Planning, Learning and Decision Making

Lecture 11. POMDPs (conc.)

Representing $J^{(k)}$

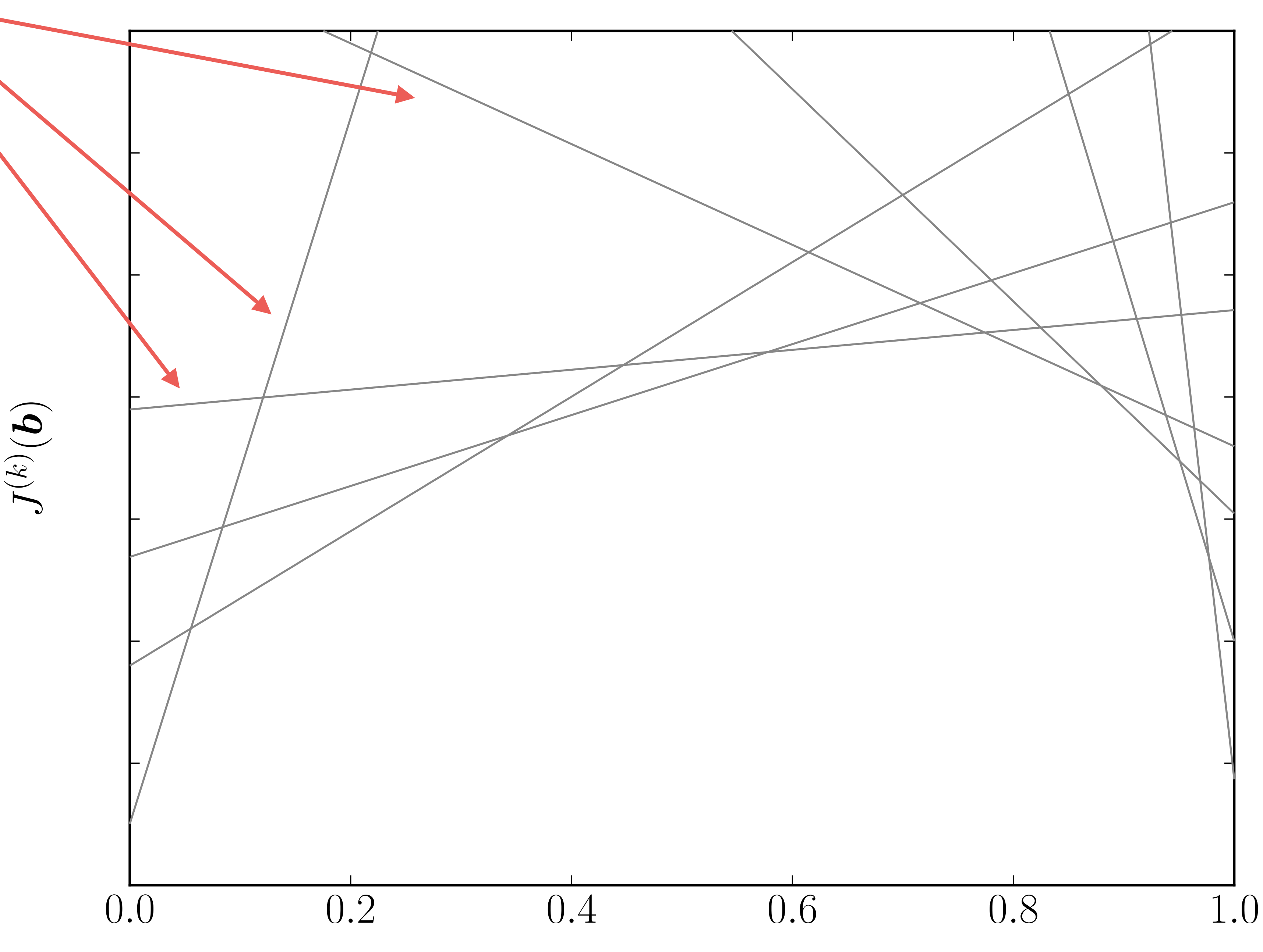
- The cost-to-go at each iteration of VI is always PWLC
  - Can always be written in the form

$$
\begin{aligned}
J^{(k)}(\boldsymbol{b}) &= \min_{\alpha \in \boxed{\Gamma}} \boldsymbol{b} \cdot \boldsymbol{\alpha} \\
&= \min_{\alpha \in \boxed{\Gamma}} \sum_{x \in \mathcal{X}} \boldsymbol{b}(x)\boldsymbol{\alpha}(x)
\end{aligned}
$$

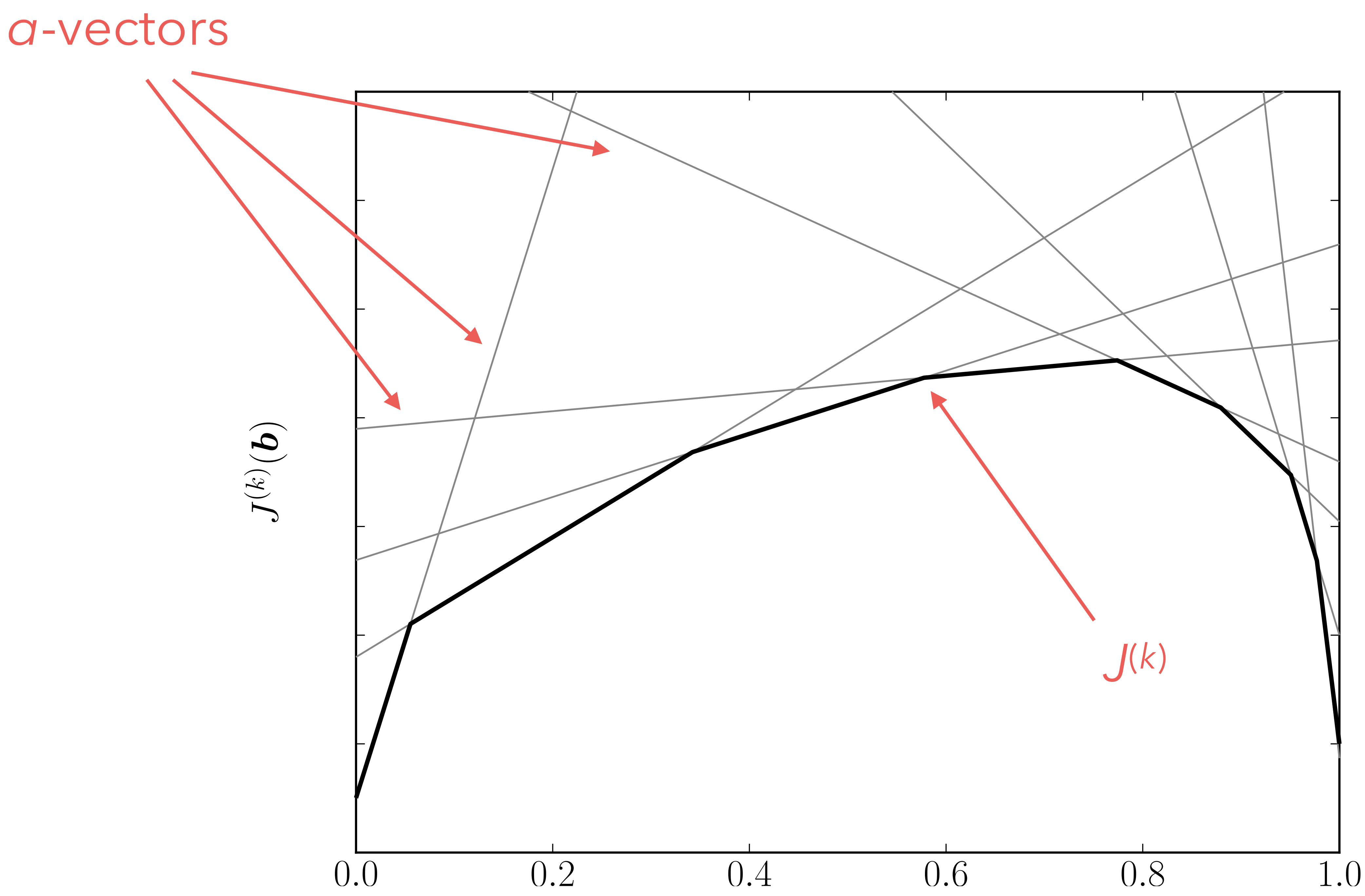
Set of vectors used in the representation

Representing $J^{(k)}$

α-vectors



Representing $J^{(k)}$



Value iteration

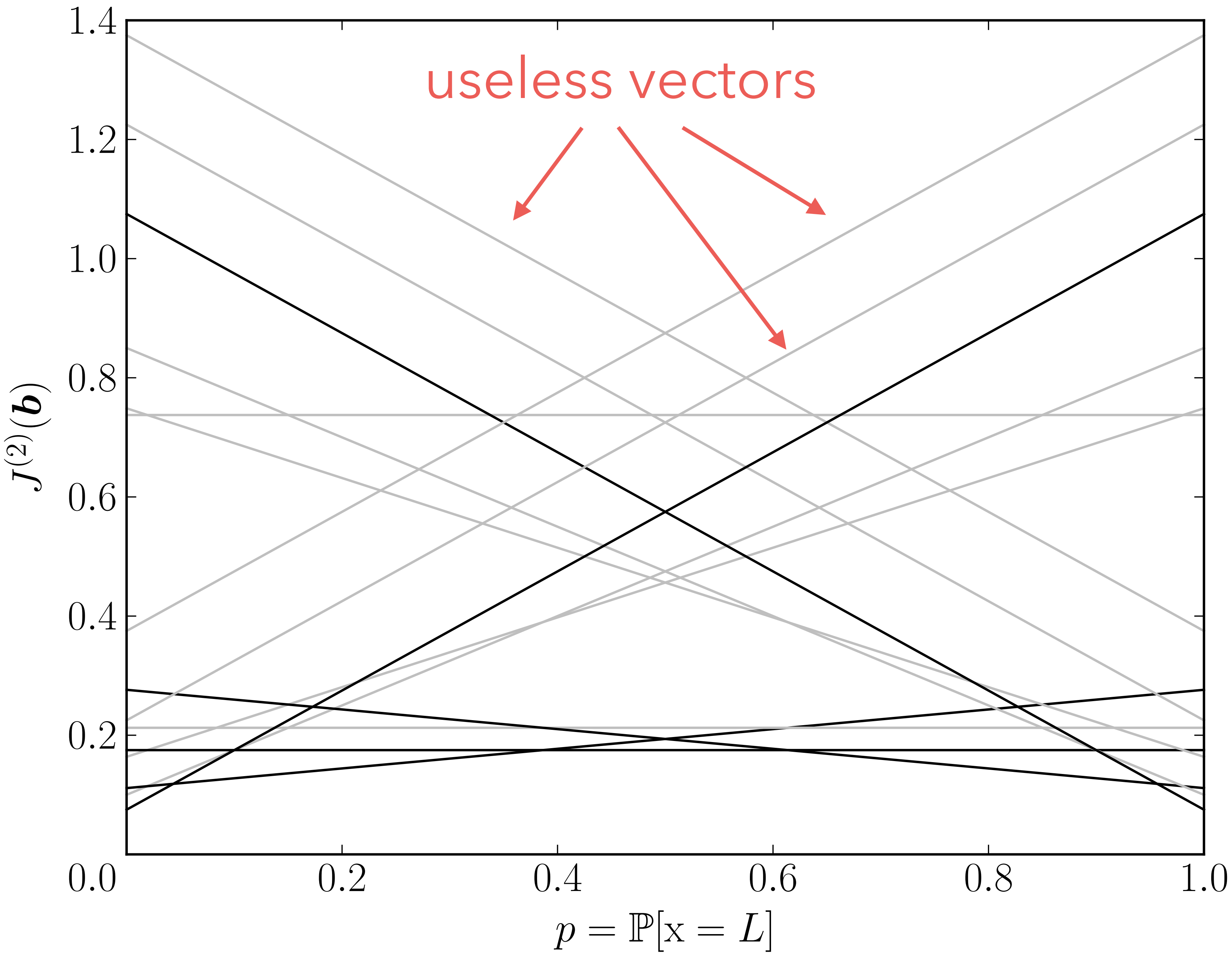
- Compute, at each iteration $k + 1$, the set $\Gamma^{(k+1)}$ from $\Gamma^{(k)}$
  - For each $\boldsymbol{\alpha} \in \Gamma^{(k)}$, compute

