WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

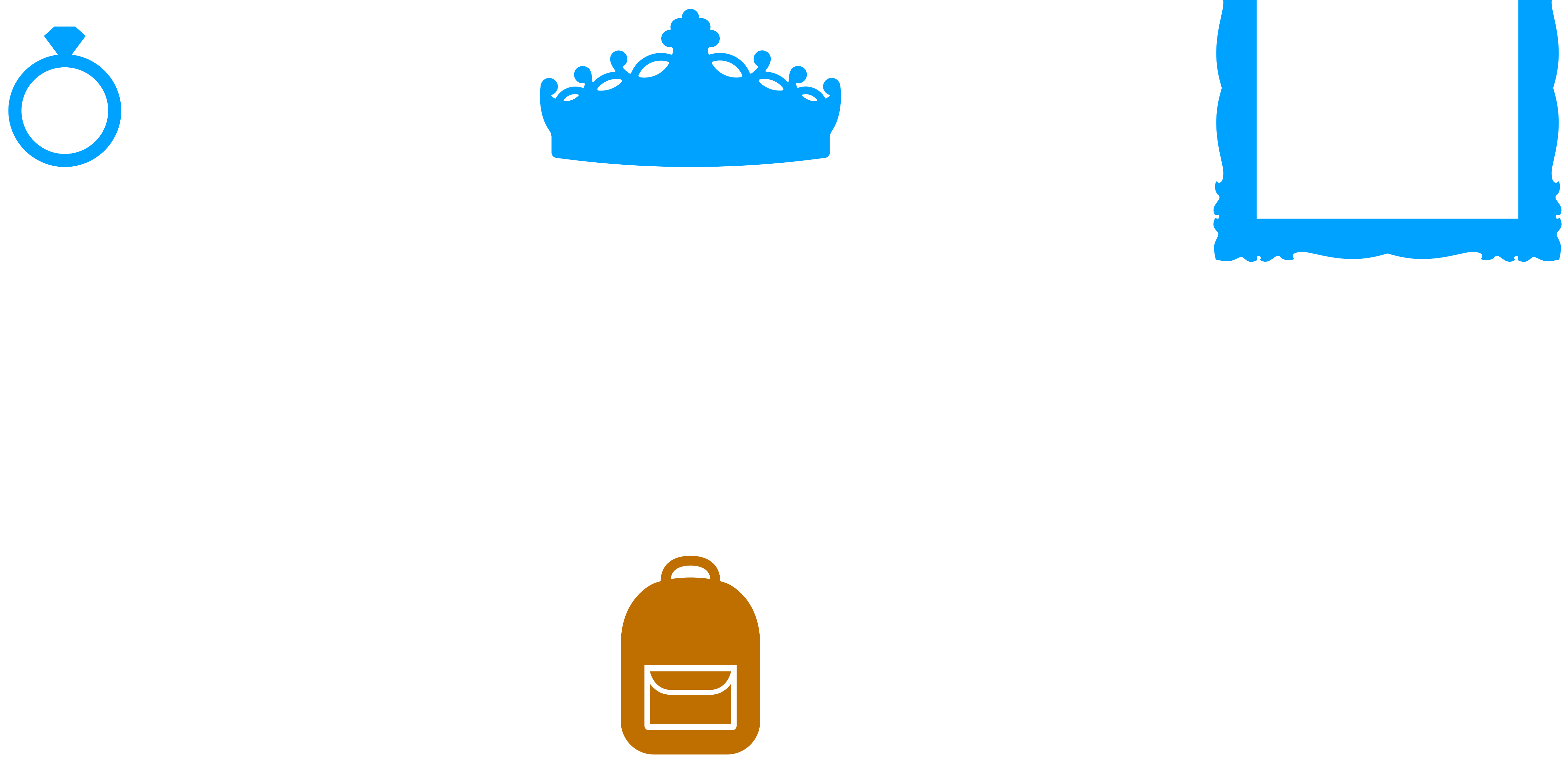
NEURAL INFORMATION PROCESSING SYSTEMS

From Knapsacks to Self-Driving: FPTAS Recipes for Constrained Reinforcement Learning

