Learning Partial Models for Planning

Neville Mehta
Prasad Tadepalli

School of Electrical Engineering and Computer Science
Oregon State University

Motivation

- Partial models
  - Compact size
  - Focused (limited) knowledge about the system dynamics
  - Easier to learn

Outline

- Planning framework
  - Partial model
  - State-variable hierarchy
  - Partial model planning (PMP)
- Learning partial models
- Empirical evaluation

Planning Framework

- Deterministic factored state space
  - S = Total state space = $D(v_1) \times \ldots \times D(v_n)$
- Partial action model
  - Set of rules = $\{r = op : pre \longrightarrow post\}$
  - Relevant variables $\rho(r)$ = variables that are checked = variables that are changed
  - Example
    - r = North: $* \; 0 \; * \longrightarrow * \; 1 \; *$
    - $\rho(r) = \{v_2\}$

Partial Model

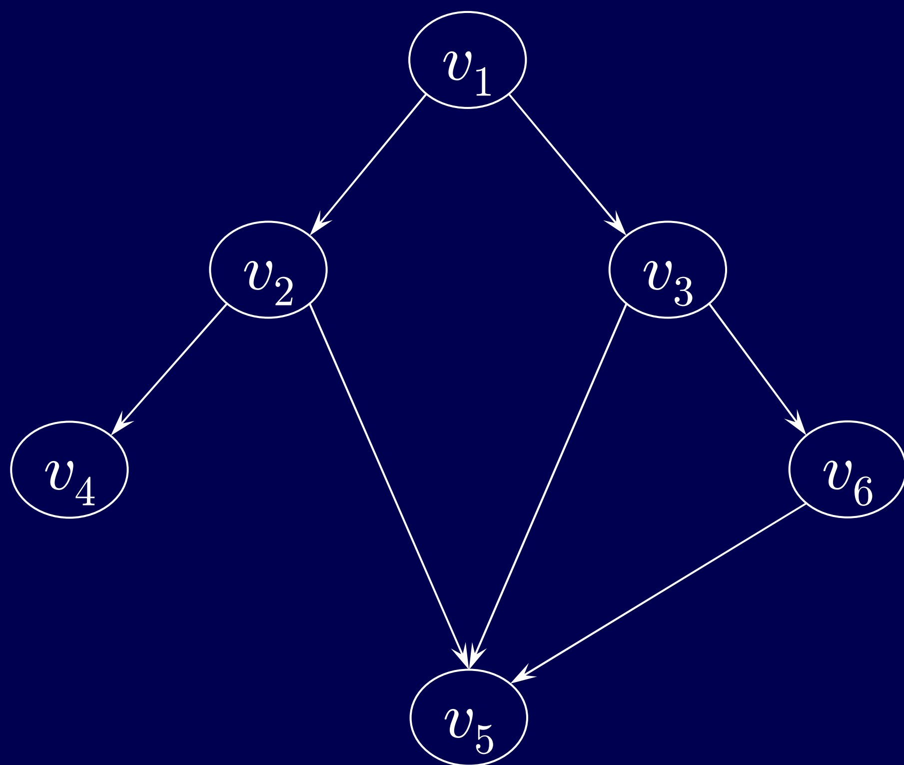
- Given ordering Ω on variables ($i < j \Leftrightarrow v_i \prec v_j$)
  - Primary variable ψ(r, Ω) = first variable in $\rho(r)$
  - Secondary variables = $\rho(r) - \psi(r, \Omega)$
  - $r.op$ has no effect on any variable preceding $\psi(r, \Omega)$
- Projection of a rule $r(v) = (pre[v], post[v])$
- Component graph (CG) for a variable

$$G_v = (D(v), \{r(v) : v = \psi(r, \Omega)\})$$

- Partial model (PM) = $\{(G_v, R_v)\}$

Hierarchy of variables

- The ordering is a "hierarchy" over variables



Definitions

- A partial model is **adequate** iff there exists an ordering such that all CGs are strongly connected
- A domain is **serializable** iff there exists an ordering such every variable can be set to its desired value without affecting precursors