$$\boldsymbol{\alpha}_{a,z}^{(k)} = \frac{1}{|\mathcal{Z}|}\boldsymbol{C}_{:,a} + \gamma \mathrm{P}_a \mathrm{diag}(\boldsymbol{O}_{z,a})\boldsymbol{\alpha}$$

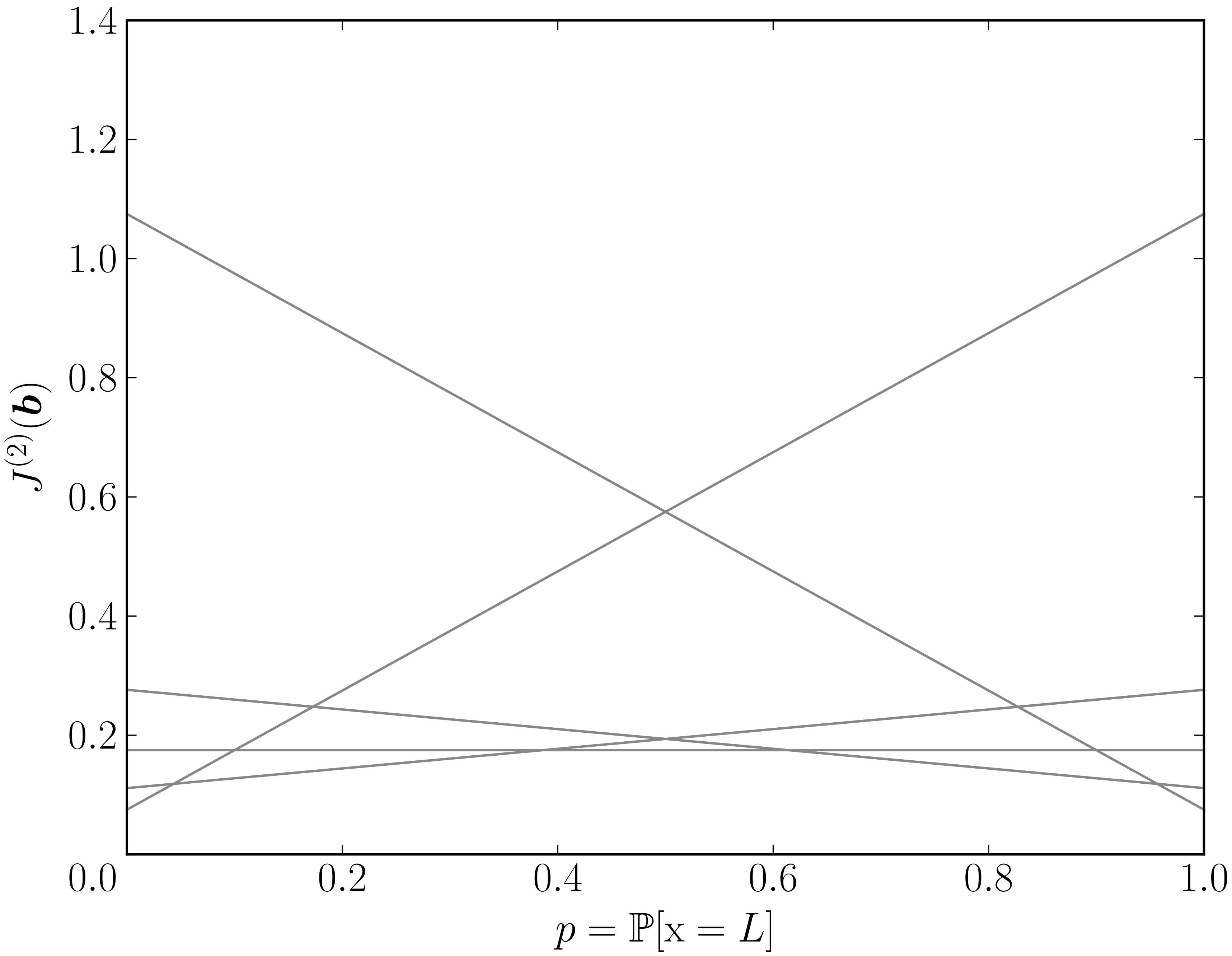
  - Compute **all possible combinations** of $\boldsymbol{\alpha}_{a,z}^{(k)}$, for each z
  - For each combination, let

$$\boldsymbol{\alpha}_a^{(k)} = \sum_{z \in \mathcal{Z}} \boldsymbol{\alpha}_{z,a}^{(k)}$$

Value iteration



Value iteration



Value iteration

- Two approaches to build $\Gamma^{(k+1)}$ from $\Gamma^{(k)}$:
  - **Region based methods:** Start with empty $\Gamma^{(k+1)}$ and only add vectors that are necessary

A vector is necessary if it represents J in a non empty belief region (witness region)

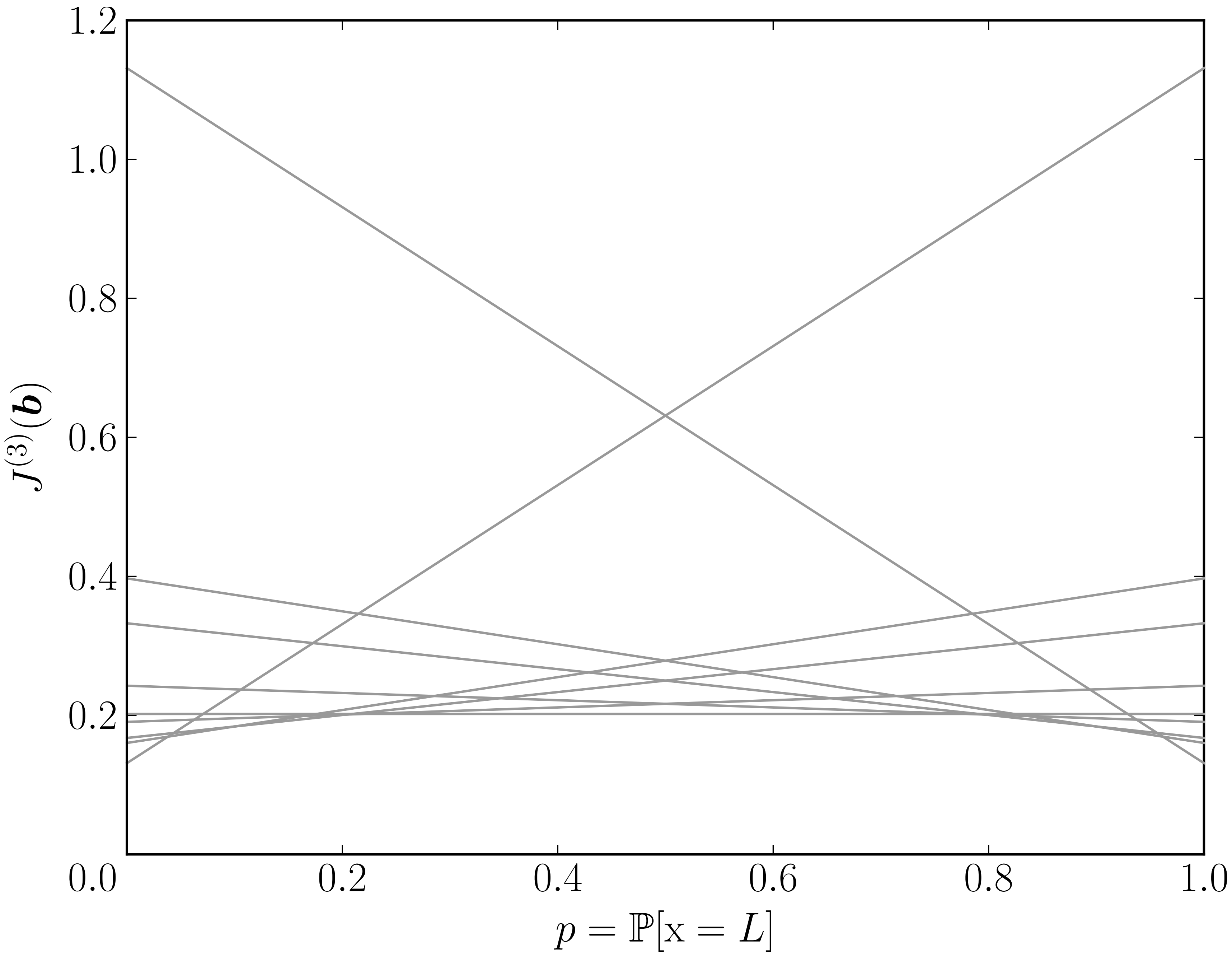
Value iteration

- Two approaches to build $\Gamma^{(k+1)}$ from $\Gamma^{(k)}$:
  - **Region based methods:** Start with empty $\Gamma^{(k+1)}$ and only add vectors that are necessary

