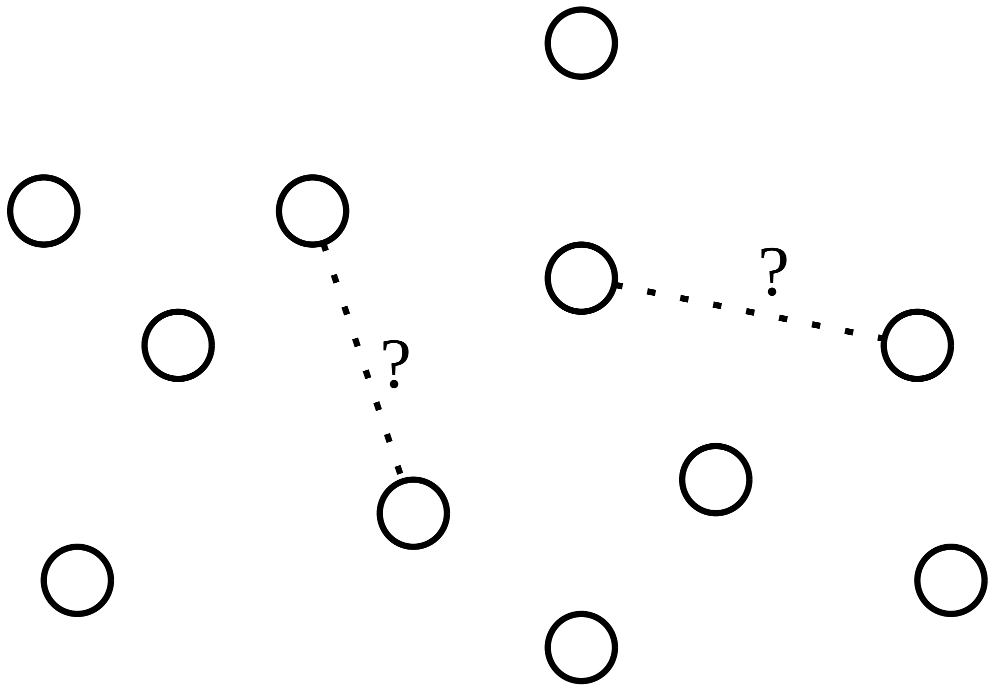
Linear Programming in Bounded Tree-width Markov Networks

Percy Liang — MIT
Nati Srebro — U. Toronto

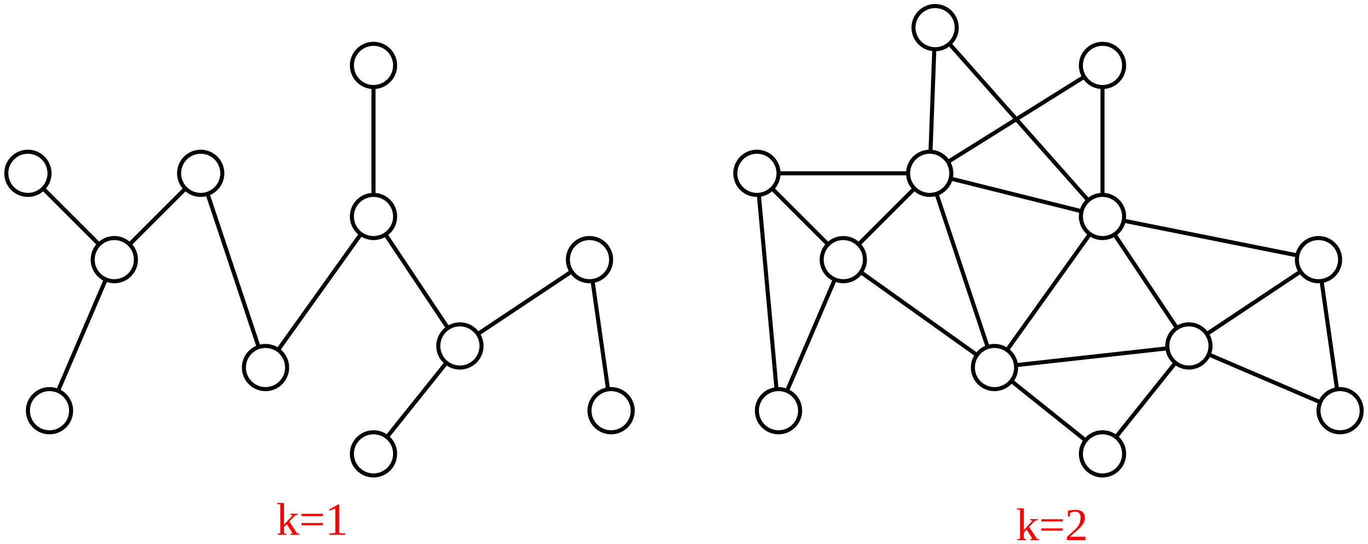
Motivation: Multivariate density estimation

Goal: to model the dependencies between a set of random variables



Hypertrees

Use Markov networks. Control complexity by limiting tree-width k.



Weight of a hyperedge (clique) quantifies the importance of modeling the dependencies between the variables in the hyperedge.

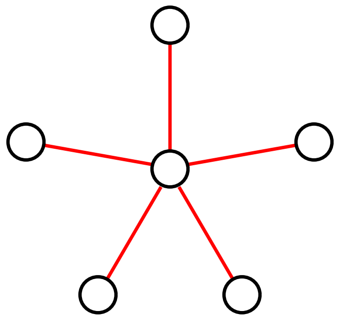
The maximum hypertree problem:

Input: weights of all candidate k-hyperedges

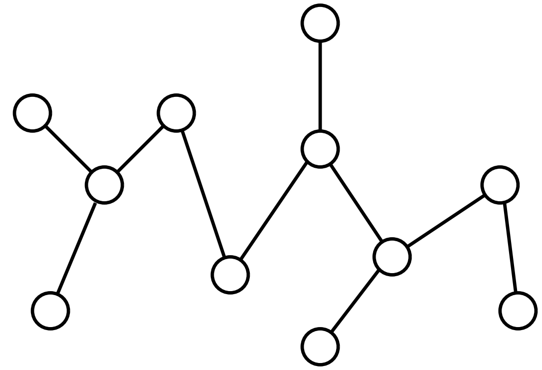
Output: a maximum weight k-hypertree

1-windmill farms

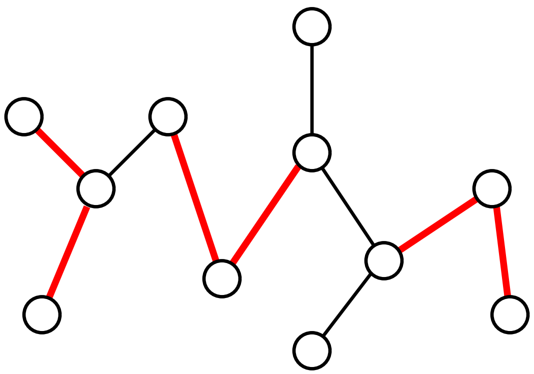
A windmill farm is a subset of the hyperedges of a hypertree.



