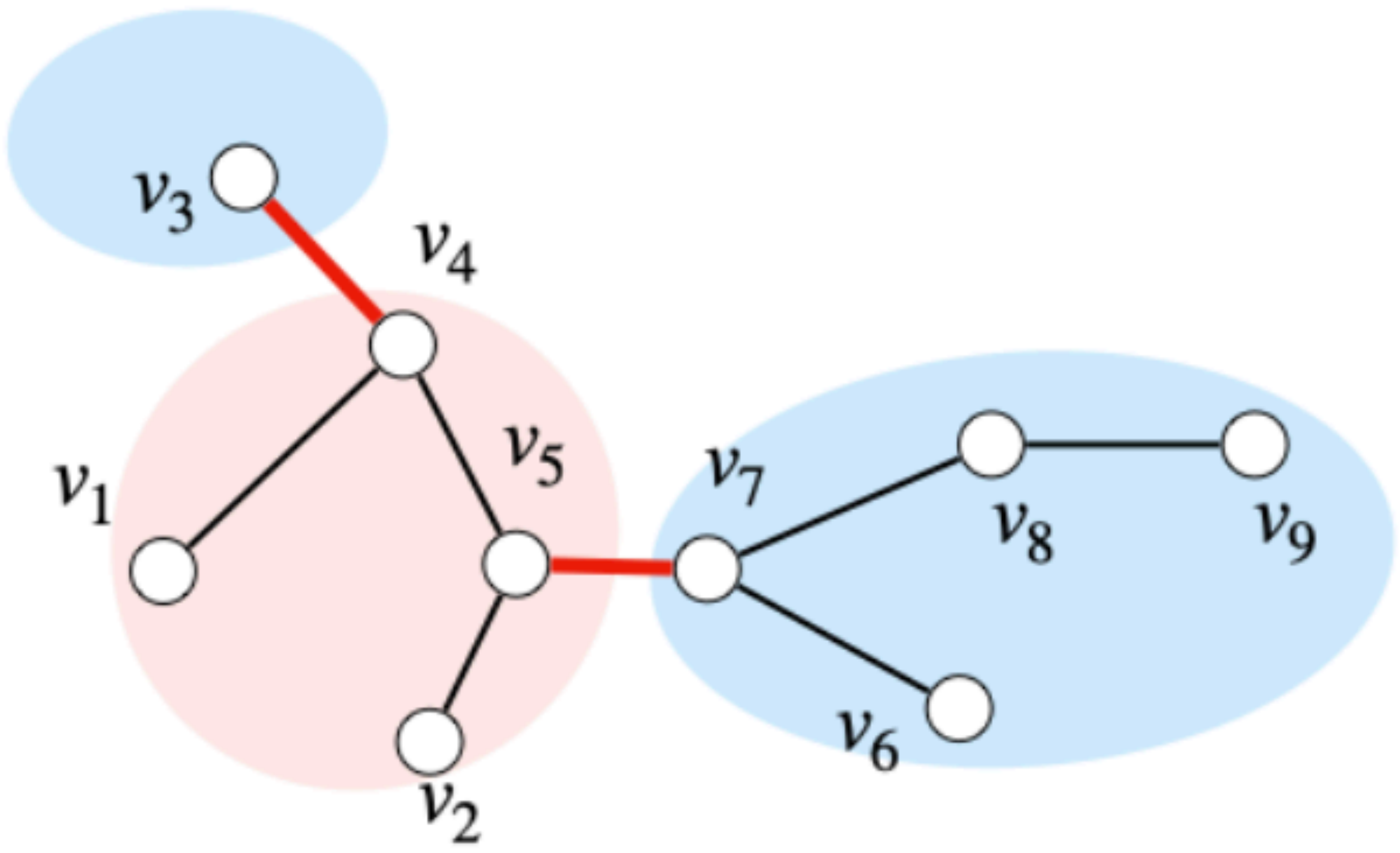
[Karger'00]

(cf. [CQ'17])

Def: A cut C k-*respects* a tree T if $|E(T) \cap C| = k$



A 1-respecting cut in a tree



A 2-respecting cut in a tree

Combinatorial Tools

The Tree Packing Theorem: There is a set of O(log n) spanning trees such that every (1+eps)-mincut 1-or-2 respects some tree

[Karger'00]

(cf. [CQ'17])

Can be computed in nearly linear work and polylog depth [Geissmann and Gianinazzi'18]

Sauce: A Corollary of [Tutte'61][Nash-Williams'61]

$$\lambda_G/2 \leq \tau_G \leq \lambda_G$$

$\tau_G :=$ The maximum number of disjoint spanning trees

$\lambda_G :=$ The edge connectivity

Combinatorial Tools

The Tree Packing Theorem: There is a set of O(log n) spanning trees such that every (1+eps)-mincut 1-or-2 respects some tree

[Karger'00]

$$\mathcal{C}^{(t)} := \{C \text{ cut } : w^{(t)}(C) < (1+\varepsilon)\lambda\}.$$

$$\left.\begin{array}{l}\textbf{While } \mathcal{C}^{(t)} \neq \emptyset\text{:}\\ \text{1. Select cut(s) from } \mathcal{C}^{(t)}.\\ \text{2. Multiplicatively increase } w^{(t)} \text{ along these cuts.}\end{array}\right\} \text{an epoch}$$

- $\lambda \leftarrow \lambda(1+\varepsilon)$ and a new epoch begins.

→

For $T \in \mathcal{T}$,

$$C_T^{(t)} := \left\{ \{e_1, e_2\} \subseteq E(T) : \mathrm{w}(\mathrm{cut}_T(e_1, e_2)) < (1+\varepsilon)\cdot\lambda \right\}$$

While $C_T^{(t)} \neq \varnothing$

1. Select pairs of tree edges from $C_T^{(t)}$

2. Multiplicatively increase $w^{(t)}$ along these cuts

Combinatorial Tools

The Tree Packing Theorem: There is a set of O(log n) spanning trees such that every (1+eps)-mincut 1-or-2 respects some tree

[Karger'00]

$$\mathcal{C}^{(t)} := \{C \text{ cut } : w^{(t)}(C) < (1+\varepsilon)\lambda\}.$$

While $\mathcal{C}^{(t)} \neq \emptyset$:

1. Select cut(s) from $\mathcal{C}^{(t)}$.
2. Multiplicatively increase $w^{(t)}$ along these cuts.

} an epoch

- $\lambda \leftarrow \lambda(1+\varepsilon)$ and a new epoch begins.

For $T \in \mathcal{T}$,

$$C_T^{(t)} := \left\{ \{e_1, e_2\} \subseteq E(T) : \mathrm{w}(\mathrm{cut}_T(e_1, e_2)) < (1+\varepsilon) \cdot \lambda \right\}$$

While $C_T^{(t)} \neq \varnothing$

1. Select pairs of tree edges from $C_T^{(t)}$

2. Multiplicatively increase $w^{(t)}$ along these cuts

Goal for the inner loop:

$\tilde{O}(n)$ work per iteration, $\tilde{O}(1)$ iterations

Combinatorial Tools

The Tree Packing Theorem: There is a set of O(log n) spanning trees such that every (1+eps)-mincut 1-or-2 respects some tree

[Karger'00]

$$\mathcal{C}^{(t)} := \{C \text{ cut } : w^{(t)}(C) < (1+\varepsilon)\lambda\}.$$

While $\mathcal{C}^{(t)} \neq \emptyset$:

1. Select cut(s) from $\mathcal{C}^{(t)}$.
2. Multiplicatively increase $w^{(t)}$ along these cuts.

an epoch

- $\lambda \leftarrow \lambda(1+\varepsilon)$ and a new epoch begins.

n² work, 1 iteration if select all pairs

For $T \in \mathcal{T}$,

$$C_T^{(t)} := \left\{ \{e_1, e_2\} \subseteq E(T) : \mathrm{w}(\mathrm{cut}_T(e_1, e_2)) < (1+\varepsilon)\cdot\lambda \right\}$$

While $C_T^{(t)} \neq \varnothing$

1. Select pairs of tree edges from $C_T^{(t)}$
2. Multiplicatively increase $w^{(t)}$ along these cuts

Goal for the inner loop:

$\tilde{O}(n)$ work per iteration, $\tilde{O}(1)$ iterations

Intuition: Select a representative/maximal set of cuts so that

Updating weights of these cuts = increase weights of every cut

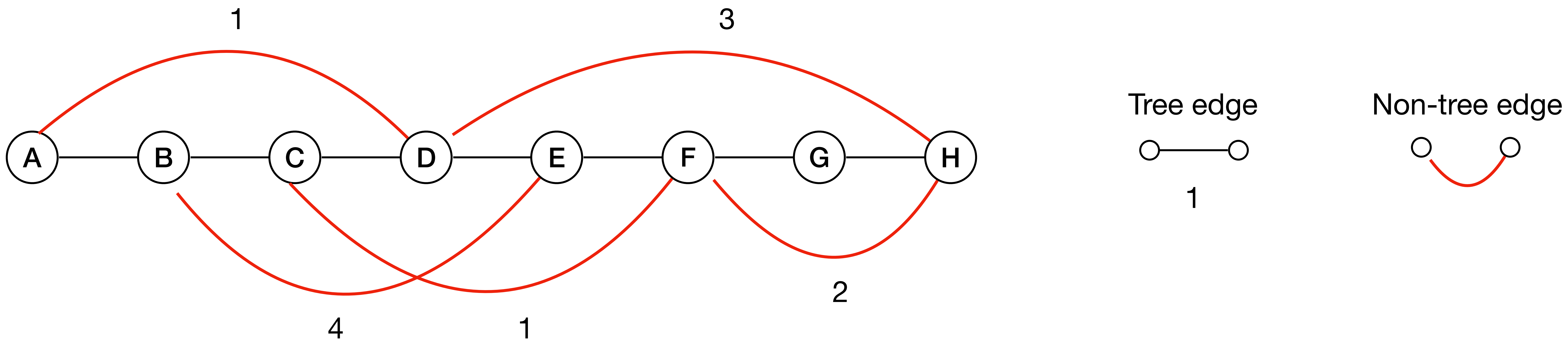
Intuition: Select a representative/maximal set of cuts so that

Updating weights of these cuts = increase weights of every cut

This talk: assume T is a path and show 'good' core sequence exists

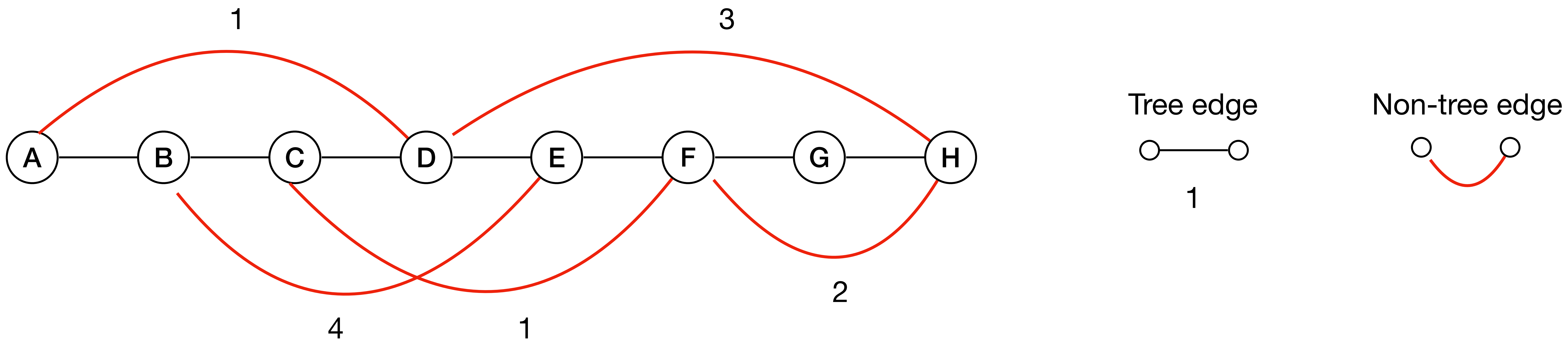
(In general, reduce to path via heavy/light decomposition [MN'20].)

