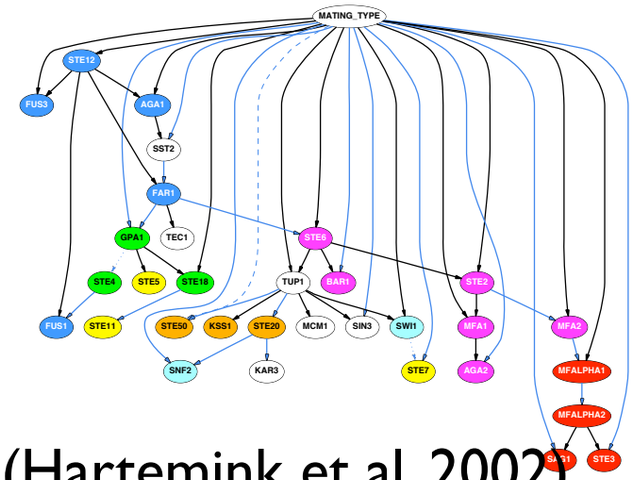
Graphs and polytopes: learning Bayesian networks with LP relaxations

Tommi Jaakkola
MIT CSAIL

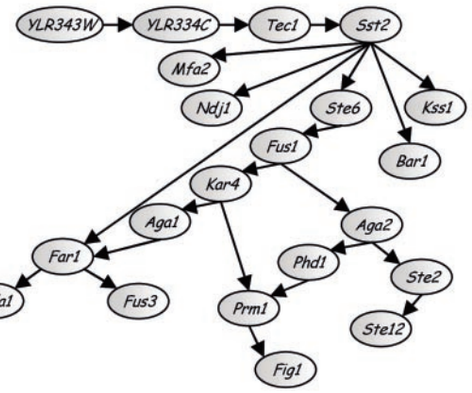
based on joint work with
David Sontag, MIT
Amir Globerson, HUJI
Marina Meila, U Washington

Learning Bayesian networks

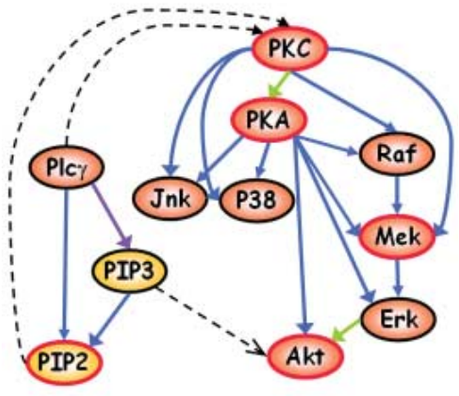
- Bayesian networks are widely used as modeling tools across applied areas, including computational biology



(Hartemink et al. 2002)



(Friedman 2004)



(Sachs et al. 2005)

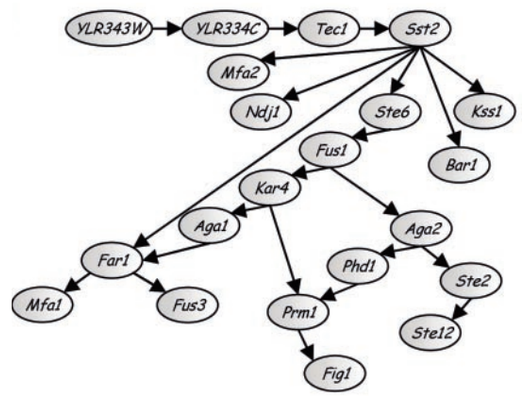
...

- Learned structures (especially causal) can lead to useful insights about the domain

Structure learning: basics

$$D = \begin{array}{cccccccccccc} 2 & 2 & 0 & 1 & 1 & 2 & 0 & 0 & 2 & 0 & 2 & \ldots \\ 0 & 2 & 2 & 2 & 2 & 2 & 0 & 1 & 1 & 2 & 2 & \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & \\ 2 & 2 & 0 & 1 & 0 & 2 & 0 & 0 & 2 & 0 & 2 & \\ 0 & 2 & 2 & 2 & 2 & 2 & 0 & 1 & 1 & 2 & 2 & \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & \\ & & & & & \ldots & & & & & & \end{array}$$

complete data



highest scoring
acyclic graph

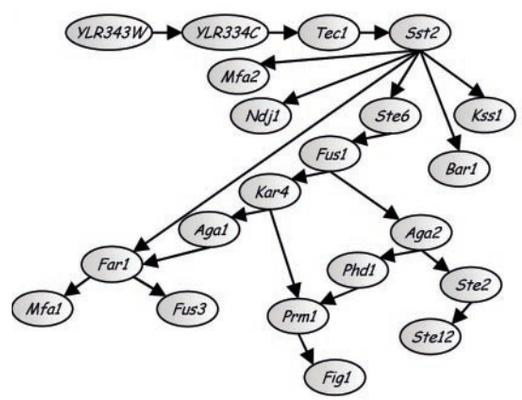
Structure learning: basics

$$D = \begin{array}{cccccccccccc} 2 & 2 & 0 & 1 & 1 & 2 & 0 & 0 & 2 & 0 & 2 & \ldots \\ 0 & 2 & 2 & 2 & 2 & 2 & 0 & 1 & 1 & 2 & 2 & \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & \\ 2 & 2 & 0 & 1 & 0 & 2 & 0 & 0 & 2 & 0 & 2 & \\ 0 & 2 & 2 & 2 & 2 & 2 & 0 & 1 & 1 & 2 & 2 & \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & \\ & & & & & \ldots & & & & & & \end{array}$$

complete data

$$\text{score}(G; D) = \sum_{i=1}^{n} \text{score}(i|pa_i, D)$$

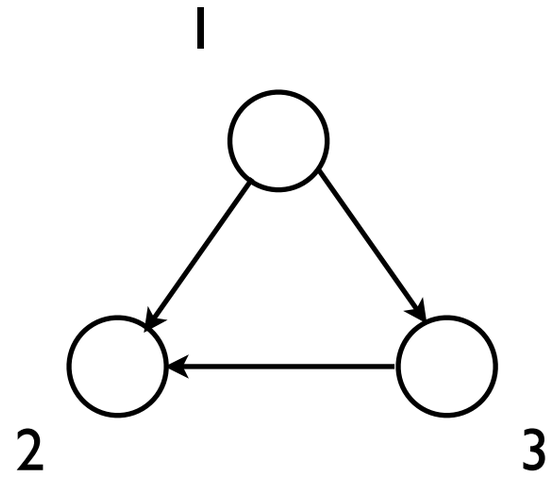
decomposable scoring function



$$\arg\max_{G} \text{score}(G; D)$$

highest scoring **acyclic** graph

Structure learning as inference



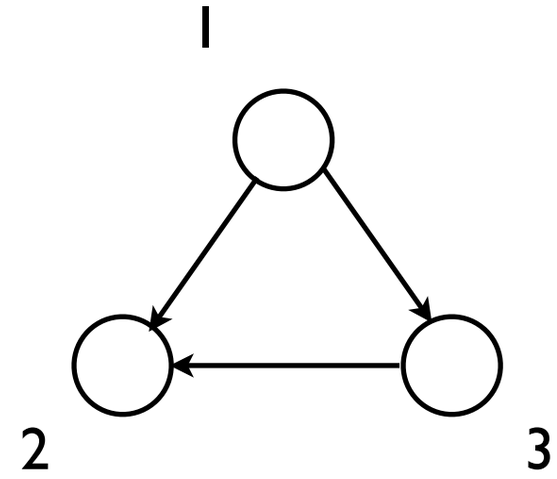
- Each node can select a subset of the other nodes as "parents"

Structure learning as inference

$$pa_1 \in \begin{matrix} \emptyset \\ \{2\} \\ \{3\} \\ \{2,3\} \end{matrix}$$

$$pa_2 \in \begin{matrix} \emptyset \\ \{1\} \\ \{3\} \\ \{1,3\} \end{matrix}$$

$$pa_3 \in \begin{matrix} \emptyset \\ \{1\} \\ \{2\} \\ \{1,2\} \end{matrix}$$



- Each node can select a subset of the other nodes as "parents"

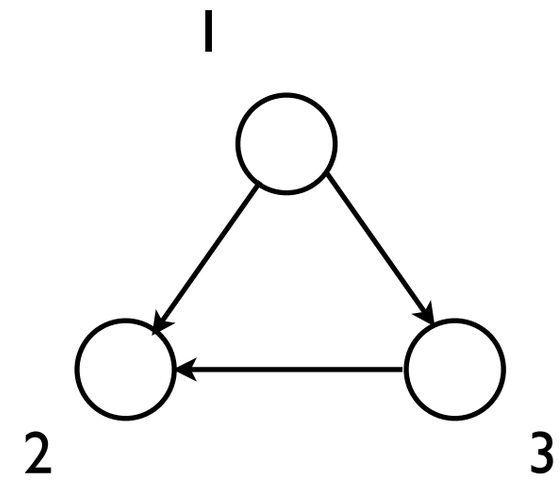
Structure learning as inference

$$pa_1 \in \begin{matrix} \{\emptyset\} \\ \{2\} \\ \{3\} \\ \{2,3\} \end{matrix}$$

$$pa_2 \in \begin{matrix} \emptyset \\ \{1\} \\ \{3\} \\ \{1,3\} \end{matrix}$$

$$pa_3 \in \begin{matrix} \emptyset \\ \{1\} \\ \{2\} \\ \{1,2\} \end{matrix}$$

$\Rightarrow$



- Each node can select a subset of the other nodes as "parents"

Structure learning as inference

$$pa_1 \in \begin{matrix} \{\emptyset\} \\ \{2\} \\ \{3\} \\ \{2,3\} \end{matrix} \quad \text{score}(1|\emptyset, D)$$

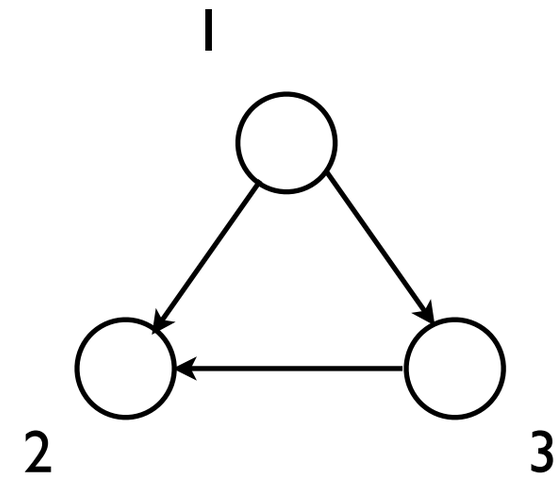
$$pa_2 \in \begin{matrix} \emptyset \\ \{1\} \\ \{3\} \\ \{1,3\} \end{matrix}$$

$\text{score}(2|\{1,3\}, D)$

$$pa_3 \in \begin{matrix} \emptyset \\ \{1\} \\ \{2\} \\ \{1,2\} \end{matrix}$$

$\text{score}(3|\{1\}, D)$

$\Rightarrow$



- Each node can select a subset of the other nodes as "parents"
- Each parent selection contributes a score; the goal is to maximize the sum of the scores (decomposable scoring metric)

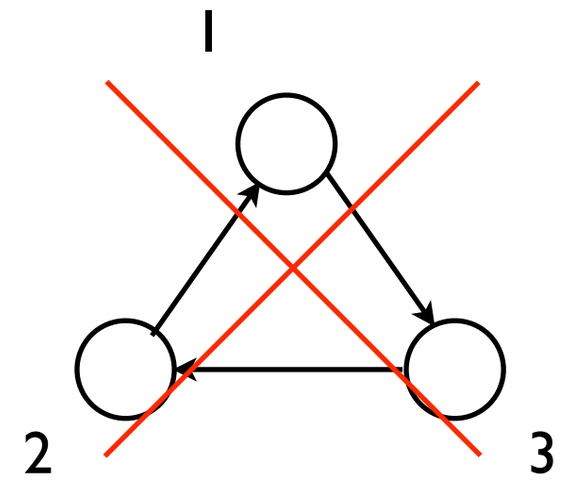
Structure learning as inference

$$pa_1 \in \begin{matrix} \emptyset \\ \{2\} \\ \{3\} \\ \{2,3\} \end{matrix}$$

$$pa_2 \in \begin{matrix} \emptyset \\ \{1\} \\ \{3\} \\ \{1,3\} \end{matrix}$$

$$pa_3 \in \begin{matrix} \emptyset \\ \{1\} \\ \{2\} \\ \{1,2\} \end{matrix}$$

$\Rightarrow$



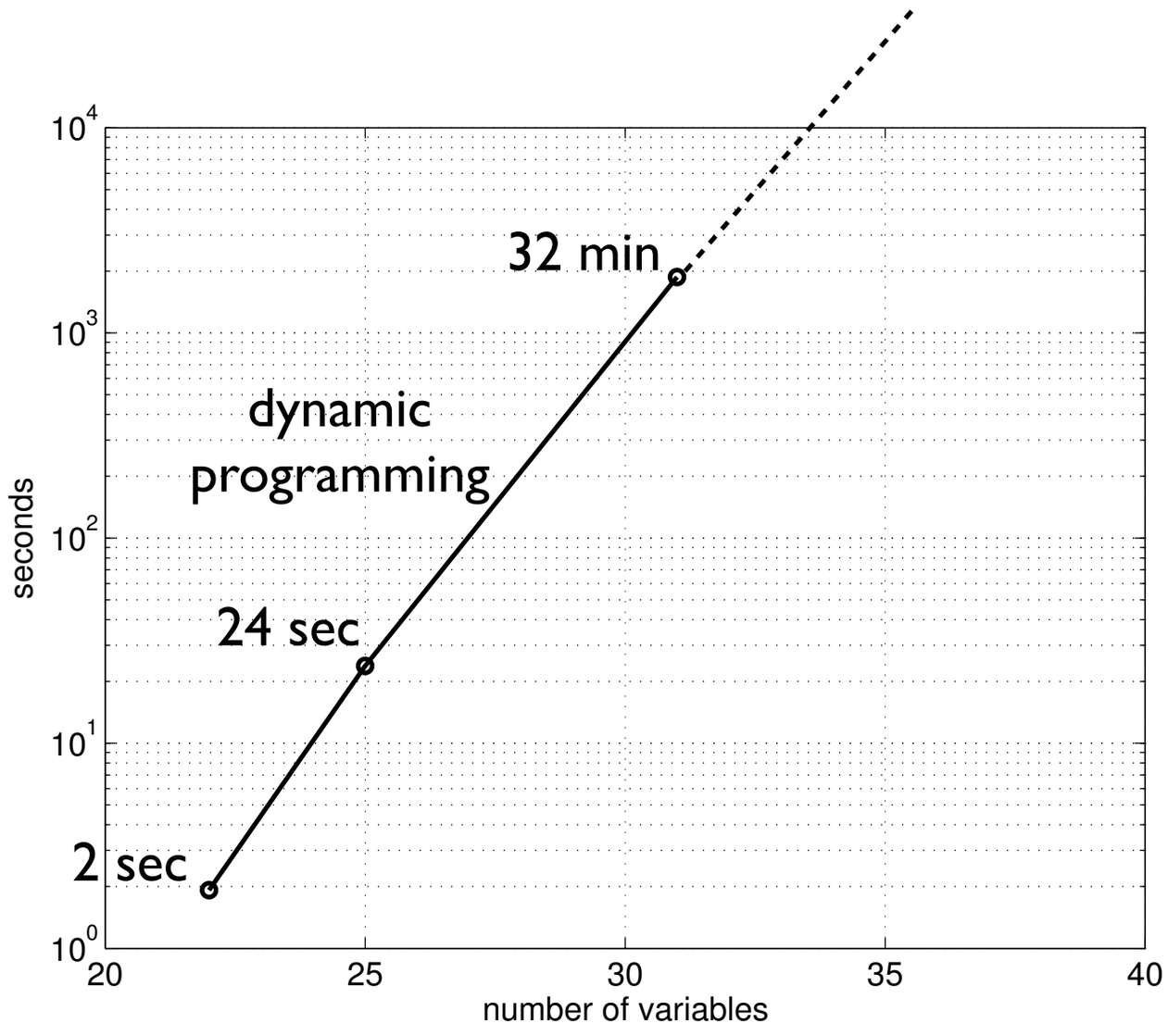
- Finding the highest scoring graph is **hard** because the graph has to be **acyclic** (the problem remains hard even if we limit the number of parents to two)

Score based approaches (briefly)

- Local search methods
  - stochastic search (e.g., Heckerman et al., 1995)
  - over equivalence classes (e.g., Chickering 2002)
  - order based search (e.g., Teyssier et al., 2005)
- Exact search methods
  - dynamic programming (e.g., Koivisto et al., 2004, Singh et al., 2005, Silander et al., 2006)
  - partial order covers (Parviainen et al., 2009)
  - branch and bound (e.g., de Campos et al., 2009)