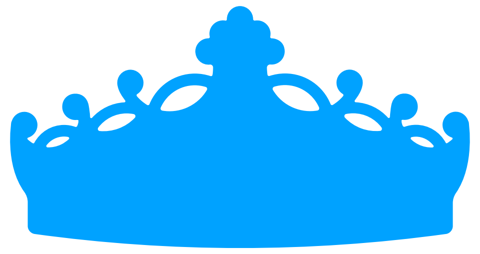
Jeremy McMahan

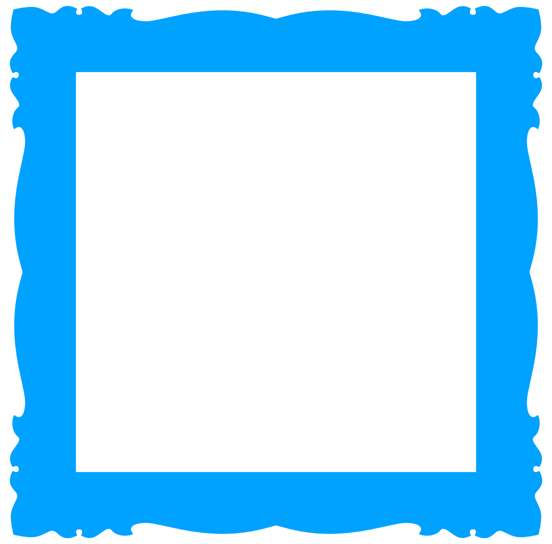
Smithsonian Bandits

Knapsack Problem

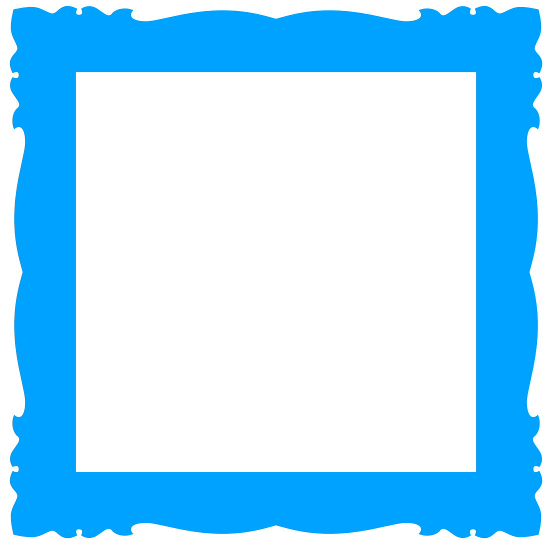


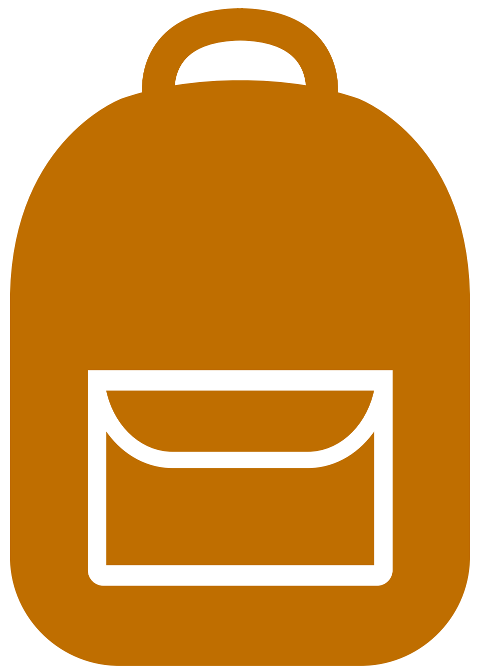
Knapsack Problem



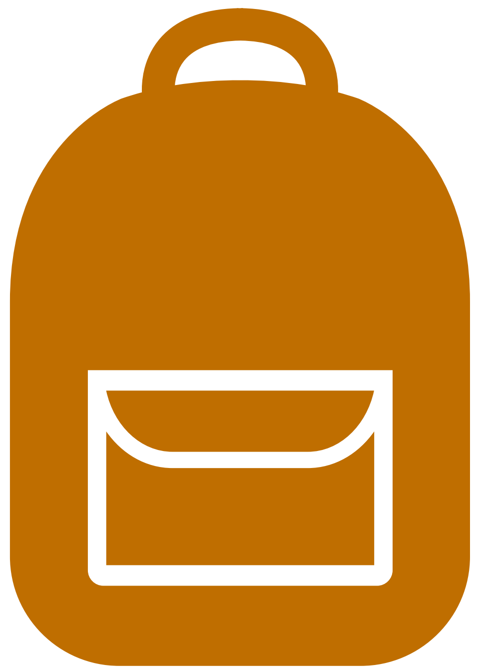


Knapsack Problem





Knapsack Problem



Optimization Formulation

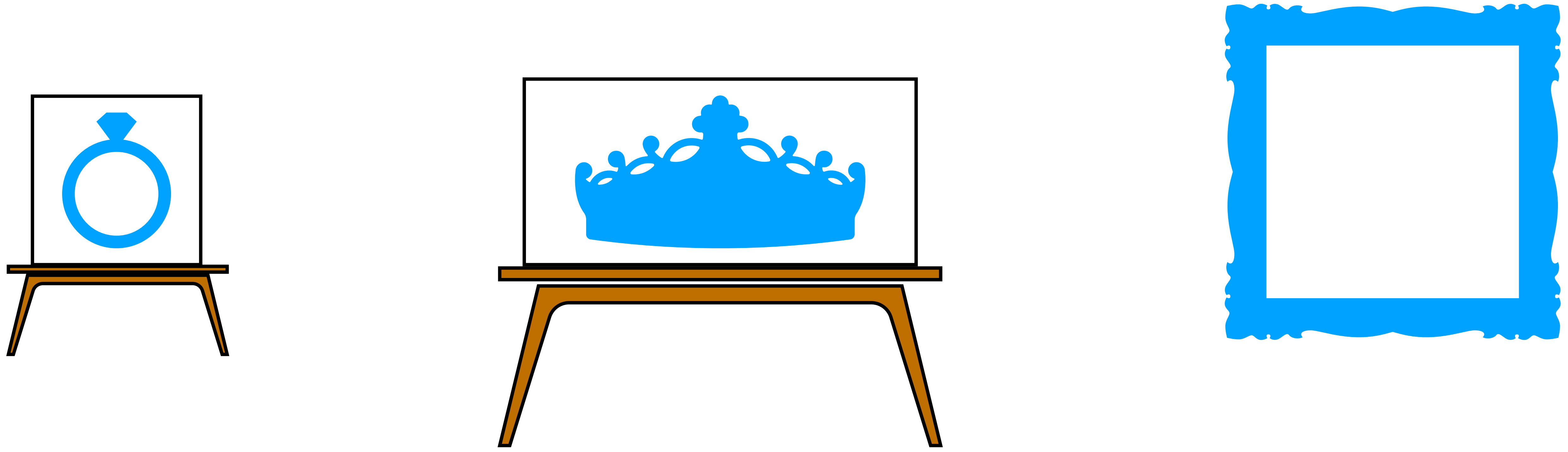
Optimization Formulation

$$\max_{x \in \{0,1\}^n} \quad \sum_{i=1}^{n} x_i v_i$$

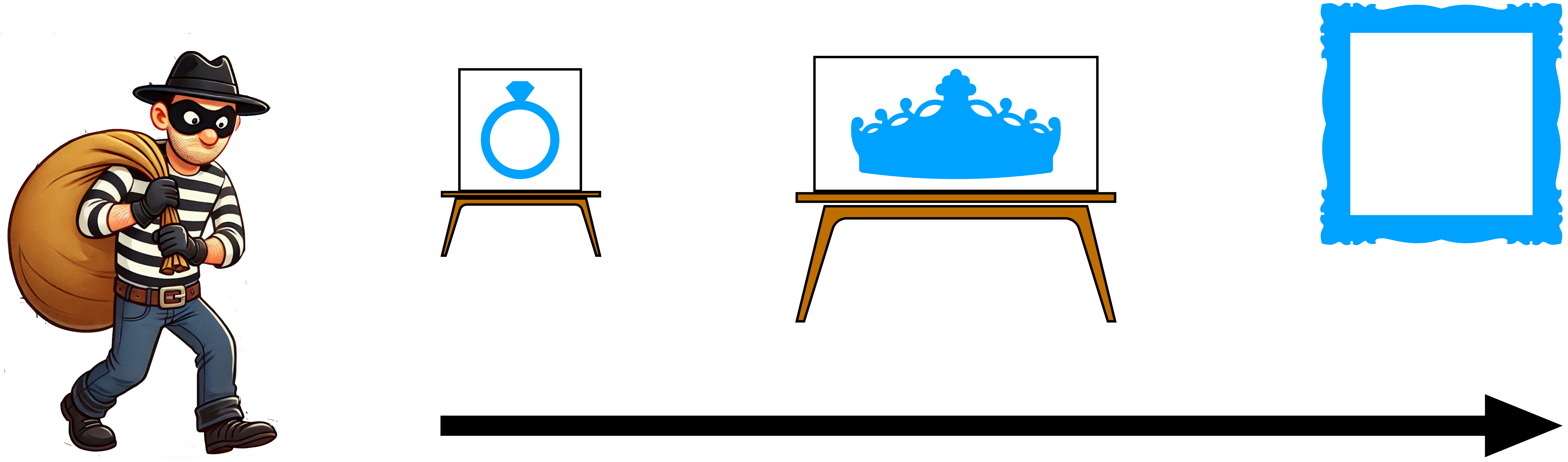
$$\text{s.t.} \quad \sum_{i=1}^{n} x_i w_i \leq B$$