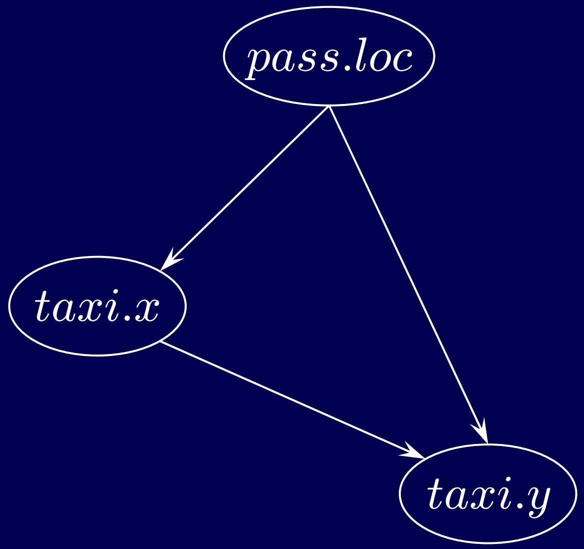
Statement of Partial Model Planning

- If a domain has an adequate partial model then
  - S is strongly connected via the operators in the PM
  - The domain is serializable
- This follows because every CG
  - Is strongly connected
  - Only affects the associated variable and successors

Taxi Domain

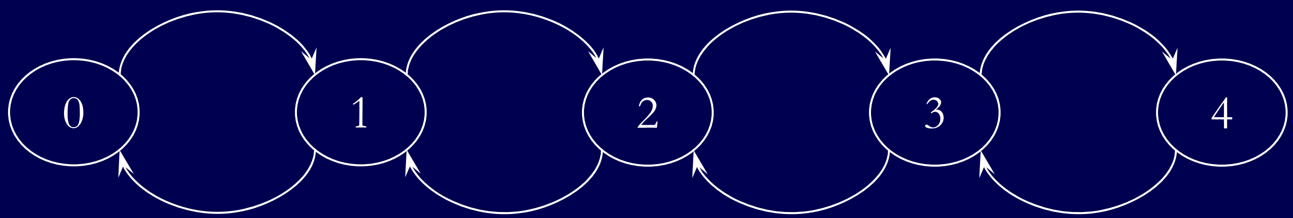
Actions:

- North, South, East, West
- Pickup
- Dropoff

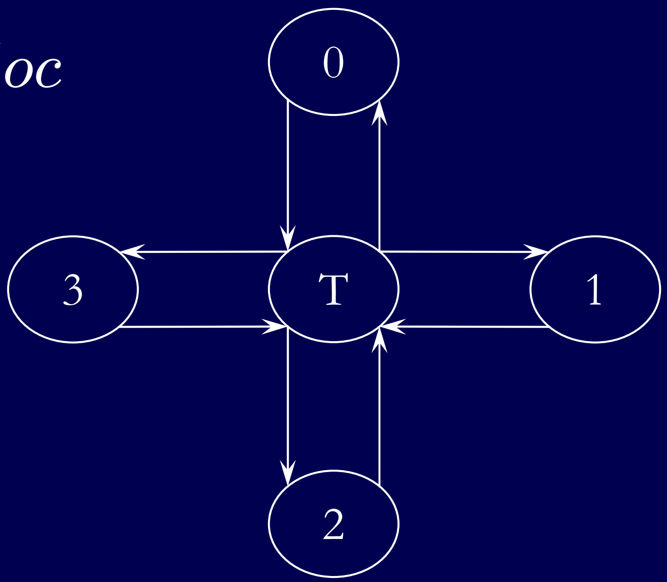


Taxi Domain CGs

$taxi.x$, $taxi.y$

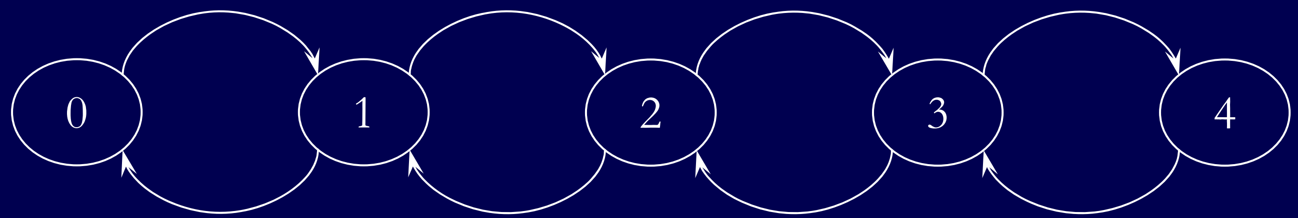


$pass.loc$

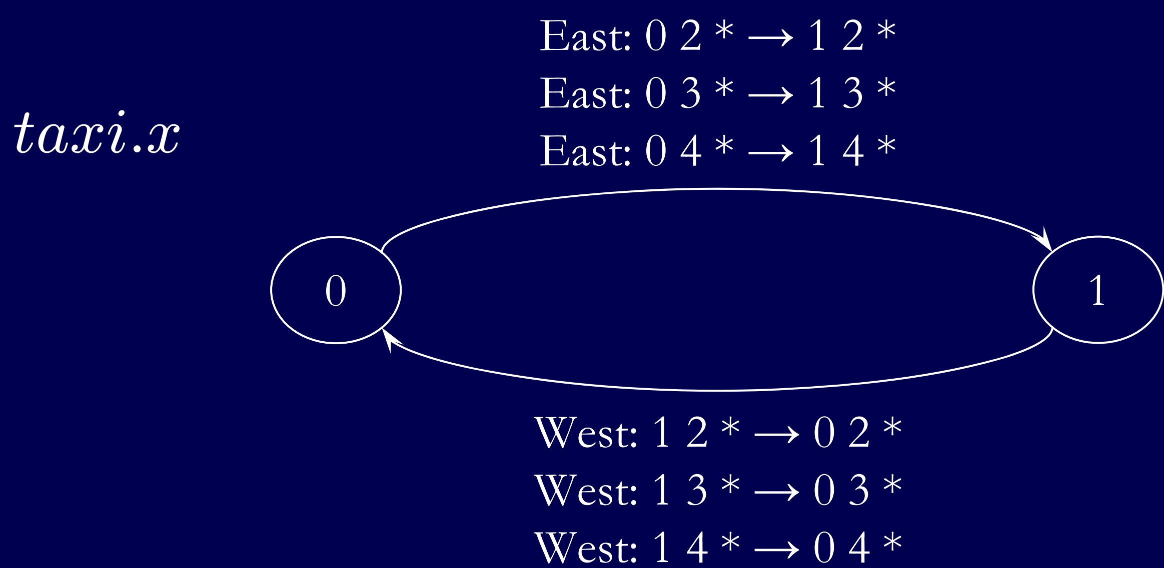


Taxi Domain CGs

$taxi.x$



Taxi Domain CGs



Partial Model Planning

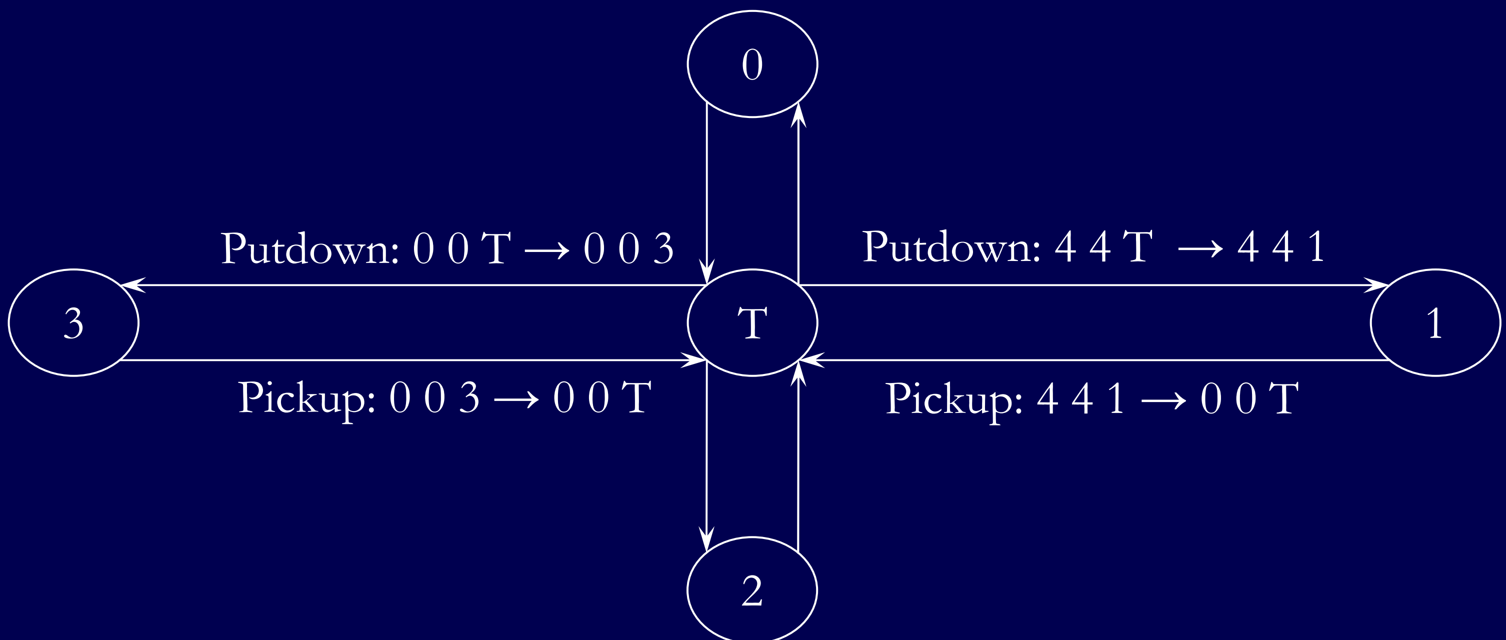
Start state: 3 0 3

Goal state: * * 1

Partial Model Planning

Start state: 3 0 3