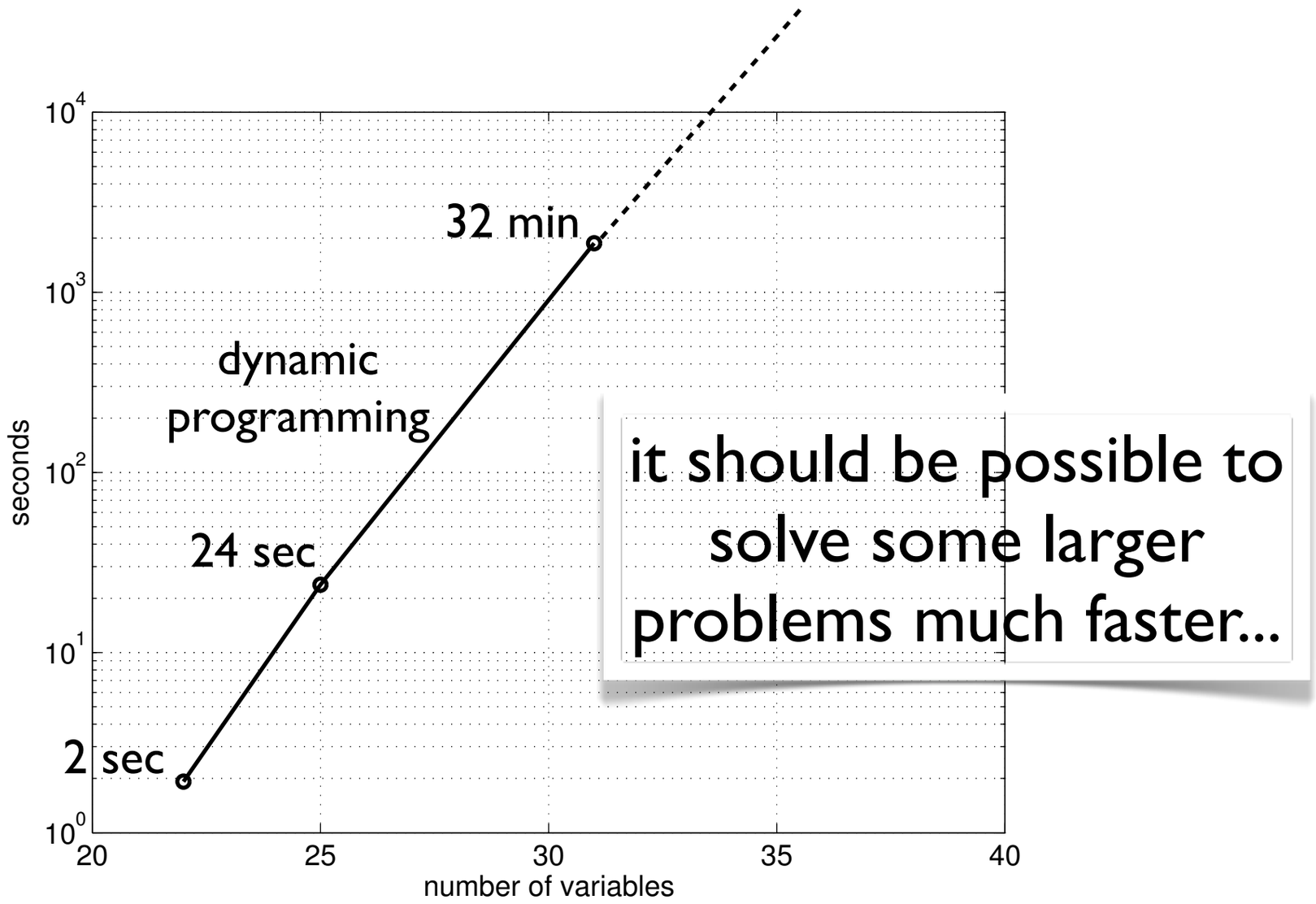
Exact methods

- Dynamic programming methods work well for small structure learning problems ...



Exact methods

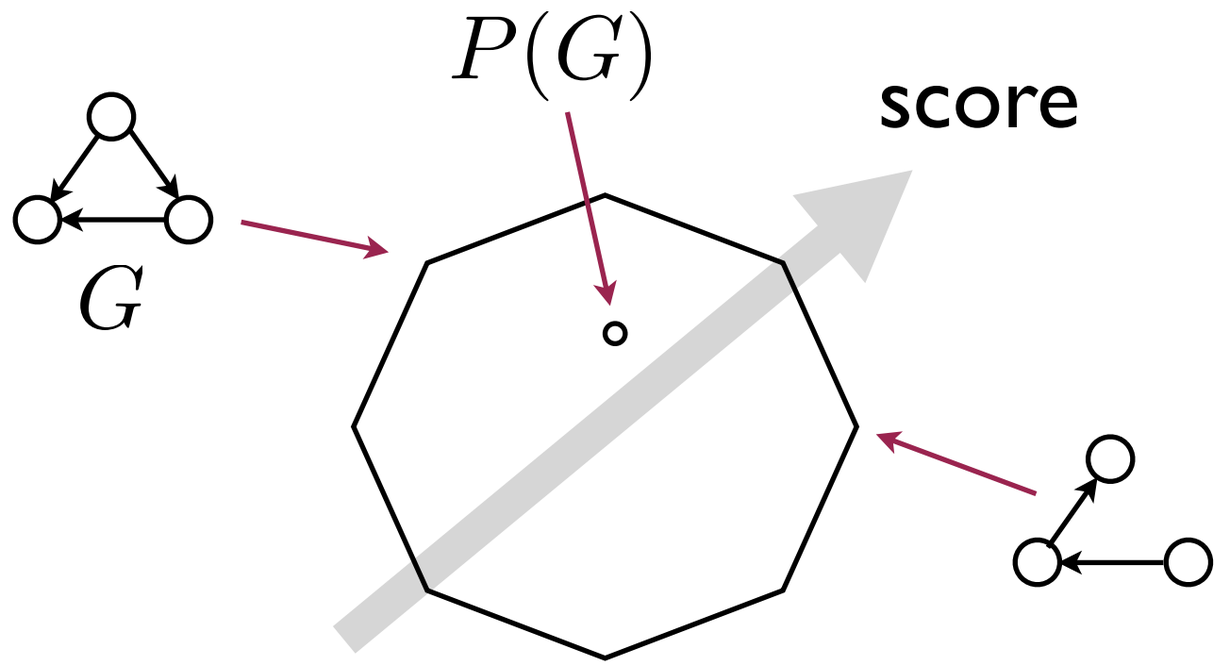
- Dynamic programming methods work well for small structure learning problems ...



it should be possible to solve some larger problems much faster...

Overview of our approach

- We reduce the search over graph structures to a linear program over a polytope representing acyclic graphs
  - each vertex corresponds to an acyclic graph
  - interior points correspond to distributions over graphs



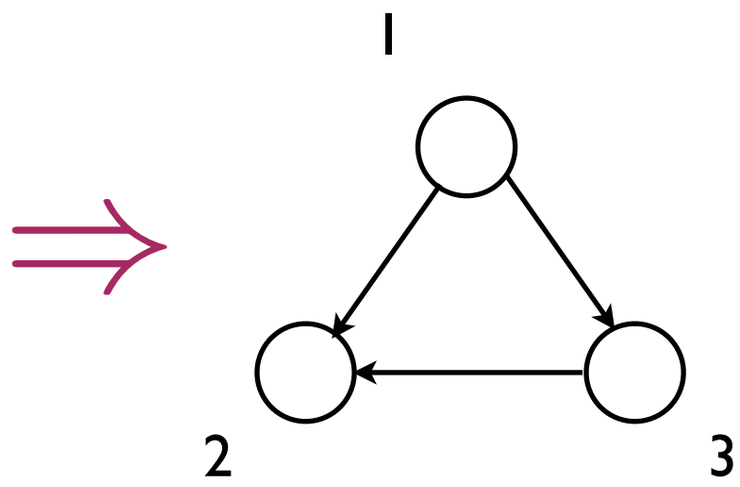
- Any solution obtained at a vertex is guaranteed to be optimal ("certificate of optimality")

Graphs and vectors

$$pa_1 \in \begin{matrix} \{\emptyset\} \\ \{2\} \\ \{3\} \\ \{2,3\} \end{matrix}$$

$$pa_2 \in \begin{matrix} \emptyset \\ \{1\} \\ \{3\} \\ \{1,3\} \end{matrix}$$

$$pa_3 \in \begin{matrix} \emptyset \\ \{1\} \\ \{2\} \\ \{1,2\} \end{matrix}$$

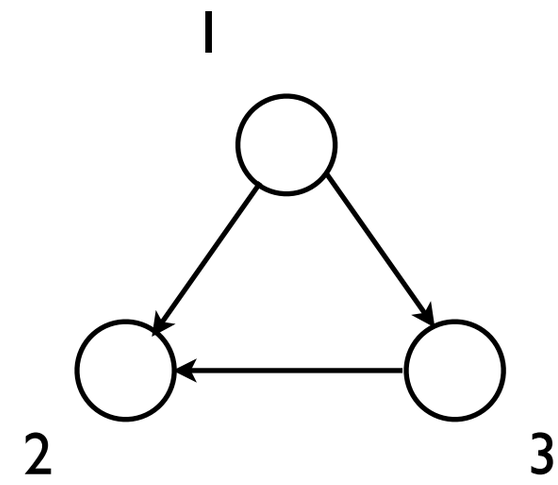


Graphs and vectors

$$pa_1 \in \begin{matrix} \{\emptyset\} \\ \{2\} \\ \{3\} \\ \{2,3\} \end{matrix}$$

$$pa_2 \in \begin{matrix} \emptyset \\ \{1\} \\ \{3\} \\ \{1,3\} \end{matrix}$$

$$pa_3 \in \begin{matrix} \emptyset \\ \{1\} \\ \{2\} \\ \{1,2\} \end{matrix}$$



An equivalent representation of the directed graph as a binary vector

$$\eta = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \hdashline 0 \\ 0 \\ 0 \\ 1 \\ \hdashline 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

parent set selection for variable 1

parent set selection for variable 2

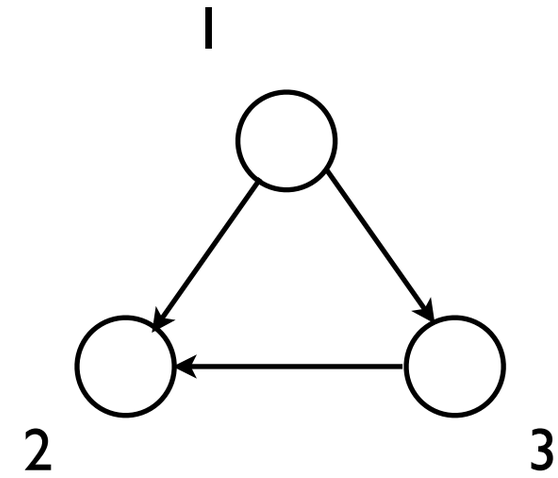
parent set selection for variable 3

Graphs and vectors

$$pa_1 \in \begin{matrix} \{\} \\ \{2\} \\ \{3\} \\ \{2,3\} \end{matrix} \quad \text{score}(1|\emptyset, D)$$

$$pa_2 \in \begin{matrix} \{\} \\ \{1\} \\ \{3\} \\ \{1,3\} \end{matrix} \quad \text{score}(2|\{1,3\}, D)$$

$$pa_3 \in \begin{matrix} \{\} \\ \{1\} \\ \{2\} \\ \{1,2\} \end{matrix} \quad \text{score}(3|\{1\}, D)$$



An equivalent representation of the directed graph as a binary vector

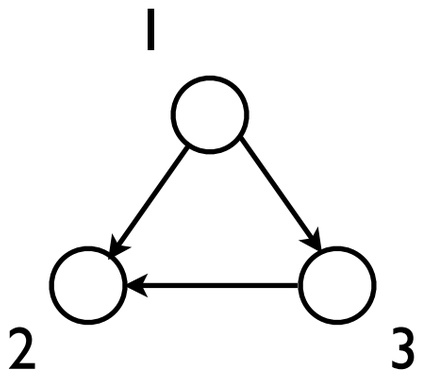
$$\eta = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \\ 1 \\ \hline 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