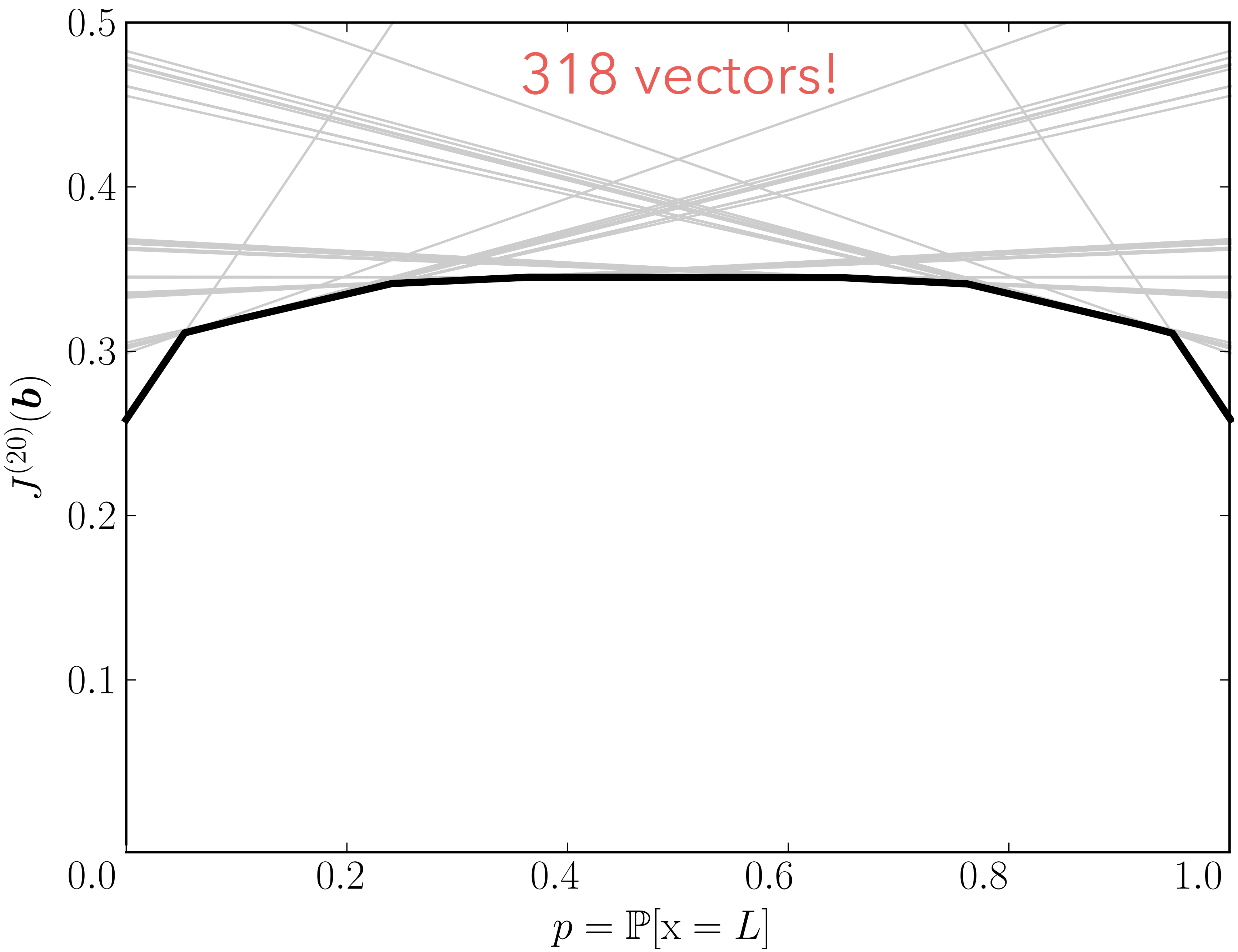
    Example: Witness algorithm
  - **Pruning-based methods:** Start with complete $\Gamma^{(k+1)}$ and remove vectors that are unnecessary

    Example: Incremental pruning

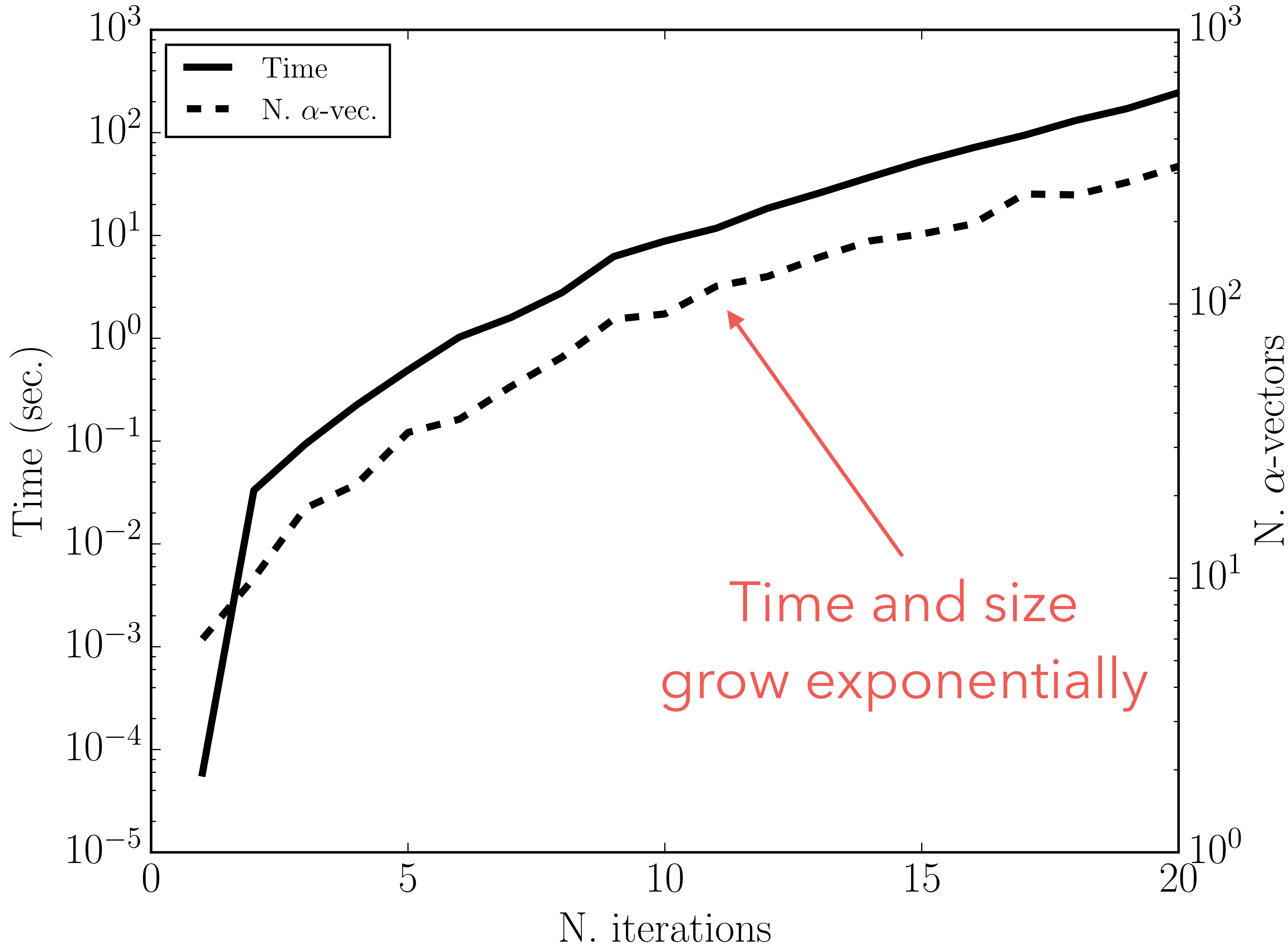
Value iteration



Value iteration



Computation time



Non-exact solutions

Idea n. 1 - Use the MDP

- MLS heuristic:

$$\pi_{\text{MLS}}(\boldsymbol{b}) = \pi_{\text{MDP}}(\operatorname*{argmax}_{x \in \mathcal{X}} \boldsymbol{b}(x))$$

- AV heuristic:

$$\pi_{\text{AV}}(\boldsymbol{b}) = \operatorname*{argmax}_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} \boldsymbol{b}(x) \mathbb{I}(a = \pi_{\text{MDP}}(x))$$

- Q-MDP heuristic:

$$\pi_{\text{Q}-\text{MDP}}(\boldsymbol{b}) = \operatorname*{argmin}_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} \boldsymbol{b}(x) Q_{\text{MDP}}(x, a)$$

- FIB heuristic:

$$\pi_{\text{FIB}}(\boldsymbol{b}) = \operatorname*{argmax}_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} \boldsymbol{b}(x) Q_{\text{FIB}}(x, a)$$

Idea n. 1 - Use the MDP

- MLS heuristic:

$$\pi_{\text{MLS}}(\boldsymbol{b}) = \pi_{\text{MDP}}(\operatorname*{argmax}_{x \in \mathcal{X}} \boldsymbol{b}(x))$$

- AV heuristic:

$$\pi_{\text{AV}}(\boldsymbol{b}) = \operatorname*{argmax}_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} \boldsymbol{b}(x)\mathbb{I}(a = \pi_{\text{MDP}}(x))$$

- Q-MDP heuristic:

$$\pi_{\text{Q-MDP}}(\boldsymbol{b}) = \operatorname*{argmin}_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} \boldsymbol{b}(x)Q_{\text{MDP}}(x, a)$$

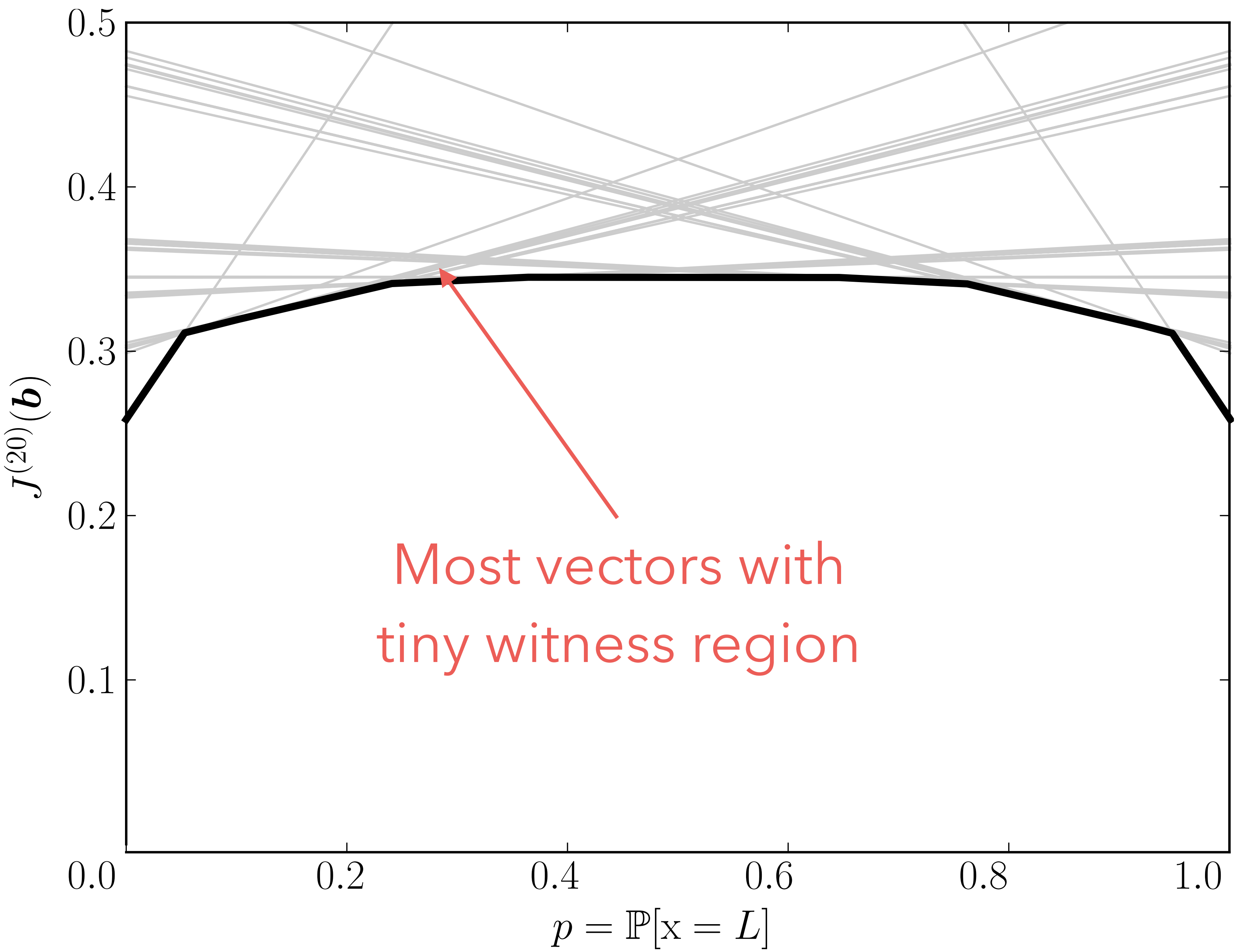
- FIB heuristic:

$$\pi_{\text{FIB}}(\boldsymbol{b}) = \operatorname*{argmax}_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} \boldsymbol{b}(x)Q_{\text{FIB}}(x, a)$$