Fixed Order

Fixed Order

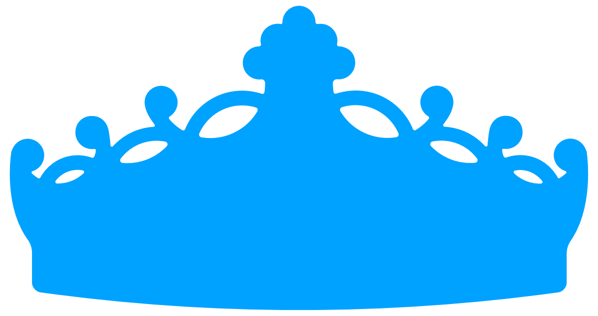


Fixed Order



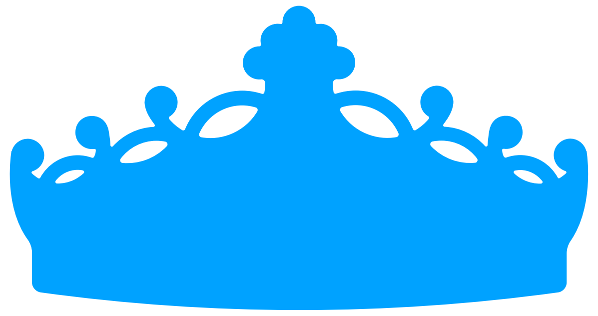
Stochastic Weights

Stochastic Weights



$$4lbs \pm 3lbs$$

Stochastic Weights



$4lbs \pm 3lbs$

Constraints

Constraints

Expectation: $$\mathbb{E}_w \left[\sum_{i=1}^{n} x_i w_i \right] \leq B$$

Constraints