T = path



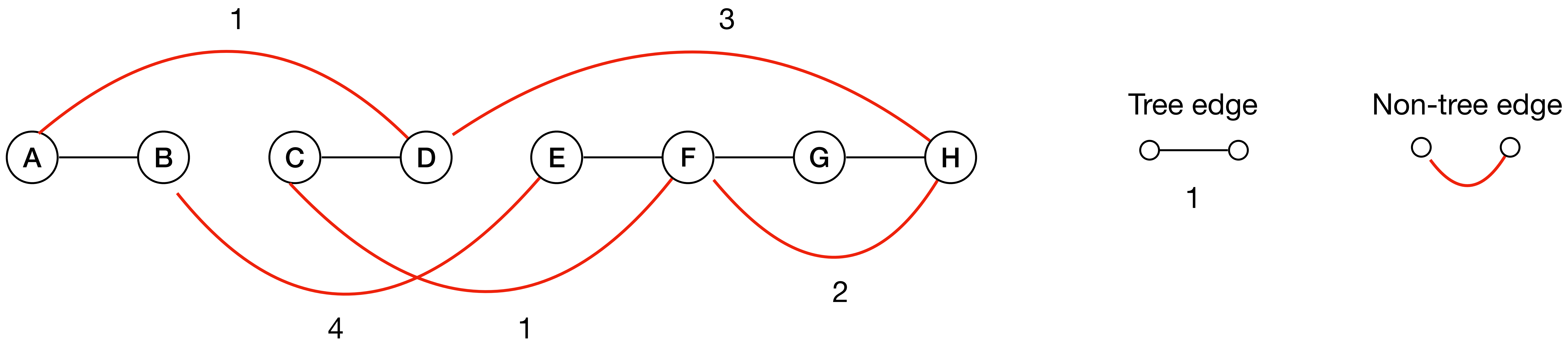
$w(\mathsf{cut}_T(e_1, e_2))$

T = path



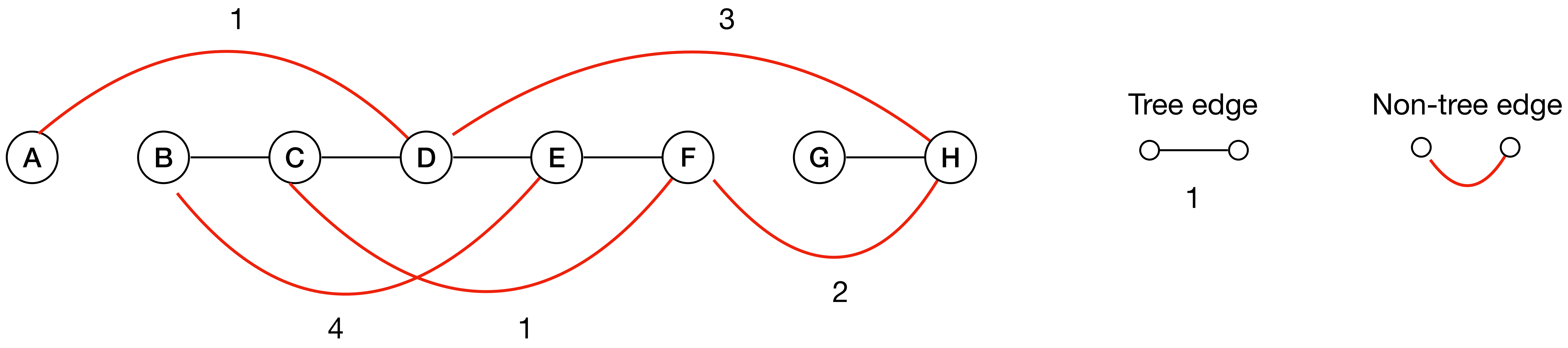
$w(\mathrm{cut}_T(BC, DE)) = ?$

T = path



$$w(\text{cut}_T(BC, DE)) = 1 + 3 + 1 + 2 = 7$$

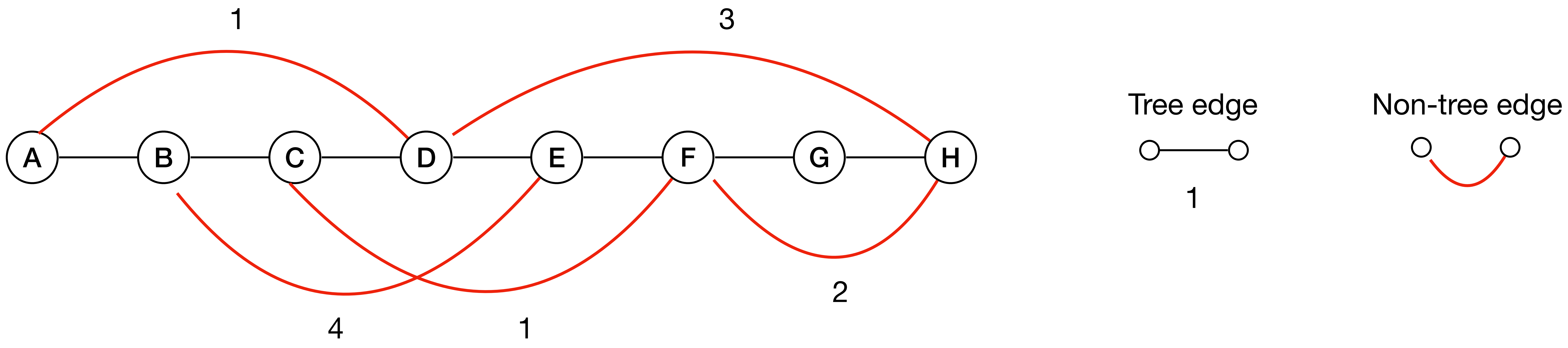
T = path



$$w(\text{cut}_T(BC, DE)) = 7$$

$$w(\text{cut}_T(AB, FG)) = 1 + 3 + 2 + 2 = 8$$

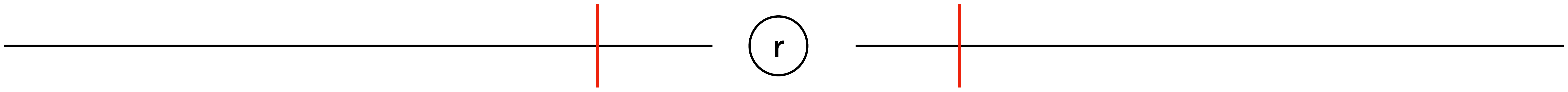
T = path



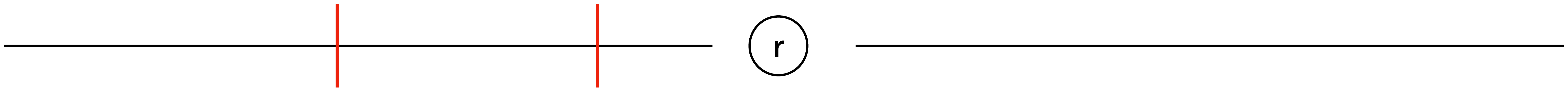
$w(\mathsf{cut}_T(e_1, e_2))$

$$C_T^{(t)} := \left\{ \{e_1, e_2\} \subseteq E(T) \colon \mathsf{w}(\mathsf{cut}_T(e_1, e_2)) < (1+\varepsilon)\cdot\lambda \right\}$$