parent set selection for variable 1

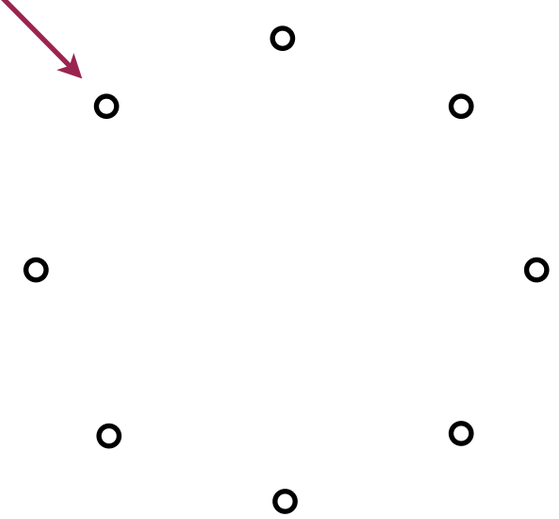
parent set selection for variable 2

parent set selection for variable 3

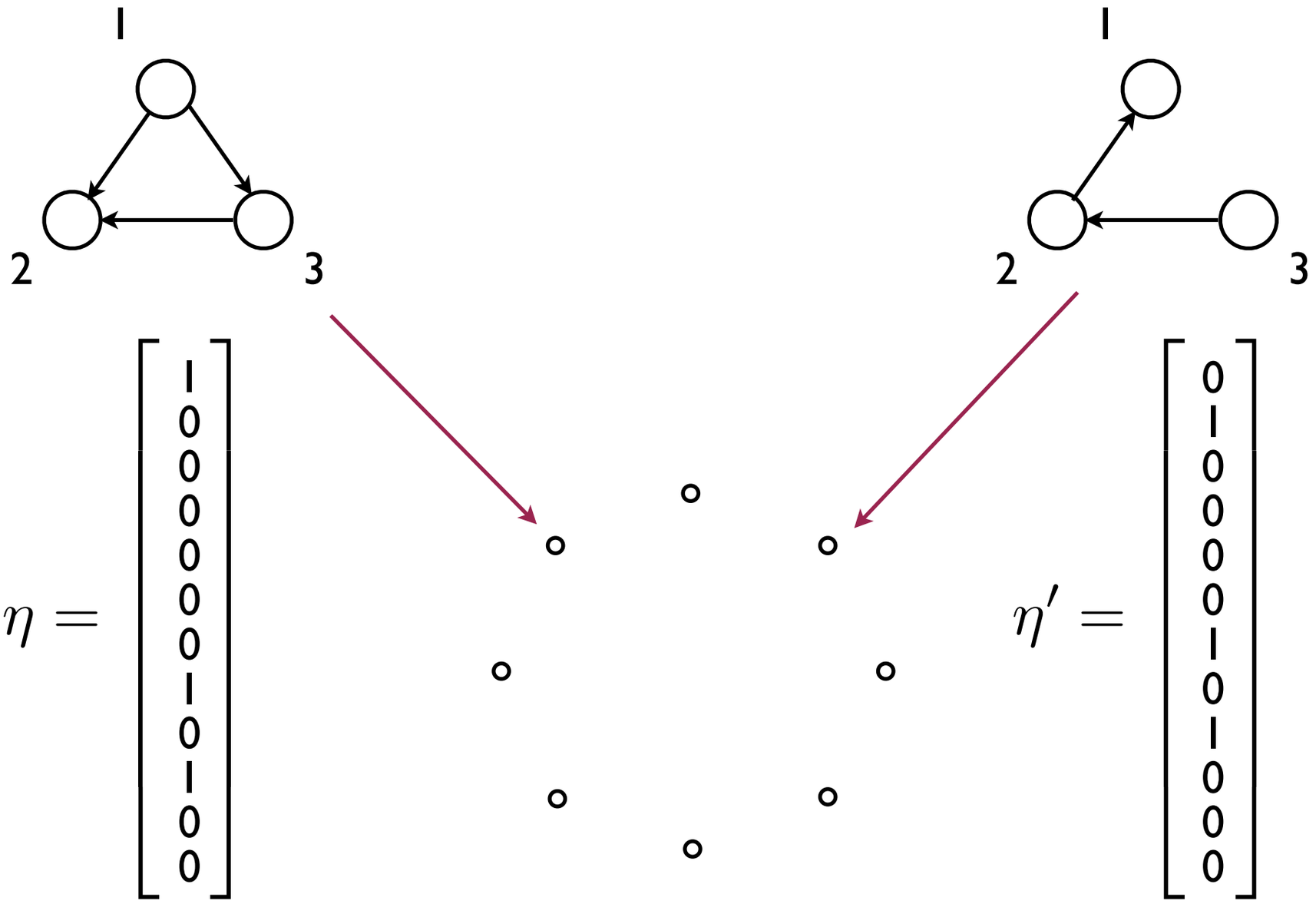
Graphs and polytopes



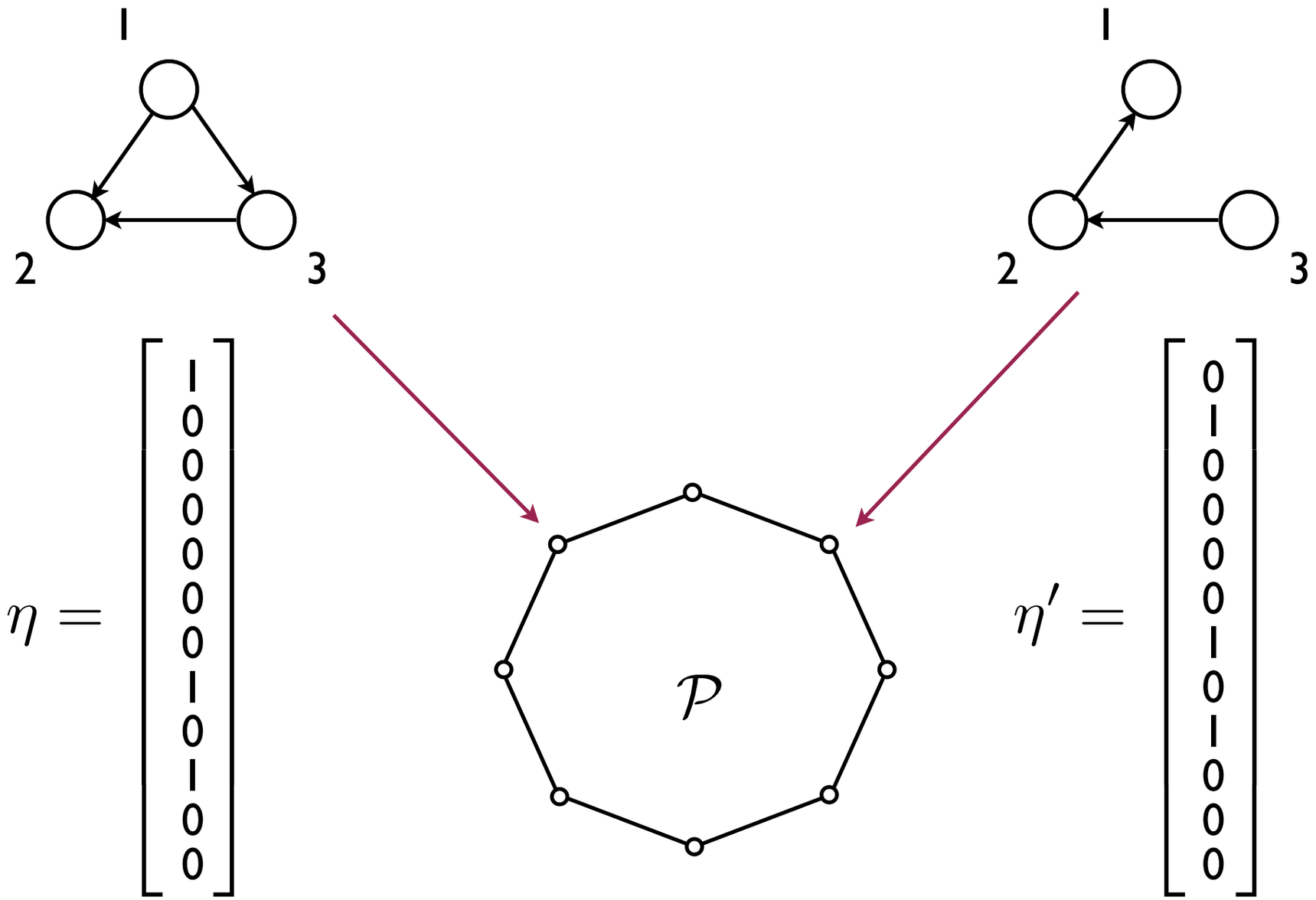
$$\eta = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$



Graphs and polytopes

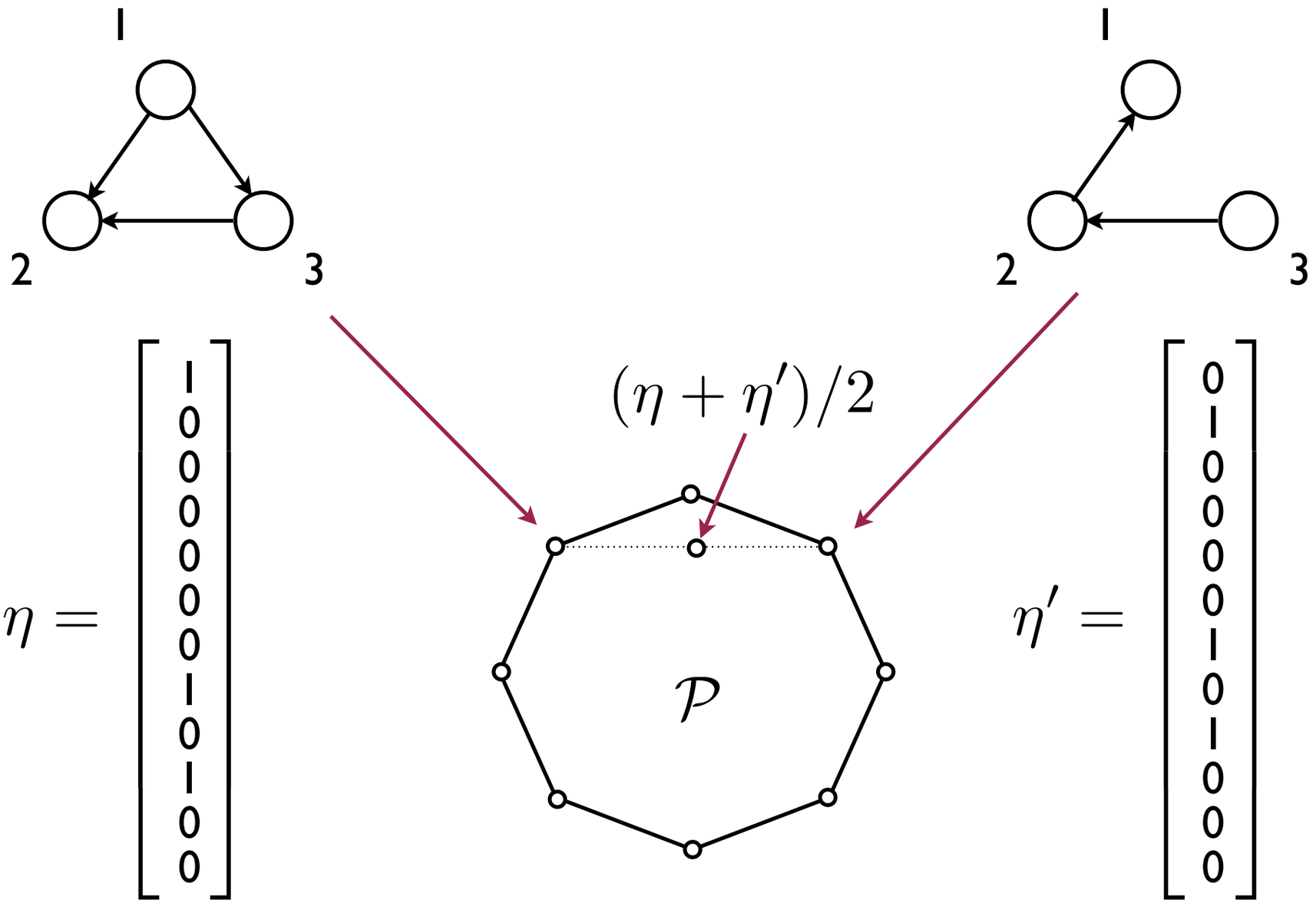


Graphs and polytopes



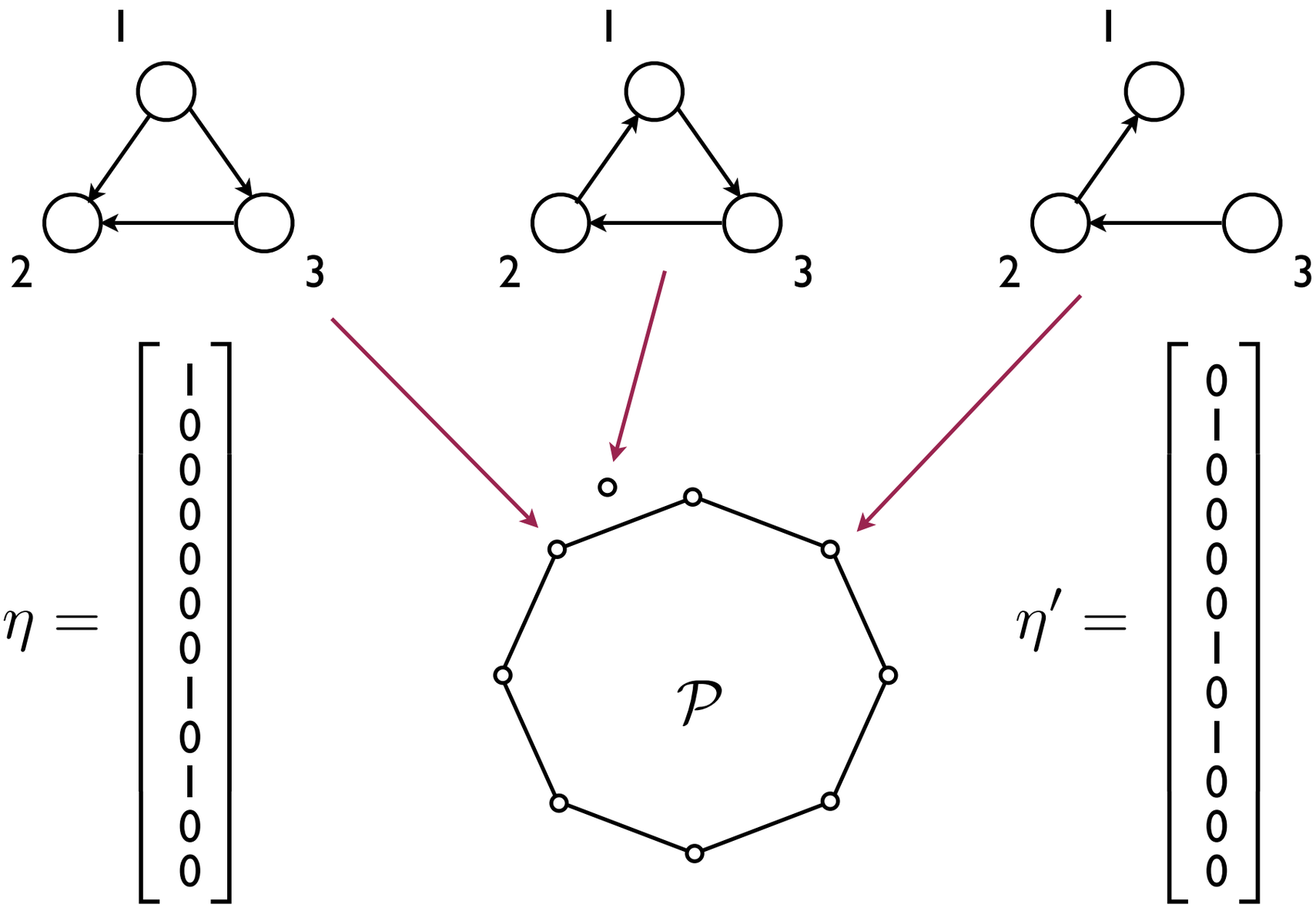
Vertices of the polytope are binary vectors corresponding to acyclic graphs; interior points are averages

Graphs and polytopes



Vertices of the polytope are binary vectors corresponding to acyclic graphs; interior points are averages

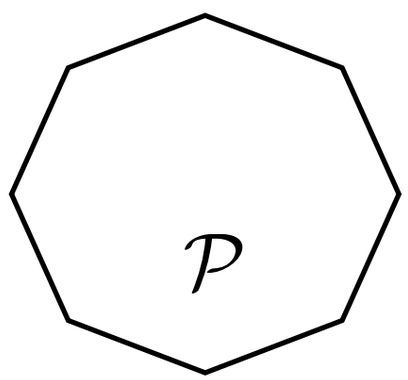
Graphs and polytopes



Vertices of the polytope are binary vectors corresponding to acyclic graphs; interior points are averages

LP for structure learning

- Maximize $$\sum_{i=1}^{n} \sum_{pa_i} \eta_i(pa_i) \text{score}(i|pa_i, D)$$ "expected" score

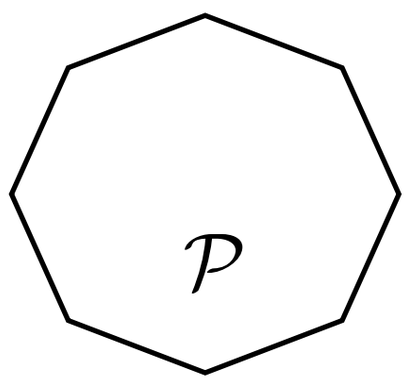
subject to $\eta \in$ 

vertices are binary vectors corresponding to acyclic graphs

- Integral solution is optimal ("certificate of optimality")

LP for structure learning

- Maximize $$\sum_{i=1}^{n} \sum_{pa_i} \eta_i(pa_i) \text{score}(i|pa_i, D)$$ "expected" score

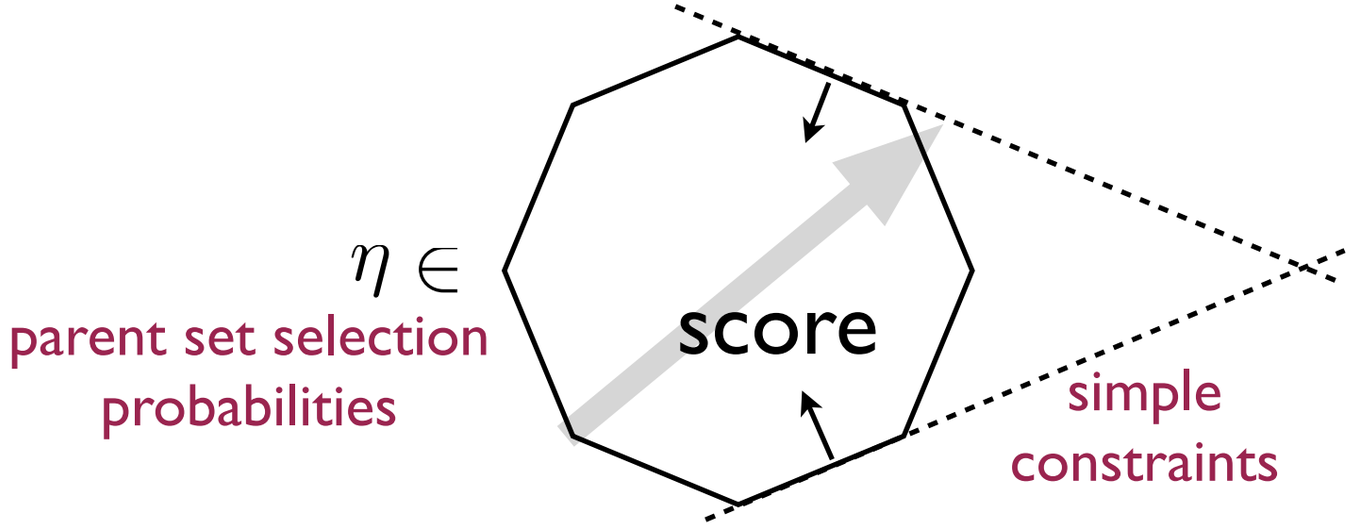
subject to $\eta \in$ 

vertices are binary vectors corresponding to acyclic graphs

- Integral solution is optimal ("certificate of optimality")
- But the polytope has exponentially many facets...