Expectation: $\mathbb{E}_w\left[\sum_{i=1}^{n} x_i w_i\right] \leq B$

Chance: $\Pr_w\left[\sum_{i=1}^{n} x_i w_i \leq B\right] \geq 95\%$

Constraints

Expectation: $\mathbb{E}_w \left[\sum_{i=1}^{n} x_i w_i \right] \leq B$

Chance: $\Pr_w \left[\sum_{i=1}^{n} x_i w_i \leq B \right] \geq 95\%$

Almost Sure: $\Pr_w \left[\sum_{i=1}^{n} x_i w_i \leq B \right] = 1$

Adaptive Policies

Adaptive Policies

x can adapt to realized weights

Adaptive Policies

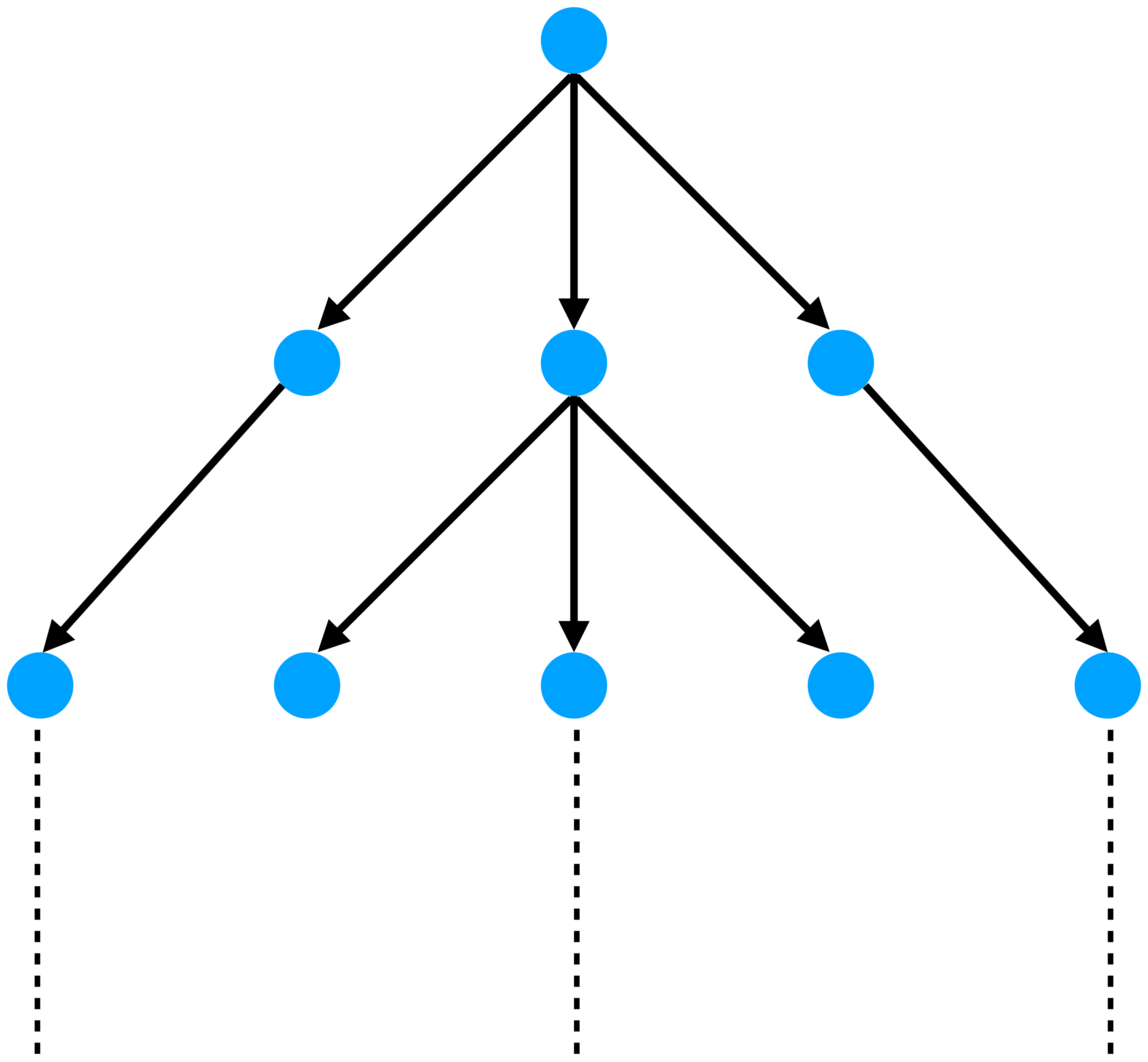
x can adapt to realized weights

$B = 15$

Adaptive Policies

x can adapt to realized weights

$B = 15$

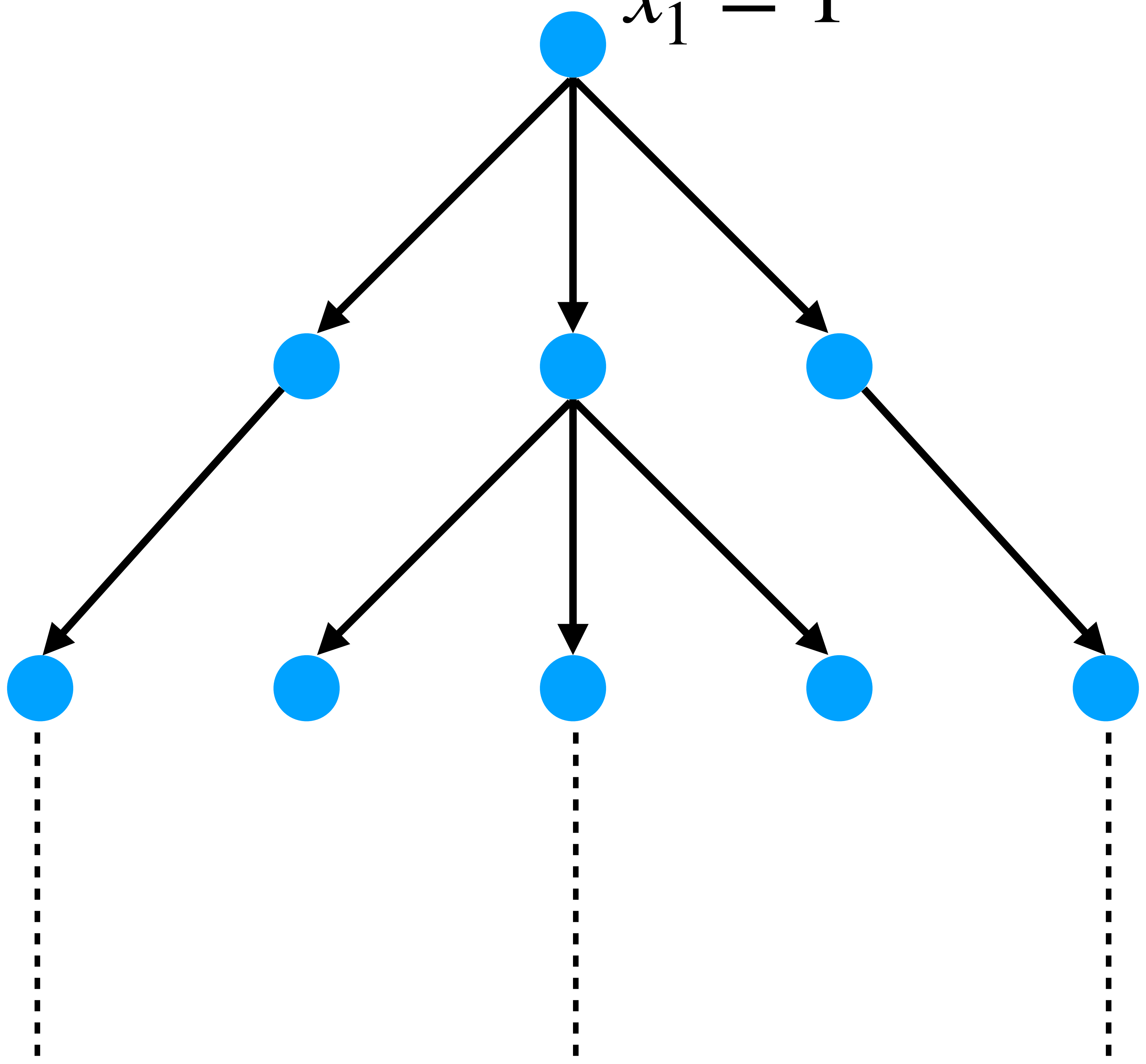


Adaptive Policies