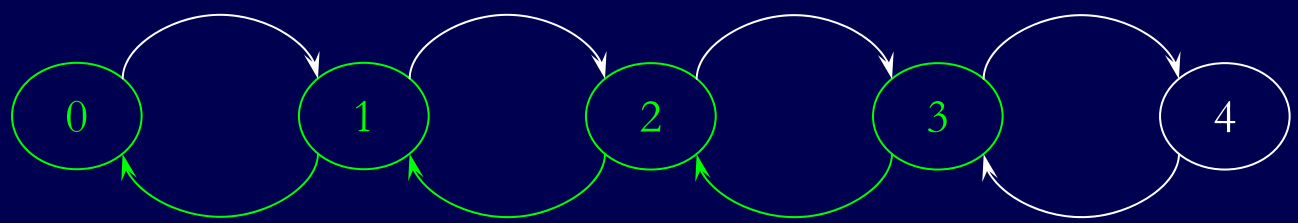
Goal state: * * 1



Partial Model Planning

Start state: 3 0 *

Goal state: 0 * *

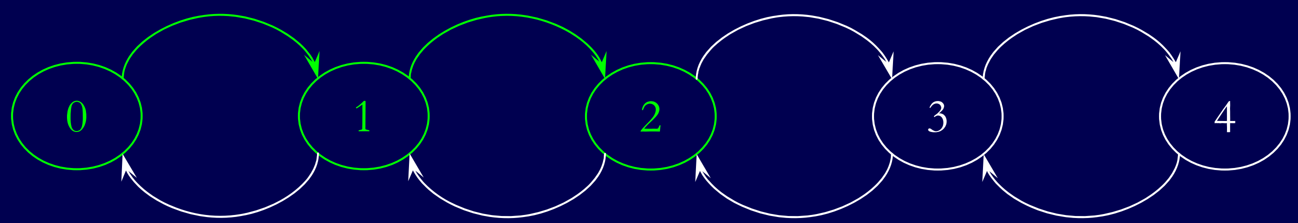


$taxi.x$

Planning Algorithm

Start state: * 0 *

Goal state: * 2 *

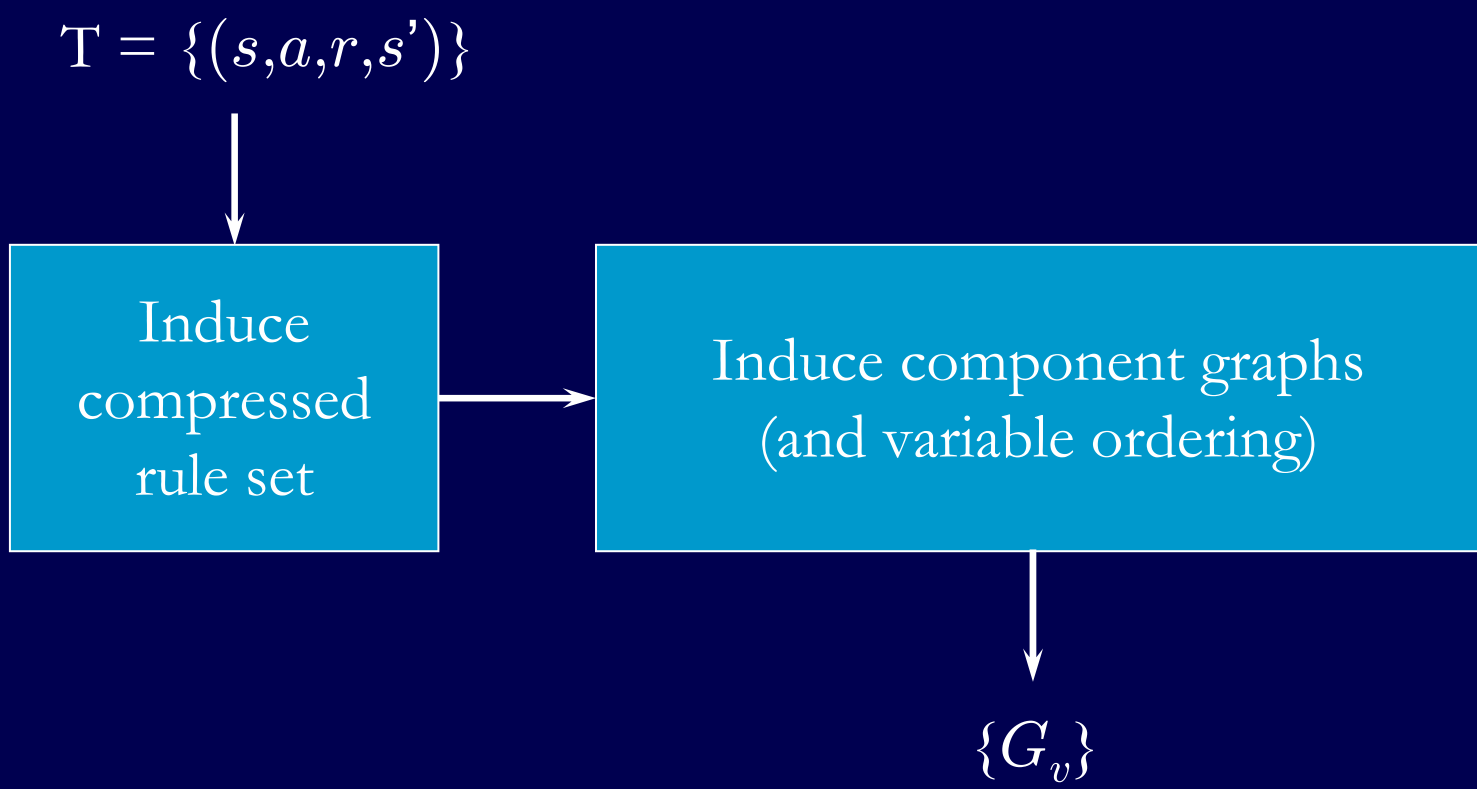


$taxi.y$

Notions Of Optimality

- Parameters:
  - $D = \max_v D(v)$
  - $b = \max_r |\rho(r)| - 1$
  - d = depth of variable hierarchy
  - k = max number of rules on an edge
- Global optimality: $O(D^n)$
- Hierarchical optimality: $O((Dbk)^n)$
  - Pick best rule to minimize pre and post cost
- Recursive optimality: $O((Dbk)^d)$
  - Pick best rule to minimize pre cost
- Agnostic(?) optimality: $O((Db)^d)$
  - Pick an arbitrary rule on the shortest path
  - Assumes all pre costs are equivalent

Learning Partial Models



Inducing Compressed Rule Set

- R $\leftarrow \emptyset$
- While T
  - Pick a transition (s,a,r,s') such that $s \neq s'$
  - Let $r = a : s \rightarrow s'$
  - Initialize $\rho = \{v : r.pre[v] \neq r.post[v]\}$
  - ρ = LearnContext($\{(s,a,r,s')\}$, r, ρ)
  - For $v \notin \rho$, $r.pre[v] \leftarrow r.post[v] \leftarrow *$
  - $R \leftarrow R \cup \{r\}$
  - $T \leftarrow T - \{(s,a,r,s')$ consistent with $r\}$