No guarantees

TÉCNICO LISBOA

Idea n. 2



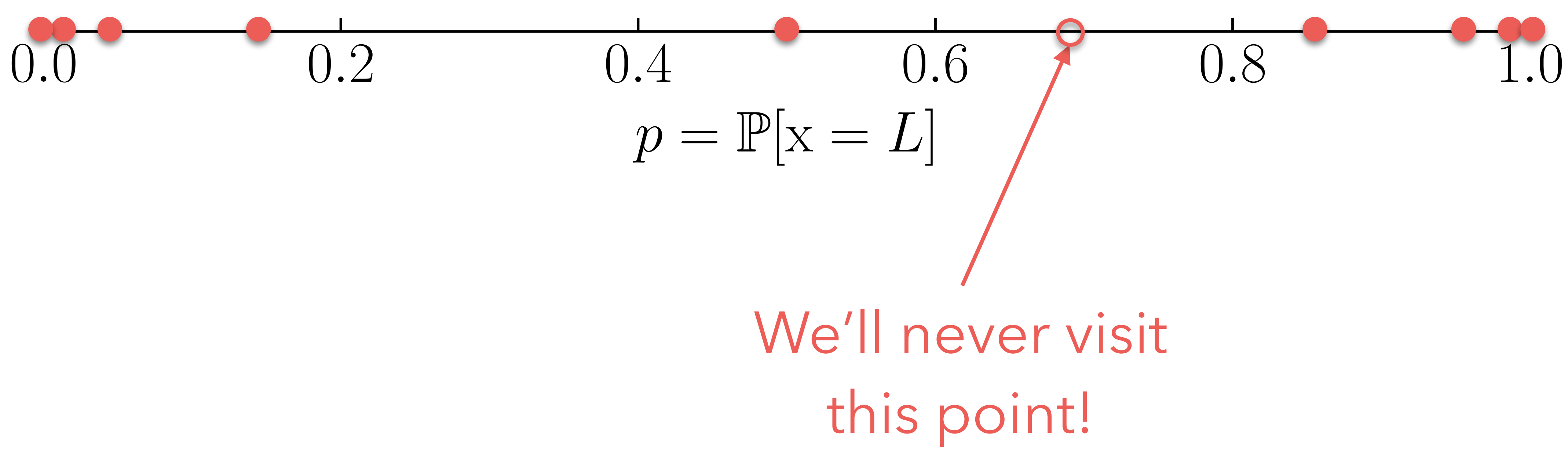
Idea n. 2

- Most α-vectors play little role in representing J
- What if we only compute the vectors that “truly matter”?

TÉCNICO LISBOA

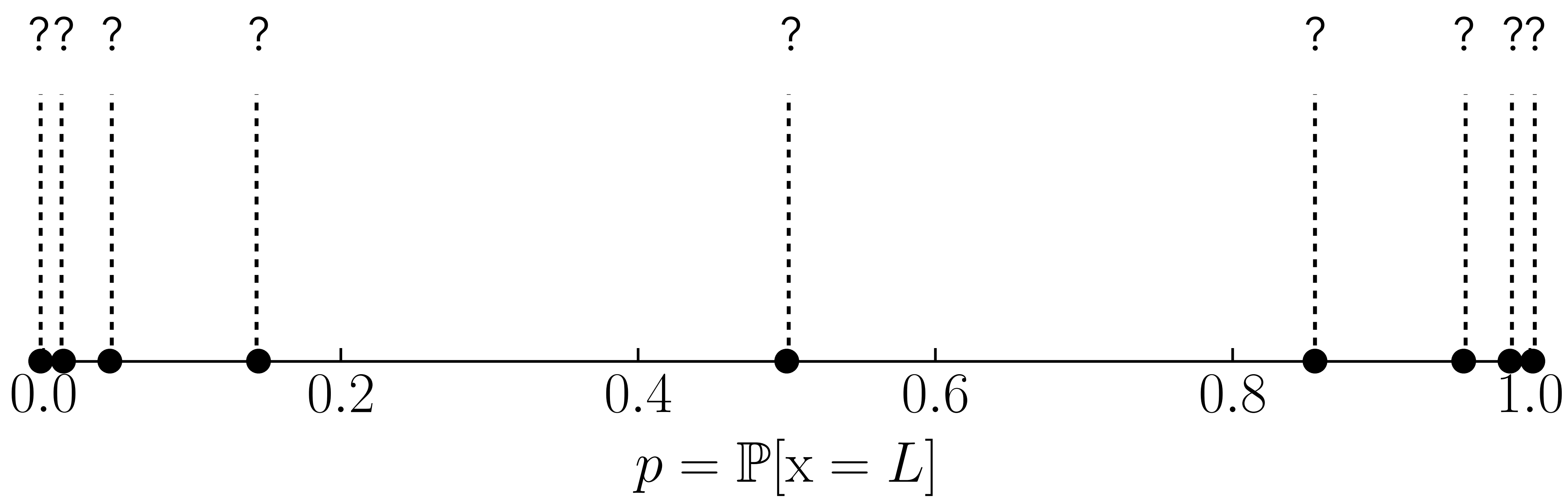
Which ones?

Idea n. 2


0.0
0.2
0.4
0.6
0.8
1.0
$p = \mathbb{P}[\mathrm{x} = L]$
We'll never visit
this point!

Idea n. 2

We only care about the vectors in these points



Point-based methods

- Select a finite set $\mathscr{B}_{\text{sample}}$ of beliefs to perform update
  - For each belief, compute the corresponding α-vector

$$\boldsymbol{\alpha}(\boldsymbol{b}) = \min_{a \in \mathcal{A}} \left[\boldsymbol{C}_{:,a} + \gamma \sum_{z \in \mathcal{Z}} \mathrm{P}_a \mathrm{diag}(\boldsymbol{O}_{z,a}) \min_{\boldsymbol{\alpha} \in \Gamma} \boldsymbol{\alpha} \cdot \boldsymbol{b}'_{za} \right]$$

Updated
belief

Point-based methods

- Select a finite set $\mathscr{B}_{\text{sample}}$ of beliefs to perform update
  - For each belief, compute the corresponding α-vector

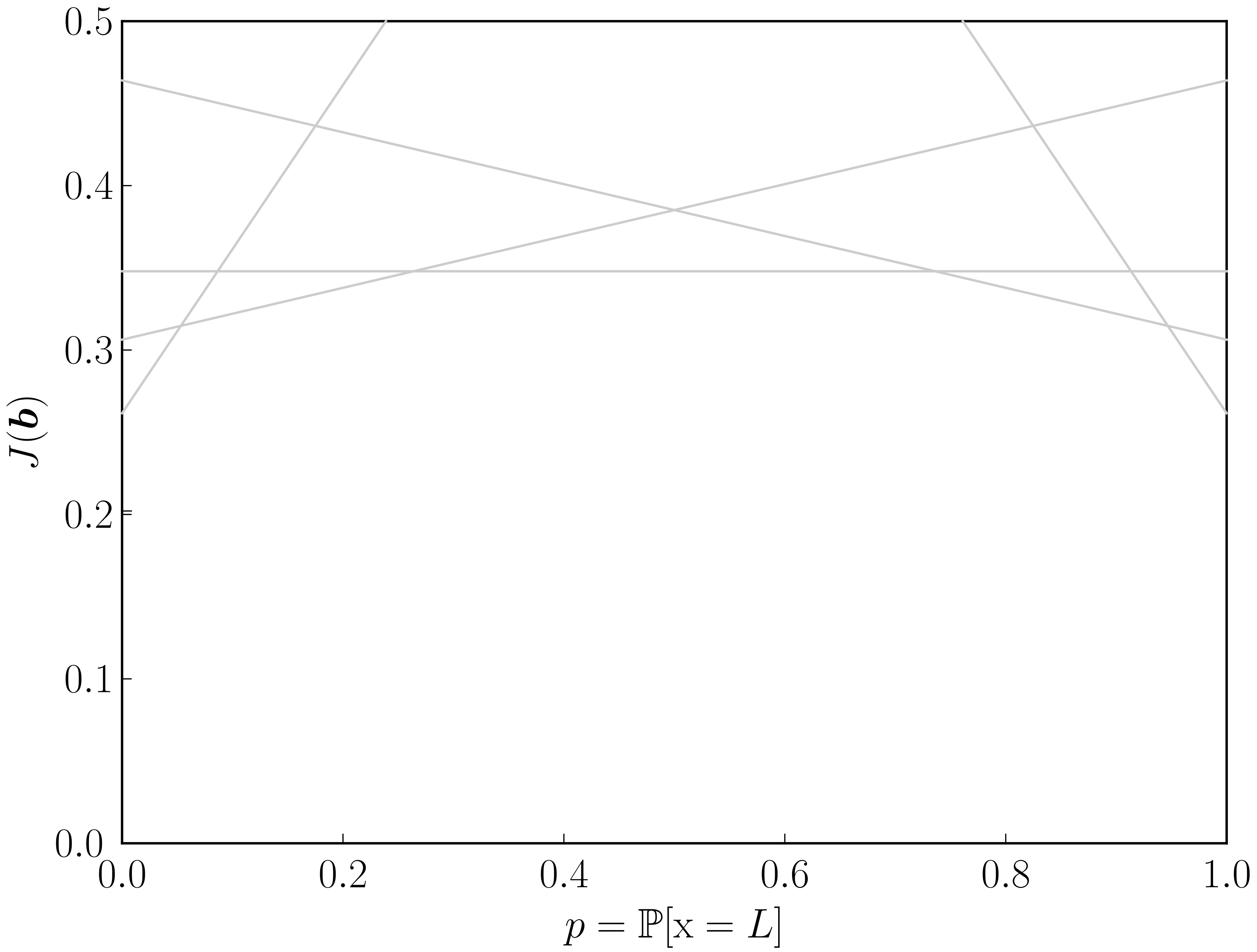
$$\boldsymbol{\alpha}(\boldsymbol{b}) = \min_{a \in \mathcal{A}} \left[\boldsymbol{C}_{:,a} + \gamma \sum_{z \in \mathcal{Z}} \mathrm{P}_a \mathrm{diag}(\boldsymbol{O}_{z,a}) \min_{\boldsymbol{\alpha} \in \Gamma} \boldsymbol{\alpha} \cdot \boldsymbol{b}'_{za} \right]$$