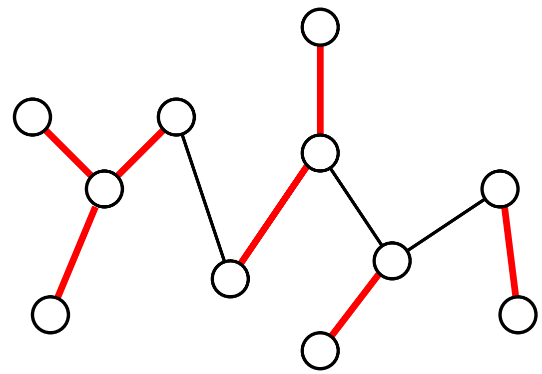
a 1-windmill (star)



a tree

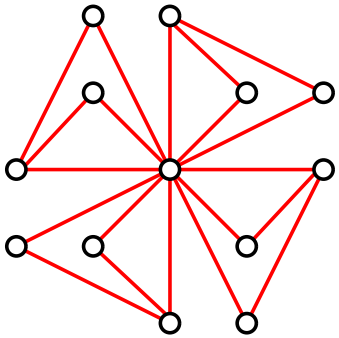


a 1-windmill farm in the tree

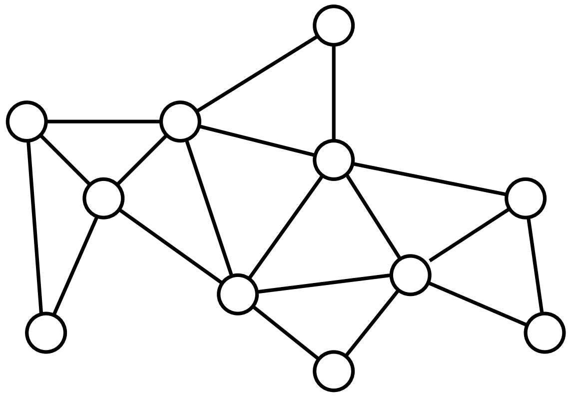


a 1-windmill farm in the tree

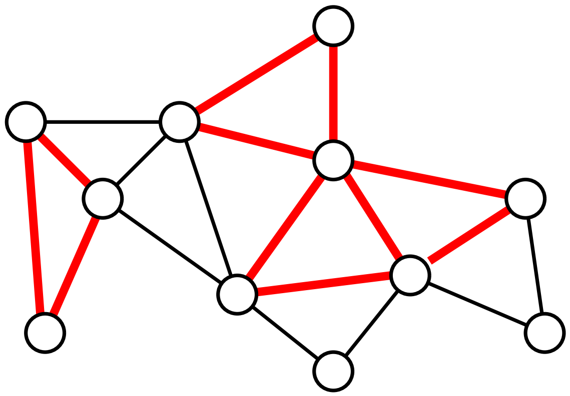
2-windmill farms



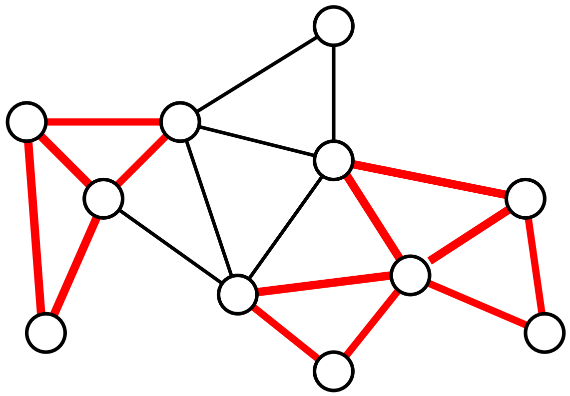
a 2-windmill



a 2-hypertree



a 2-windmill farm in a 2-hypertree



a 2-windmill farm in a 2-hypertree

Using windmills to approximate hypertrees

- A linear programming relaxation finds a windmill farm with weight $\frac{1}{8^k k!}$ of the maximum windmill farm
- The maximum windmill farm captures at least $\frac{1}{(k+1)!}$ of the weight of a hypertree
- Conclusion: The LP-based algorithm can find a hypertree with weight $\frac{1}{8^k k!(k+1)!}$ of the optimal hypertree

Analyzing the windmill coverage ratio

C_k = the fraction of the weight of a k-hypertree that can be captured by a maximum weight k-windmill farm

$$\underbrace{\frac{1}{(k+1)!}}_{\text{Previous lower bound}} \leq C_k \leq \underbrace{\frac{1}{k+1}}_{\text{Previous upper bound}}$$

Question: What is C_k?

Approach: find the "worst" hypertrees for which the weight of the maximum windmill farm is minimized

Analyzing the windmill coverage

Assume all weights are non-negative and weight of the hypertree $w(T) = 1$.

1. Given a weighted hypertree (T, w), find the maximum weight windmill farm F.

$$C_k(T, w) = \max_{F \subset T} w(F)$$

Analyzing the windmill coverage

Assume all weights are non-negative and weight of the hypertree $w(T) = 1$.

1. Given a weighted hypertree (T, w), find the maximum weight windmill farm F. $C_k(T, w) = \max_{F \subset T} w(F)$

2. Given an unweighted hypertree structure T, find the "worst" weights w. $C_k(T) = \min_{w} \max_{F \subset T} w(F)$

Analyzing the windmill coverage

Assume all weights are non-negative and weight of the hypertree $w(T) = 1$.

1. Given a weighted hypertree (T, w), find the maximum weight windmill farm F. $C_k(T, w) = \max_{F \subset T} w(F)$
2. Given an unweighted hypertree structure T, find the "worst" weights w. $C_k(T) = \min_{w} \max_{F \subset T} w(F)$
3. Find the "worst" weighted hypertree (T, w). $C_k = \inf_{T} \min_{w} \max_{F \subset T} w(F)$

Analyzing the windmill coverage

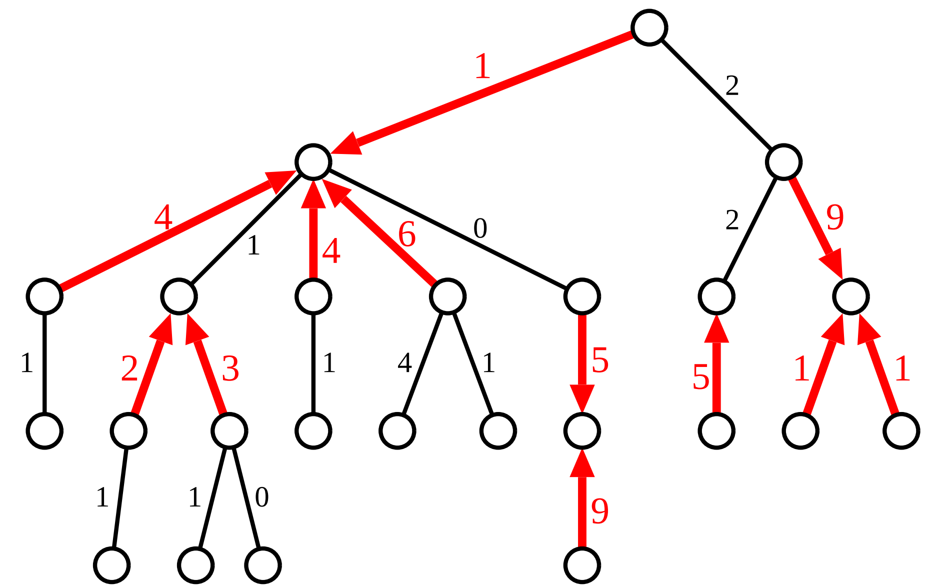
Assume all weights are non-negative and weight of the hypertree $w(T) = 1$.

1. Given a weighted hypertree (T, w), find the maximum weight windmill farm F. $$C_k(T, w) = \max_{F \subset T} w(F)$$
2. Given an unweighted hypertree structure T, find the "worst" weights w. $$C_k(T) = \min_{w} \max_{F \subset T} w(F)$$
3. Find the "worst" weighted hypertree (T, w). $$C_k = \inf_{T} \min_{w} \max_{F \subset T} w(F)$$

Plan: solve problems 1, 2, and 3 for trees and then for hypertrees.