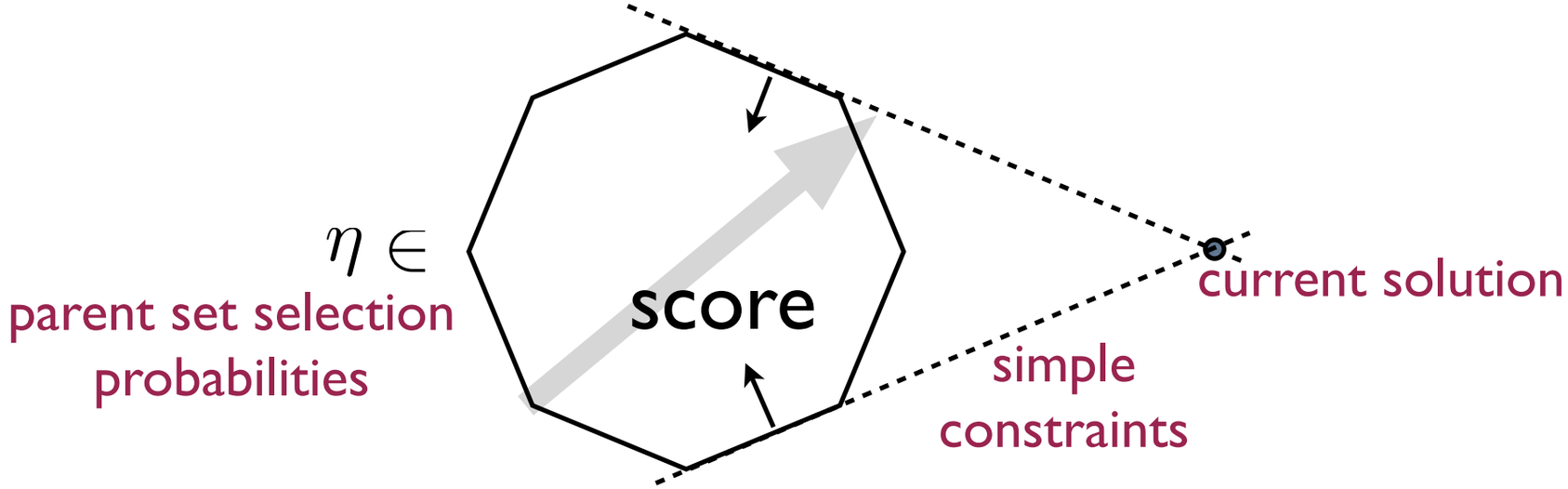
A cutting plane approach

- We only need to fully characterize the polytope (linear constraints) near the actual solution



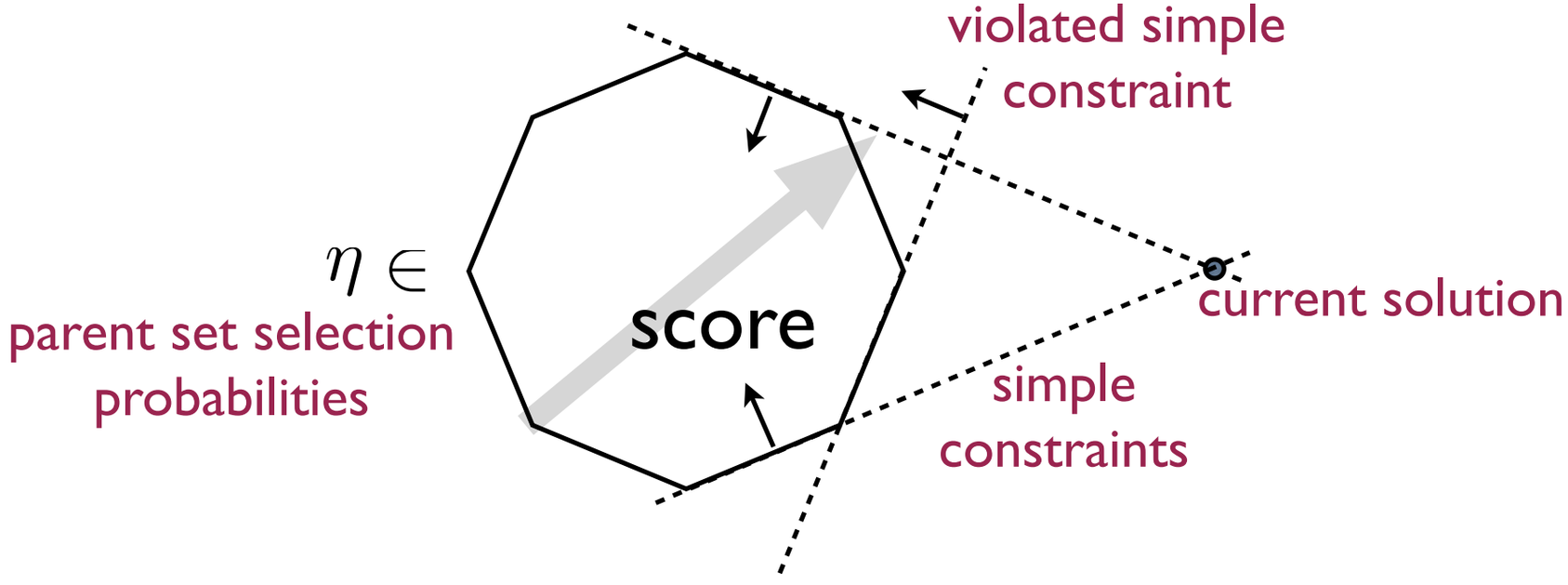
A cutting plane approach

- We only need to fully characterize the polytope (linear constraints) near the actual solution
  - solve first with the current constraints



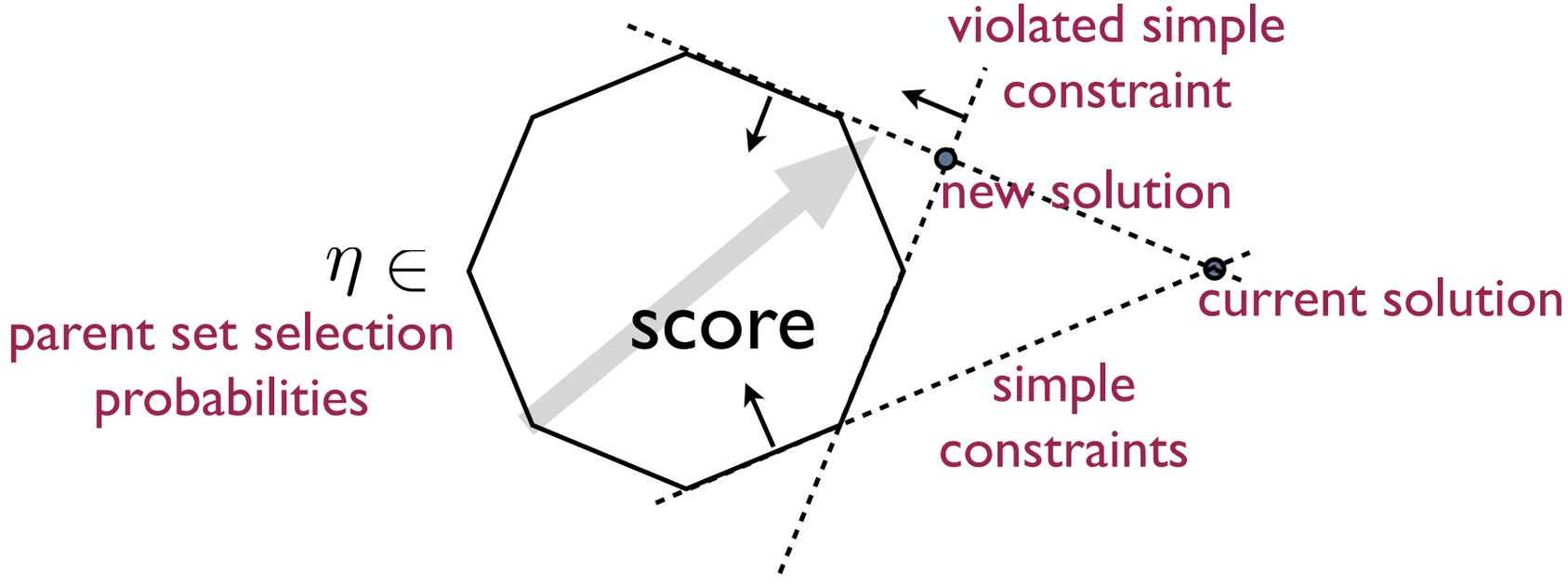
A cutting plane approach

- We only need to fully characterize the polytope (linear constraints) near the actual solution
  - solve first with the current constraints
  - find a violated constraint (separation problem)



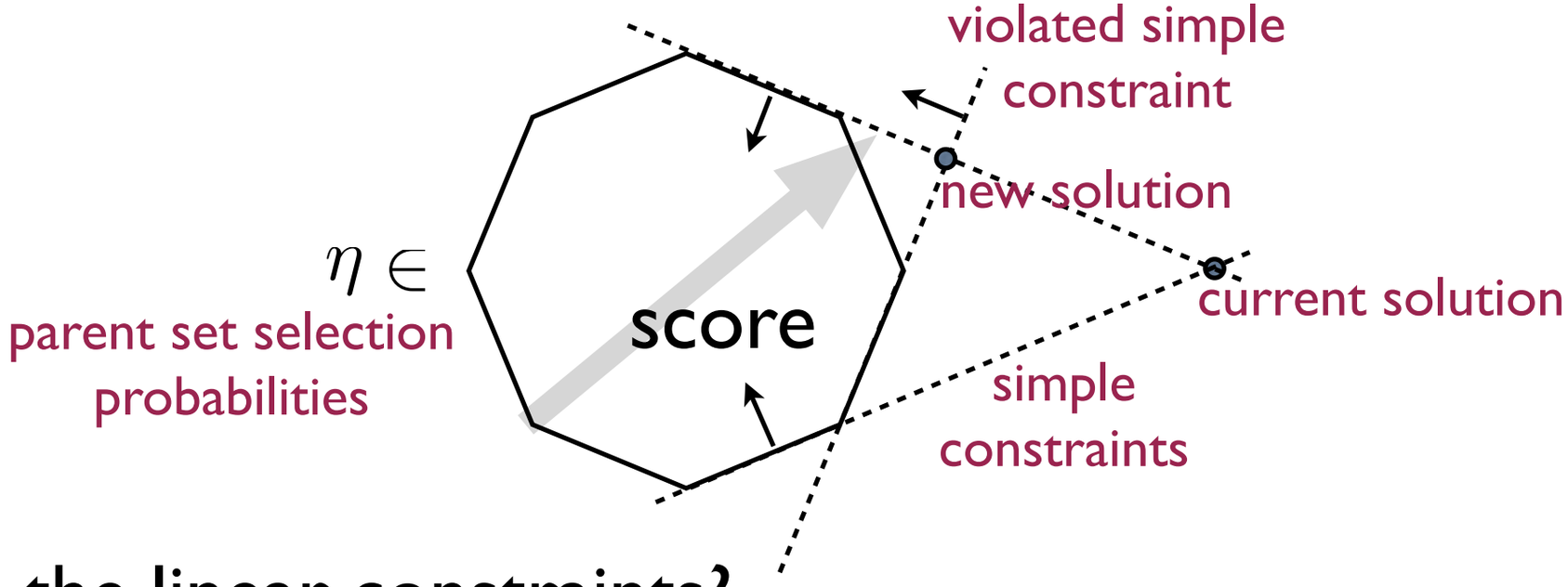
A cutting plane approach

- We only need to fully characterize the polytope (linear constraints) near the actual solution
  - solve first with the current constraints
  - find a violated constraint (separation problem)
  - resolve



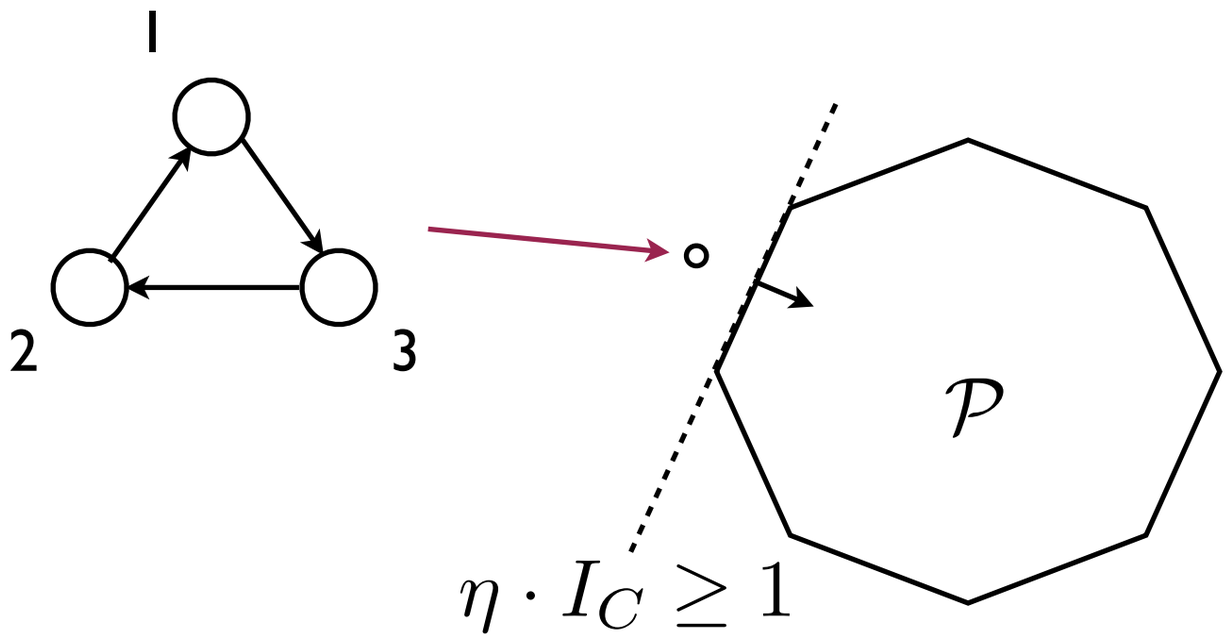
A cutting plane approach

- We only need to fully characterize the polytope (linear constraints) near the actual solution
  - solve first with the current constraints
  - find a violated constraint (separation problem)
  - resolve

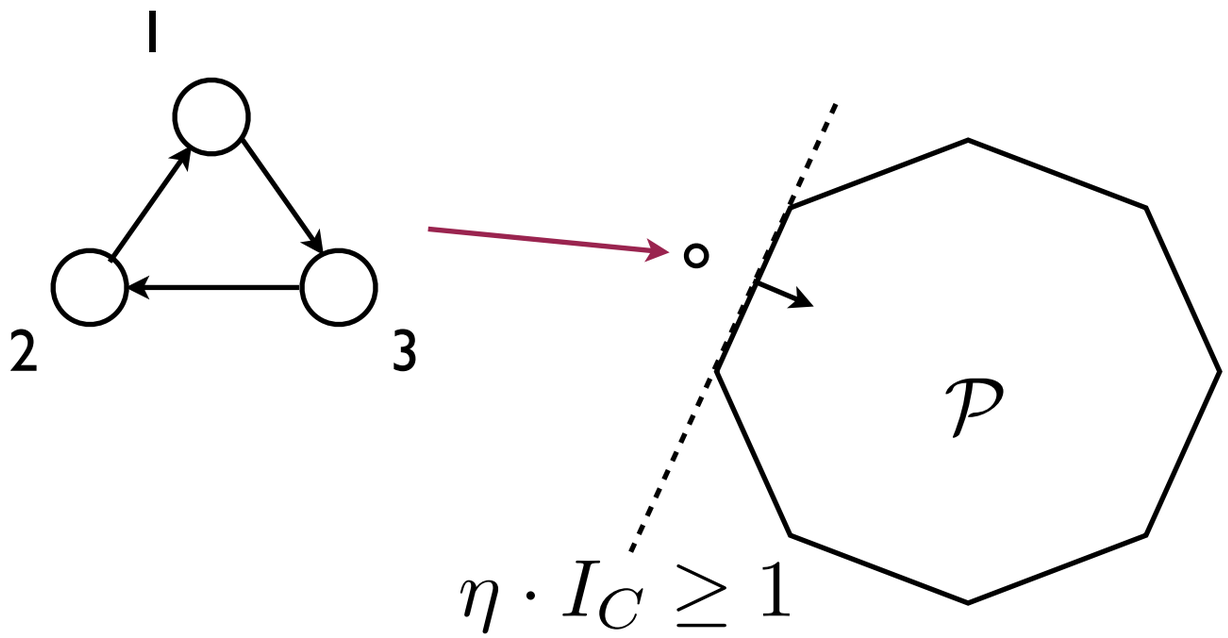


What are the linear constraints?
How to find a violated constraint?
How to solve the resulting (simpler) LP?