Inducing Compressed Rule Set

0 2 2
East
1 2 2
West
0 2 2
South
0 1 2
West
0 1 2

Inducing Compressed Rule Set

0 2 2
East
1 2 2
West
0 2 2
South
0 1 2
West
0 1 2

r = East: 0 * * $\longrightarrow$ 1 * *

Inducing Compressed Rule Set

0 2 2
East
1 2 2
West
0 2 2
South
0 1 2
East
0 1 2

r = East: 0 * * ⟶ 1 * *

Inducing Compressed Rule Set

0 2 2
East
1 2 2
West
0 2 2
South
0 1 2
East
0 1 2

r = East: 0 2 * $\longrightarrow$ 1 2 *

Inducing Compressed Rule Set

1 2 2
West
0 2 2
South
0 1 2
East
0 1 2

r = East: 0 2 * $\longrightarrow$ 1 2 *

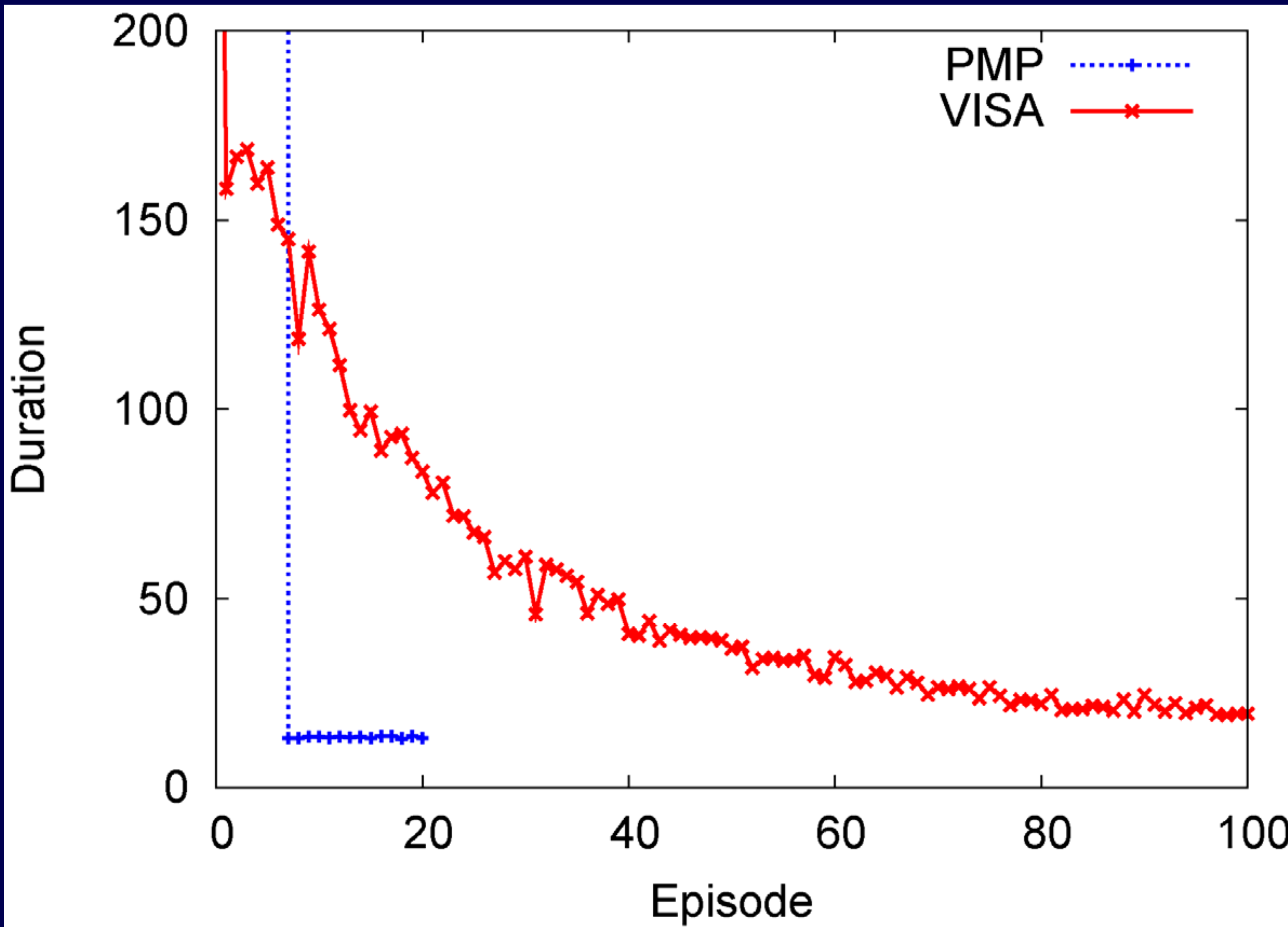
Inducing Component Graphs

- While Ω does not contain all variables
  - For $r \in R$
    - $v \in \rho(r) - \{u : u \text{ in } \Omega\}$ // Assuming a single variable
    - Insert edge $(r.pre[v], r.post[v])$ in G_v
  - For $v : v \notin \Omega \wedge G_v$
    - If G_v is strongly connected, append v to Ω
  - Delete all malformed component graphs