Problem 1: $C_{k=1}(T, w) = \max_{F \subset T} w(F)$

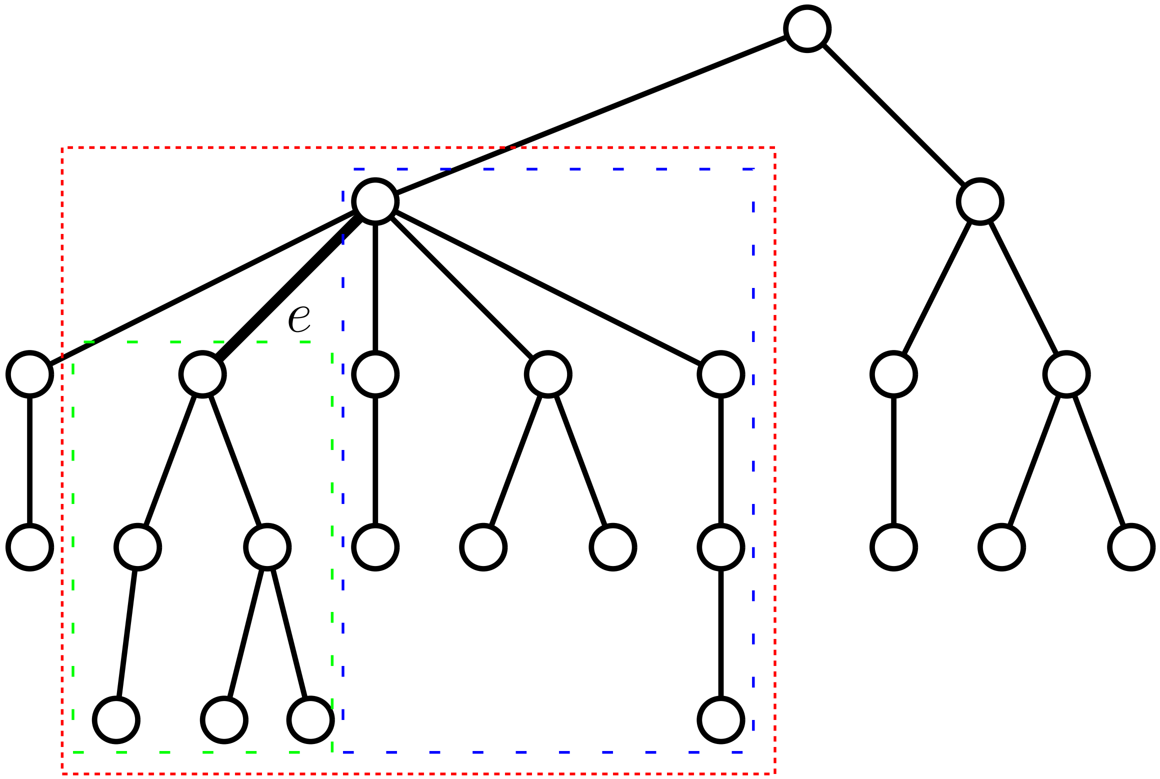
Goal: find the maximum weight windmill farm in a weighted tree.



$$w(F) = 1 + 4 + 4 + 6 + 2 + 3 + 5 + 9 + 9 + 5 + 1 + 1 = 50$$

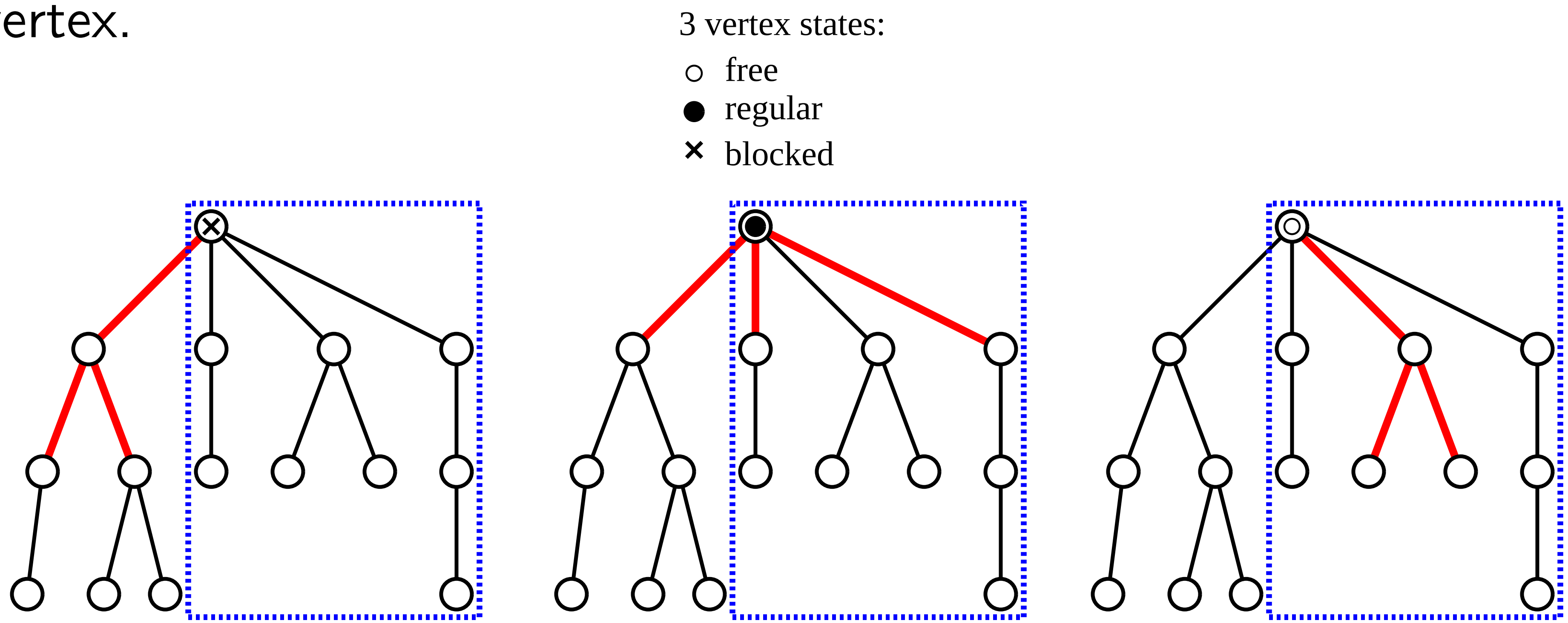
Problem 1: $C_{k=1}(T, w) = \max_{F \subset T} w(F)$

Solve using dynamic programming:



Problem 1: $C_{k=1}(T, w) = \max_{F \subset T} w(F)$

Find the maximum weight windmill farm given the state of the root vertex.

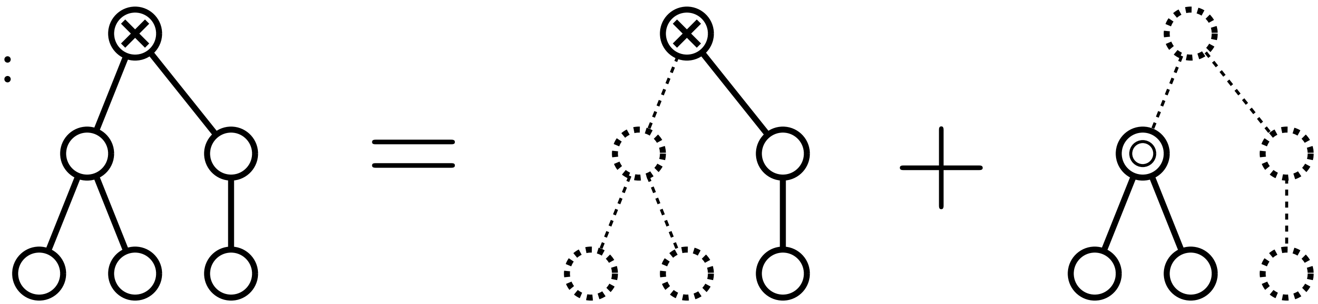


$f_{v,i,s}$ = maximum weight of a 1-windmill farm in subtree (v, i) with vertex v in state s