$$|C_T^{(t)}| \leq n^2$$

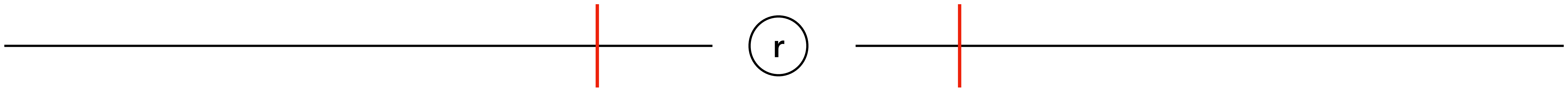


r-crossing



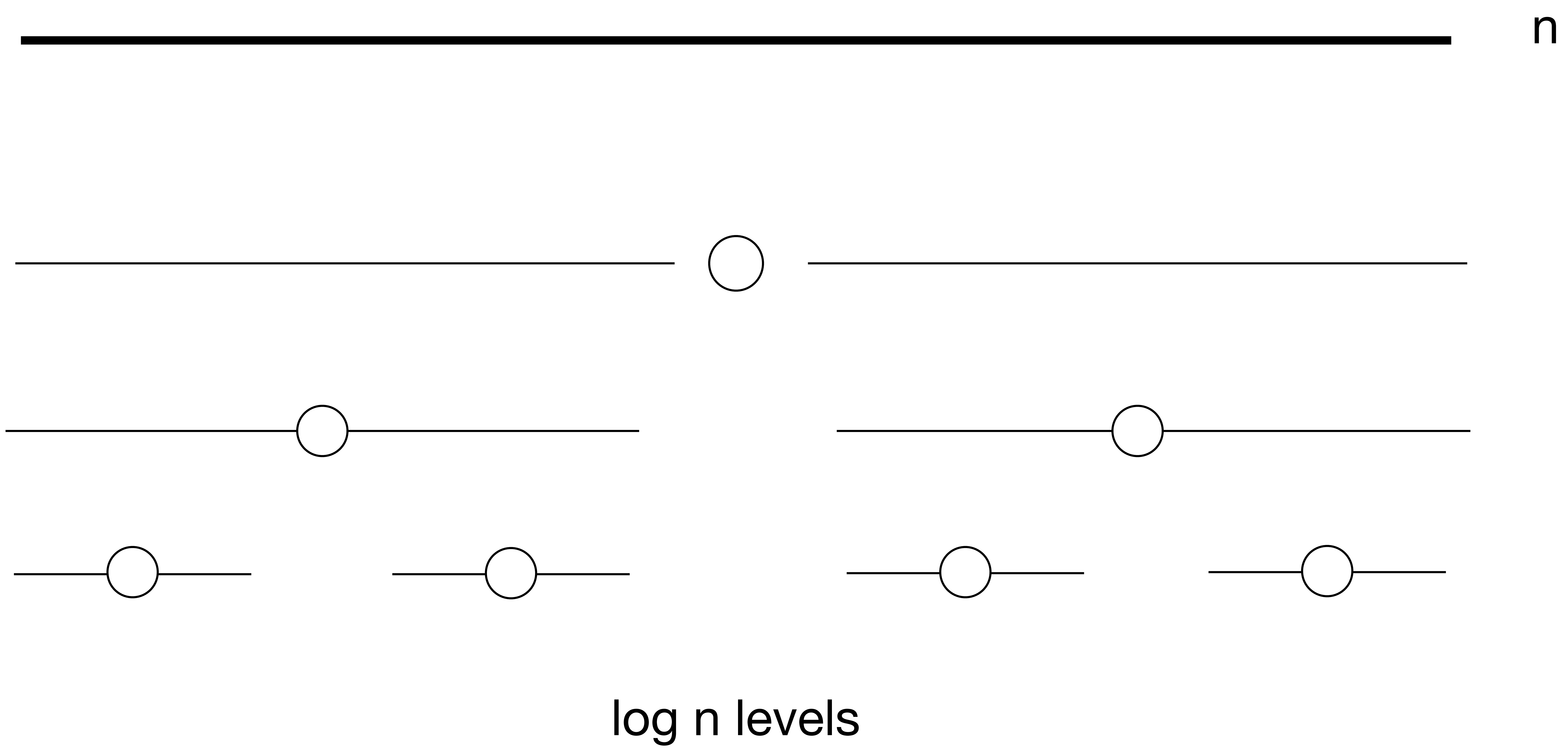
NOT r-crossing

Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs



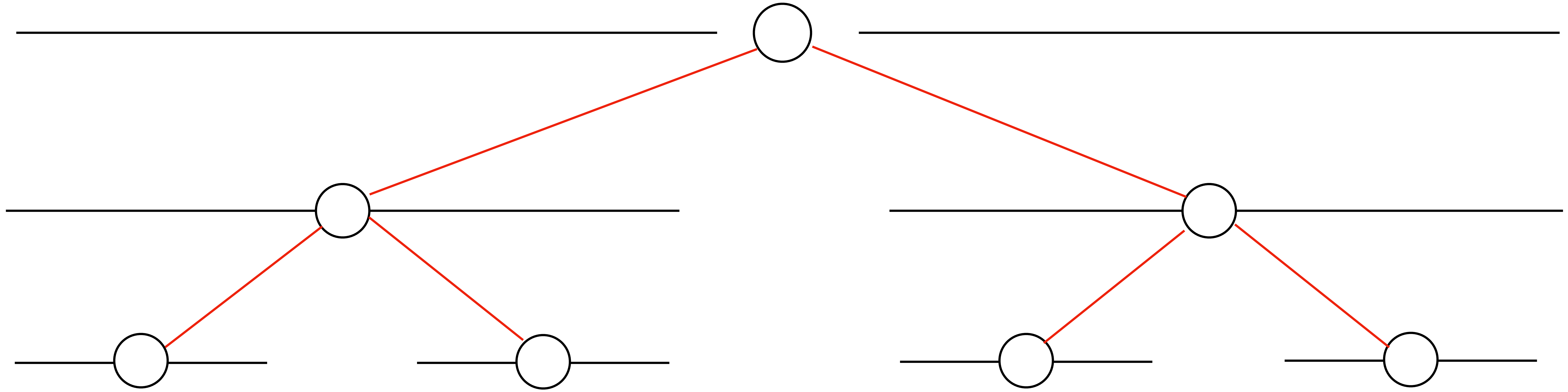
r-crossing

Representation of all pairs



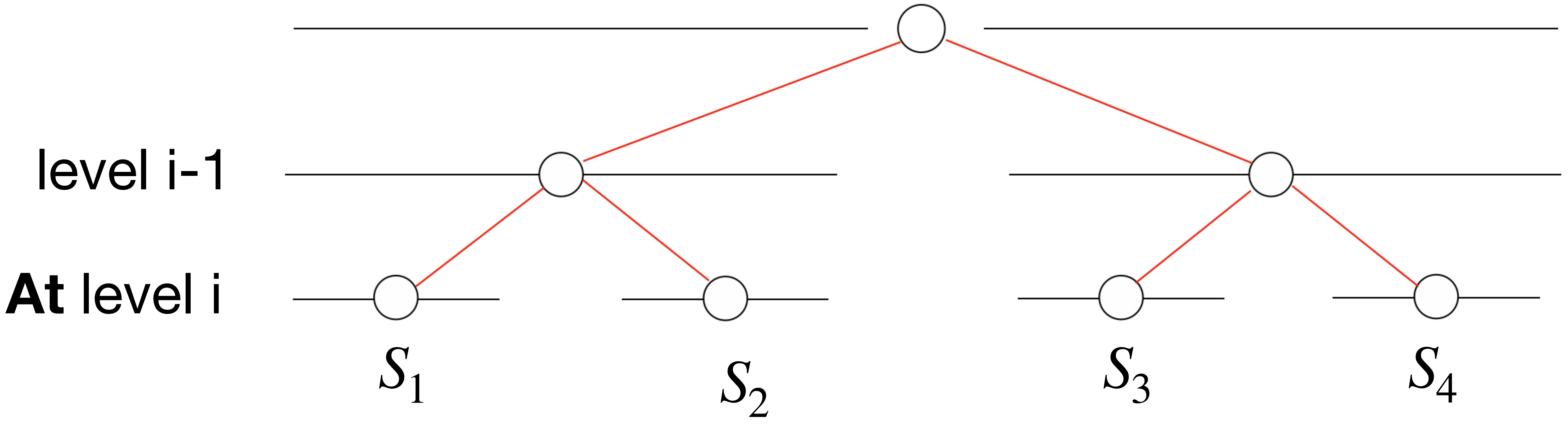
Representation of all pairs

The recursion tree



Property: every pair of edges in T is mapped to the first node that splits it

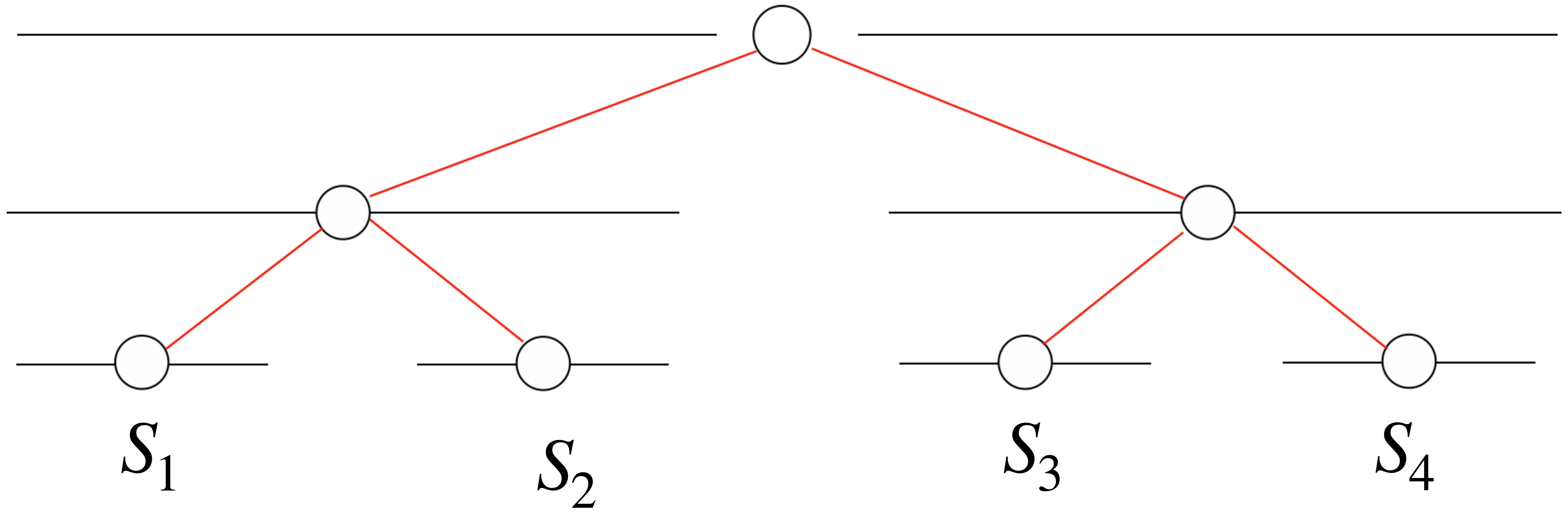
Lemma 1 : Fix a node r in a path, if every small pair is r-crossing, then there are O(n) small pairs



Lemma 1 : Fix a node r in a path, if every small pair is r-crossing, then there are O(n) small pairs

level i-1

At level i



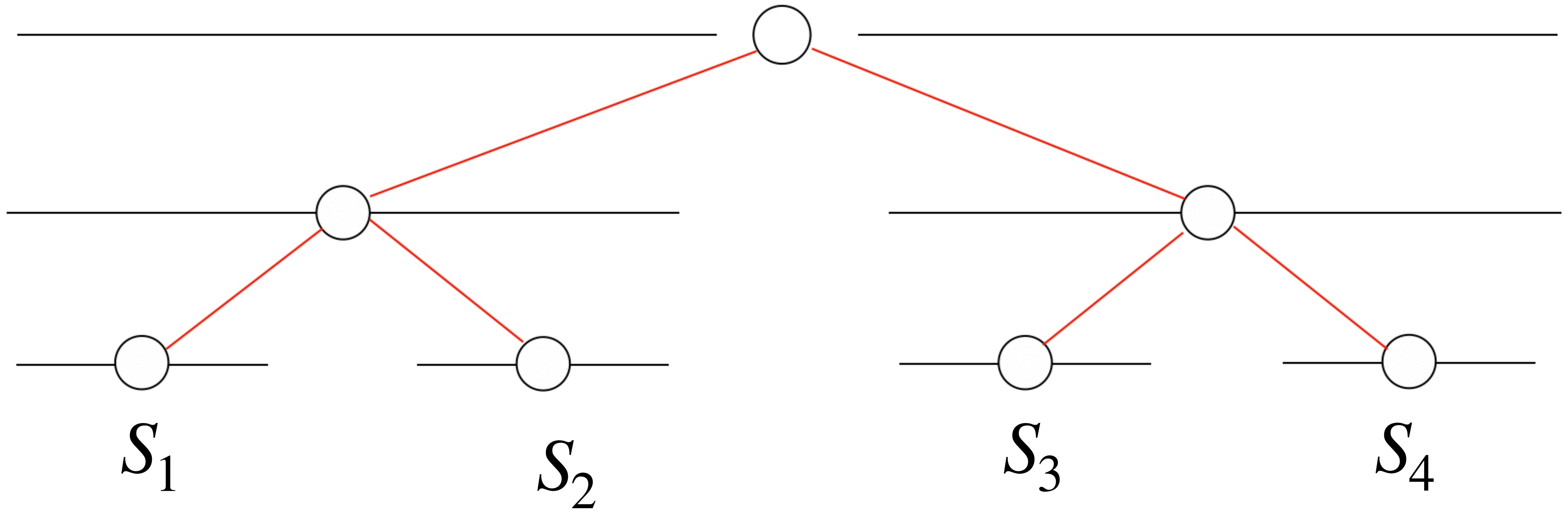
At level i

Compute $B_i := \bigcup_j S_j$ where $S_j :=$ the set of small pairs in j-th path