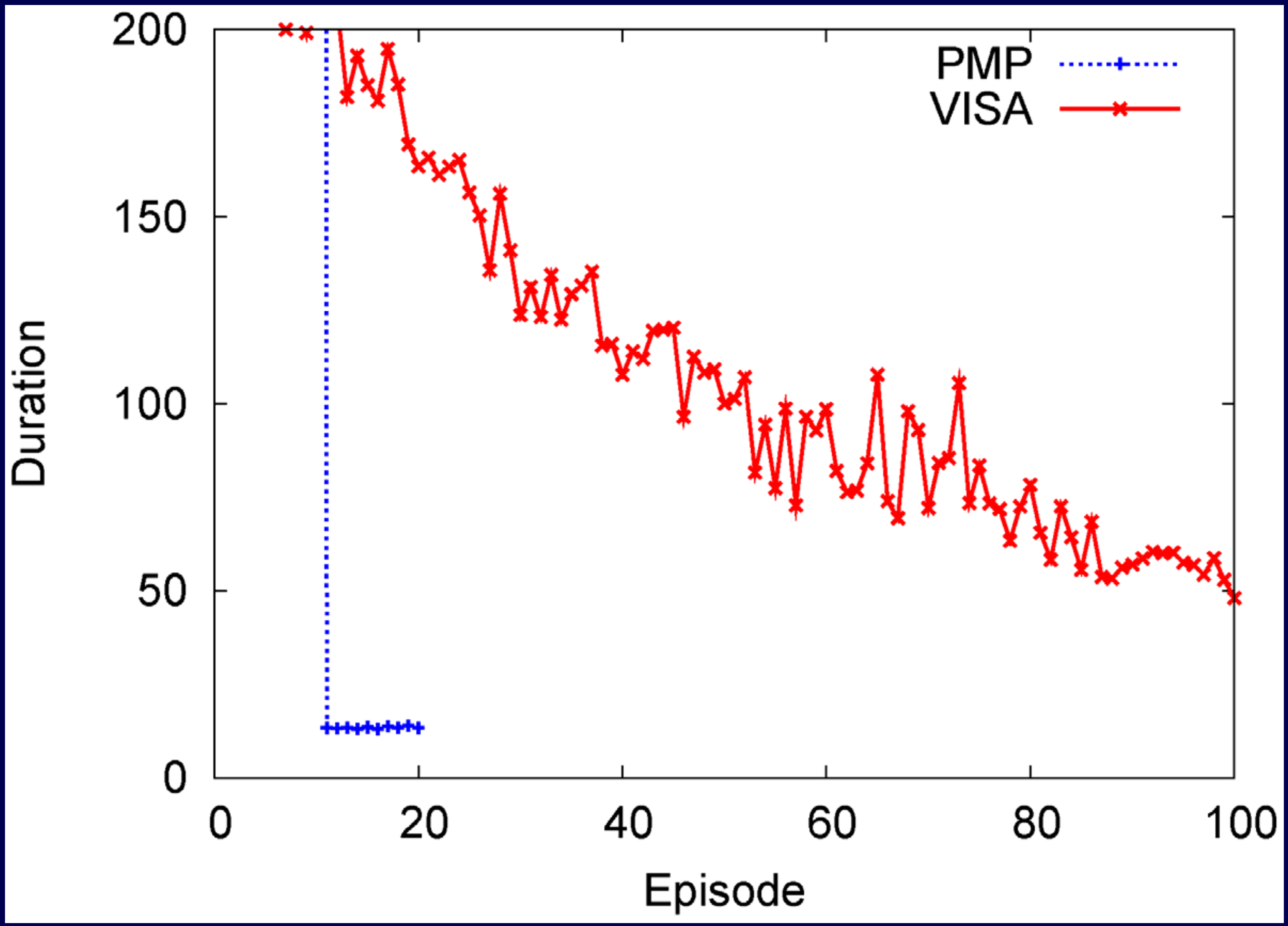
Empirical Evaluation

- Compare against VISA (Jonsson & Barto, 06)
  - Learns an exit-option hierarchy from fully specified DBN models
  - Does intra-option Q learning
- Domains
  - Regular Taxi domain
  - Modified Taxi domain
    - Pickup when $pass.loc$ = T causes $taxi.y$++

Empirical Evaluation: Regular Taxi



Empirical Evaluation: Modified Taxi



Results

- PMP learns the appropriate variable ordering and an adequate model in ~8 episodes
- Recursive optimality coincides with global optimality in the Taxi domain
- VISA induces tight structure in Regular Taxi but only the shallowest hierarchy for Modified Taxi
  - Entire causal graph is strongly connected
- PMP's partial model stays unchanged

Conclusion

- Partial models can be learned from random trajectories
- PMP can be made exponential in d even when VISA and factored planning are exponential in n
- Future work
  - Bounding PMP's plan length vs. optimal
  - Generalizing factors to sets of variables
  - Easing the requirement of strong connectedness