Problem 1: $C_{k=1}(T, w) = \max_{F \subset T} w(F)$

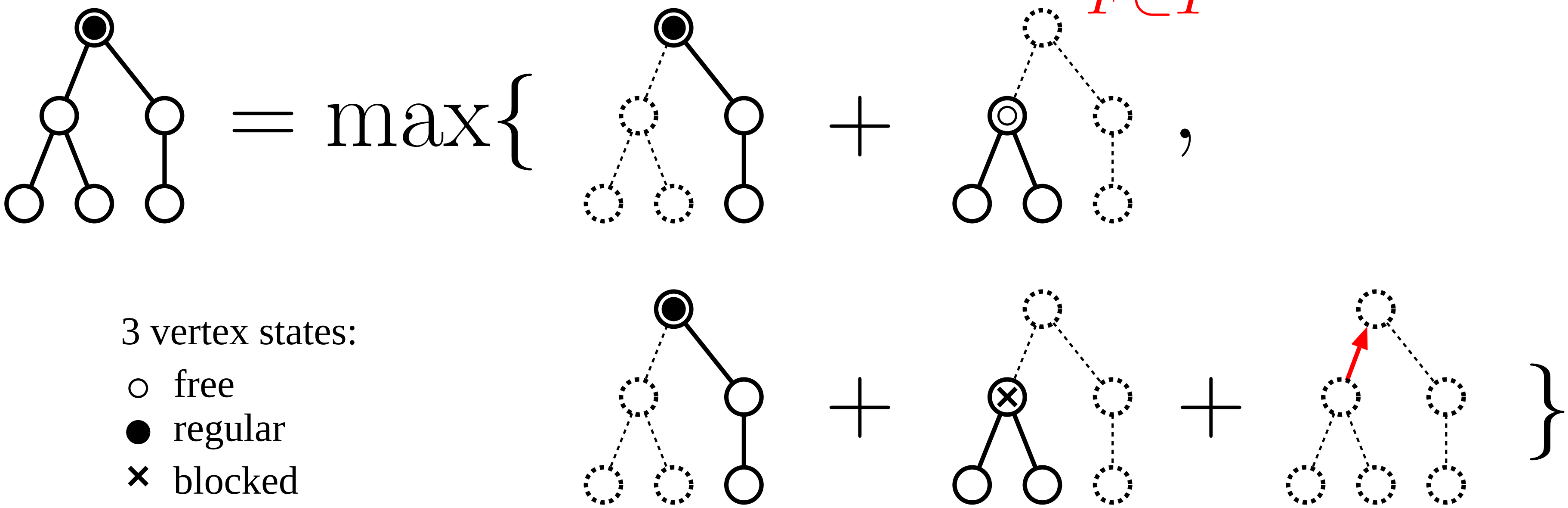
3 vertex states:

- ○ free
- ● regular
- × blocked



$$f_{v,i,\times} = f_{v,i+1,\times} + f_{c_i,1,\circ}$$

Problem 1: $C_{k=1}(T,w) = \max_{F \subset T} w(F)$

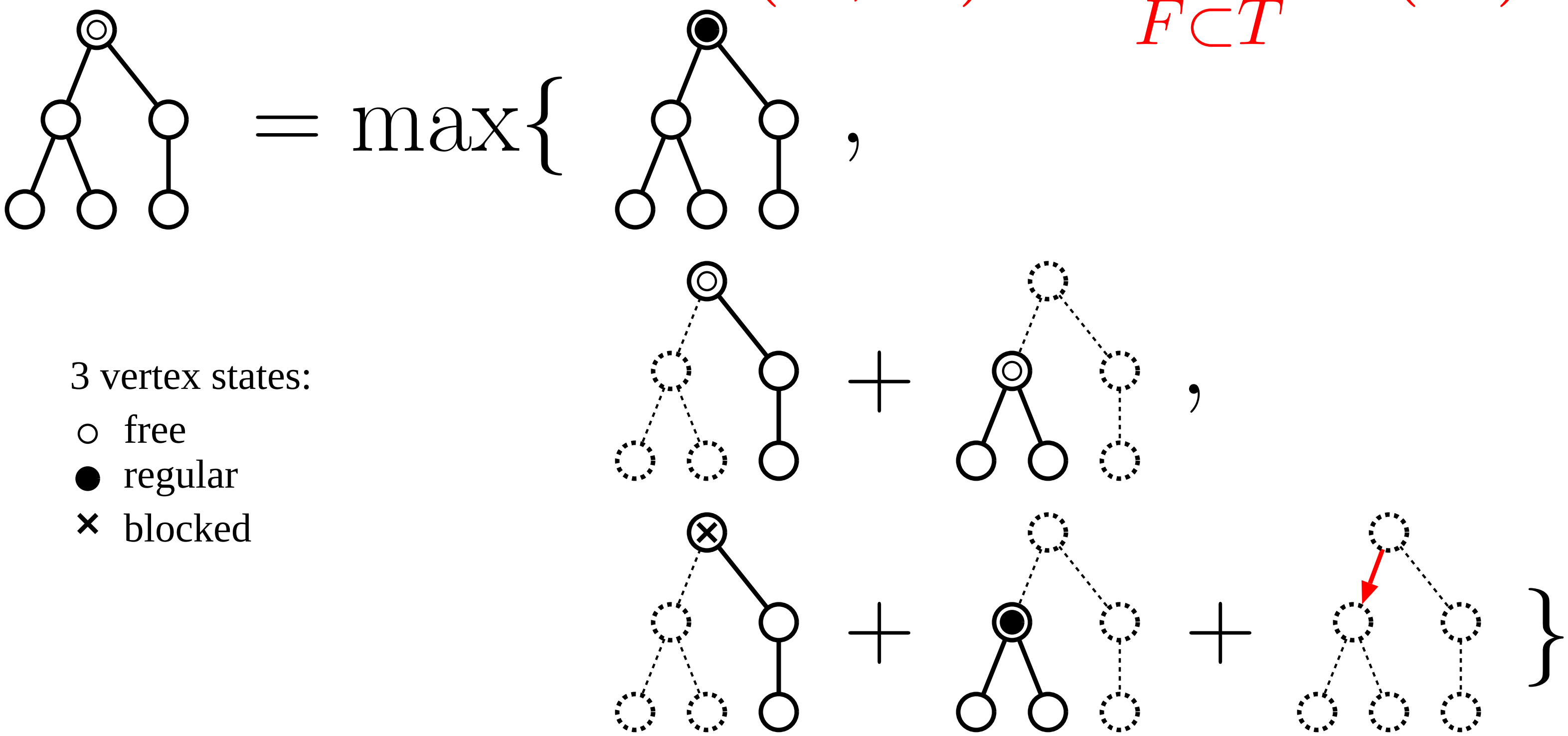


3 vertex states:

- ○ free
- ● regular
- × blocked

$$f_{v,i,\bullet} = \max\{\, f_{v,i+1,\bullet} + f_{c_i,1,\circ},\quad f_{v,i+1,\bullet} + f_{c_i,1,\times} + w_{v,c_i} \,\}$$

Problem 1: $C_{k=1}(T, w) = \max_{F \subset T} w(F)$



$$f_{v,i,\circ} = \max\{\; f_{v,i,\bullet},\; f_{v,i+1,\circ} + f_{c_i,1,\circ},\; f_{v,i+1,\times} + f_{c_i,1,\bullet} + w_{v,c_i} \;\}$$

Problem 1: $C_{k=1}(T, w) = \max_{F \subset T} w(F)$

Compute all dynamic programming states $f_{v,i,s}$ in $O(|V|)$ time:

$$
\begin{aligned}
f_{v,i,\times} &= \quad f_{v,i+1,\times} + f_{c_i,1,\circ} \\
f_{v,i,\bullet} &= \max\{\, f_{v,i+1,\bullet} + f_{c_i,1,\circ}, \\
&\qquad\quad f_{v,i+1,\bullet} + f_{c_i,1,\times} + w_{v,c_i} \,\} \\
f_{v,i,\circ} &= \max\{\, f_{v,i,\bullet}, \\
&\qquad\quad f_{v,i+1,\circ} + f_{c_i,1,\circ}, \\
&\qquad\quad f_{v,i+1,\times} + f_{c_i,1,\bullet} + w_{v,c_i} \,\}
\end{aligned}
$$

Problem 2: $C_{k=1}(T) = \min_{w} \max_{F \subset T} w(F)$

Preliminary step: convert the dynamic program into an equivalent linear program.

Compute $f_{\text{root}(T),1,\circ}$

$$
\begin{aligned}
f_{v,i,\times} &= f_{v,i+1,\times} + f_{c_i,1,\circ} \\
f_{v,i,\bullet} &= \max\{\, f_{v,i+1,\bullet} + f_{c_i,1,\circ}, \\
&\qquad f_{v,i+1,\bullet} + f_{c_i,1,\times} + w_{v,c_i} \,\} \\
f_{v,i,\circ} &= \max\{\, f_{v,i,\bullet}, \\
&\qquad f_{v,i+1,\circ} + f_{c_i,1,\circ}, \\
&\qquad f_{v,i+1,\times} + f_{c_i,1,\bullet} + w_{v,c_i} \,\}
\end{aligned}
$$

Problem 2: $C_{k=1}(T) = \min_{w} \max_{F \subset T} w(F)$

Preliminary step: convert the dynamic program into an equivalent linear program.