The separation problem



The separation problem



parent set selection for variable 1

parent set selection for variable 2

parent set selection for variable 3

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ \hdashline 0 \\ 1 \\ 0 \\ 0 \\ \hdashline 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = 0$$

η I_C

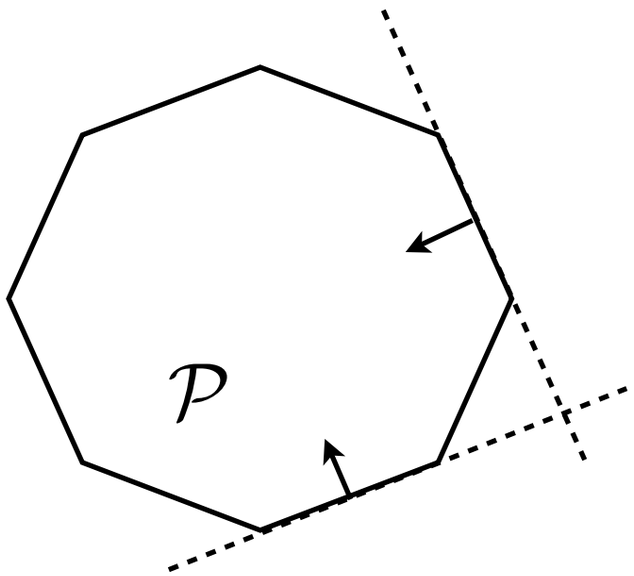
In an acyclic graph, any subset (cluster) C must have at least one variable with all its parents outside C

LP relaxation for structure learning

- Maximize $$\sum_{i=1}^{n} \sum_{pa_i} \eta_i(pa_i) \text{score}(i|pa_i, D)$$ "expected" score

subject to $$\eta_i(pa_i) \geq 0, \quad \sum_{pa_i} \eta_i(pa_i) = 1$$ parent set selections

$$\eta \cdot I_C \geq 1, \quad \forall C$$ cluster constraints



These constraints are facet defining but not sufficient to fully specify the polytope

Dual LP for structure learning

- Minimize

local scores adjusted based on clusters

$$\sum_{i=1}^{n} \max_{pa_i} \left[\text{score}(i|pa_i, D) + \sum_{C:\, i \in C} \lambda_C I_C(pa_i) \right] - \sum_{C} \lambda_C$$

subject to $\lambda_C \geq 0, \quad \forall C$

Dual LP for structure learning

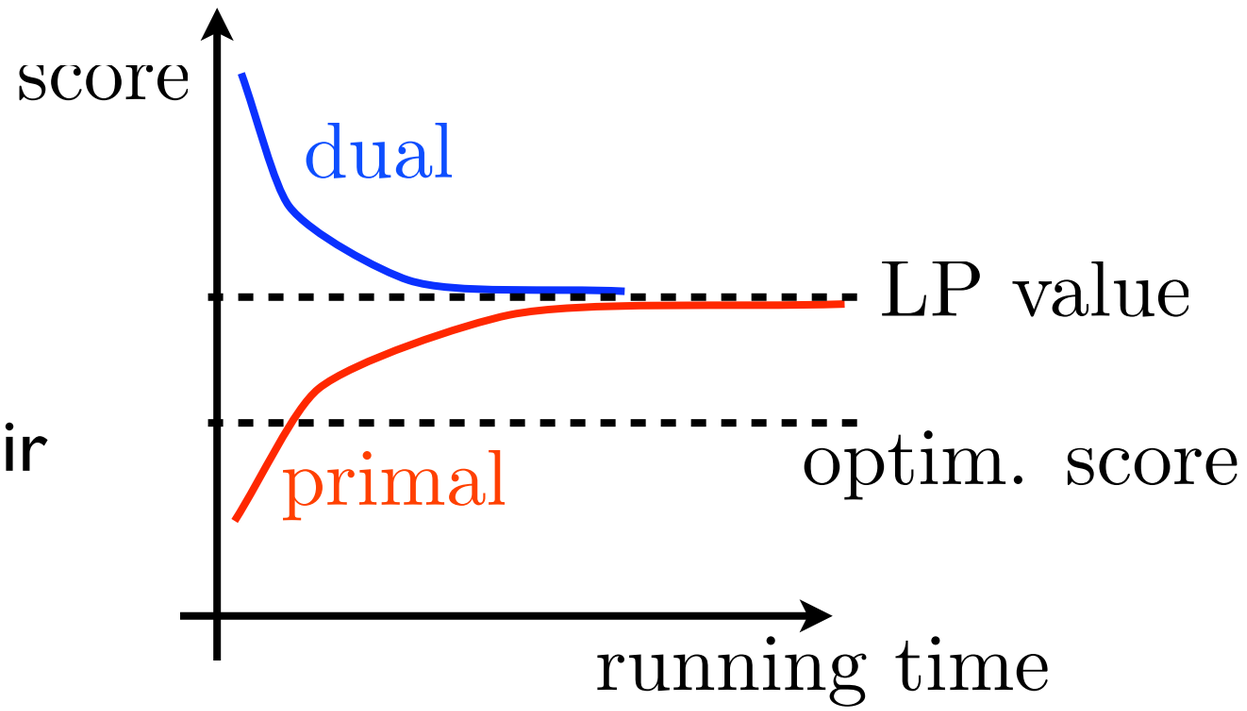
- Minimize

local scores adjusted based on clusters

$$\sum_{i=1}^{n} \max_{pa_i} \left[\text{score}(i|pa_i, D) + \sum_{C:\, i \in C} \lambda_C I_C(pa_i) \right] - \sum_{C} \lambda_C$$

subject to $\lambda_C \geq 0, \quad \forall C$

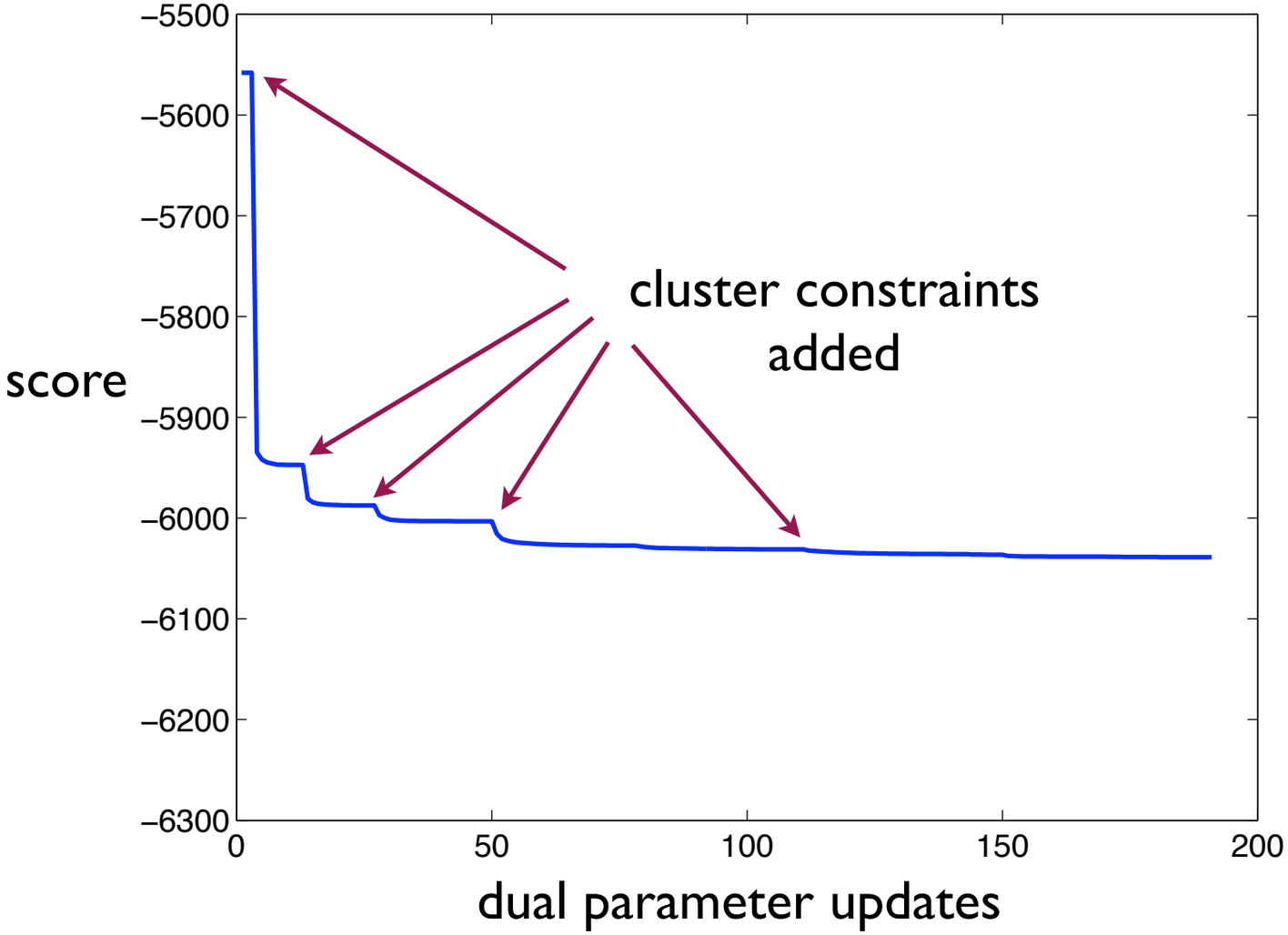
- Why the dual?
  - simpler constraints, closed form coordinate updates
  - clusters can be ranked by their effect on value
  - any dual feasible point upper bounds the LP value (and the optimal score)



Simple example problem

- RNAi gene silencing experiments in C. elegans
- 25 phenotypic indicators/markers that characterize each experimental outcome
- We seek to build a Bayesian network model over the phenotypic markers in order to capture their coordinate variation across experiments
- Technical constraint: each variable can have at most four parents

Dual solution



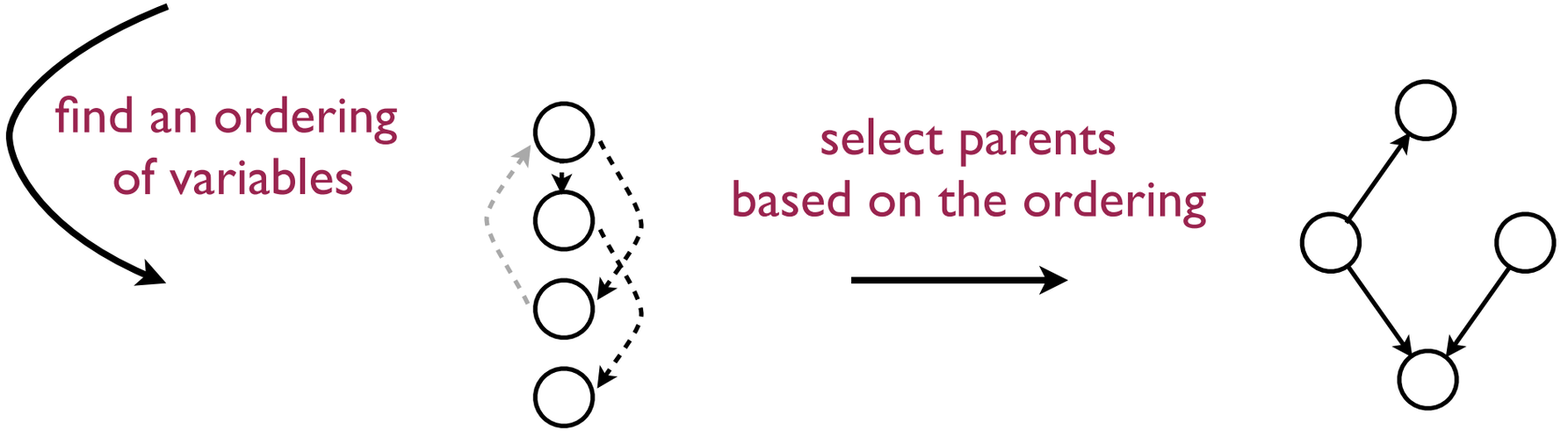
- The LP is tight if we can find a structure (integral solution) that attains the dual value

Minimum regret decoding

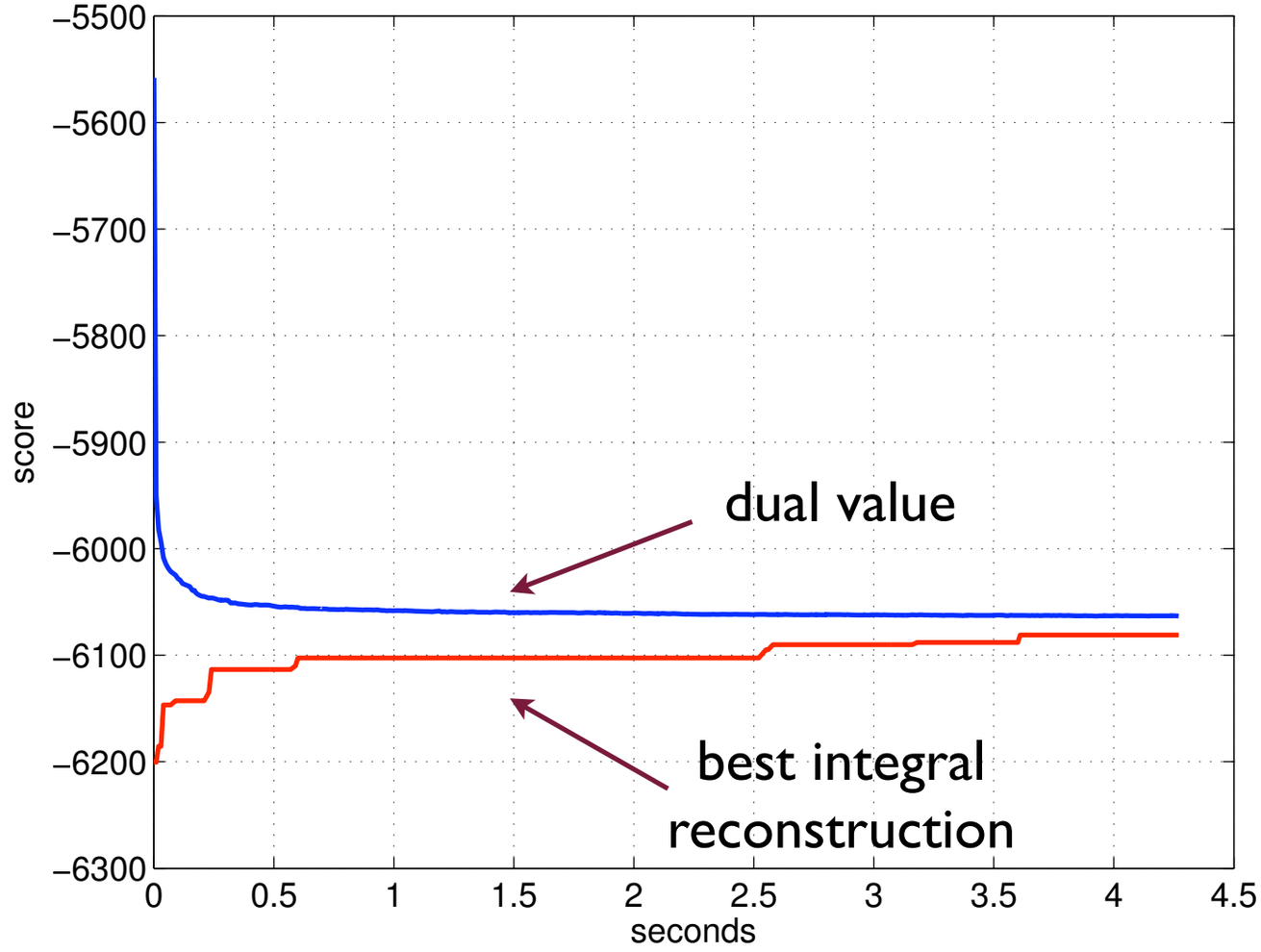
- We can use adjusted local scores from the dual solution to find good candidate structures (decoding)

local scores adjusted based on clusters

$$\text{score}(i|pa_i, D) + \sum_{C:\, i \in C} \lambda_C I_C(pa_i)$$



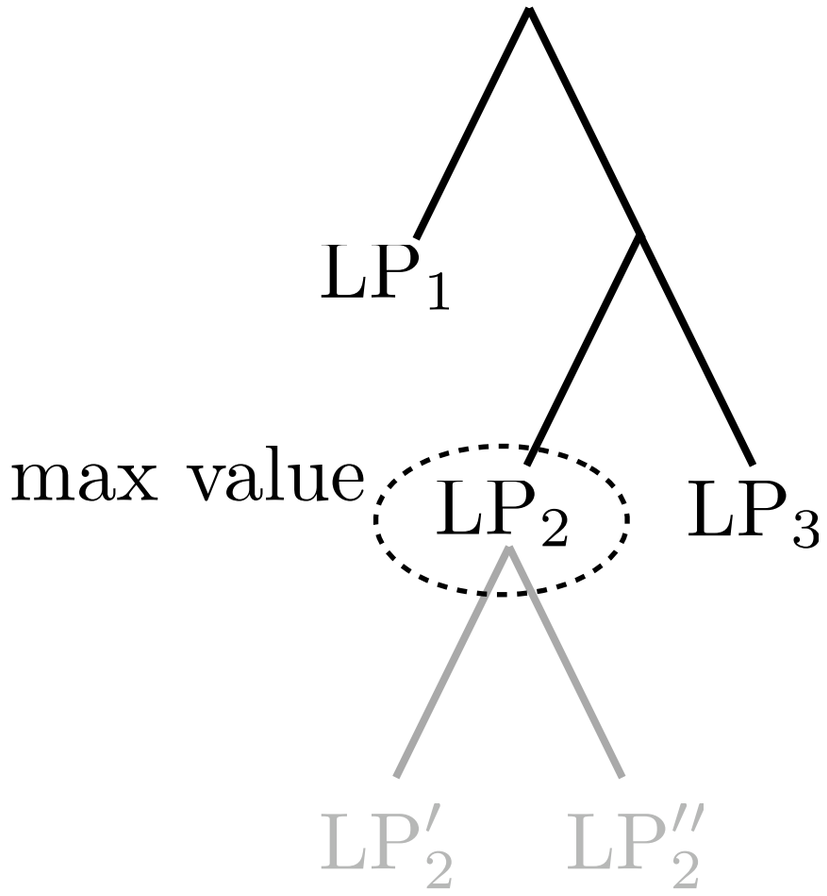
Dual solution with decoding



- A structure is proved optimal if it attains the dual value
- The simple LP relaxation is (almost) tight for this problem