  - If necessary, rebuild $\mathscr{B}_{\text{sample}}$

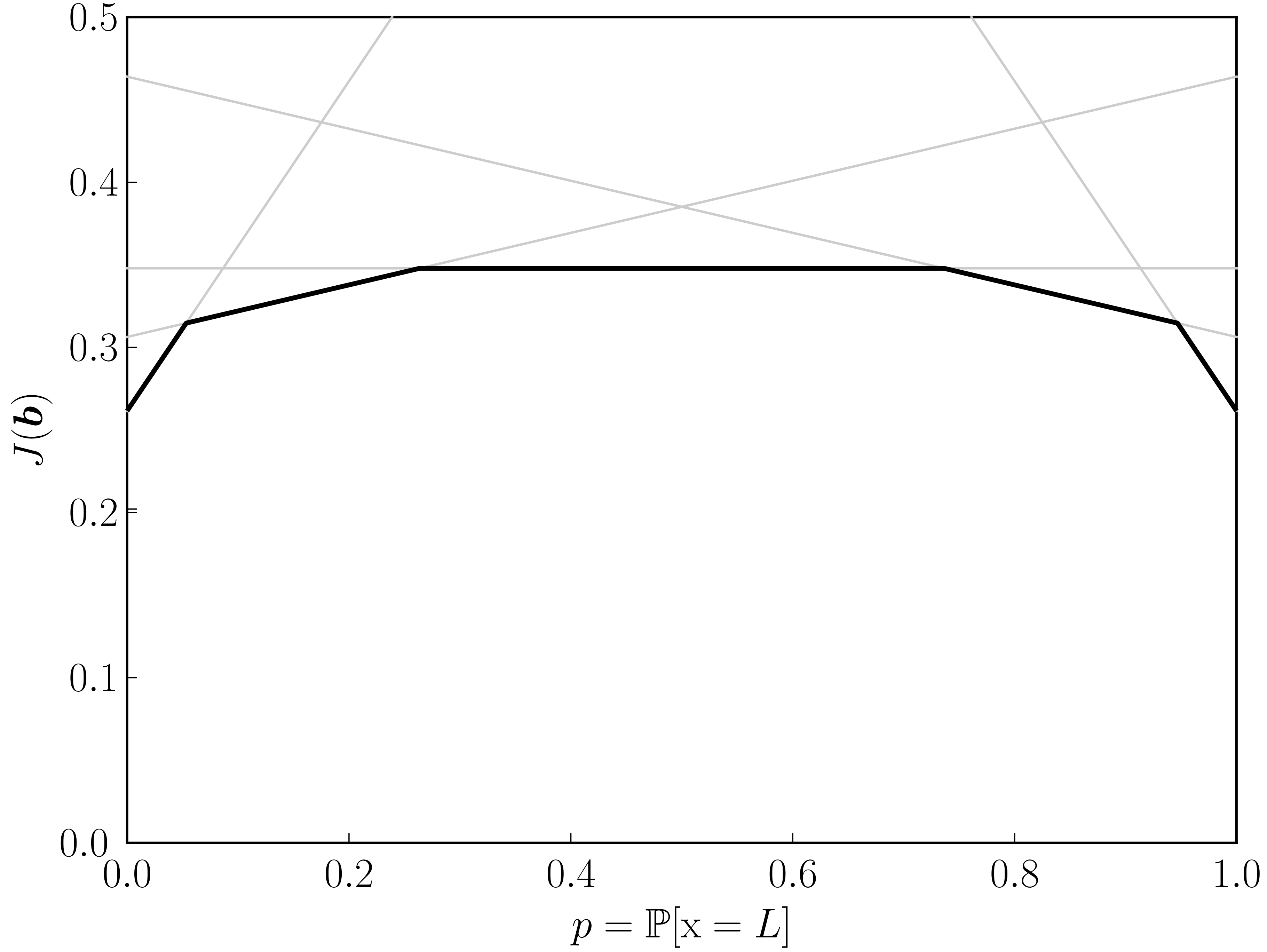
Point-based methods

- Many point-based methods:
  - PBVI (Pineau et al., 2003)
  - Perseus (Spaan & Vlassis, 2005)
  - HSVI (Smith & Simmons, 2005)
  - FSVI (Shani et al., 2007)
  - SARSOP (Kurniawati et al., 2008)
  - GapMin (Poupart et al., 2011)
- Much code available