Compute $f_{\text{root}(T),1,\circ}$

$$
\begin{aligned}
f_{v,i,\times} &= \quad f_{v,i+1,\times} + f_{c_i,1,\circ} \\
f_{v,i,\bullet} &= \max\{ f_{v,i+1,\bullet} + f_{c_i,1,\circ}, \\
&\qquad f_{v,i+1,\bullet} + f_{c_i,1,\times} + w_{v,c_i} \} \\
f_{v,i,\circ} &= \max\{ f_{v,i,\bullet}, \\
&\qquad f_{v,i+1,\circ} + f_{c_i,1,\circ}, \\
&\qquad f_{v,i+1,\times} + f_{c_i,1,\bullet} + w_{v,c_i} \}
\end{aligned}
$$

$\Leftrightarrow$

Minimize $f_{\text{root}(T),1,\circ}$

$$
\begin{aligned}
f_{v,i,\times} &\geq f_{v,i+1,\times} + f_{c_i,1,\circ} \\
f_{v,i,\bullet} &\geq f_{v,i+1,\bullet} + f_{c_i,1,\circ} \\
f_{v,i,\bullet} &\geq f_{v,i+1,\bullet} + f_{c_i,1,\times} + w_{v,c_i} \\
f_{v,i,\circ} &\geq f_{v,i,\bullet} \\
f_{v,i,\circ} &\geq f_{v,i+1,\circ} + f_{c_i,1,\circ} \\
f_{v,i,\circ} &\geq f_{v,i+1,\times} + f_{c_i,1,\bullet} + w_{v,c_i}
\end{aligned}
$$

Problem 2: $C_{k=1}(T) = \min_{w} \max_{F \subset T} w(F)$

Preliminary step: convert the dynamic program into an equivalent linear program.

Compute $f_{\text{root}(T),1,\circ}$

$$
\begin{aligned}
f_{v,i,\times} &= \phantom{\max\{}\ f_{v,i+1,\times} + f_{c_i,1,\circ} \\
f_{v,i,\bullet} &= \max\{\ f_{v,i+1,\bullet} + f_{c_i,1,\circ}, \\
&\qquad\quad f_{v,i+1,\bullet} + f_{c_i,1,\times} + w_{v,c_i}\ \} \\
f_{v,i,\circ} &= \max\{\ f_{v,i,\bullet}, \\
&\qquad\quad f_{v,i+1,\circ} + f_{c_i,1,\circ}, \\
&\qquad\quad f_{v,i+1,\times} + f_{c_i,1,\bullet} + w_{v,c_i}\ \}
\end{aligned}
$$

$\Leftrightarrow$

Minimize $f_{\text{root}(T),1,\circ}$

$$
\begin{aligned}
f_{v,i,\times} &\geq f_{v,i+1,\times} + f_{c_i,1,\circ} \\
f_{v,i,\bullet} &\geq f_{v,i+1,\bullet} + f_{c_i,1,\circ} \\
f_{v,i,\bullet} &\geq f_{v,i+1,\bullet} + f_{c_i,1,\times} + w_{v,c_i} \\
f_{v,i,\circ} &\geq f_{v,i,\bullet} \\
f_{v,i,\circ} &\geq f_{v,i+1,\circ} + f_{c_i,1,\circ} \\
f_{v,i,\circ} &\geq f_{v,i+1,\times} + f_{c_i,1,\bullet} + w_{v,c_i}
\end{aligned}
$$

$$\max_{F \subset T} w(F) = \min_{Af \geq Bw} f_{\text{root}(T),1,\circ}$$

Problem 2: $C_{k=1}(T) = \min_{w} \max_{F \subset T} w(F)$

$$\max_{F \subset T} w(F) = \qquad\qquad \min_{f : Af \geq Bw} f_{\mathsf{root}(T),1,\circ}$$

Problem 2: $C_{k=1}(T) = \min_{w} \max_{F \subset T} w(F)$

$$\min_{\substack{w \\ w \geq 0; \sum w_i = 1}} \max_{F \subset T} w(F) = \min_{\substack{w \\ w \geq 0; \sum w_i = 1}} \min_{f : Af \geq Bw} f_{\mathsf{root}(T),1,\circ}$$

Problem 2: $C_{k=1}(T) = \min_{w} \max_{F \subset T} w(F)$

$$\min_{\substack{w \\ w \geq 0; \sum w_i = 1}} \max_{F \subset T} w(F) = \min_{\substack{w \\ w \geq 0; \sum w_i = 1}} \min_{f : Af \geq Bw} f_{\mathsf{root}(T),1,\circ}$$

A single linear program:

$$\min_{\substack{w,f \\ w \geq 0; \sum w_i = 1; Af \geq Bw}} f_{\mathsf{root}(T),1,\circ}$$

Problem 3: $C_{k=1} = \inf_{T} \min_{w} \max_{F \subset T} w(F)$

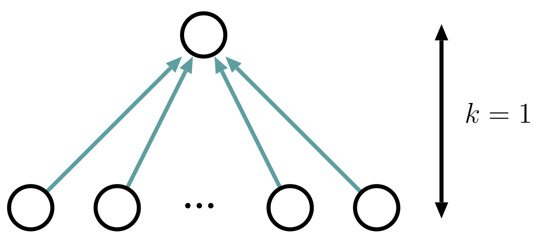
- Observation: A weighted tree with a weight 0 edge is equivalent to a weighted tree without the edge
- Construct a family of tree structures $\{T_{b,h} \mid b, h = 1, 2, 3, \ldots\}$ (branching factor b, height h) that contains each tree structure
- $C_{k=1} = \lim_{b,h \to \infty} C_{k=1}(T_{b,h})$

We solve the linear program and get $C_{k=1} = \frac{1}{2}$

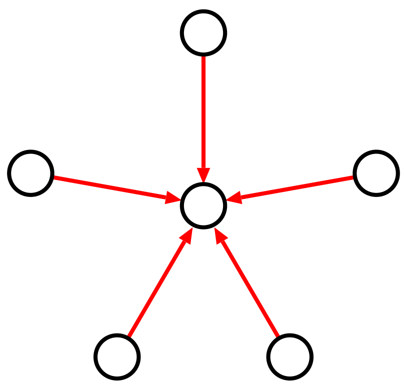
k-windmill farms (definition)

Hyperedges of windmill $=$ root-to-leaf paths in representing tree

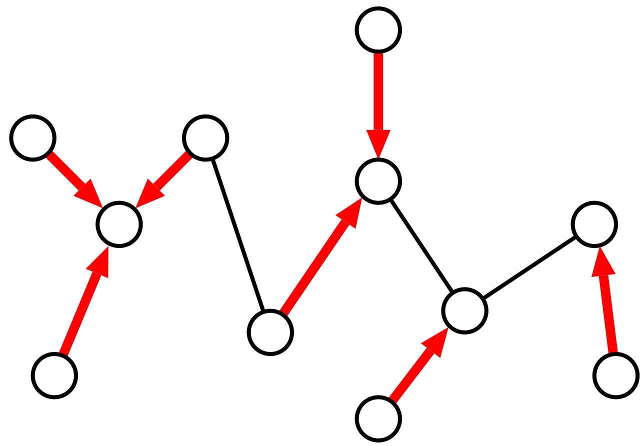
$k = 1$



representing tree of a 1-windmill



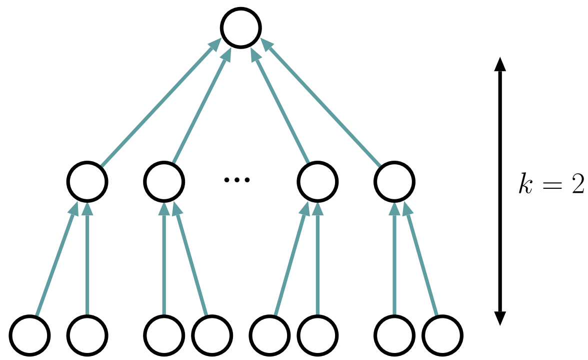
1-windmill



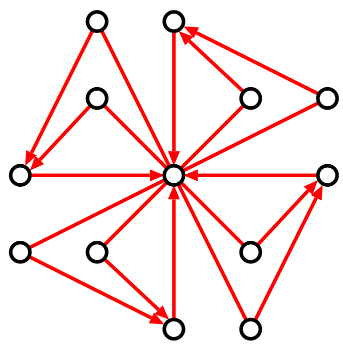
1-windmill farm in a tree

k-windmill farms (definition)