Lemma 1 : Fix a node r in a path, if every small pair is r-crossing, then there are O(n) small pairs

level i-1

At level i

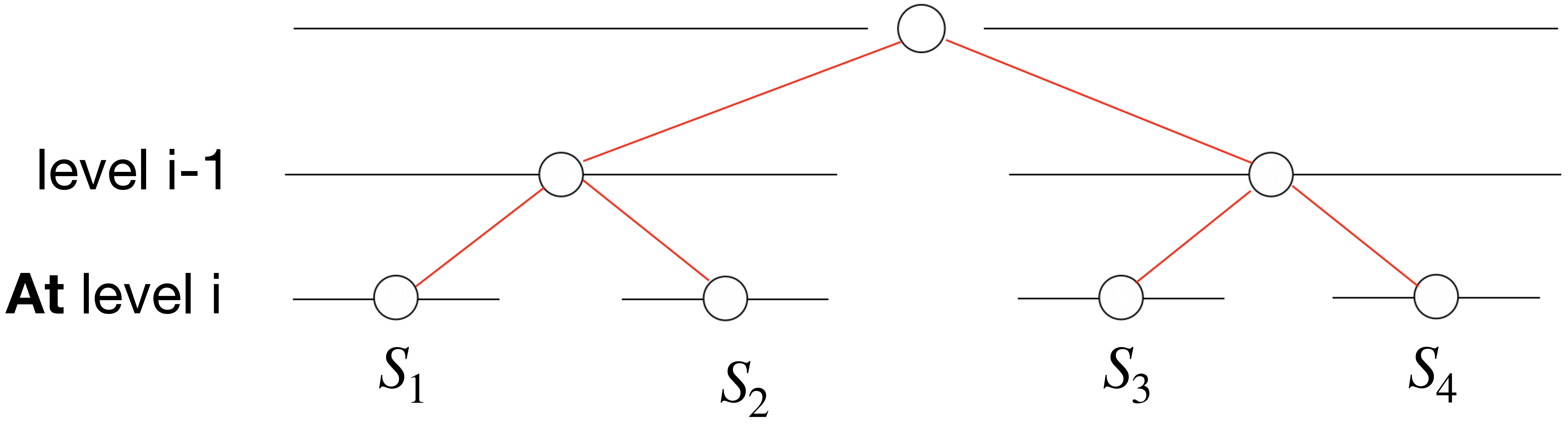


At level i

Compute $B_i := \bigcup_j S_j$ where $S_j :=$ the set of small pairs in j-th path

By **Lemma 1,** $|B_i| = \sum_j |S_j| = \sum_j O(n_i) = O(n)$

Lemma 1 : Fix a node r in a path, if every small pair is r-crossing, then there are O(n) small pairs



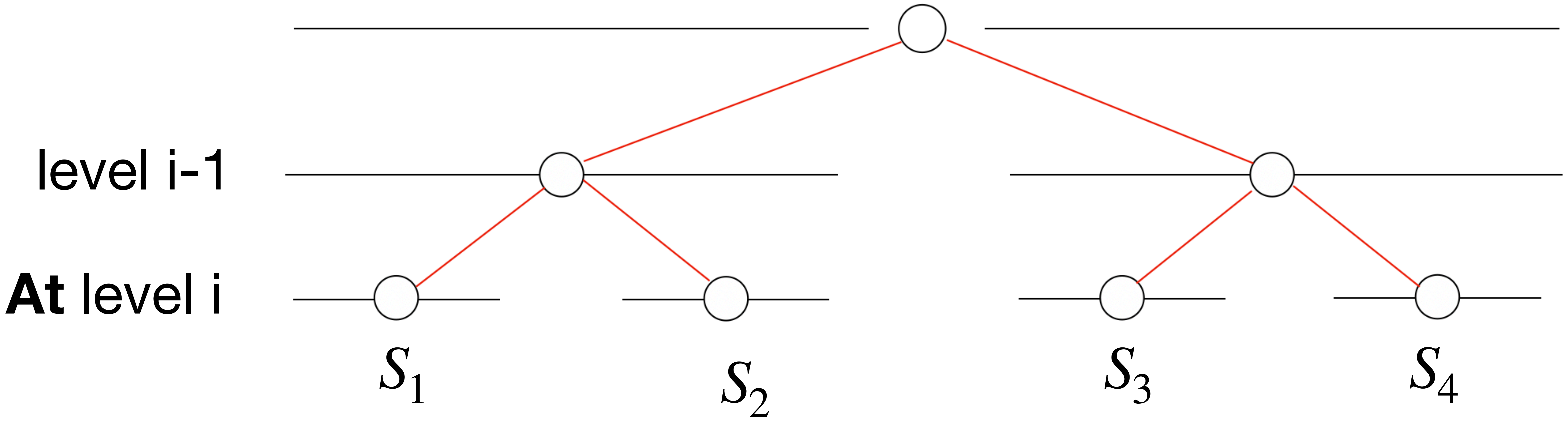
At level i

Compute $B_i := \bigcup_j S_j$ where $S_j :=$ the set of small pairs in j-th path

By **Lemma 1,** $|B_i| = \sum_j |S_j| = \sum_j O(n_i) = O(n)$

Feed B_i to the MWU framework to update weights along these cuts

Lemma 1 : Fix a node r in a path, if every small pair is r-crossing, then there are O(n) small pairs



At level i

Compute $B_i := \bigcup_j S_j$ where $S_j :=$ the set of small pairs in j-th path

By **Lemma 1,** $|B_i| = \sum_j |S_j| = \sum_j O(n_i) = O(n)$

Feed B_i to the MWU framework to update weights along these cuts

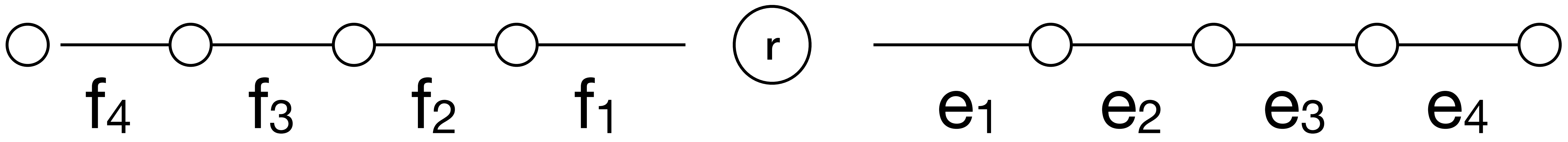
Repeat at level i-1 and so on $(B_{\log n}, B_{\log n-1}, \ldots, B_1)$ is a good core sequence