x can adapt to realized weights

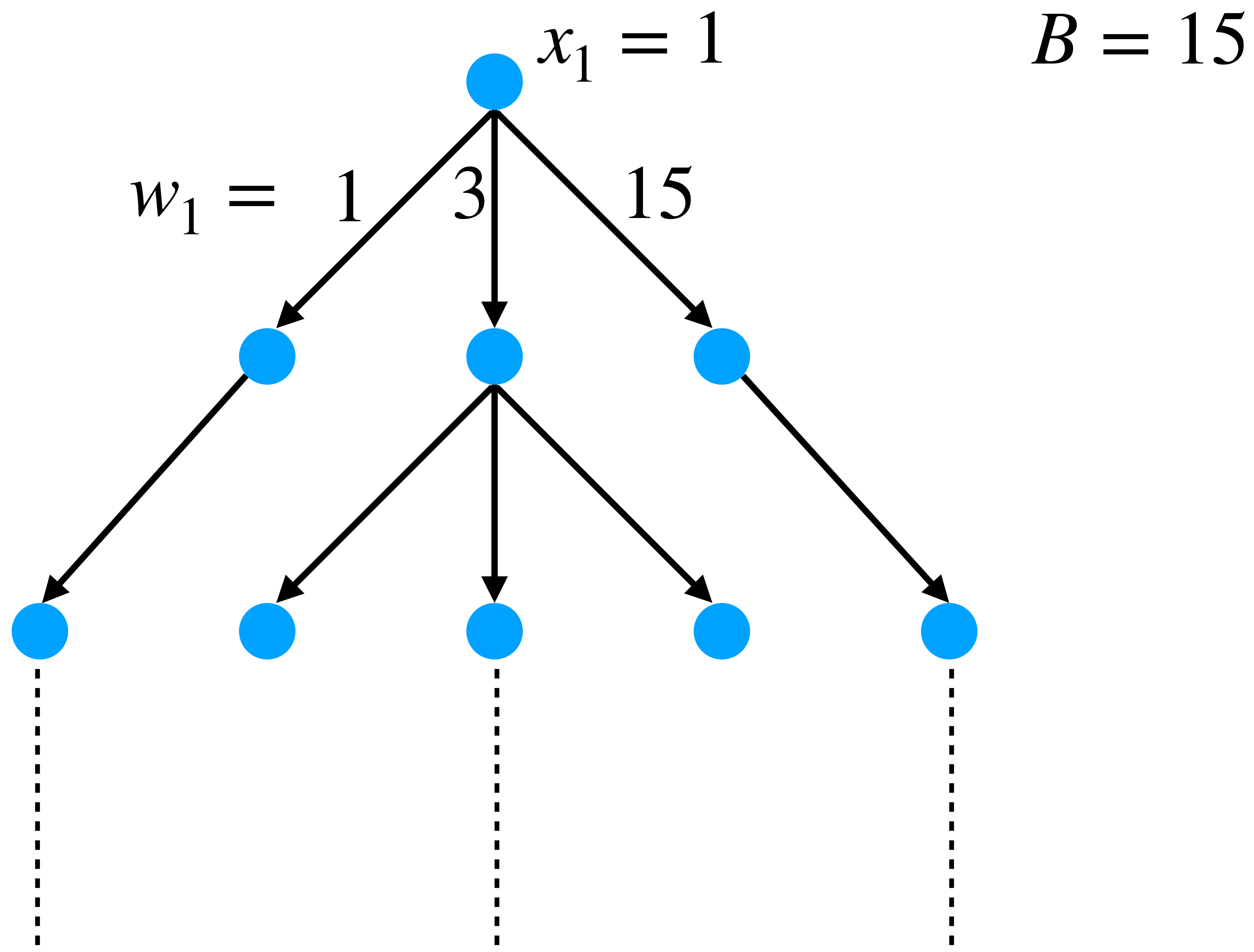
$x_1 = 1$

$B = 15$



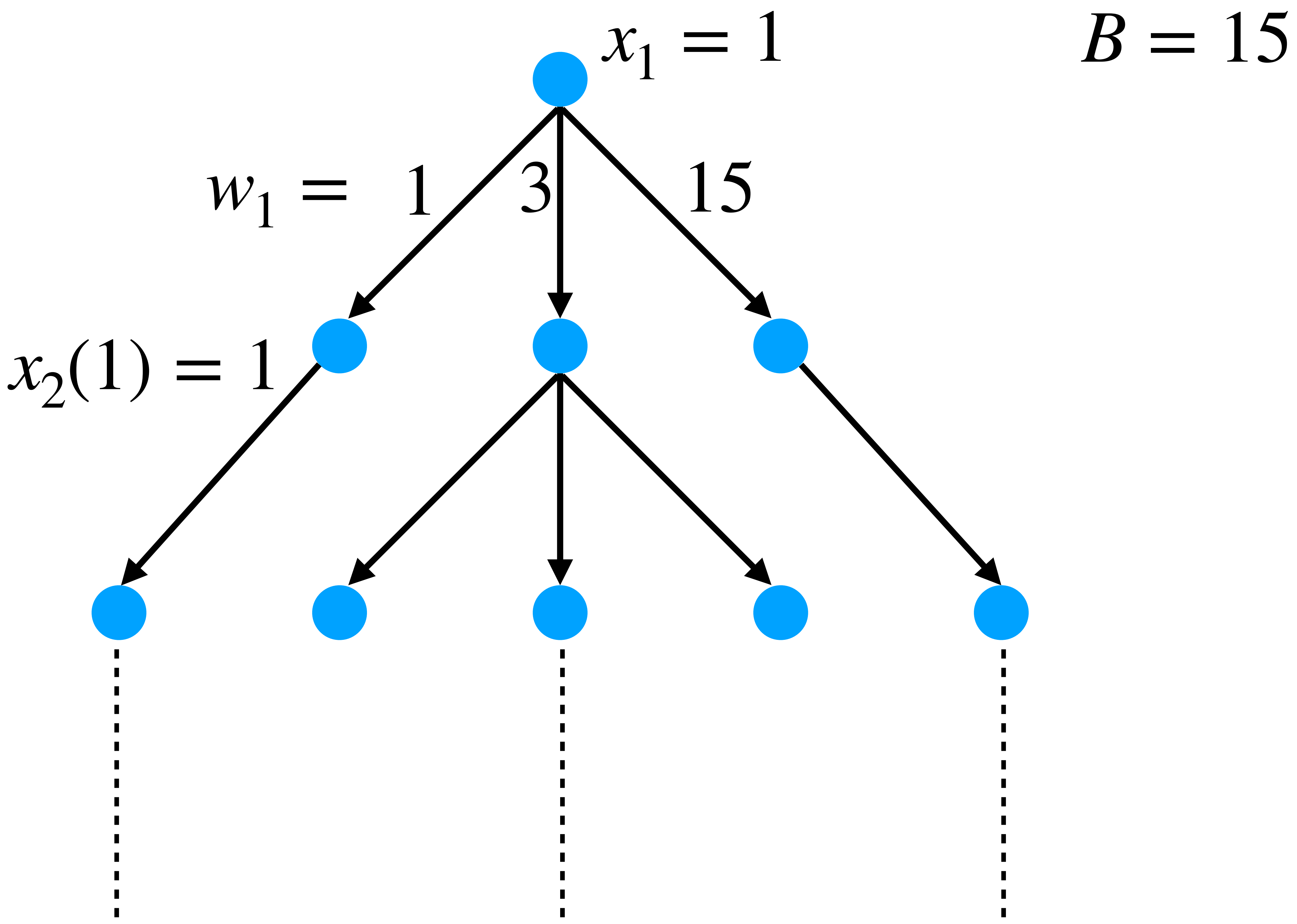
Adaptive Policies

x can adapt to realized weights



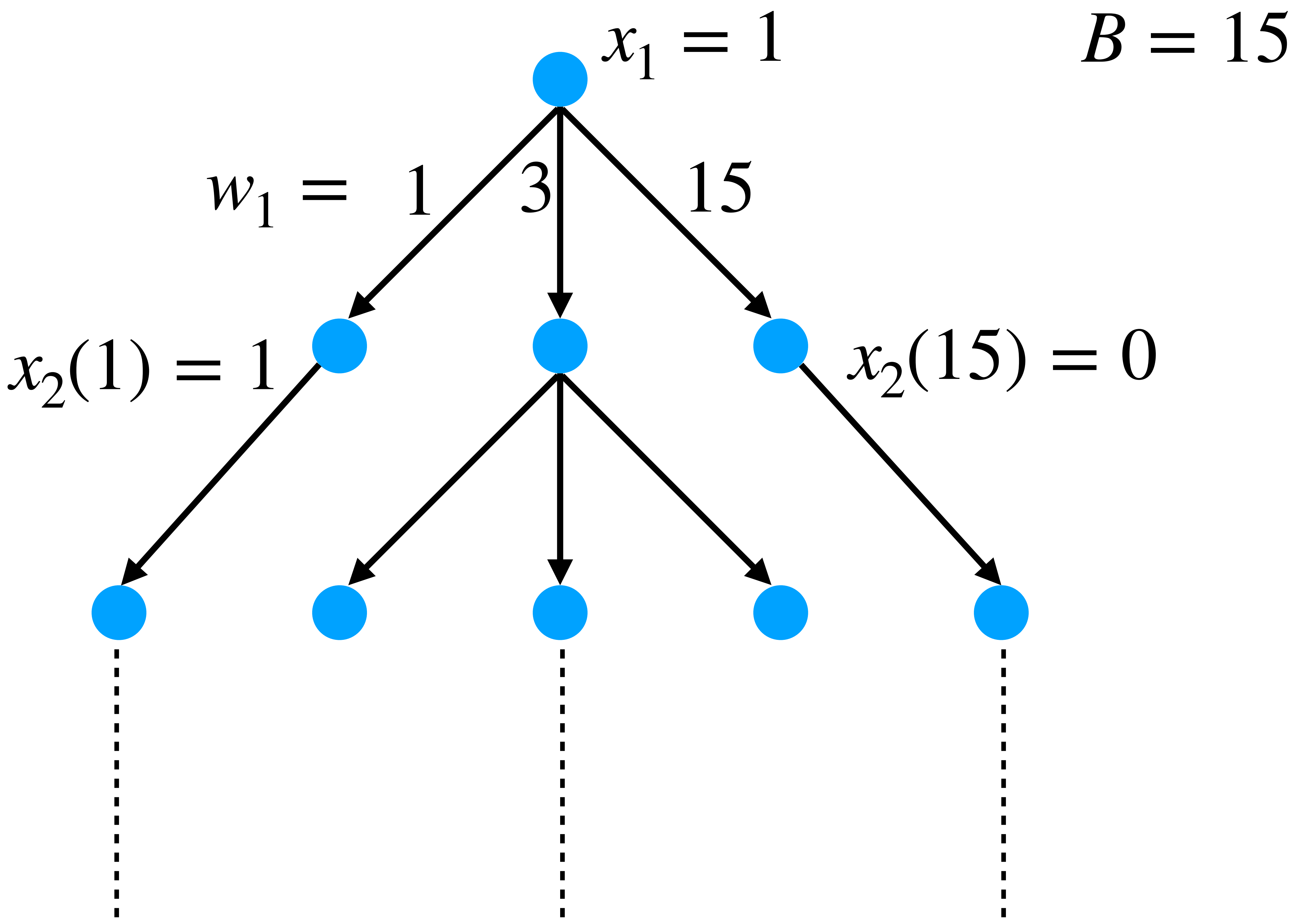
Adaptive Policies

x can adapt to realized weights



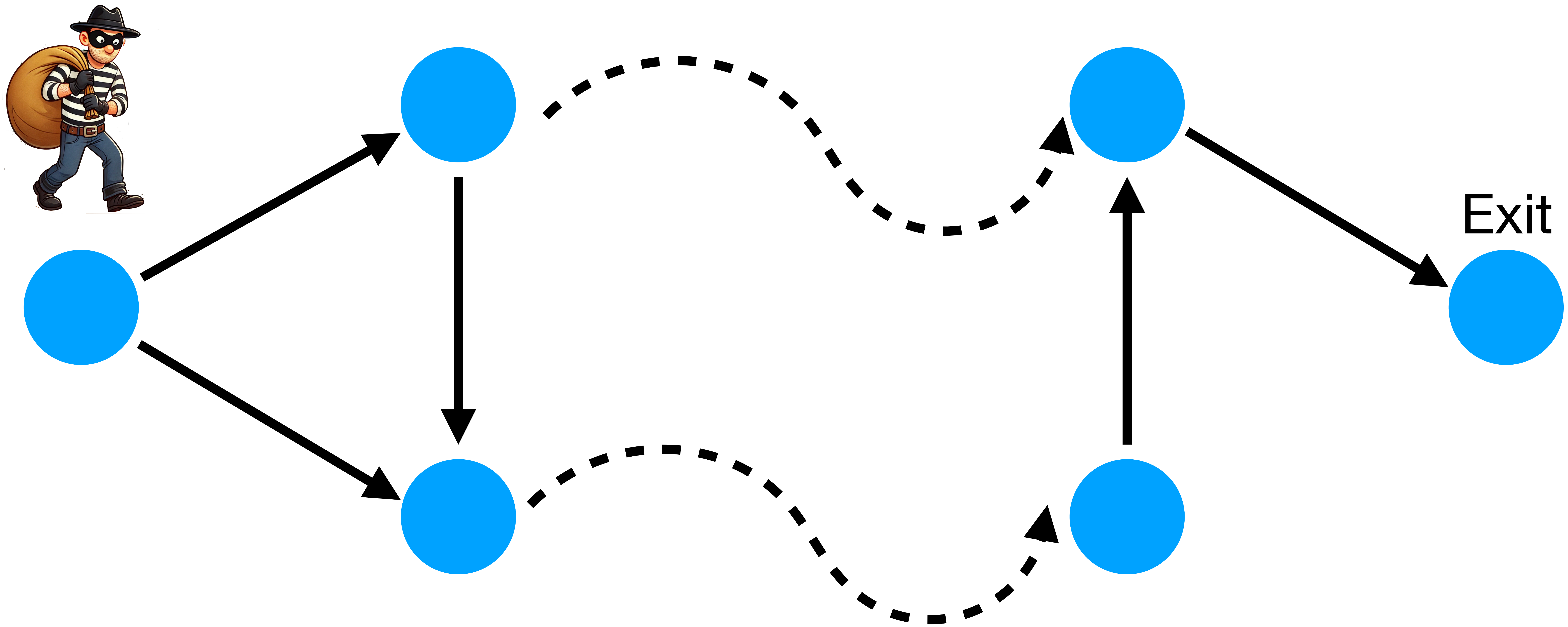