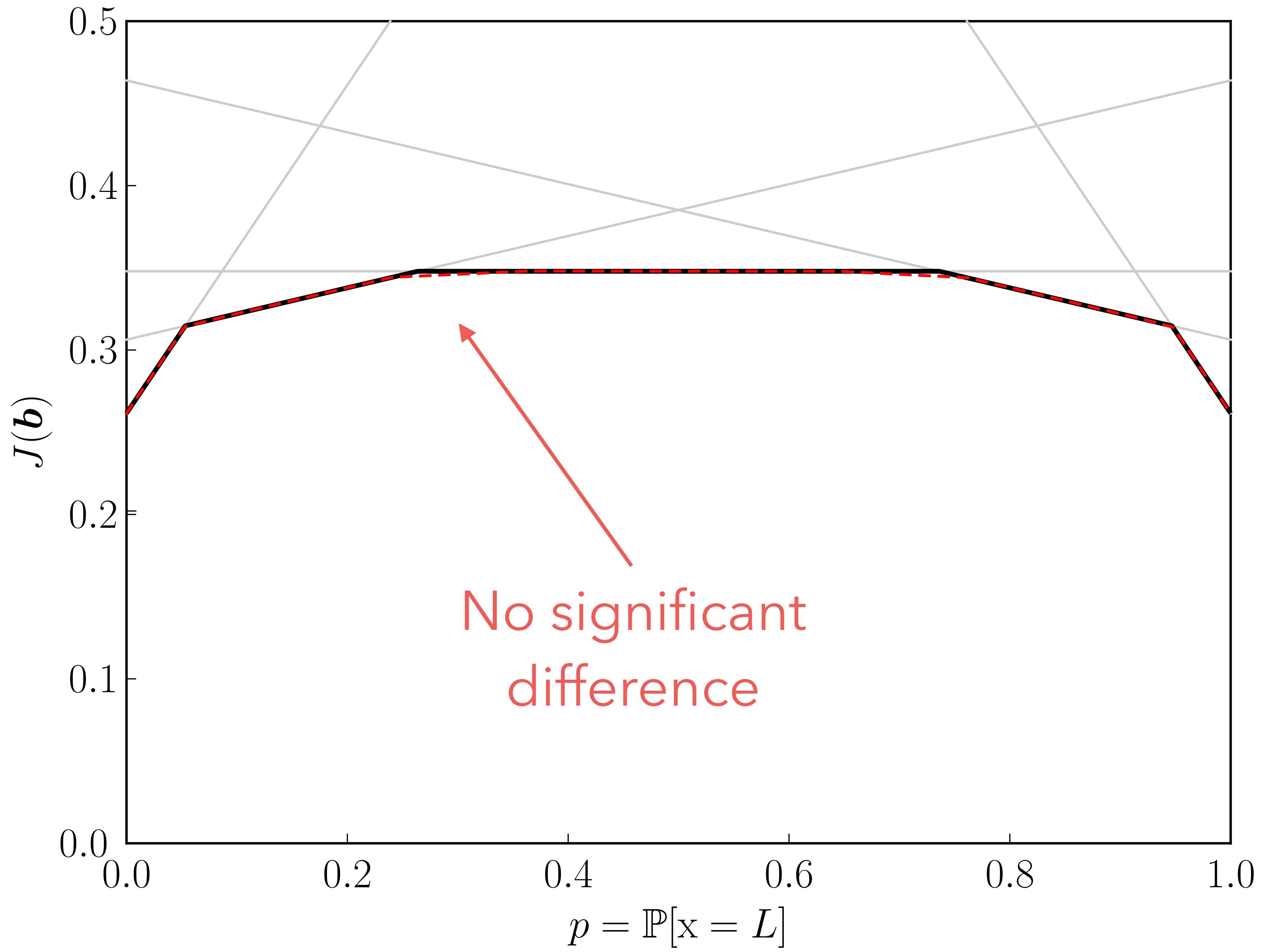
Point-based methods



Point-based methods



Point-based methods



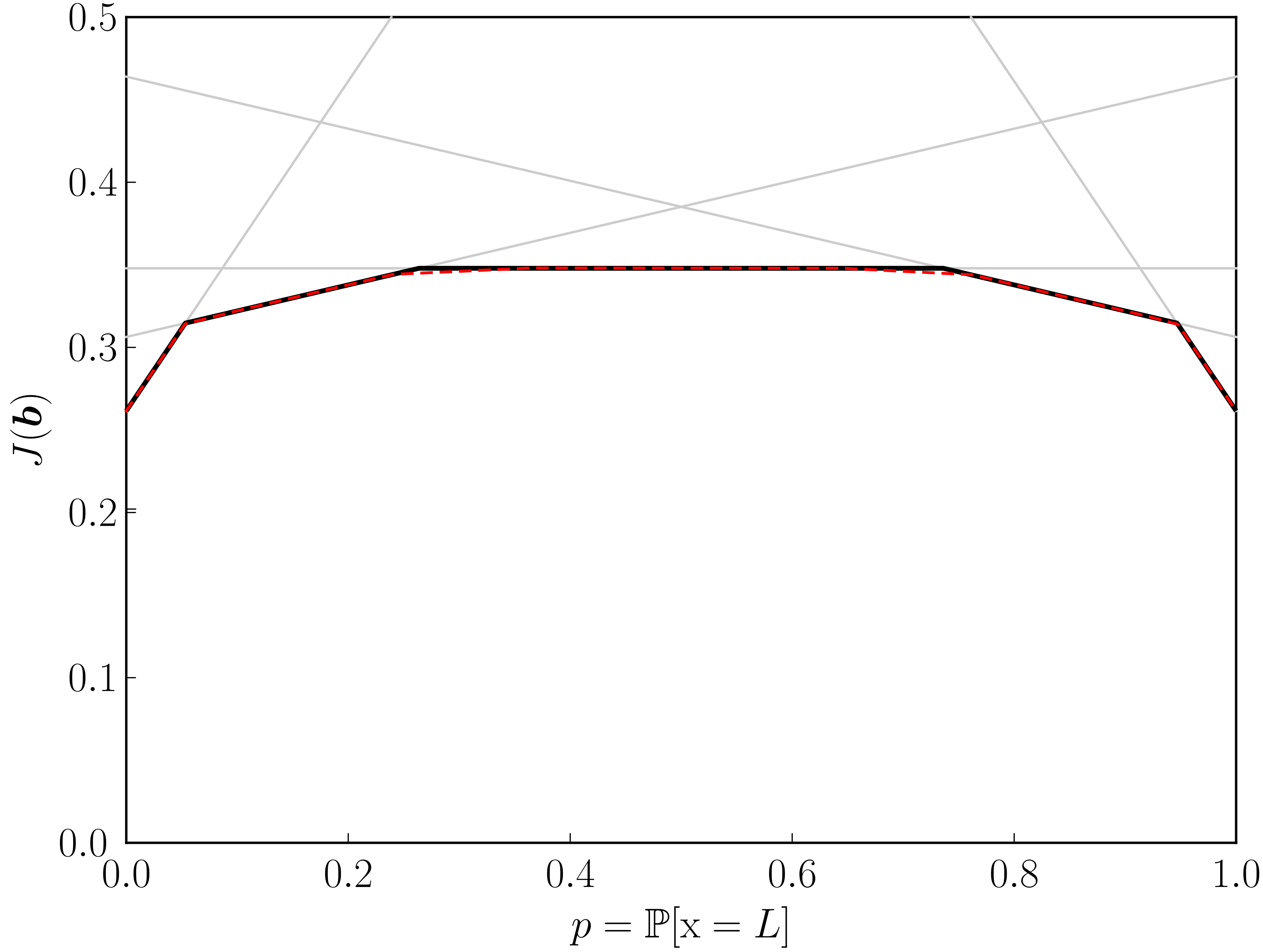
Point-based methods

VI:

- 318 vectors
- ~4 minutes

PERSEUS:

- 5 vectors
- 226 ms



What about policy iteration?

Policy iteration?

- Value iteration for POMDPs
  - How do we represent a cost-to-go function?
  - At each iteration of VI, the cost-to-go is PWLC

- Policy iteration for POMDPs
  - How do we represent a policy?

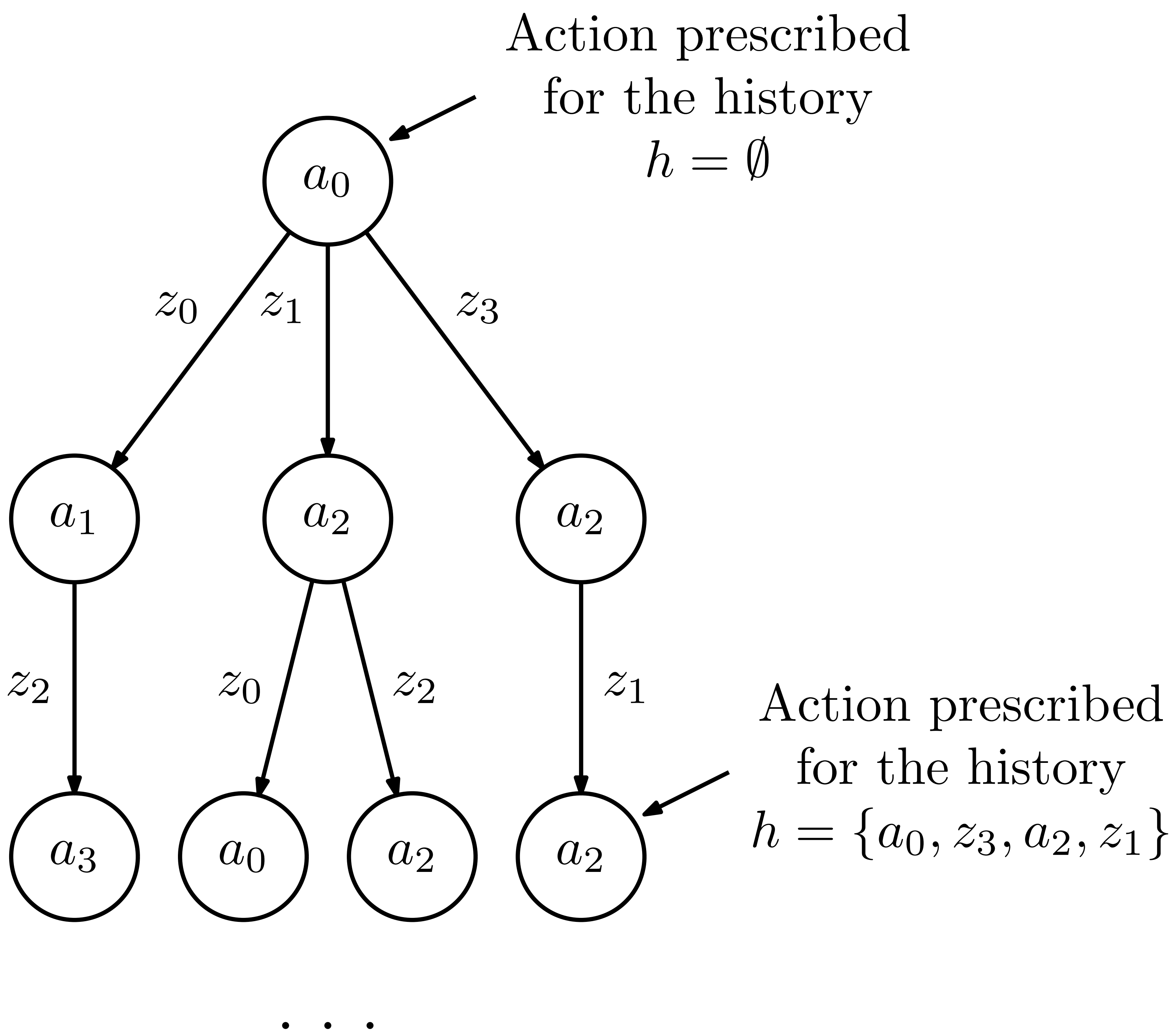
Policy graphs

- How can we represent a POMDP policy?
  - Compute it in runtime from J
  - Alternatively, we can consider policies as mapping **histories** to **actions**

Policy graphs

- We can represent the possible histories in a **policy tree**
  - Each node contains the **action** for that history
  - Branches correspond to **observations** from the node