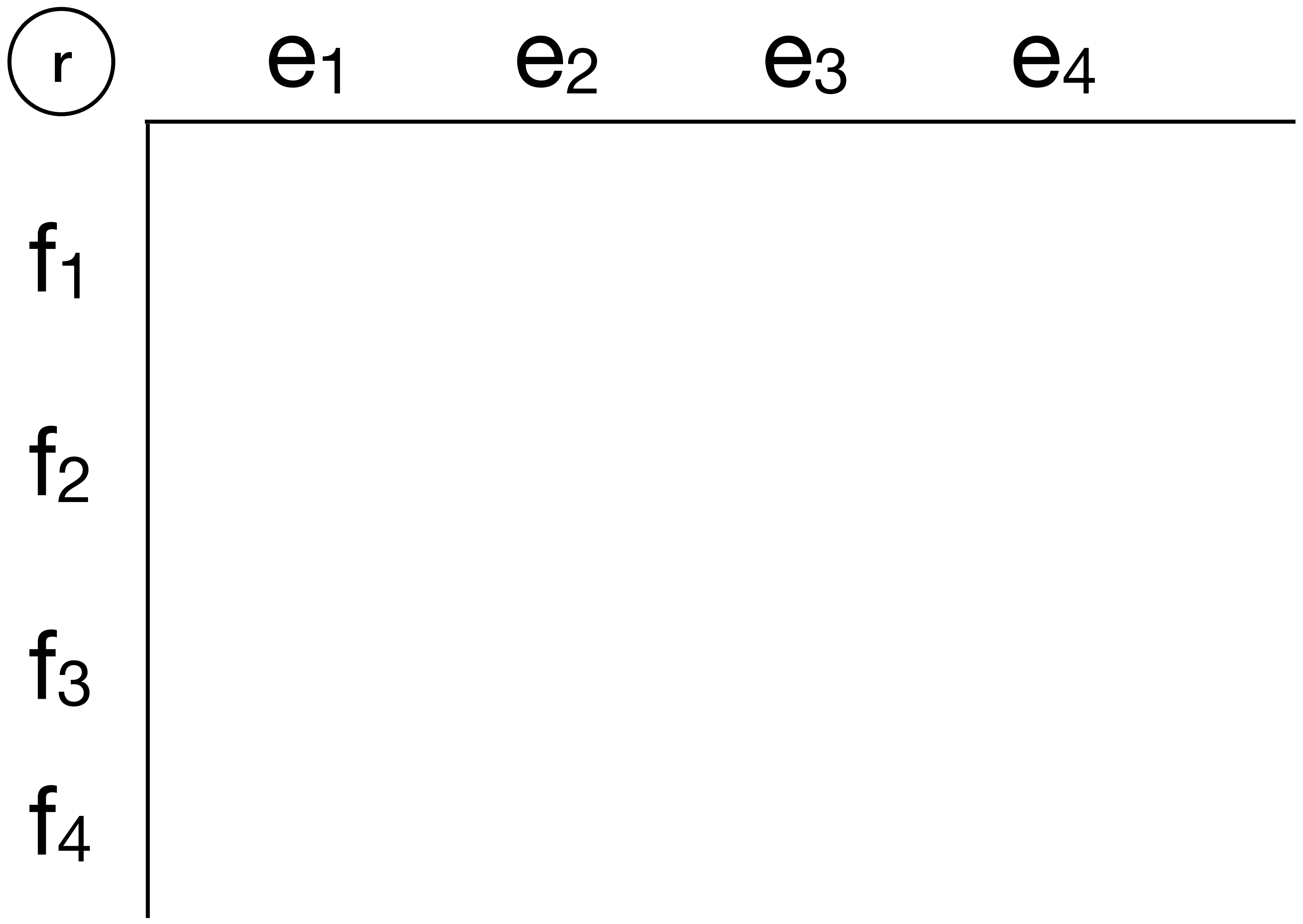
Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs

Proof:

Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs

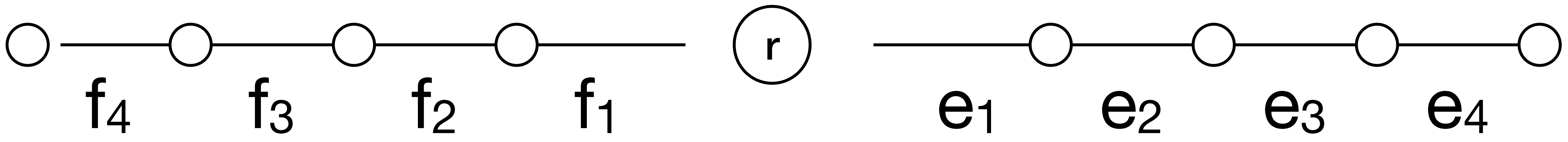
Proof:





Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs

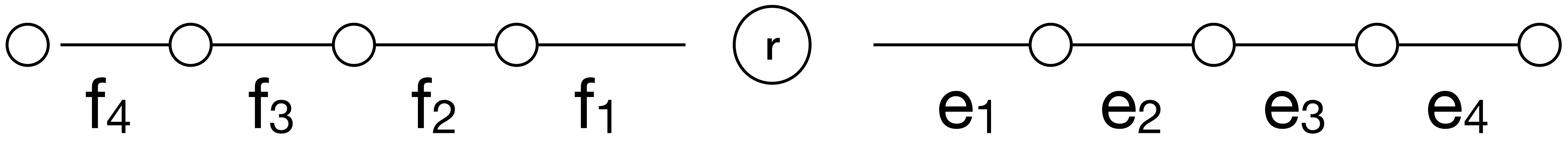
Proof:



r	e_1	e_2	e_3	e_4
f_1				
f_2				
f_3				
f_4				

Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs

Proof:



Define M_r

$M_r(f_i, e_j) = w(\text{cut}_T(f_i, e_j))$

r	e_1	e_2	e_3	e_4
f_1				
f_2				
f_3				
f_4				

Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs

Proof:

$$\begin{array}{c|cccc} r & e_1 & e_2 & e_3 & e_4 \\ \hline f_1 & & & & \\ f_2 & & & & \\ f_3 & & & & \\ f_4 & & & & \end{array}$$

M_r

$M_r(f_i, e_j) = w(\text{cut}_T(f_i, e_j))$

define Q_r

$Q_r(f_i, e_j) = 1$ if $M_r(f_i, e_j) < (1 + \varepsilon)\lambda$

$Q_r(f_i, e_j) = 0$ else

Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs

Proof:

r	e_1	e_2	e_3	e_4
f_1				
f_2				
f_3		1		
f_4				

define Q_r

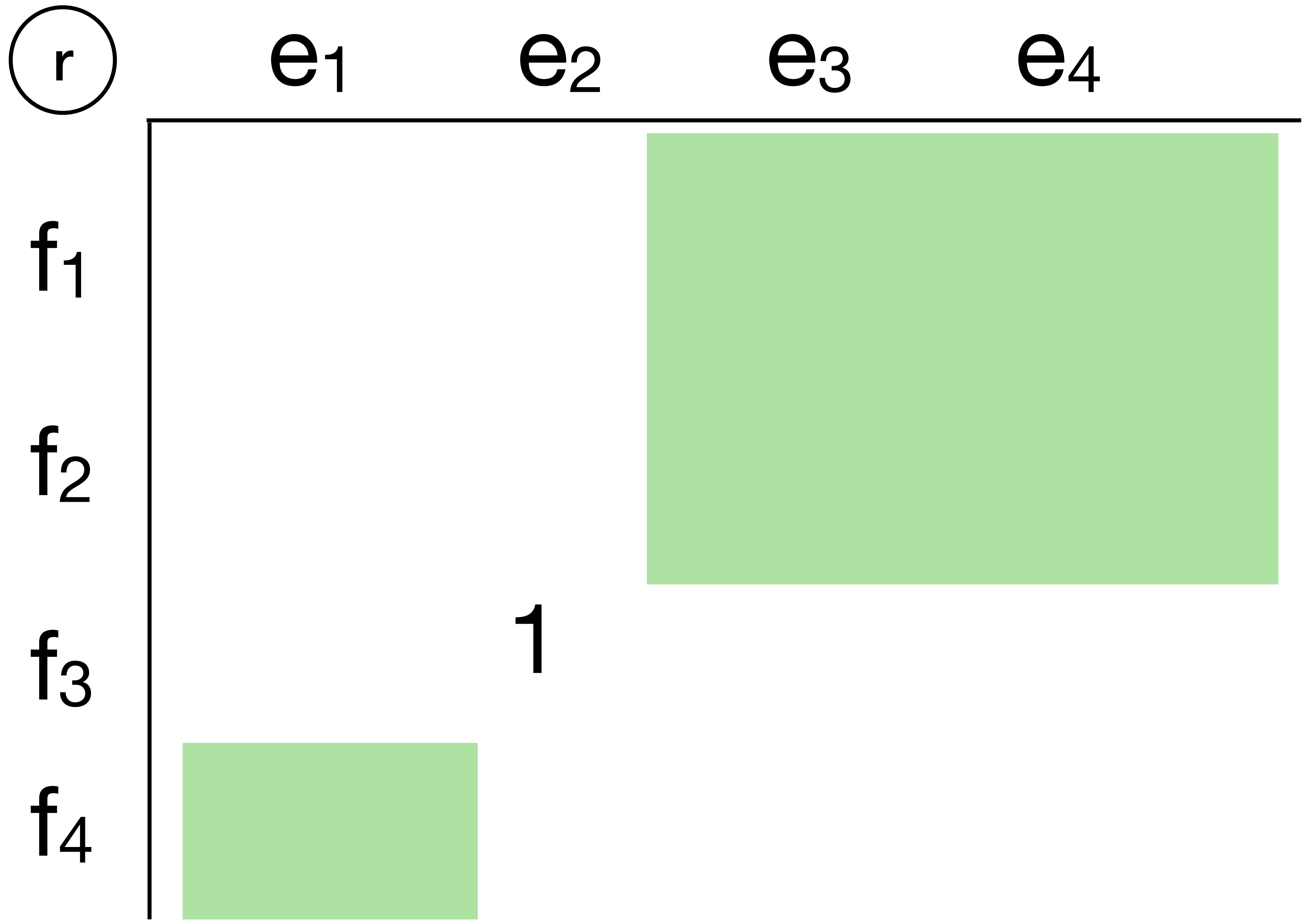
$$Q_r(f_i, e_j) = 1 \text{ if } M_r(f_i, e_j) < (1+\varepsilon)\lambda$$