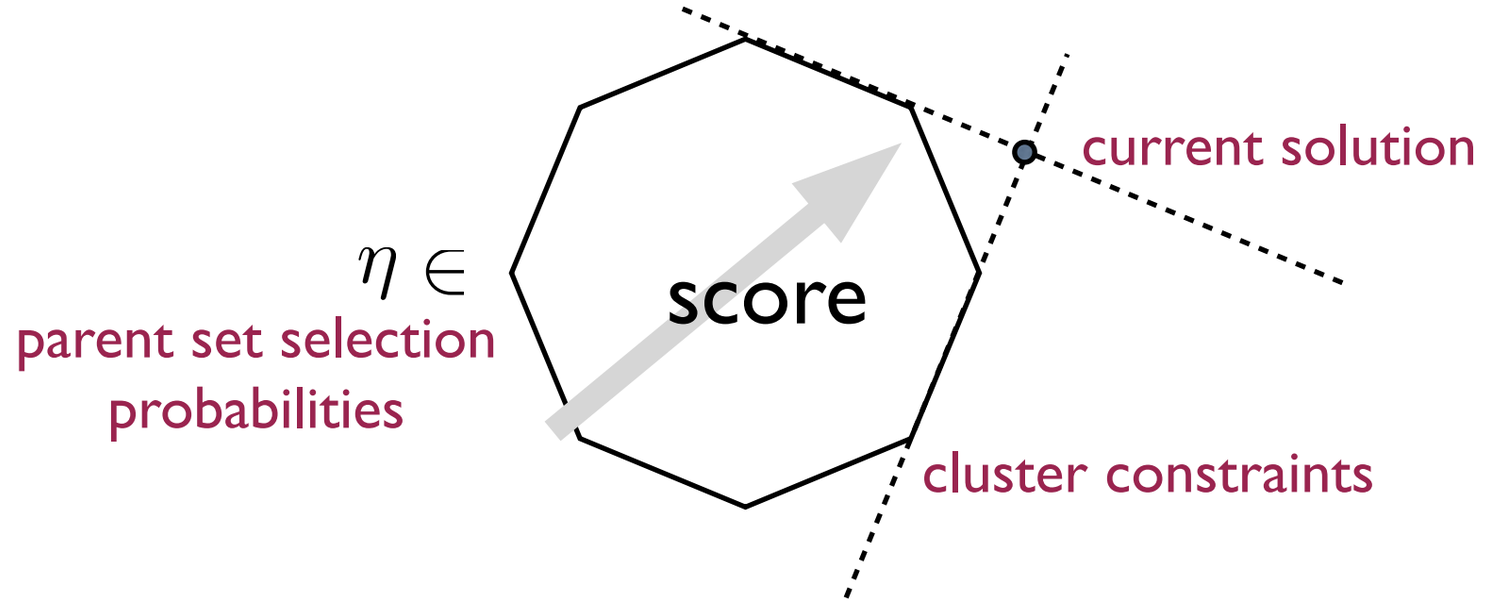
Branch and Bound

- We can further tighten the approximation by iteratively partitioning the space of possible solutions and using the LP relaxation separately for each partition
- The dual LP (upper bound) is particularly effective for determining how the tree of partitions should be expanded



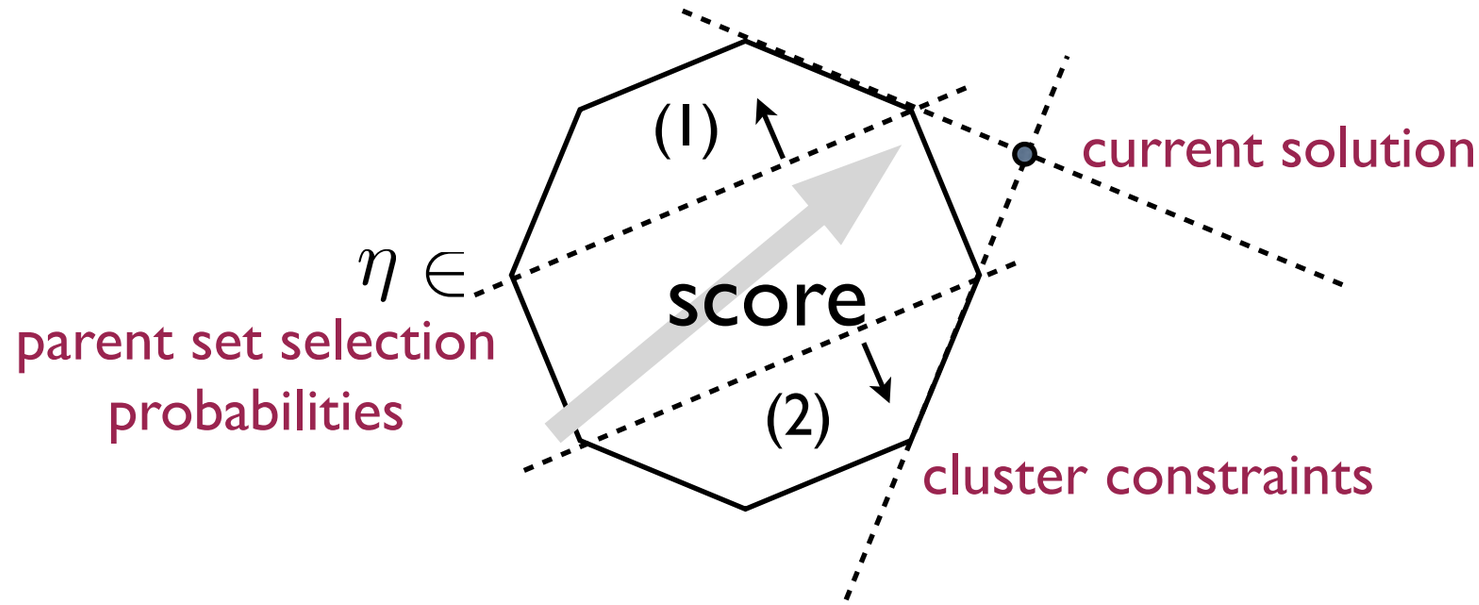
Partitions

- We can tighten the approximation further by partitioning the polytope into segments and solving each segment separately



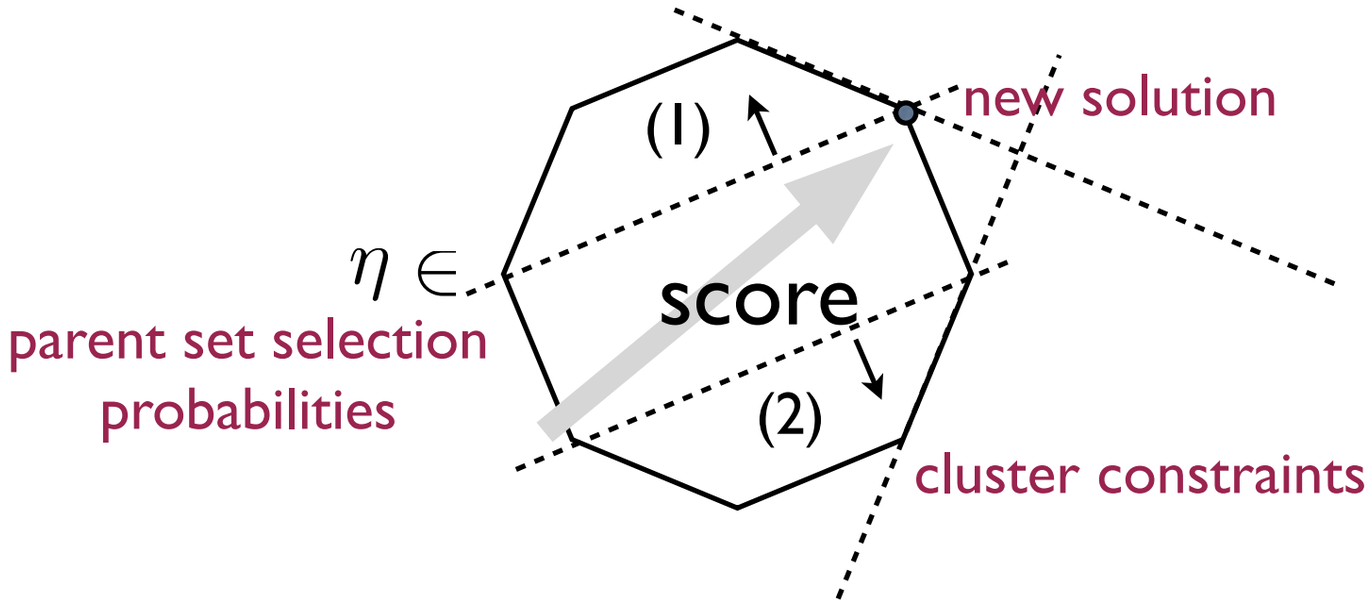
Partitions

- We can tighten the approximation further by partitioning the polytope into segments and solving each segment separately



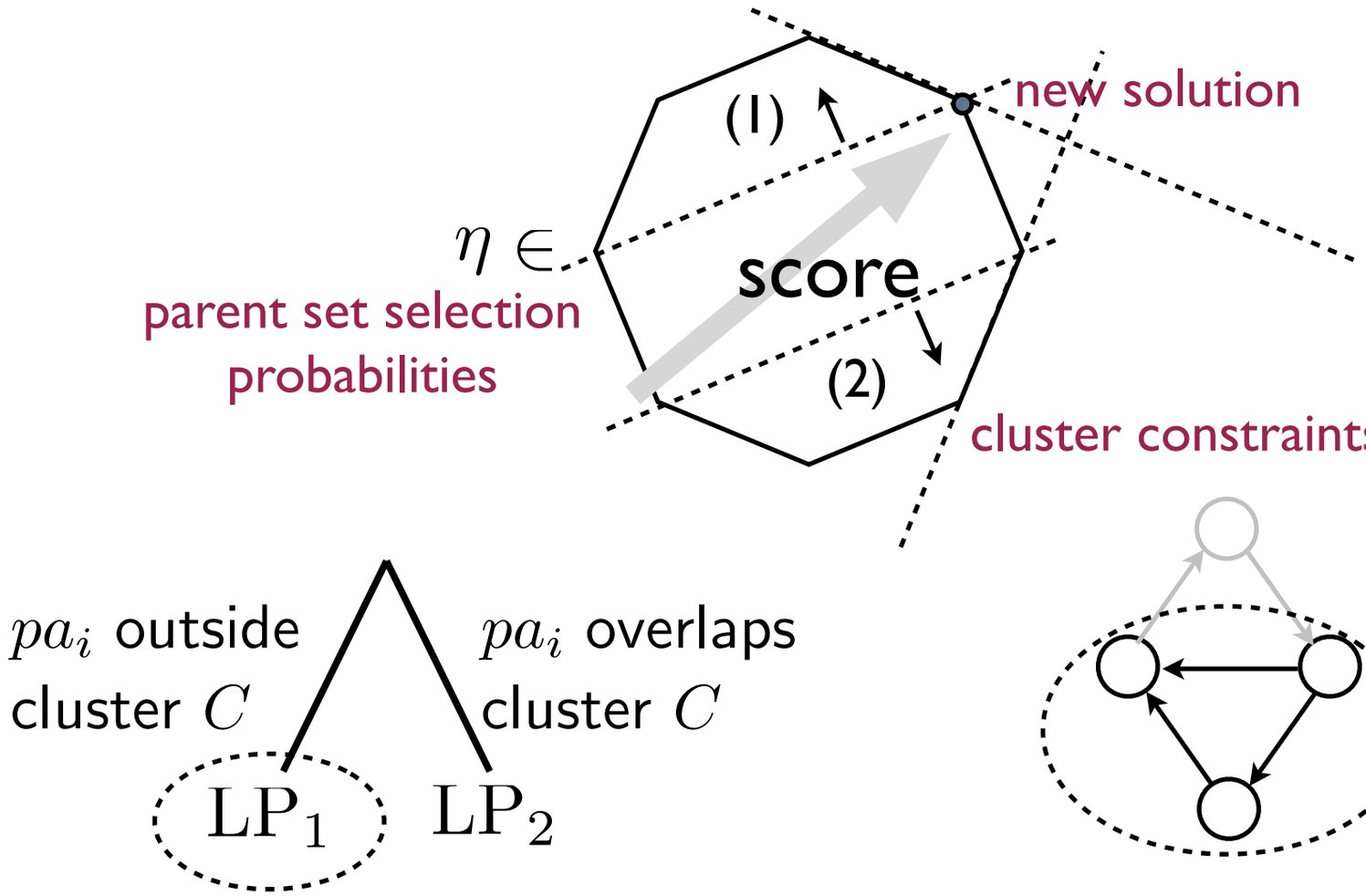
Partitions

- We can tighten the approximation further by partitioning the polytope into segments and solving each segment separately