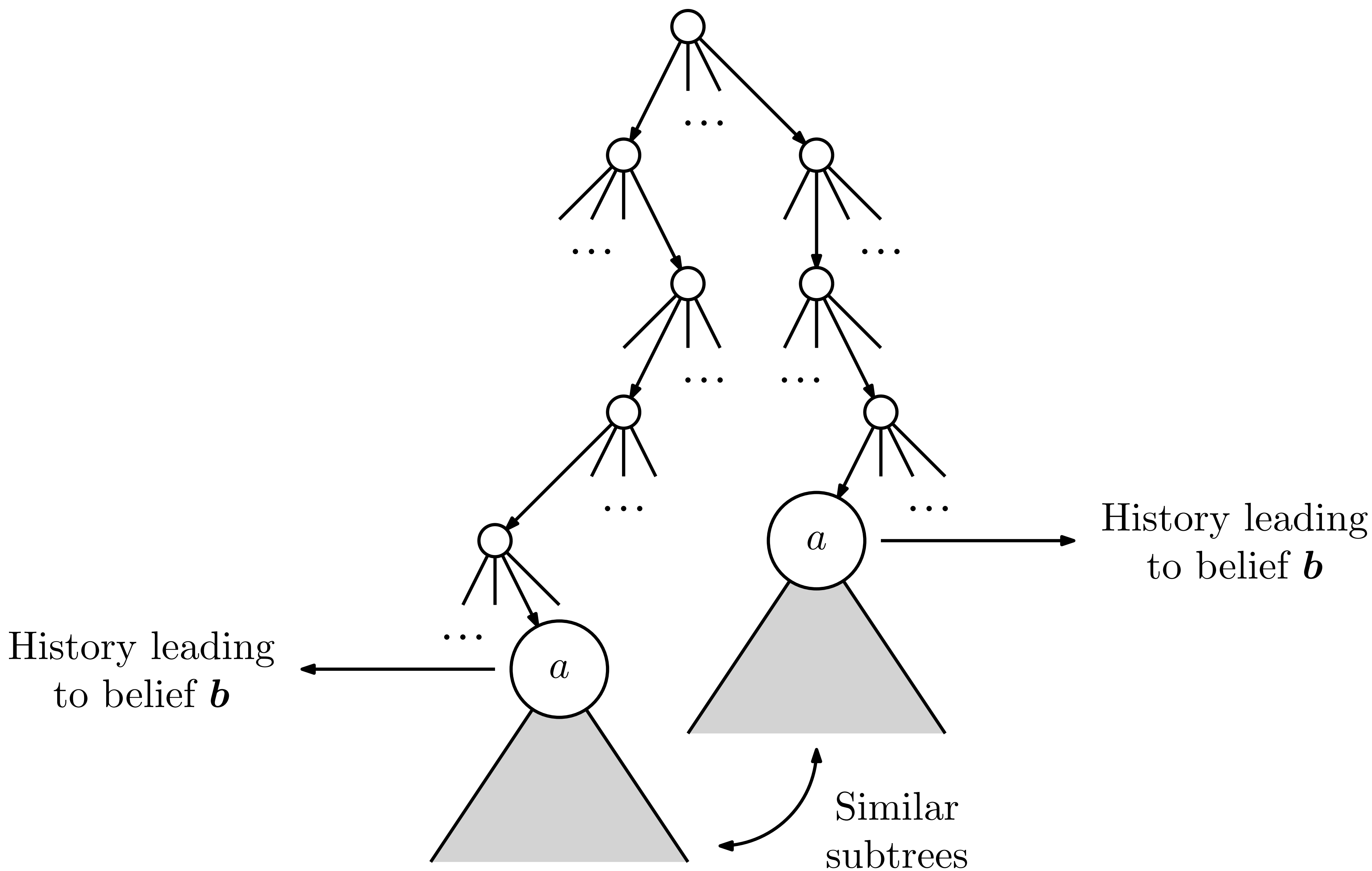
Policy graphs



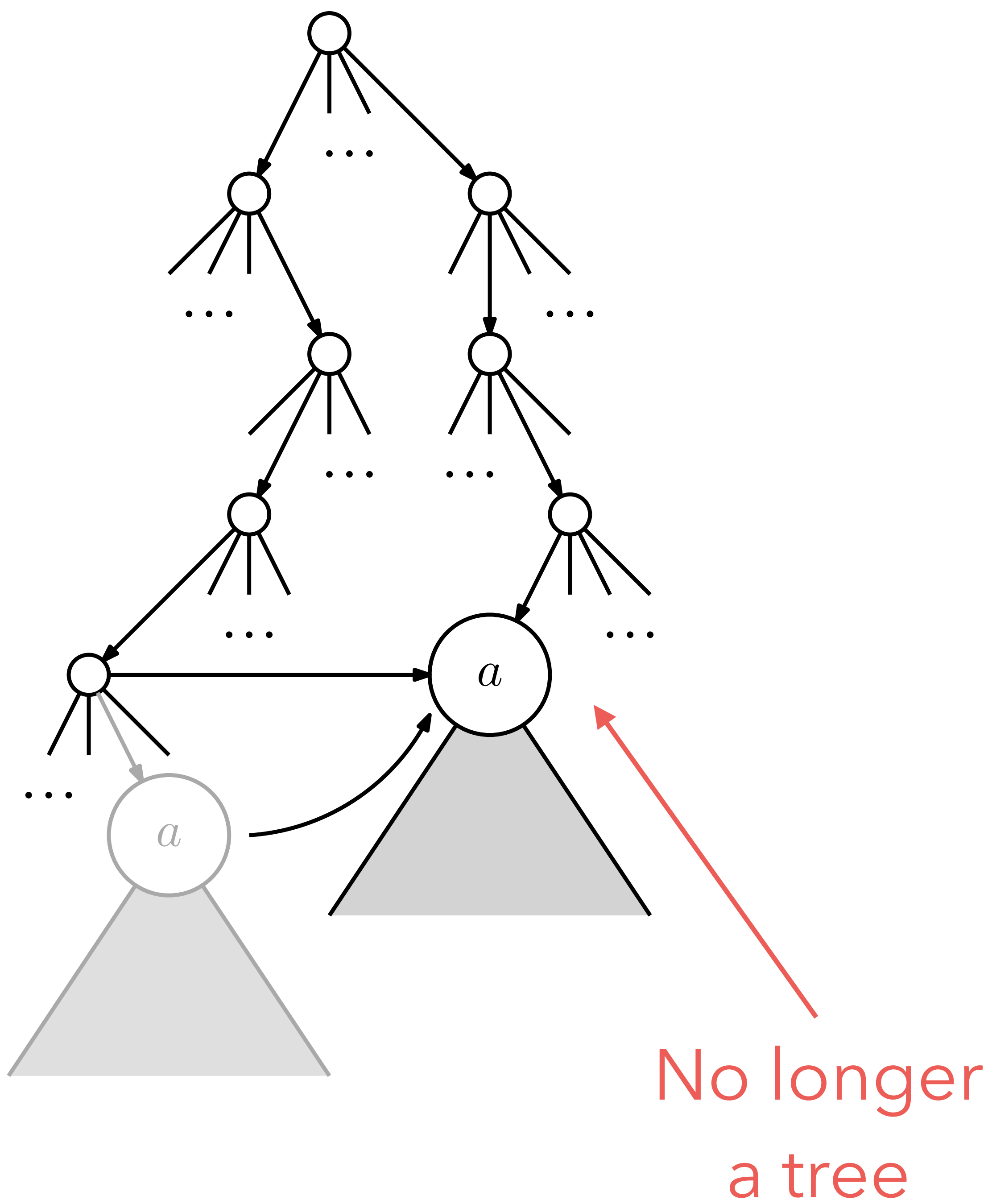
Policy graphs

- There is a lot of redundancy in policy trees
  - Histories leading to the same belief will have equivalent subtrees

Policy graphs



Policy graphs

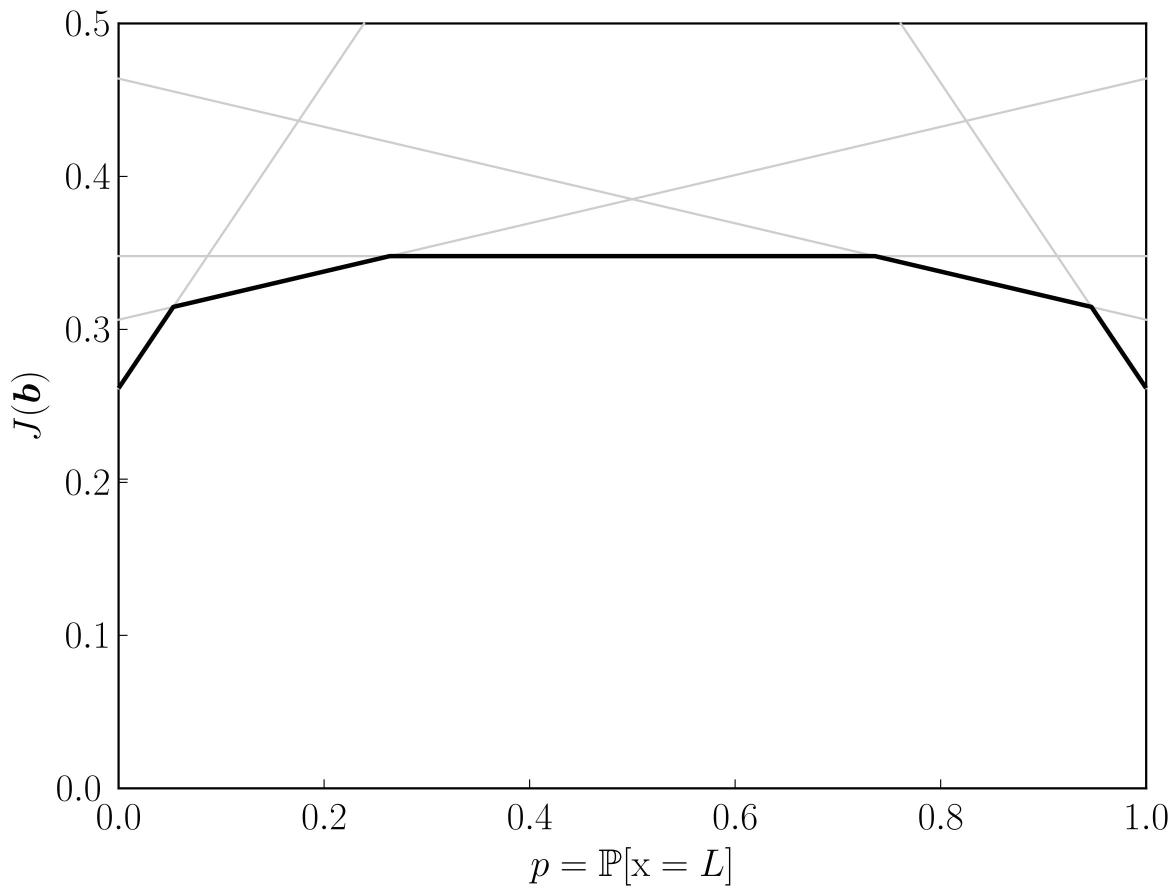


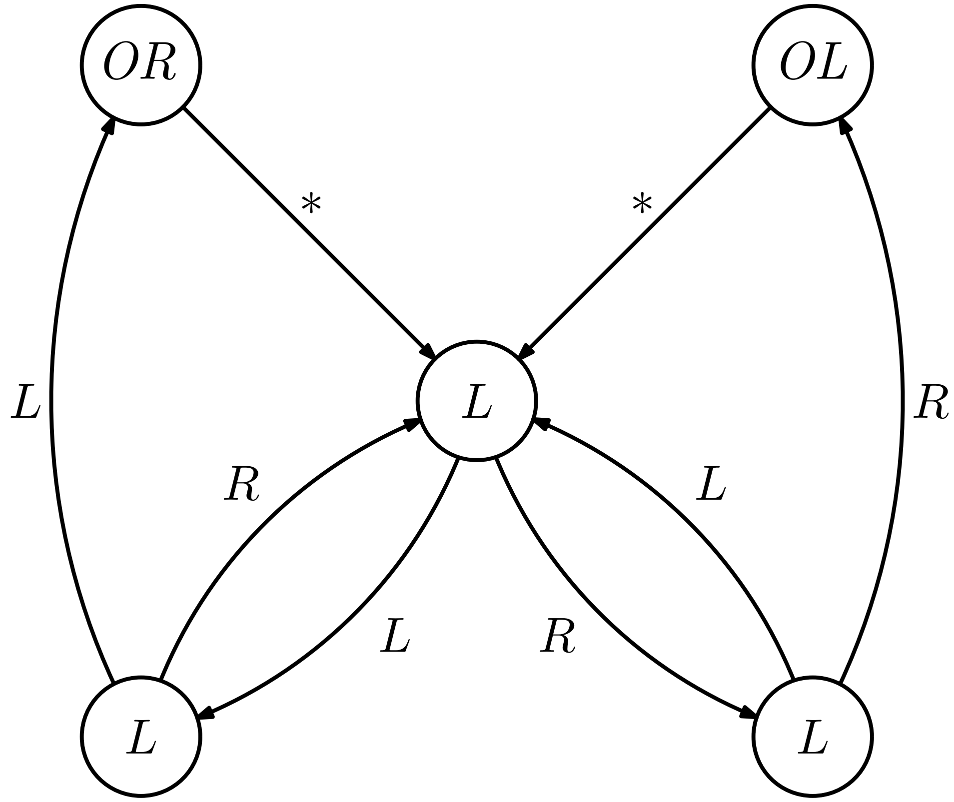
Policy graphs

- Policy graphs provide convenient representations for POMDP policies
- Close relation between policy graphs and α-vector
  - Each node in the graph corresponds to an α-vector and vice-versa

Policy graphs

- Example:





Key points about POMDPs

- Very general model for decision making under uncertainty
- Very hard to solve
- Beliefs provide a **summary** of the history
- POMDP $\longleftrightarrow$ Belief MDP
  - We can use VI → Cost-to-go is PWLC (finite representation)
  - We can use PI → Policy graphs w/ finite number of nodes
- Approximate methods:
  - MDP heuristics
  - Point-based methods

Comments on complexity

How hard is an MDP?