$$Q_r(f_i, e_j) = 0 \text{ else}$$

claim: Q avoids $\begin{array}{|c|c|}\hline & 1 \\ \hline 1 & \\ \hline\end{array} \implies |Q| \leq 2n$

Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs

Proof:



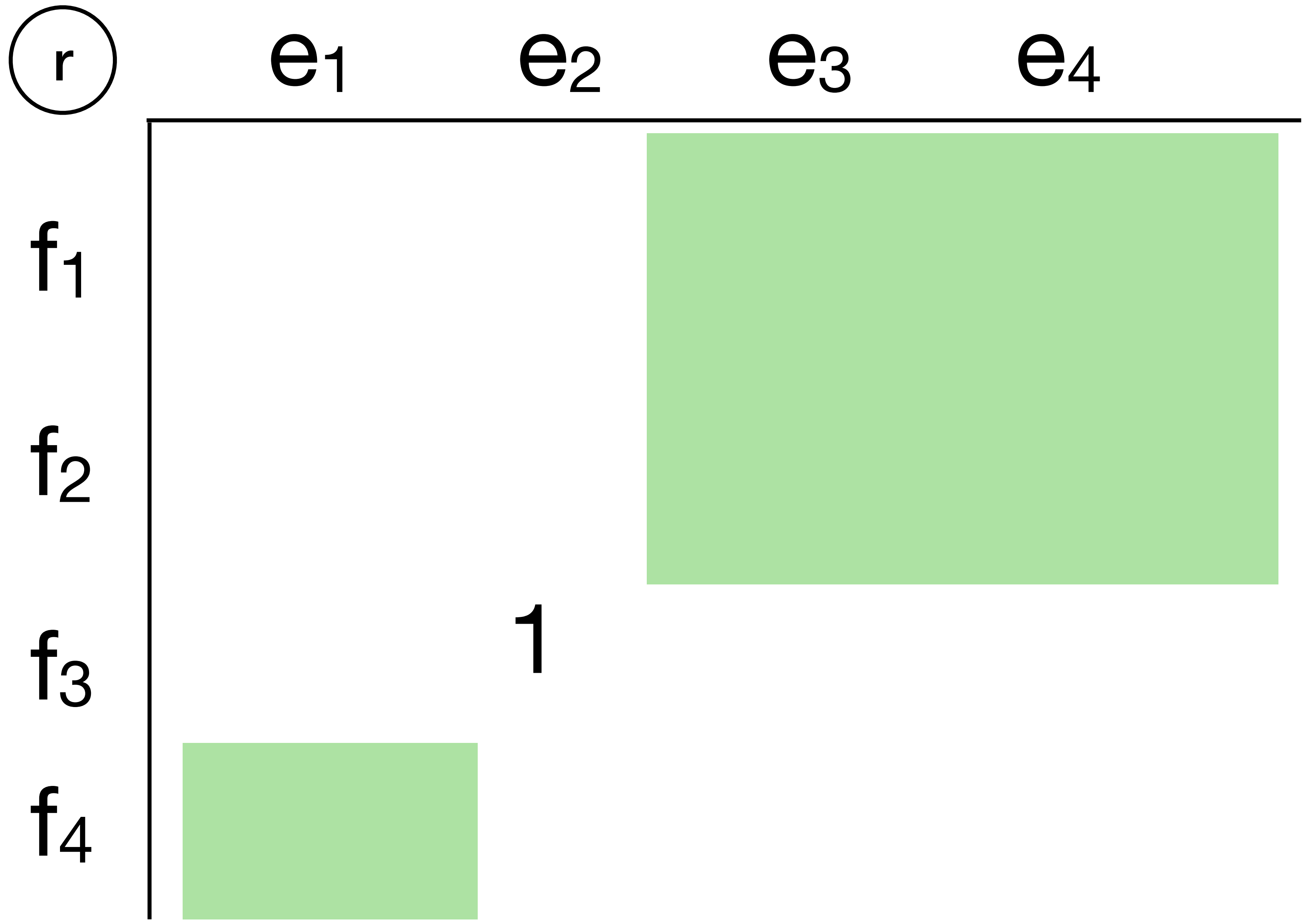
define Q_r

$$Q_r(f_i, e_j) = 1 \text{ if } M_r(f_i, e_j) < (1+\varepsilon)\lambda$$
$$Q_r(f_i, e_j) = 0 \text{ else}$$

claim: Q avoids $\begin{array}{|c|c|}\hline & 1 \\ \hline 1 & \\ \hline\end{array} \implies |Q| \le 2n$

Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs

Proof:



define Q_r

$$Q_r(f_i, e_j) = 1 \text{ if } M_r(f_i, e_j) < (1+\varepsilon)\lambda$$
$$Q_r(f_i, e_j) = 0 \text{ else}$$

claim: Q avoids $\begin{array}{|c|c|}\hline & 1 \\ \hline 1 & \\ \hline\end{array}$ $\implies |Q| \leq 2n$

claim: Q can be computed

$$ex\left(n, \begin{array}{|c|c|c|}\hline & & 1 \\ \hline & 1 & \\ \hline 1 & & \\ \hline\end{array}\right) \leq 6n$$

number of probes in M_r

Lemma: Fix a node r in a path, if every **small** pair is r-crossing, then there are O(n) **small** pairs

Proof:

r	e_1	e_2	e_3	e_4
f_1				
f_2			1	
f_3		1		
f_4				

define Q_r

$Q_r(f_i, e_j) = 1$ if $M_r(f_i, e_j) < (1+\varepsilon)\lambda$

$Q_r(f_i, e_j) = 0$ else

claim: Q avoids $\begin{array}{|c|c|}\hline & 1 \\ \hline 1 & \\ \hline\end{array}$ $\implies |Q| \leq 2n$

proof: Suppose Q contains $\begin{array}{|c|c|}\hline & 1 \\ \hline 1 & \\ \hline\end{array}$

claim 2:

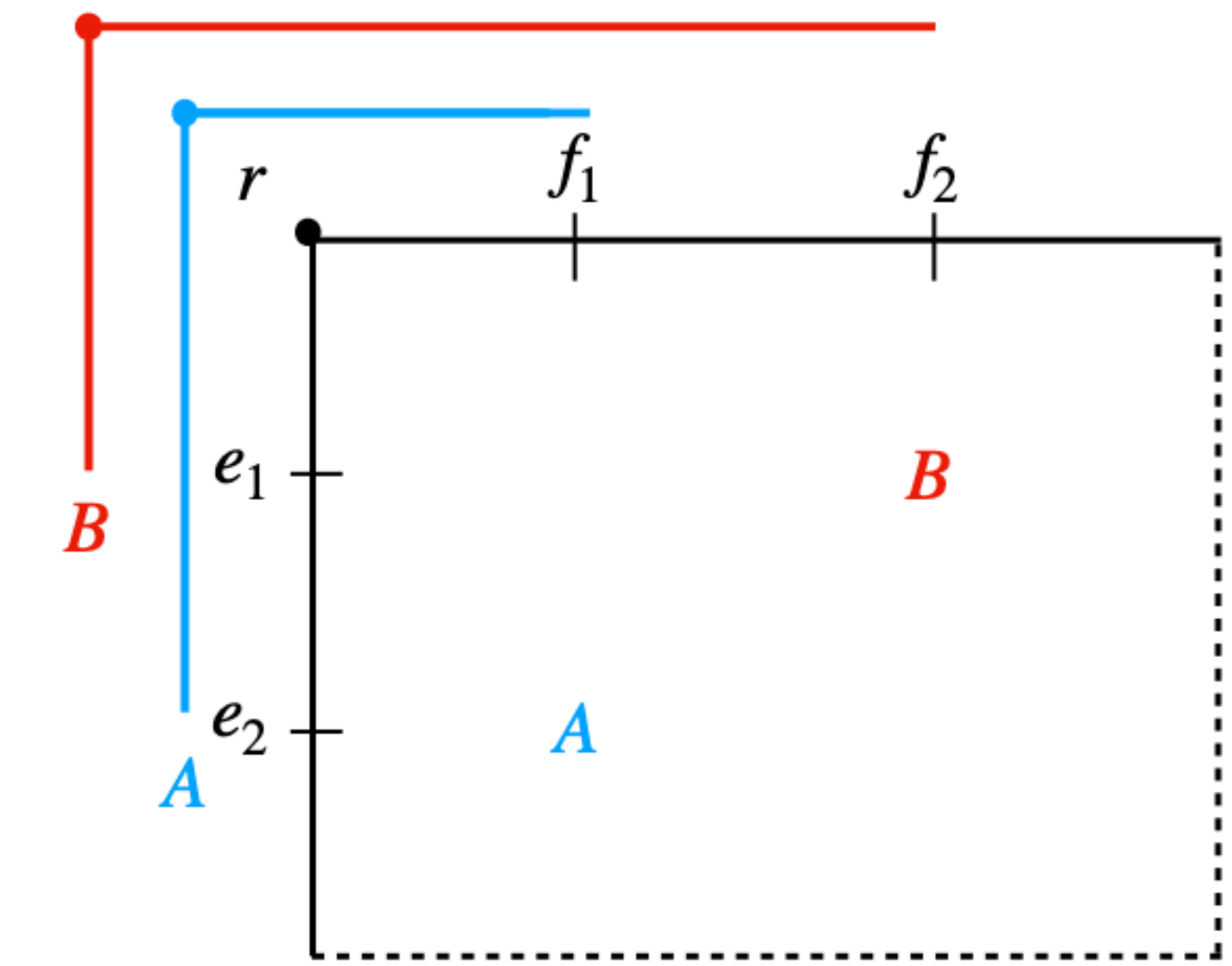
there is a small pair that does **not** cross r

contradiction

Proposition 2.2. *For every pair of subsets $X, Y \subseteq V$, we have*

- *(Submodularity)* $f_w(X) + f_w(Y) \geq f_w(X \cap Y) + f_w(X \cup Y)$, *and*
- *(Posi-modularity)* $f_w(X) + f_w(Y) \geq f_w(X \setminus Y) + f_w(Y \setminus X)$.

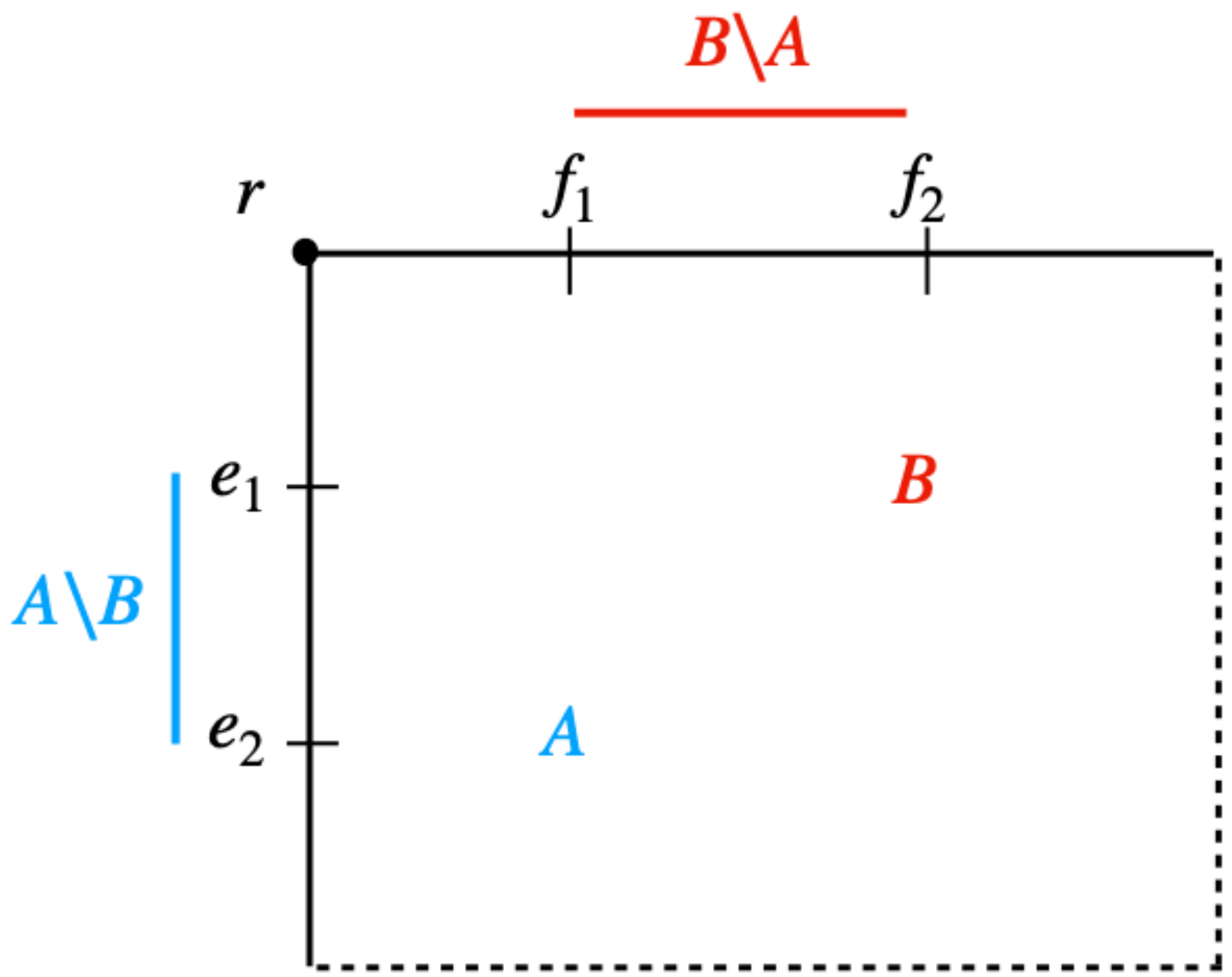
Claim 2: there is a small pair that does **not** cross r



$$2\lambda(1+\varepsilon) > f(A) + f(B) \geq f(A \backslash B) + f(B \backslash A)$$ (Posi-modularity)

Therefore, $\min\{f(A\backslash B), f(B\backslash A)\} < (1+\varepsilon)\lambda$

Contradiction!



Recall **Lemma:** if every **small** pair is r-crossing,
then there are O(n) **small** pairs

Concluding Remark

- Core-sequence is generic
- Improve: $\tilde{O}(\frac{m}{\varepsilon^4}) \rightarrow \tilde{O}(\frac{m}{\varepsilon^2})?$
- Solving kECSM with high accuracy?
- Extension to streaming and distributed algorithms? [MN'20]