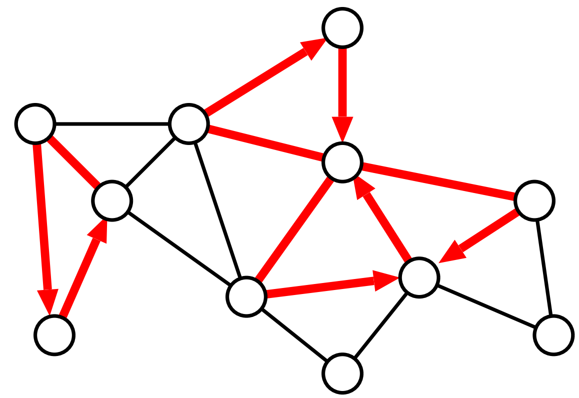
$k = 2$



representing tree of a 2-windmill



2-windmill

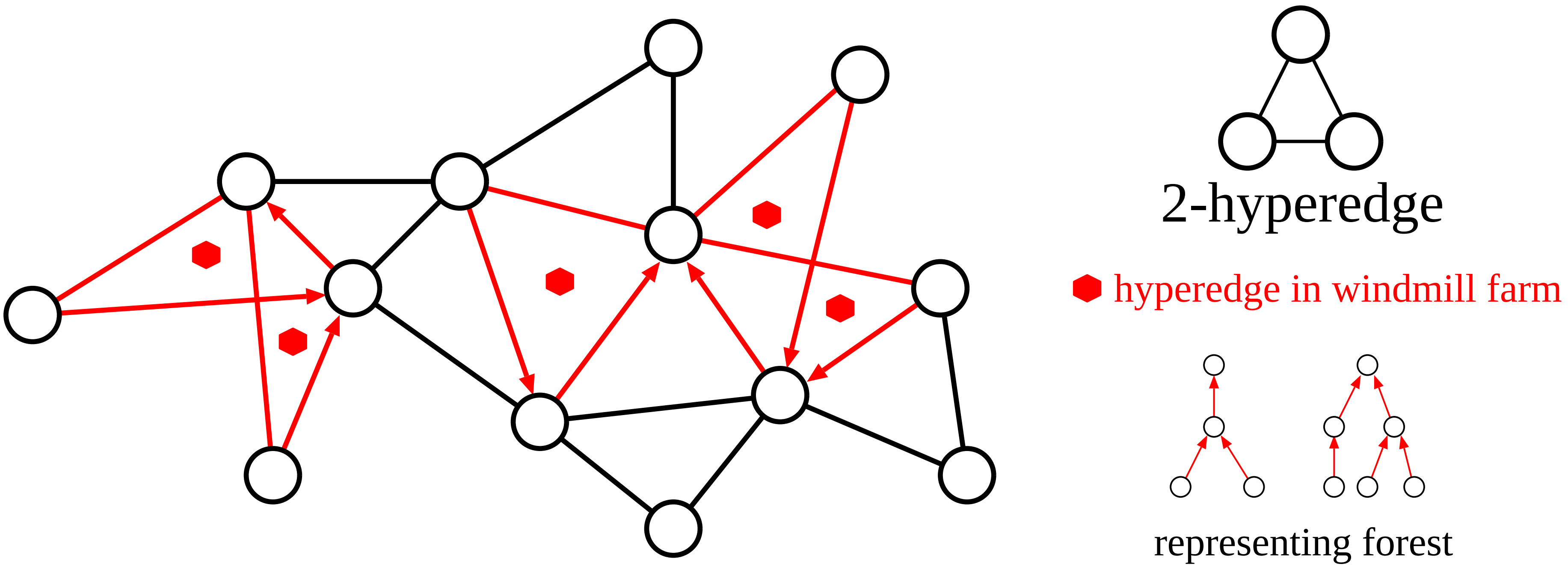


2-windmill farm in a hypertree

Problem 1: $C_k(T, w) = \max_{F \subset T} w(F)$

Analyze the windmill coverge for *hypertrees*.

$k = 2$

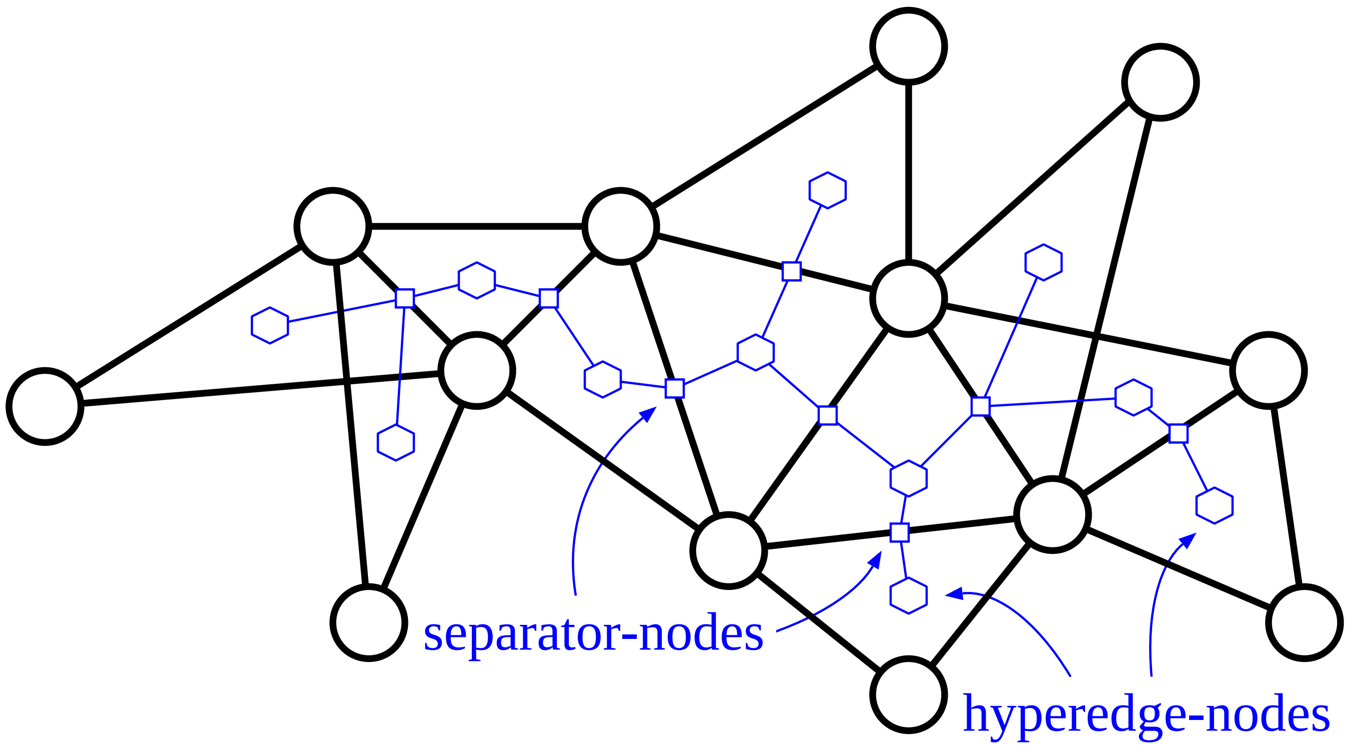


How do we decompose a hypertree?