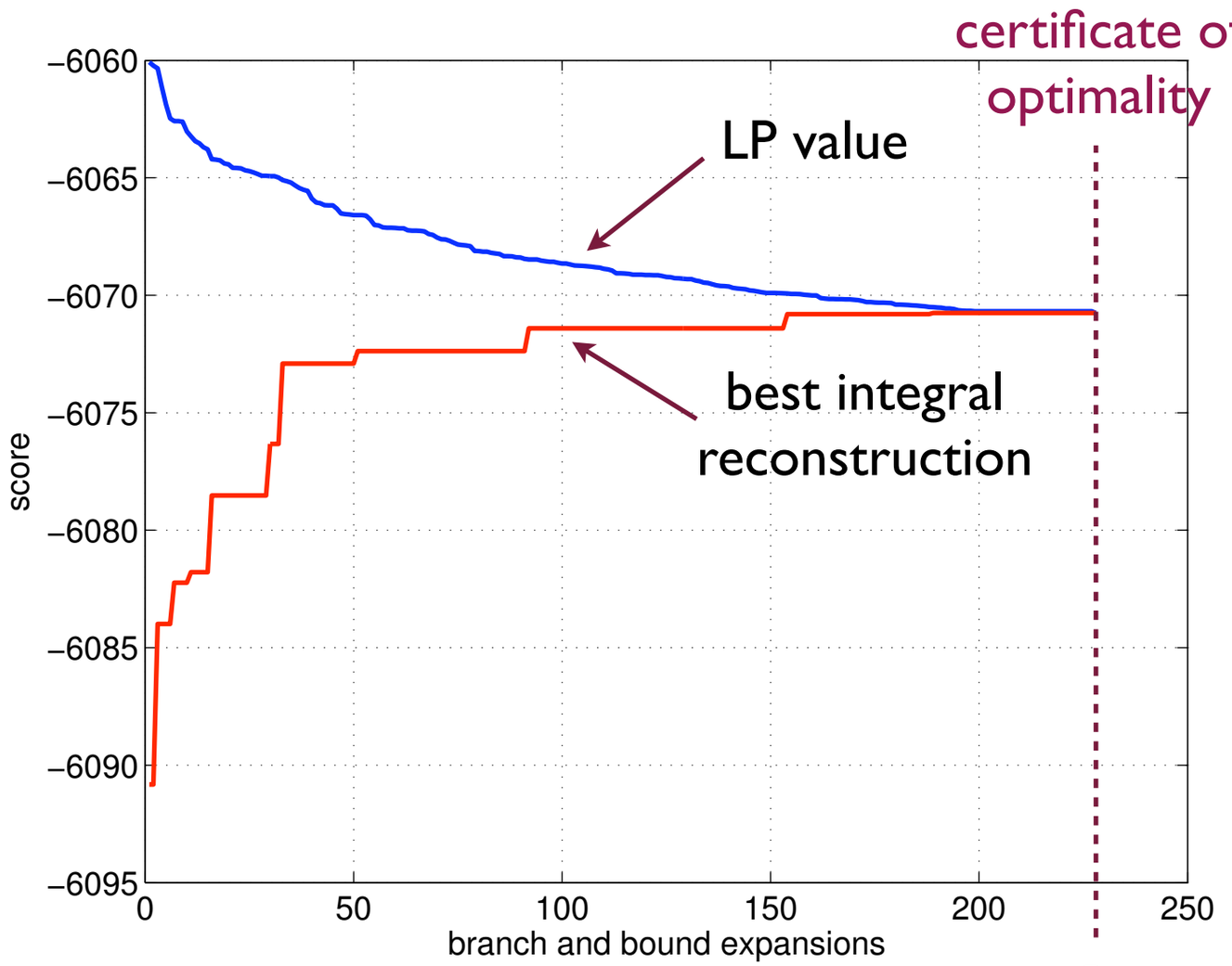
Partitions

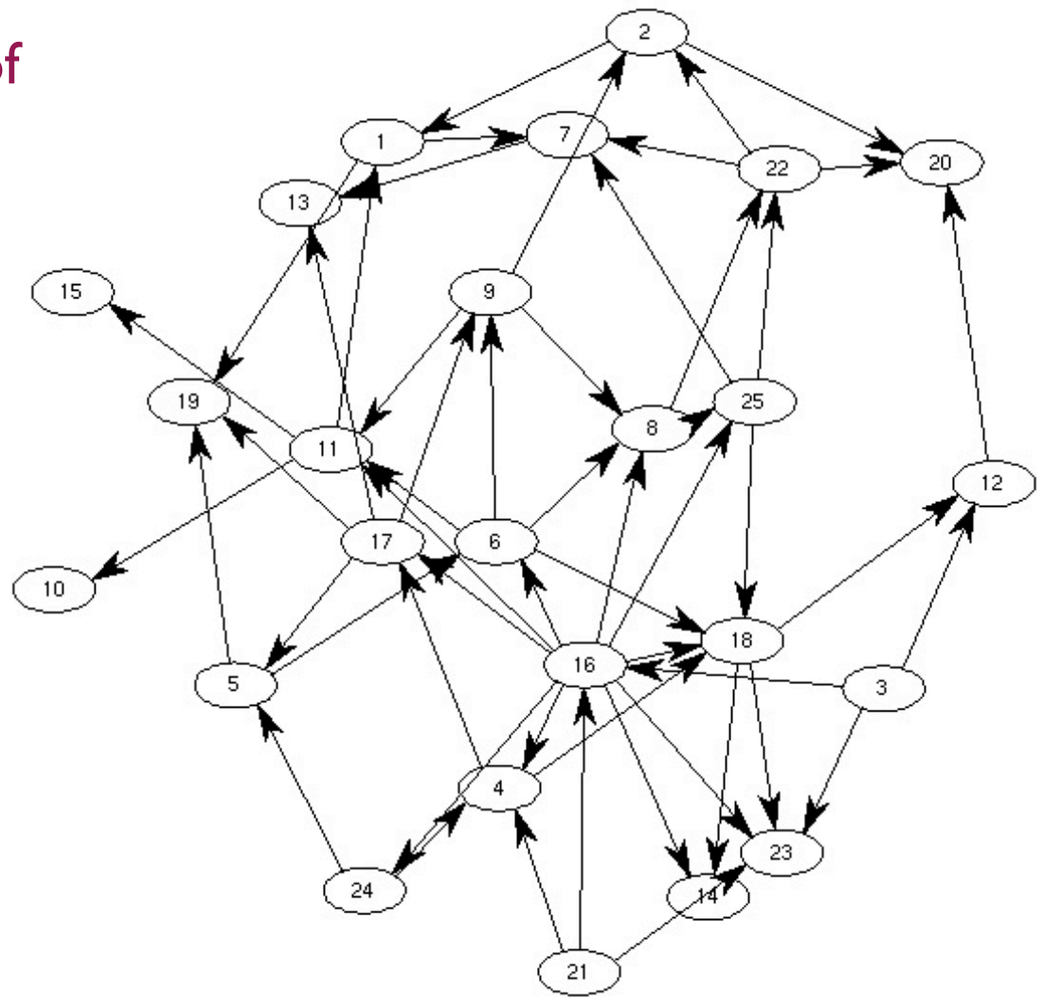
- We can tighten the approximation further by partitioning the polytope into segments and solving each segment separately



Solution...

- The highest scoring structure is found after a small number of branch and bound partitions

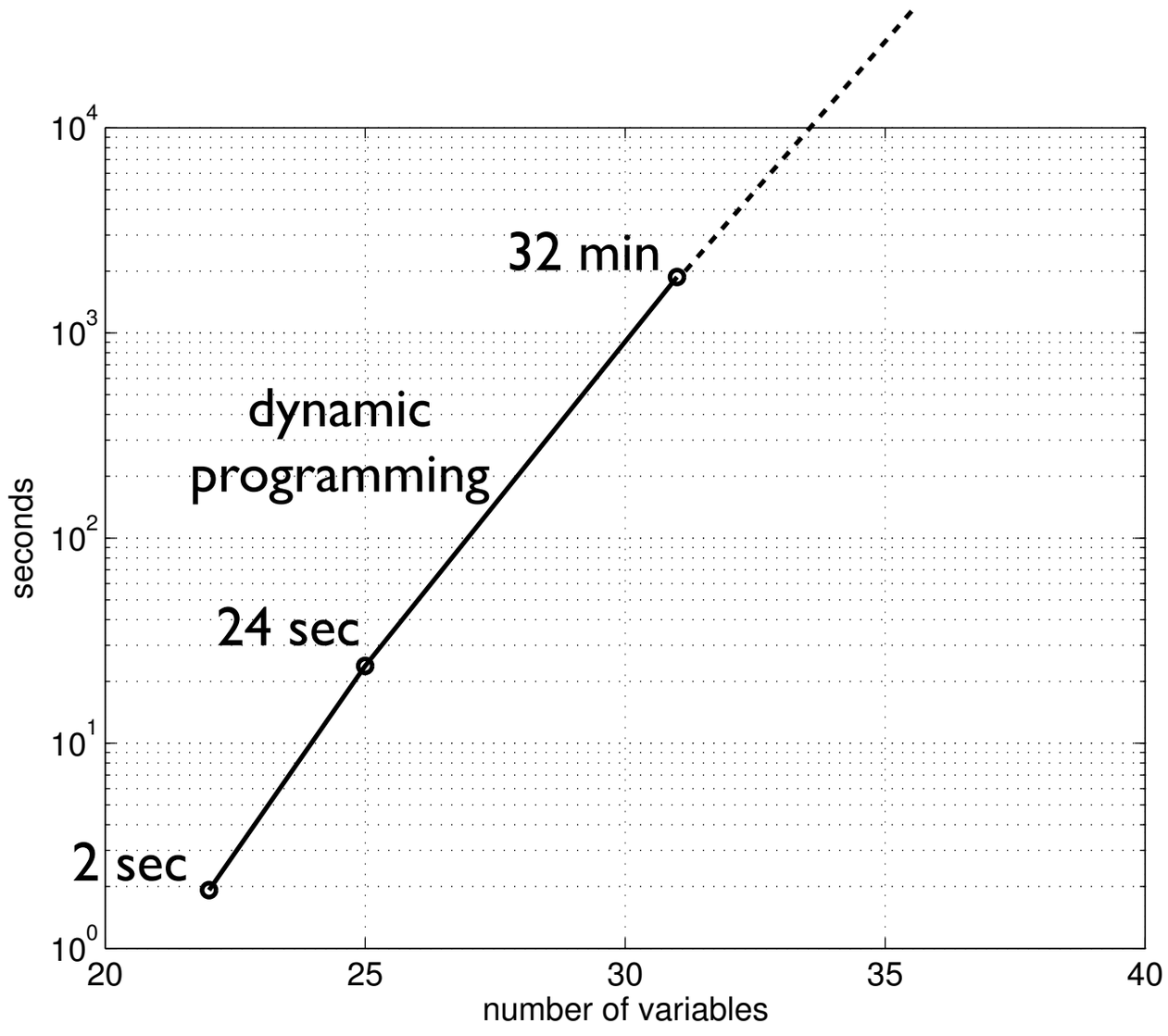




8 seconds

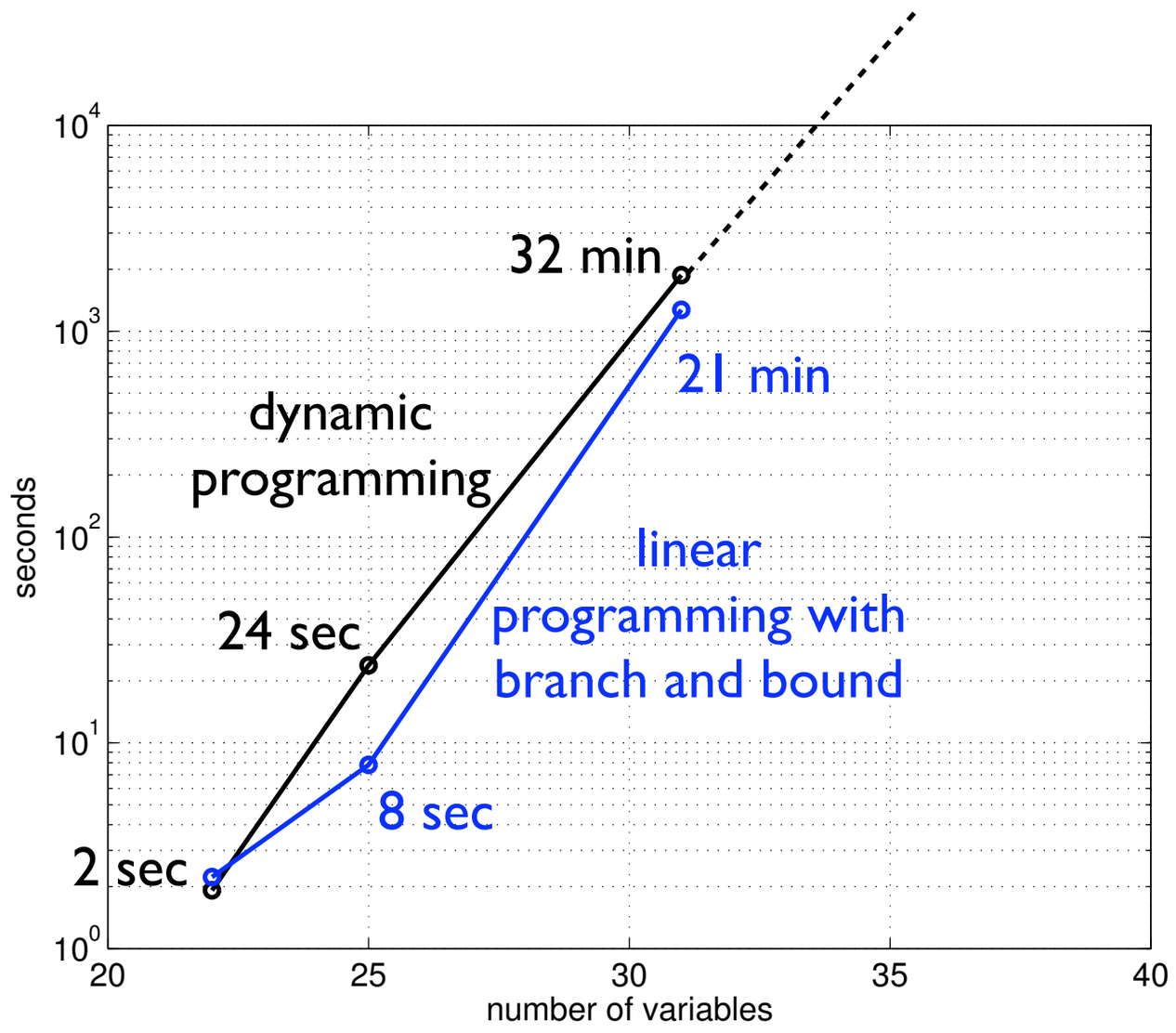
Recall: exact methods

- Dynamic programming methods work well for small structure learning problems ...



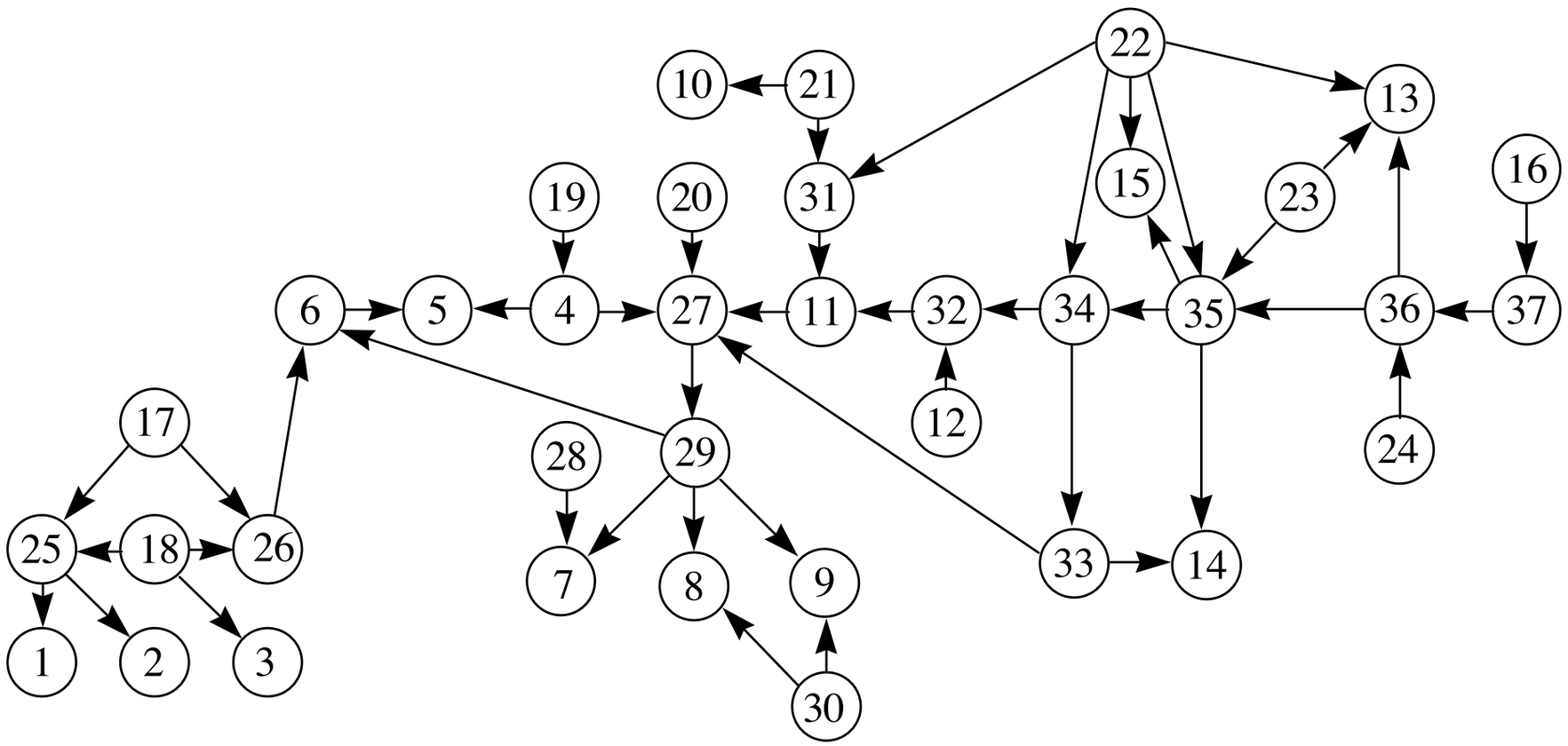
LP relaxation with BB

- LP relaxation combined with branch and bound performs similarly to dynamic programming methods



The alarm network

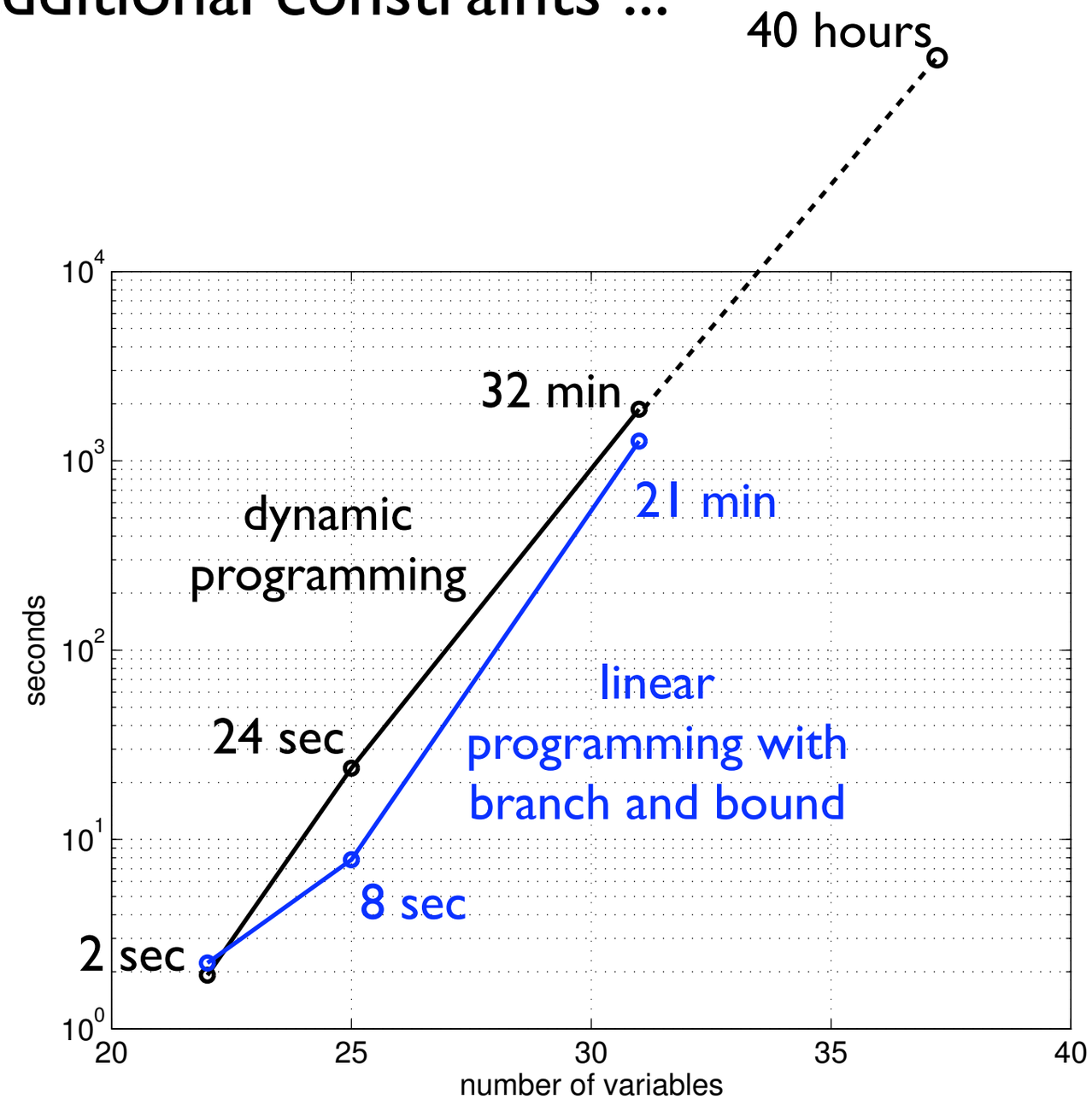
- 37 variables, 1000 data points



(Heckerman et al. 1995)