- MDPs can be solved by a linear program
  - LP is known to be **polynomial-time (P)**
  - MDPs are solvable in **polynomial-time (P)**

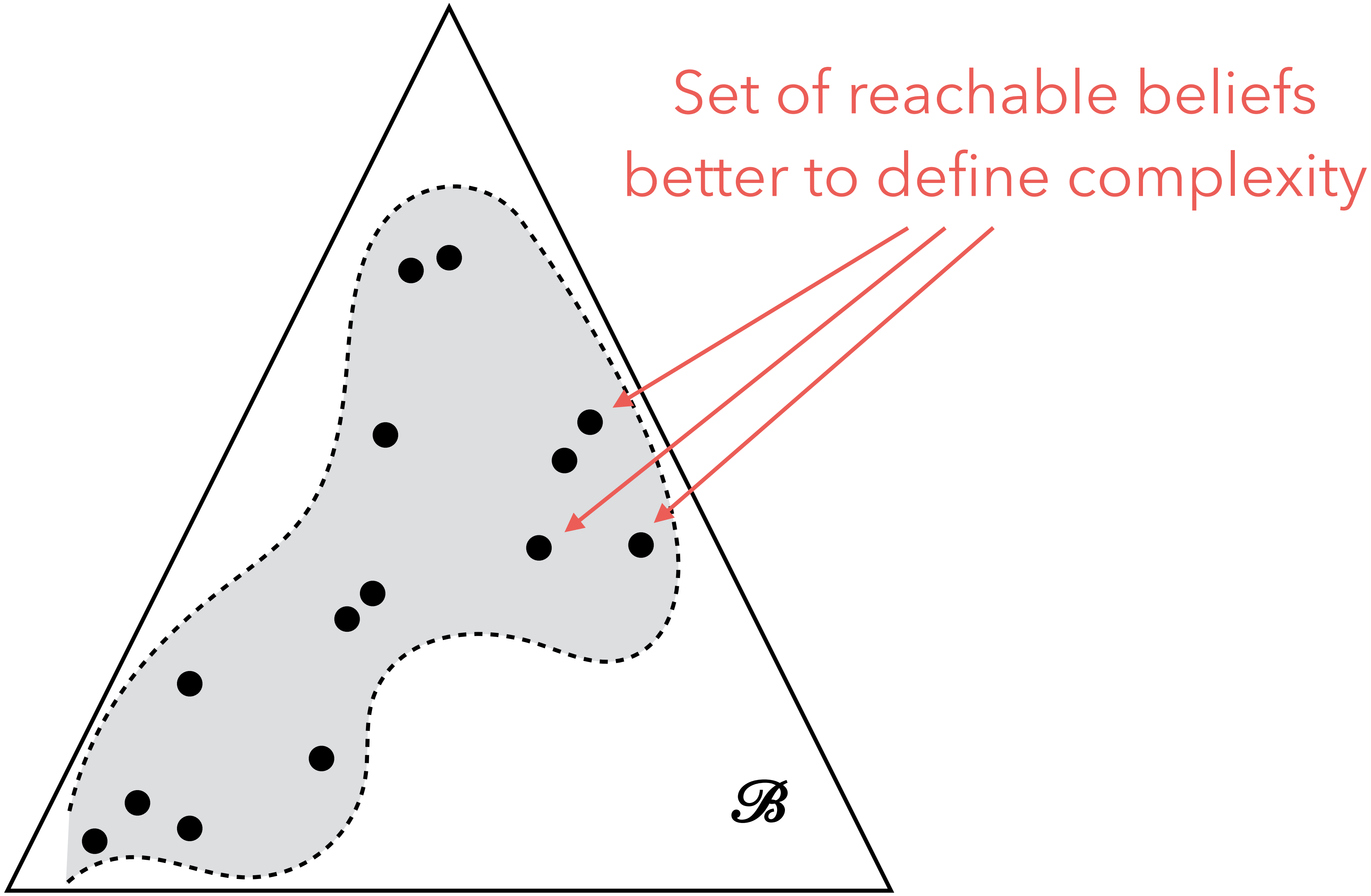
How hard is a POMDP?

- Infinite horizon POMDPs are **undecidable**
- Finite-horizon POMDPs are **PSPACE-complete**
  - … little hope for exact solution methods
- POMDPs are **non-approximable**
  - … in the worst case, you can't even guarantee a good approximation!

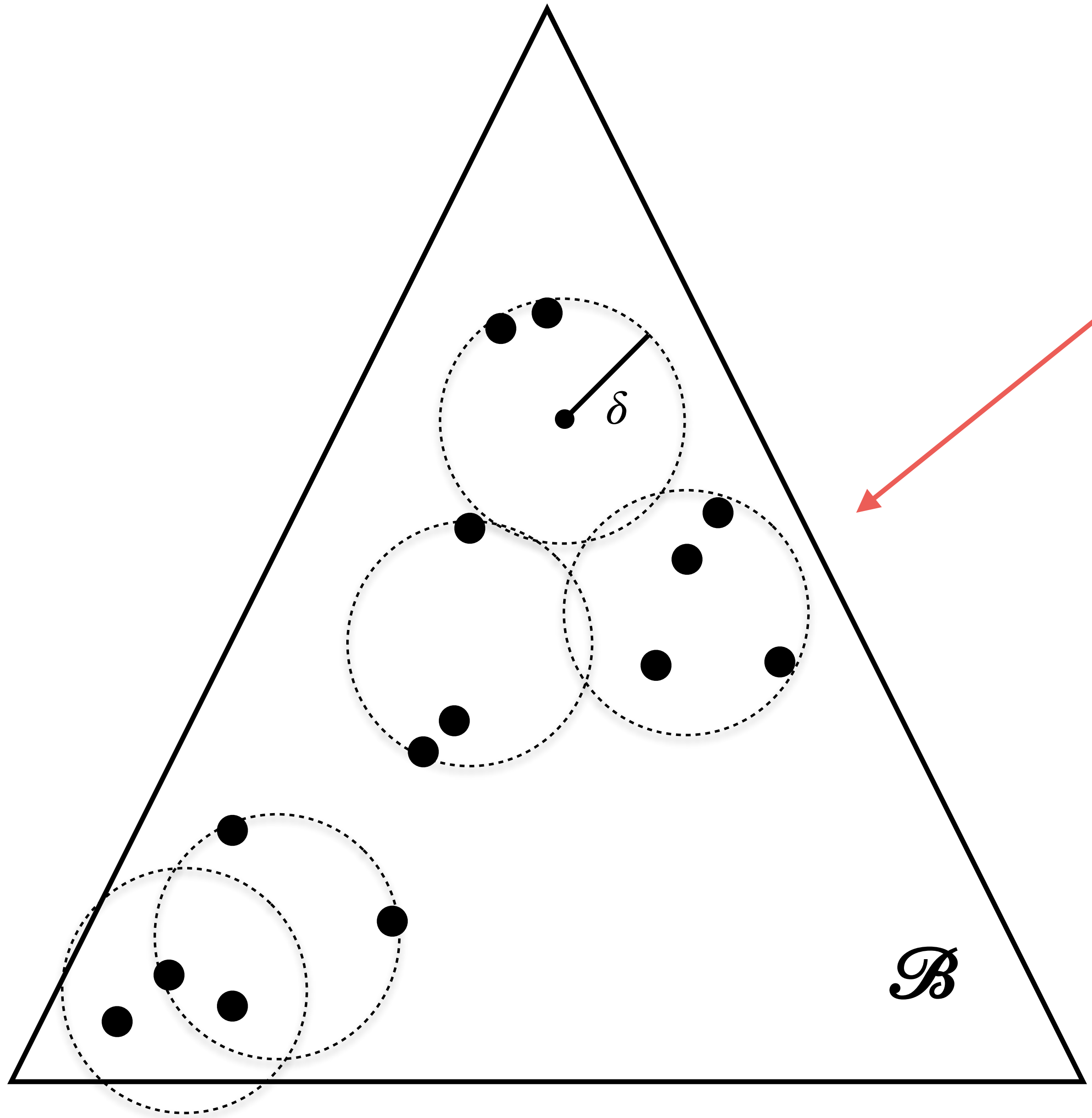
How hard is a POMDP?

- However, point-based methods work quite well!

How hard is a POMDP?

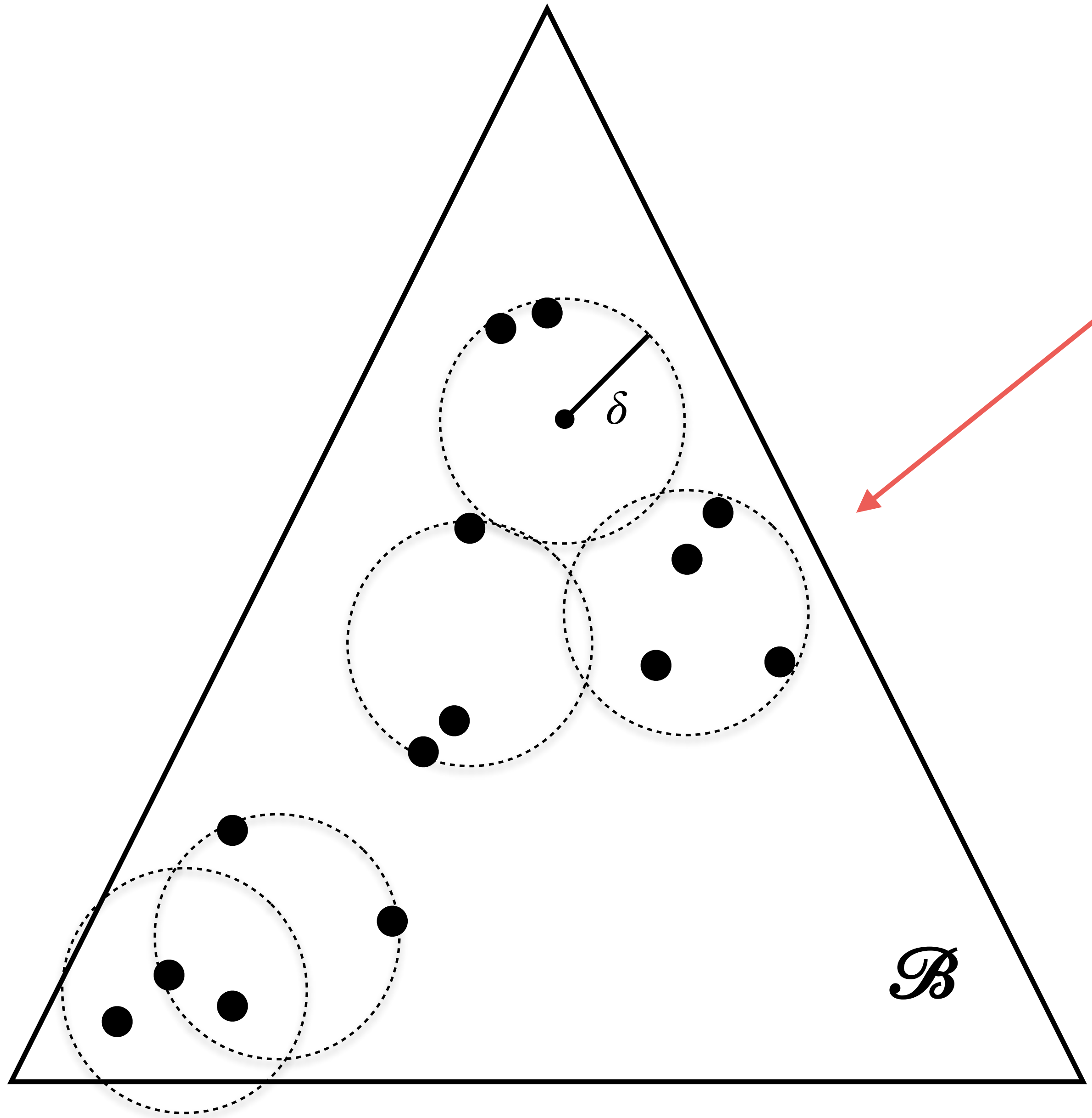


How hard is a POMDP?



Need 5 balls or radius δ to "cover" reachable points

How hard is a POMDP?



Covering number
C(δ) = 5

How hard is a POMDP?

- Complexity of POMDP planning better captured by the covering number of the reachable belief space
- Some point-based methods are built on such argument (they sample beliefs to cover reachable space)
  - Ex: SARSOP (Kurniawati et al., 2008)