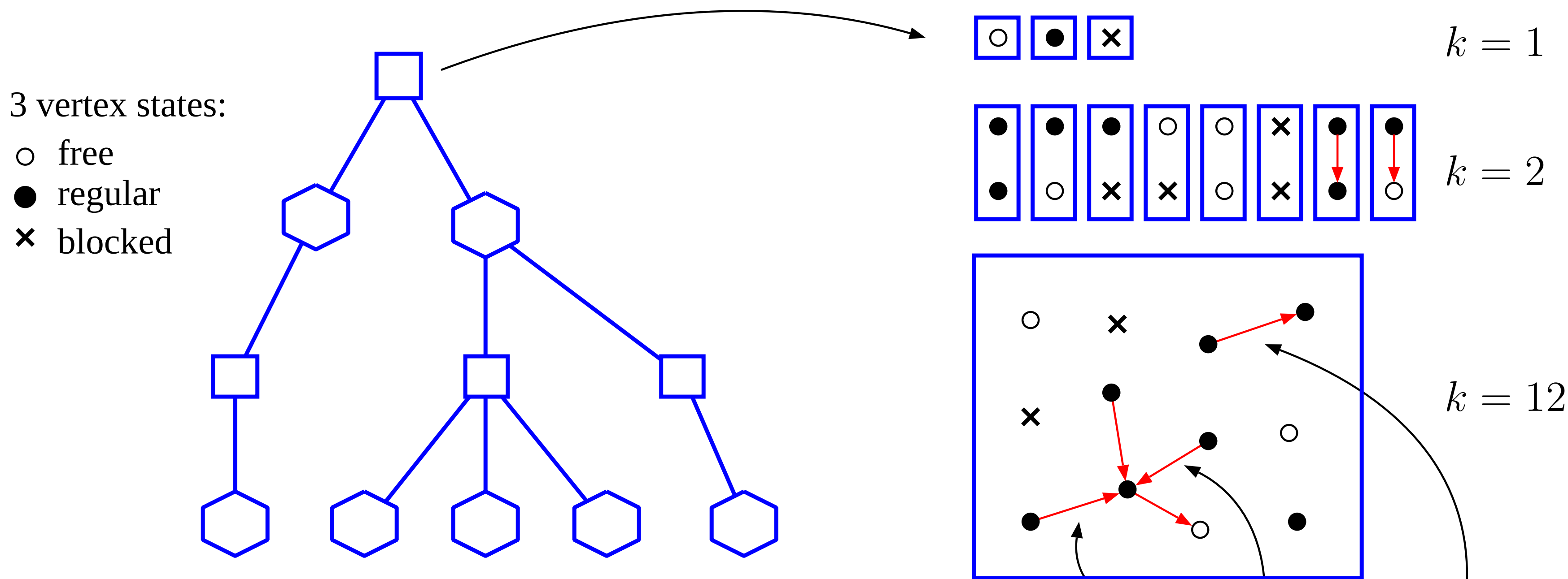
Problem 1: $C_k(T, w) = \max_{F \subset T} w(F)$

Incidence tree structure: represents how the hypertree is connected

$k = 2$



Problem 1: $C_k(T, w) = \max_{F \subset T} w(F)$



incidence tree structure

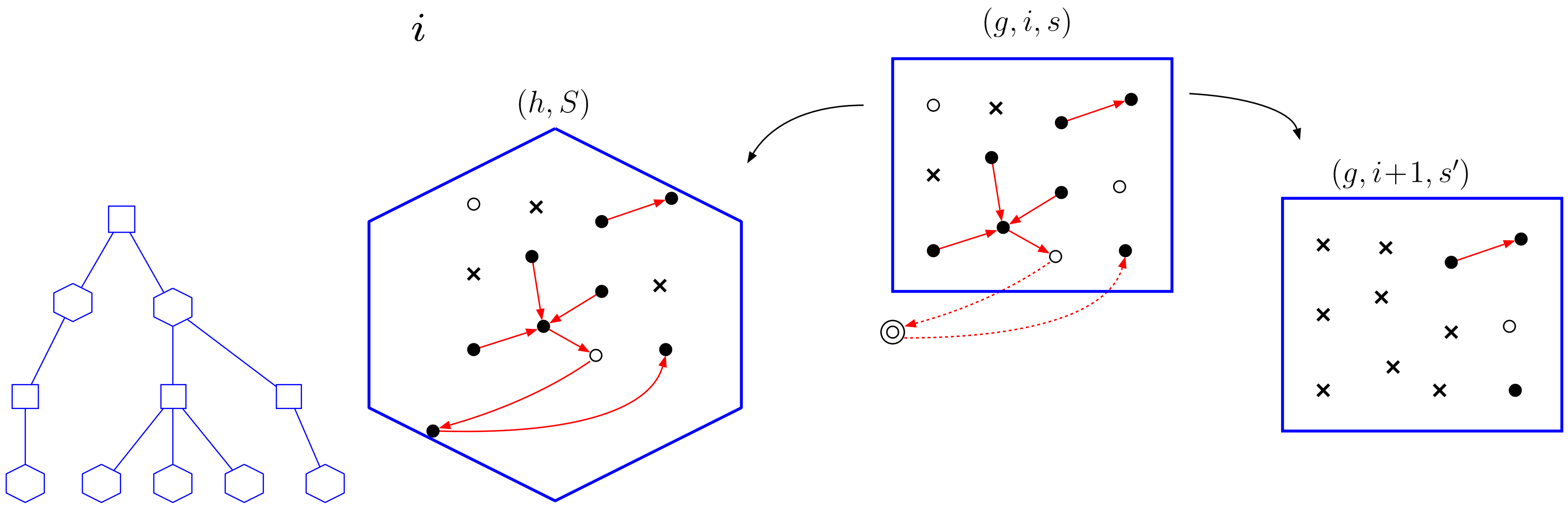
representing forest of the windmill farm

Problem 1: $C_k(T, w) = \max_{F \subset T} w(F)$

Dynamic programming states: $f_{g,i,s}$, $f_{h,S}$

$$f_{g,i,s} = \max_{s \to S} \left\{ f_{g,i+1,s'} + f_{h,S} + w(h)[[S \text{ is a path}]] \right\}$$

$$f_{h,S} = \sum_i f_{g_i,1,\text{restrict}(S,g_i)}$$



Problem 2: $C_k(T) = \min_w \max_{F \subset T} w(F)$

Apply the duality technique from before.

$$\min_{\substack{w \\ w \geq 0; \sum w_i = 1}} \max_{F \subset T} w(F) = \min_{\substack{w \\ w \geq 0; \sum w_i = 1}} \min_{s, f : Af \geq Bw} f_{\text{root}(T), 1, s}$$

A single linear program:

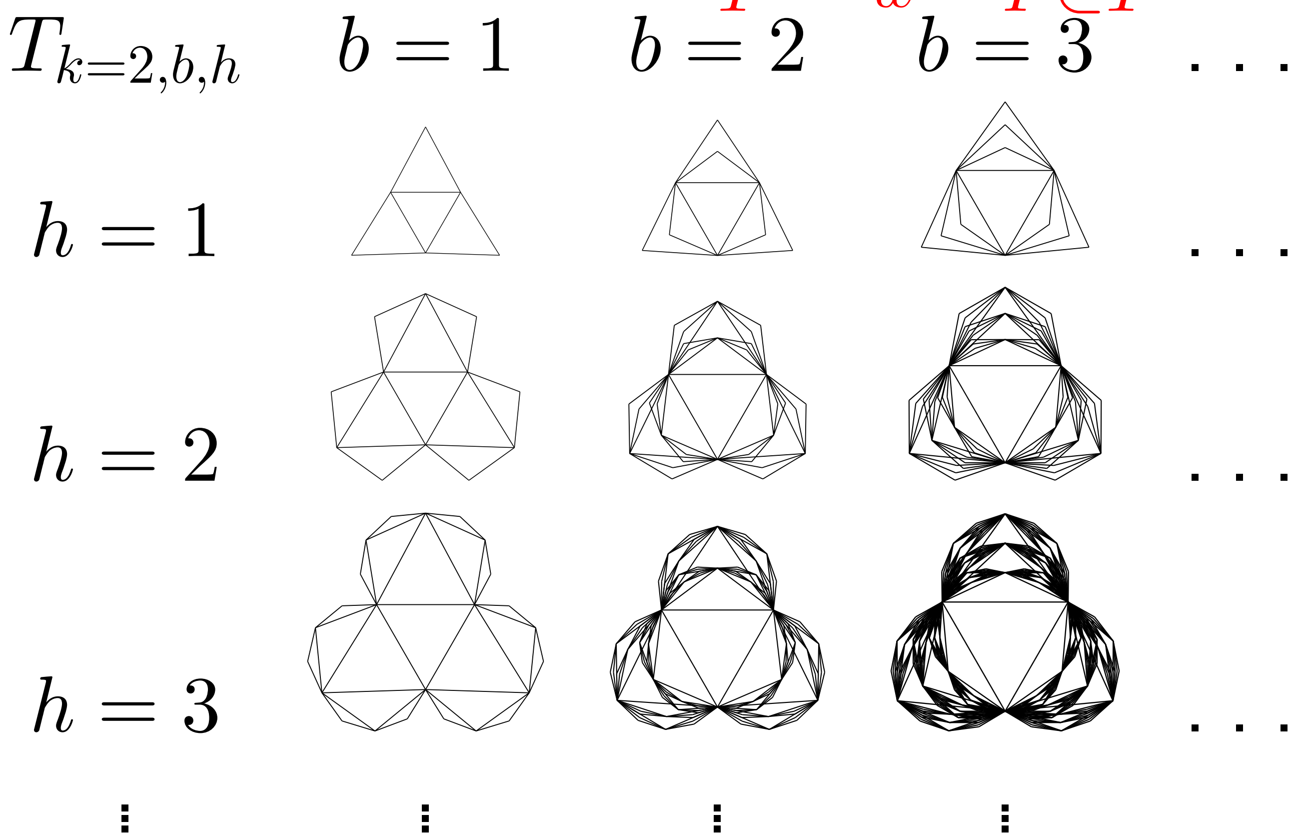
$$\min_{\substack{w, f \\ w \geq 0; \sum w_i = 1; Af \geq Bw}} f_{\text{root}(T), 1, \circ}$$