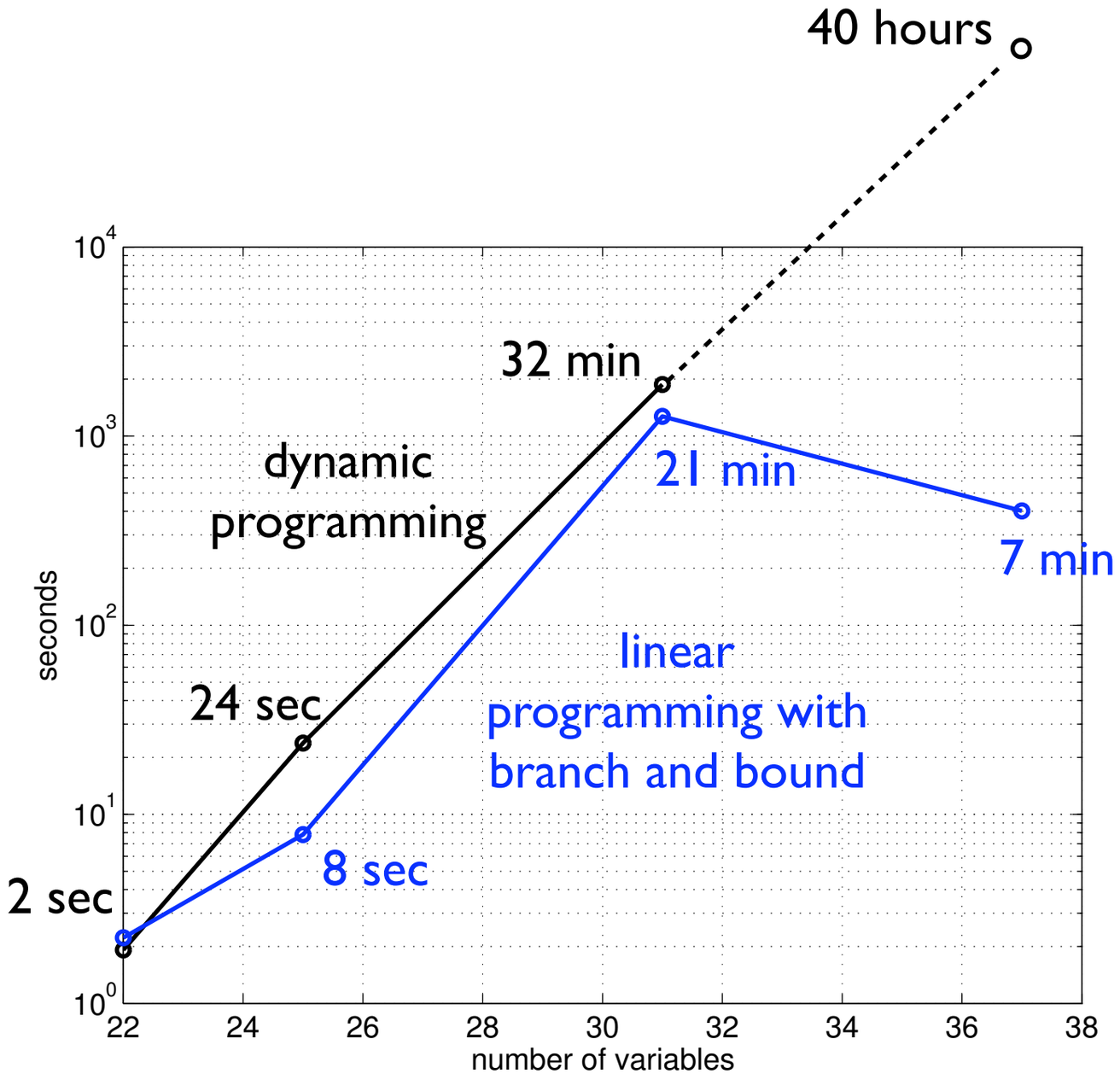
The alarm network

- Dynamic programming methods are no longer practical without additional constraints ...



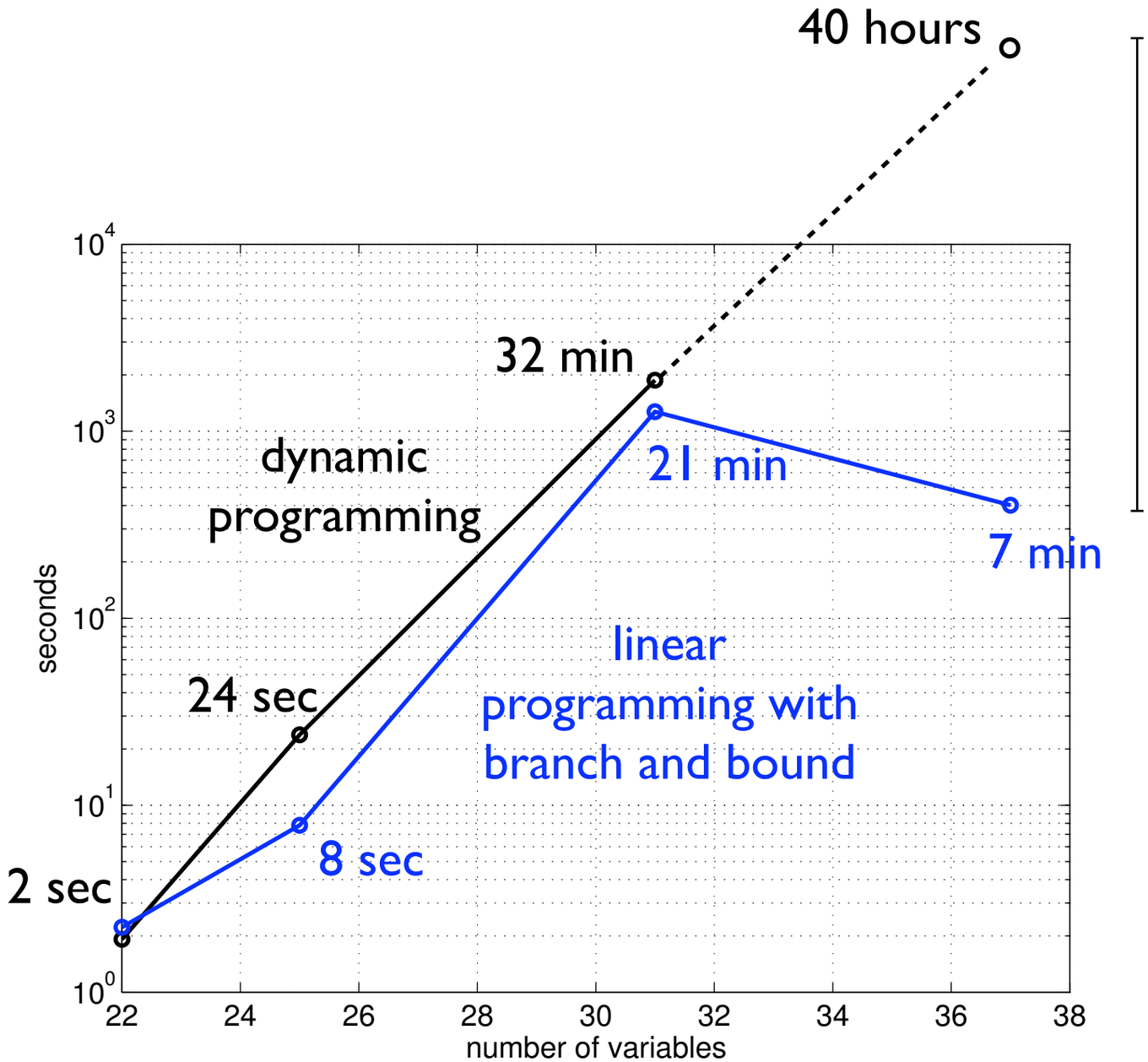
The alarm network

- Dynamic programming methods are no longer practical without additional constraints ...

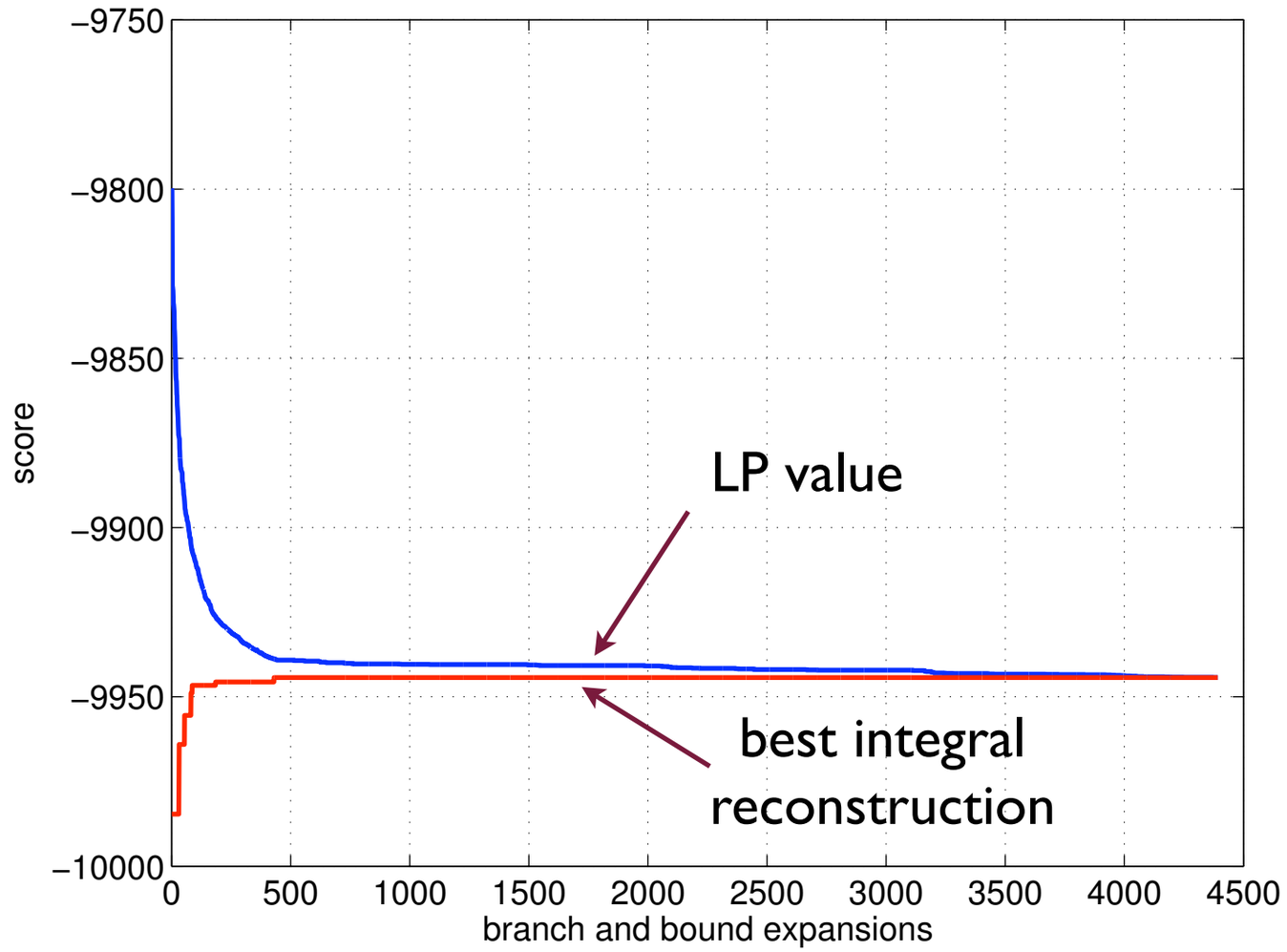


The alarm network

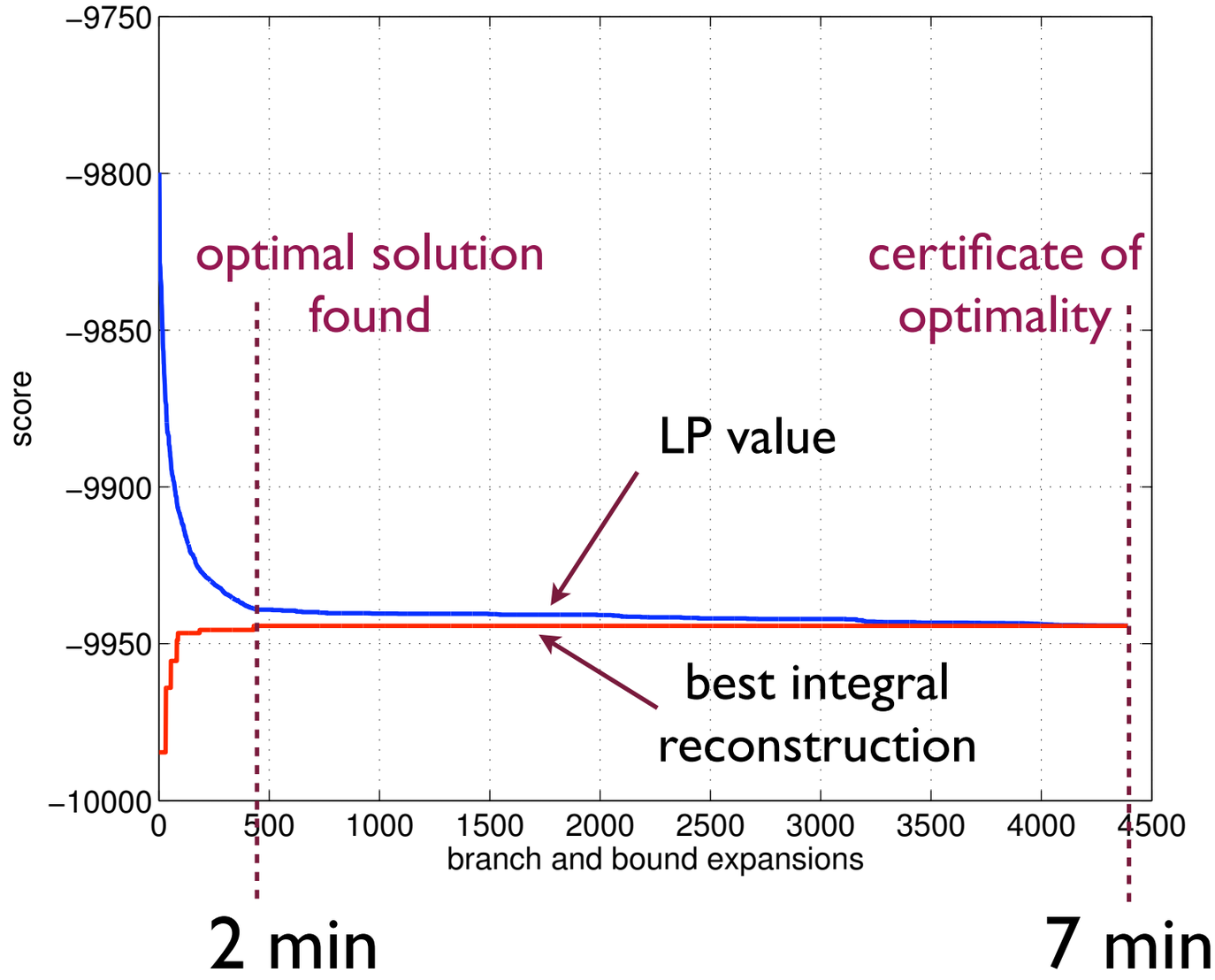
- Dynamic programming methods are no longer practical without additional constraints ...



Anytime solution



Anytime solution



Summary

- Finding the highest scoring Bayesian network structure from data is a hard combinatorial problem... but the "hard instances" may not be typical
- Our "anytime" approach to structure learning is based on linear programming relaxations that are iteratively refined in a cutting plane fashion
- The approach relies fundamentally on understanding the facets of the polytope corresponding to acyclic graphs