Adaptive Policies

x can adapt to realized weights



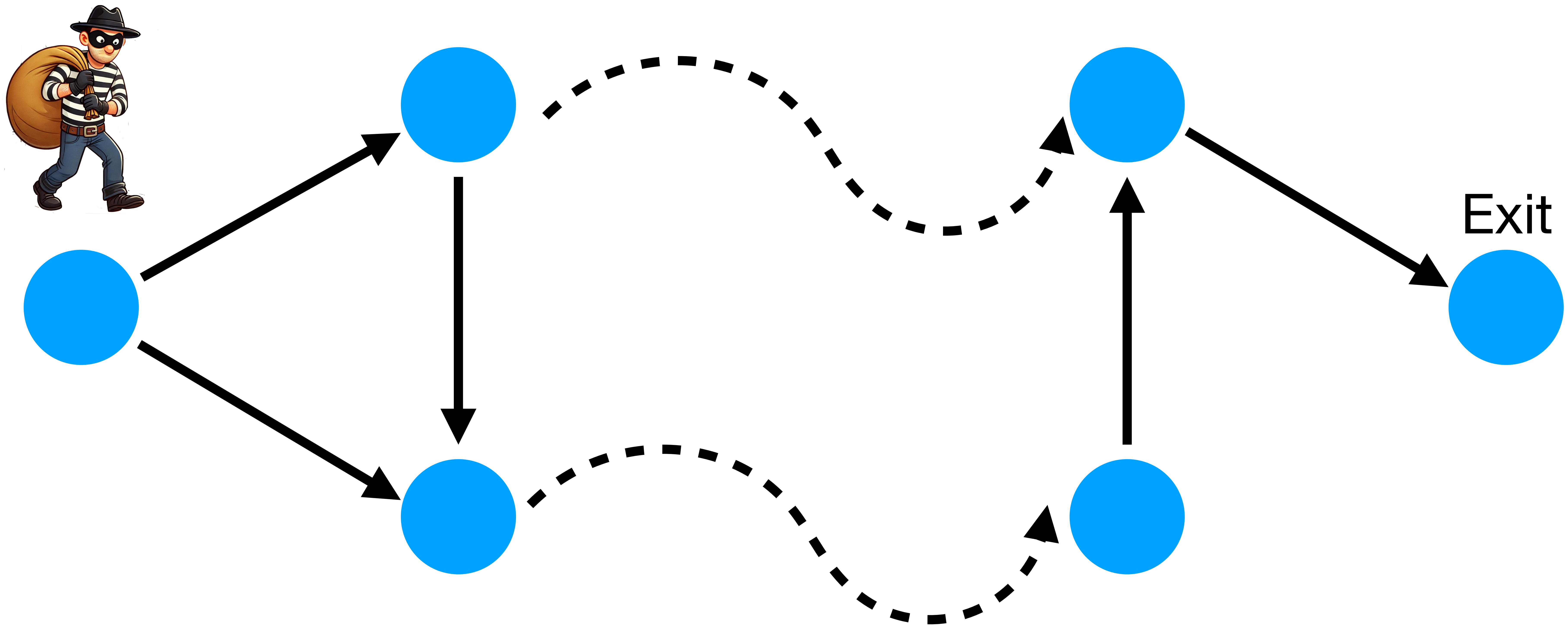
Context Dependence

Context Dependence



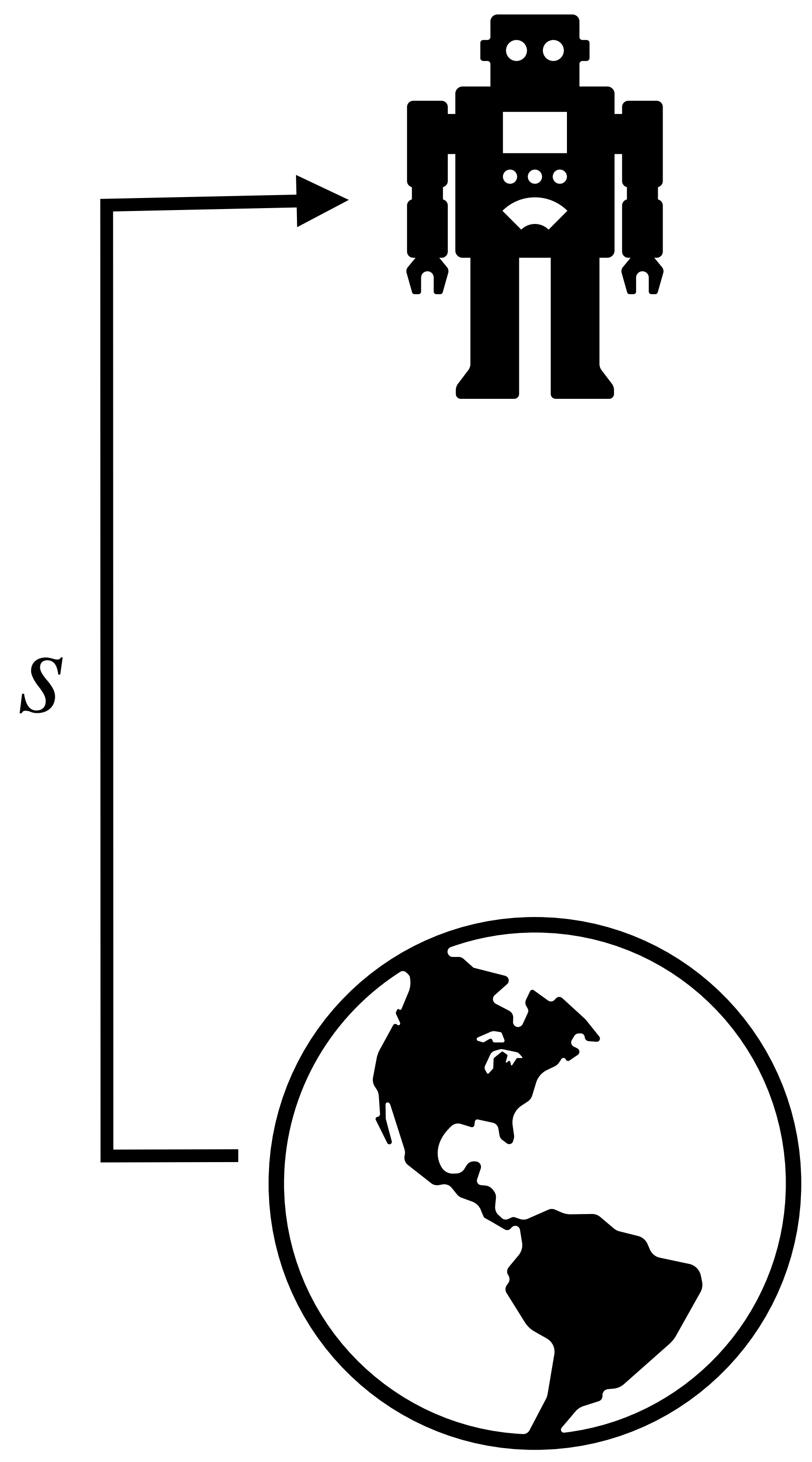
Context Dependence

$x : \text{context} \rightarrow \{0,1\}$

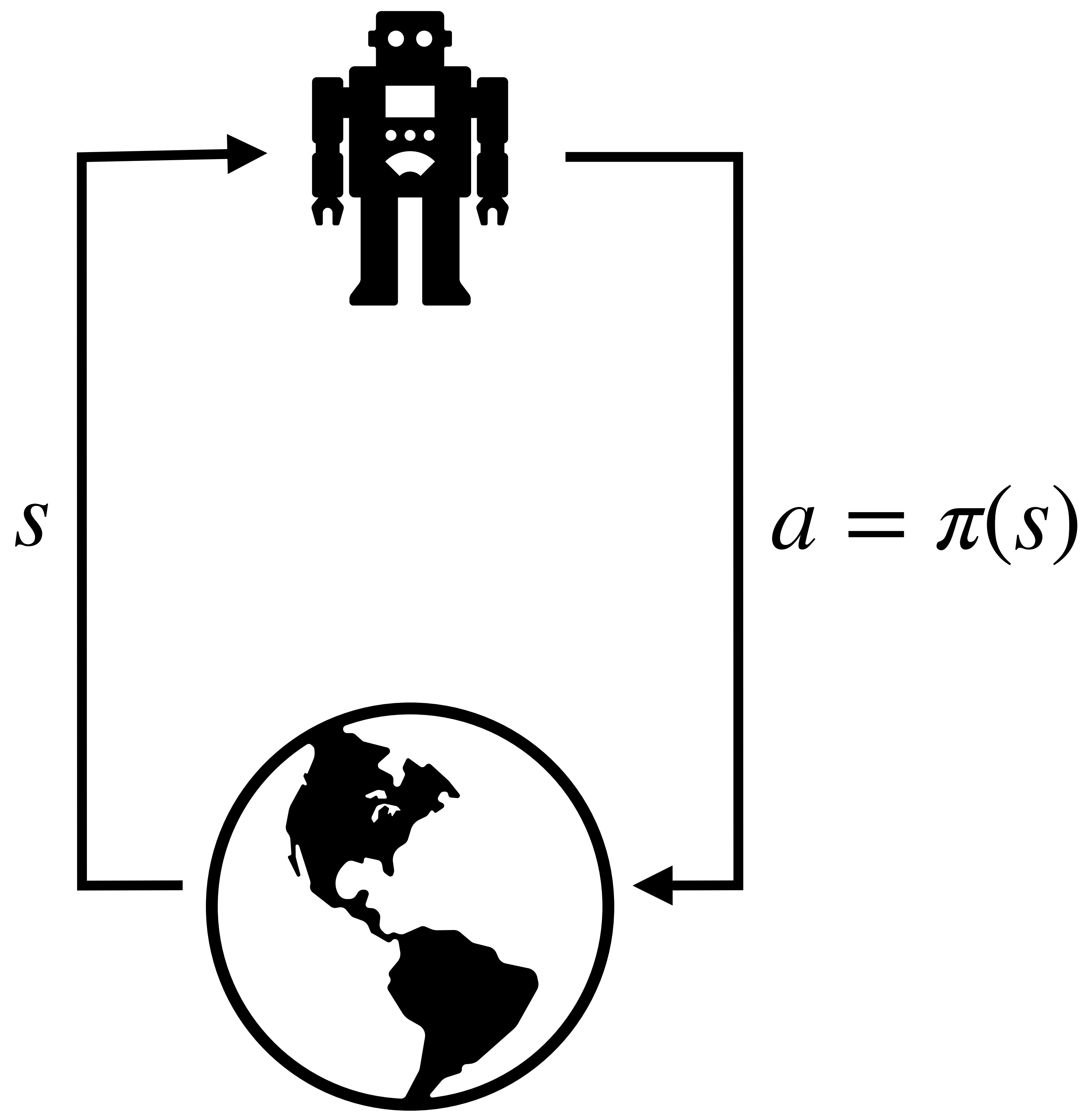


Constrained MDPs

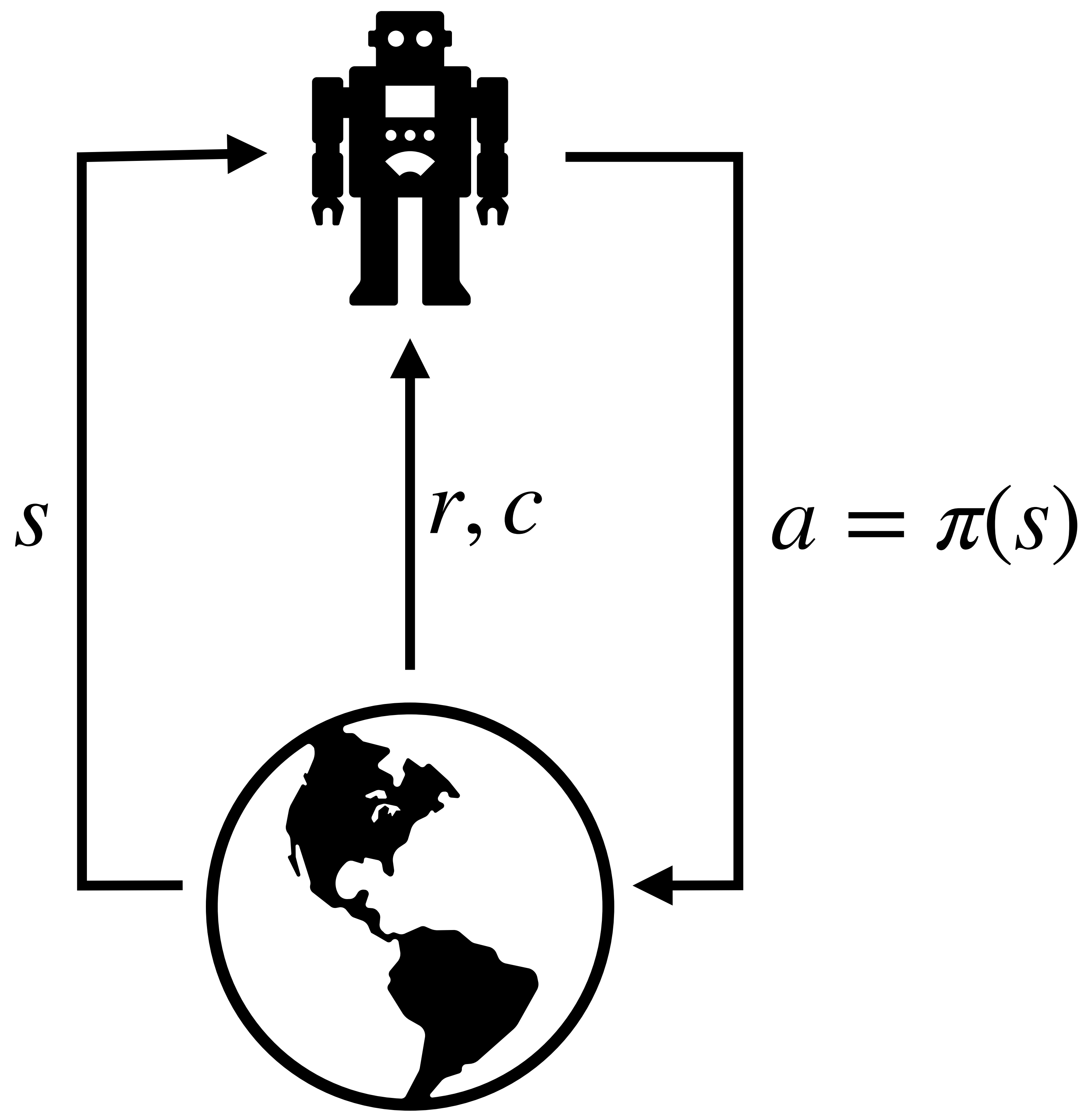