Problem 3: $C_k = \inf_T \min_w \max_{F \subset T} w(F)$

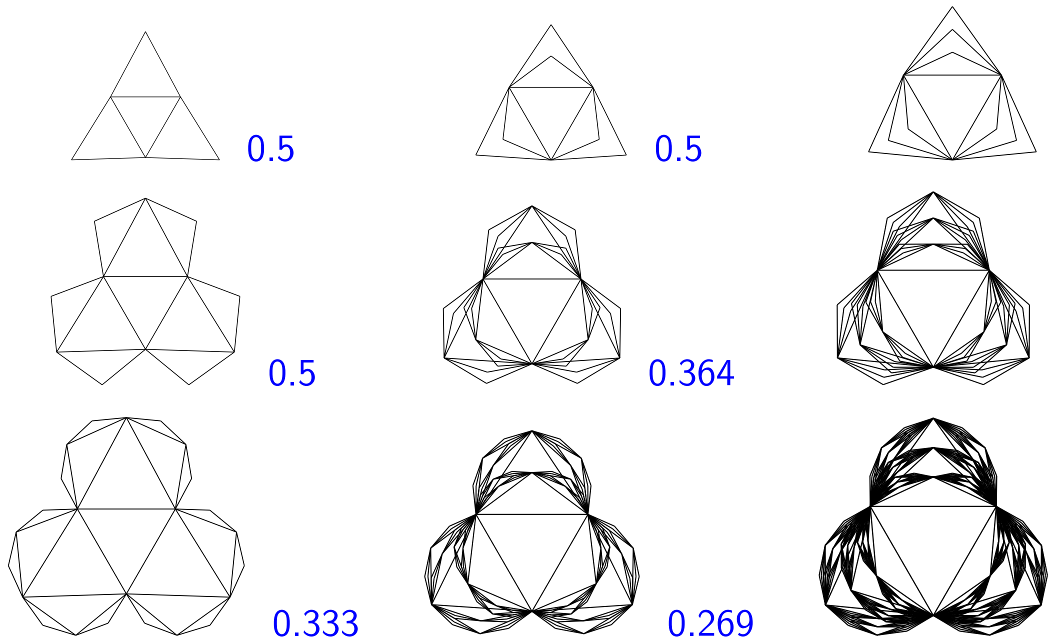
Construct a family of hypertrees $\{T_{k,b,h}\}$ such that:

- Each hypertree is contained in some $T_{k,b,h}$ (branching factor b, height h)
- $C_k = \lim_{b,h \to \infty} C_k(T_{k,b,h})$

Problem 3: $C_k = \inf_T \min_w \max_{F \subset T} w(F)$



Problem 3: $C_k = \inf_T \min_w \max_{F \subset T} w(F)$



$T_{k=2,b,h}$	$b = 1$	$b = 2$	$b = 3$	$\ldots$
$h = 1$	0.5	0.5	0.5	$\ldots$
$h = 2$	0.5	0.364	0.308	$\ldots$
$h = 3$	0.333	0.269	0.263	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Converges to $C_{k=2}$.

Achieving a tighter upper bound

- Use weights obtained from the LP solution to construct a sequence of *weighted* hypertrees $\{(T_{k,h}, w_{k,h})\}$
- Compute $\lim_{h \to \infty} C_k(T_{k,h}, w_{k,h})$ (involves solving Problem 1)

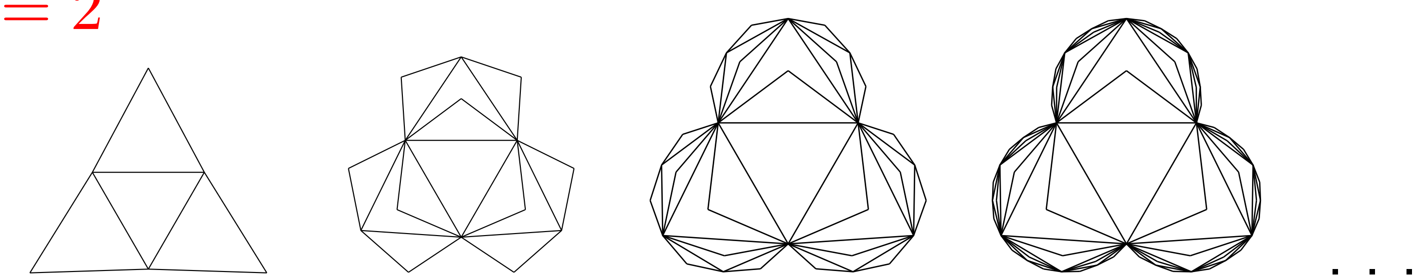
$w_{k,h}$: weight of a hyperedge is $2^{-\text{height of hyperedge}}$

Achieving a tighter upper bound

- Use weights obtained from the LP solution to construct a sequence of *weighted* hypertrees $\{(T_{k,h}, w_{k,h})\}$
- Compute $\lim_{h\to\infty} C_k(T_{k,h}, w_{k,h})$ (involves solving Problem 1)

$w_{k,h}$: weight of a hyperedge is $2^{-\text{height of hyperedge}}$

$k = 2$



$T_{k=2,h=1}$	$T_{k=2,h=2}$	$T_{k=2,h=3}$	$T_{k=2,h=4}$	...	
0.5	0.353	0.308	0.286	...	
2/4	6/17	16/52	40/140	...	$\frac{2h+2}{9h-1} \longrightarrow \frac{2}{9}$

Achieving a tighter upper bound

$$C_k = \min_{T,w} C_k(T,w) = \min_{T} \min_{w} \max_{F \subset T} w(F)$$

k	$\leq C_k$ Windmill Cover Theorem	C_k	$\lim_{h\to\infty} C_k(T_{k,h}, w_{k,h}) \geq C_k$	$\geq C_k$ Previous upper bound
1	0.5	0.5	0.5	0.5
2	0.166666. . .	?	0.2222222. . .	0.33333. . .
3	0.041666. . .	?	0.0953932. . .	0.25
4	0.008333. . .	?	0.0515625	0.2
5	0.001389. . .	?	0.0258048	0.16666. . .
6	0.000198. . .	?	0.0123157. . .	0.14286. . .
k	$1/(k+1)!$	?	$< 1/2^k$?	$1/(k+1)$

Conclusions

- Motivation: using windmill farms to approximate the maximum likelihood Markov network
- We described an algorithmic technique for providing bounds on the windmill farm coverage

Conclusions

- Motivation: using windmill farms to approximate the maximum likelihood Markov network
- We described an algorithmic technique for providing bounds on the windmill farm coverage
- The exact windmill coverage C_k is open for $k > 1$
- Future work: apply the duality technique to other problems (shortest path, minimum cut)