Constrained MDPs



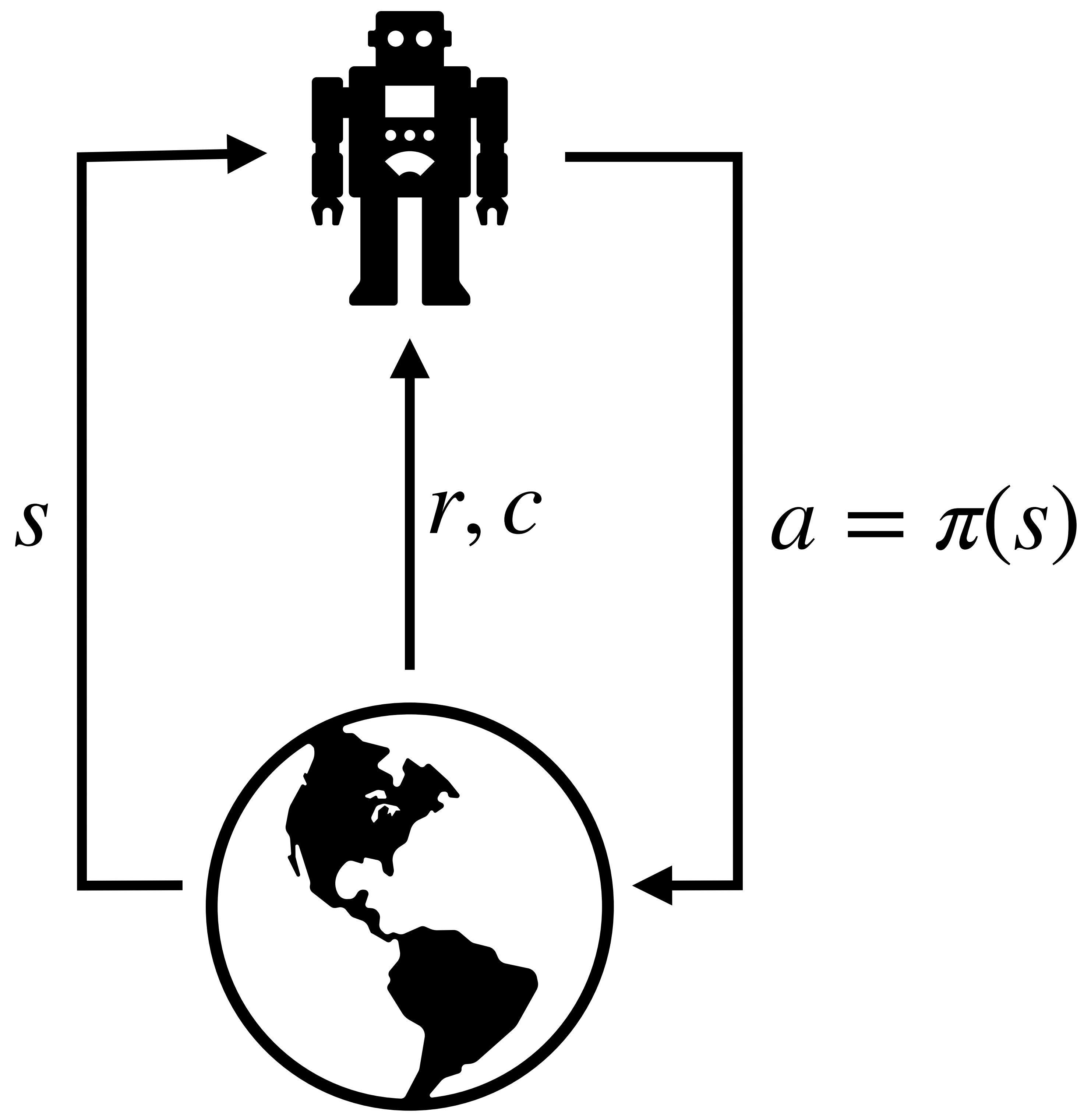
Constrained MDPs



Constrained MDPs

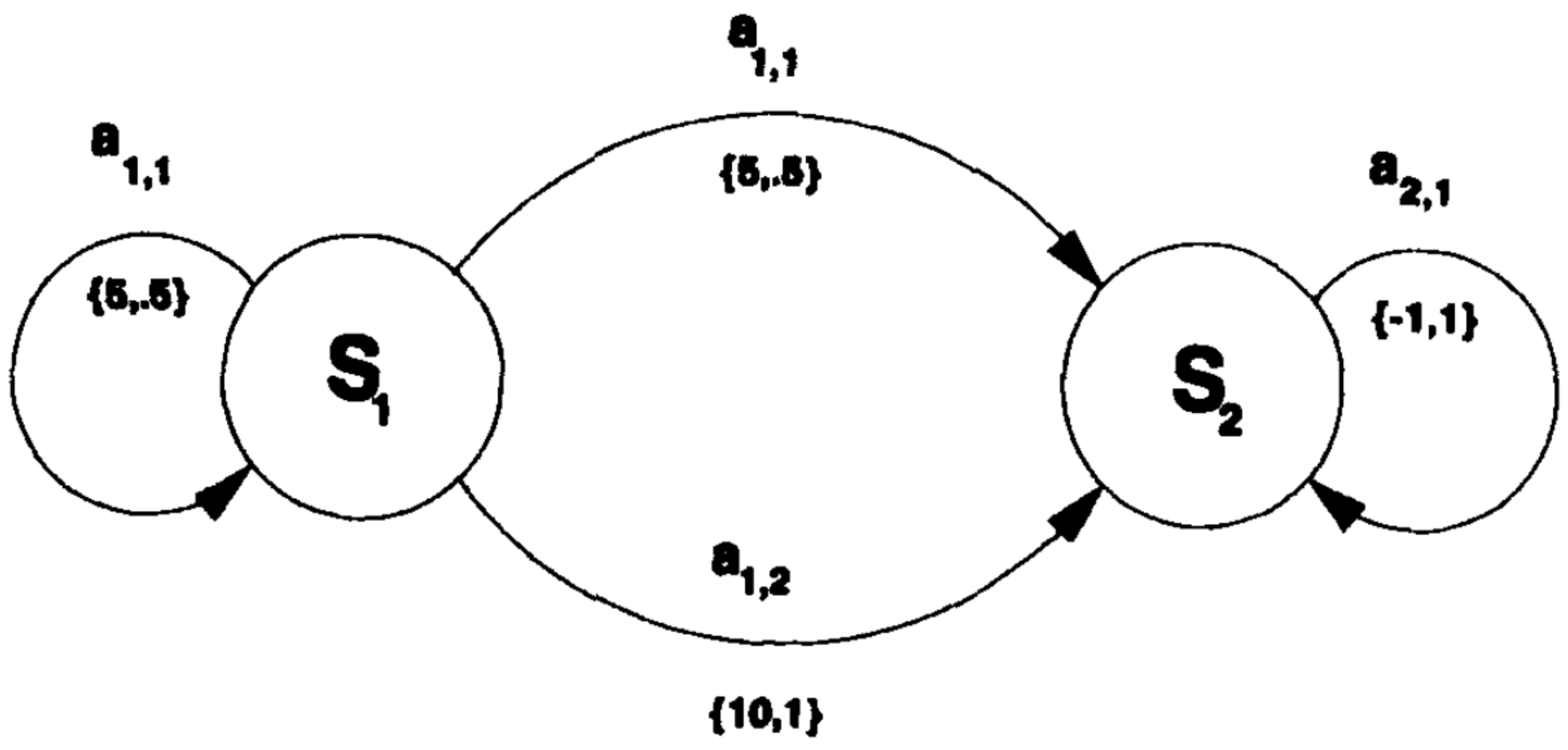


Constrained MDPs



Repeated H times

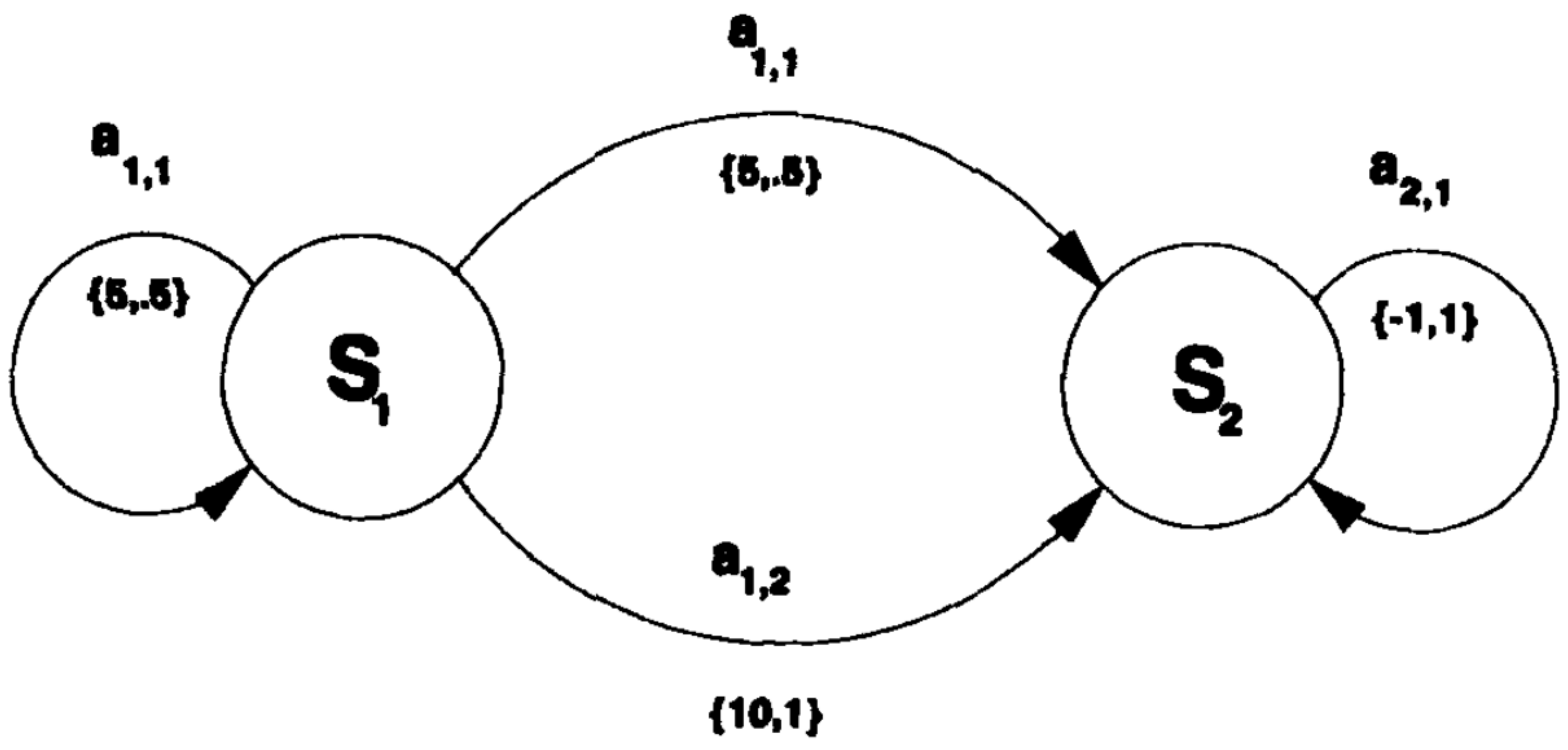
Formalism



H = 3

Formalism

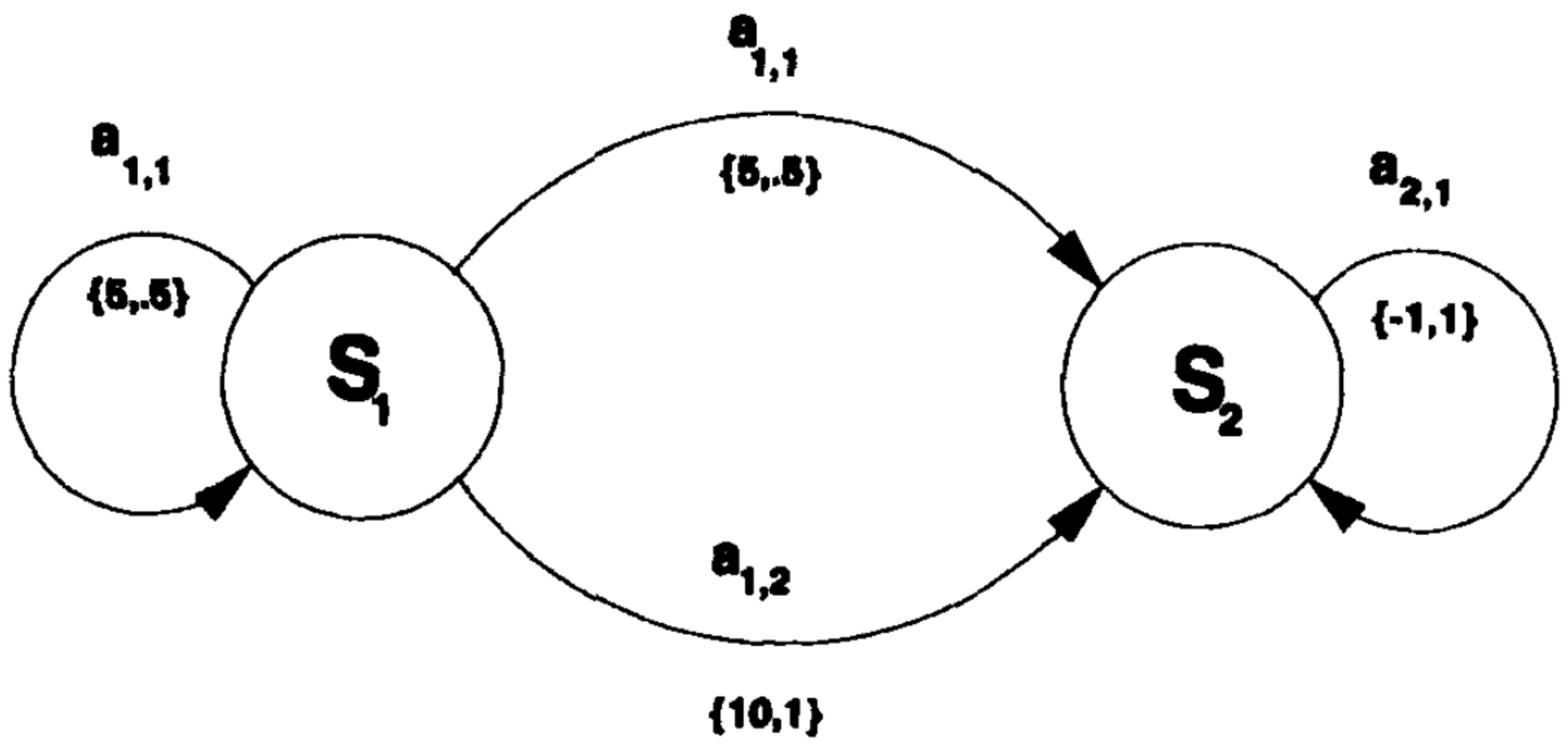
- States: S



H = 3

Formalism

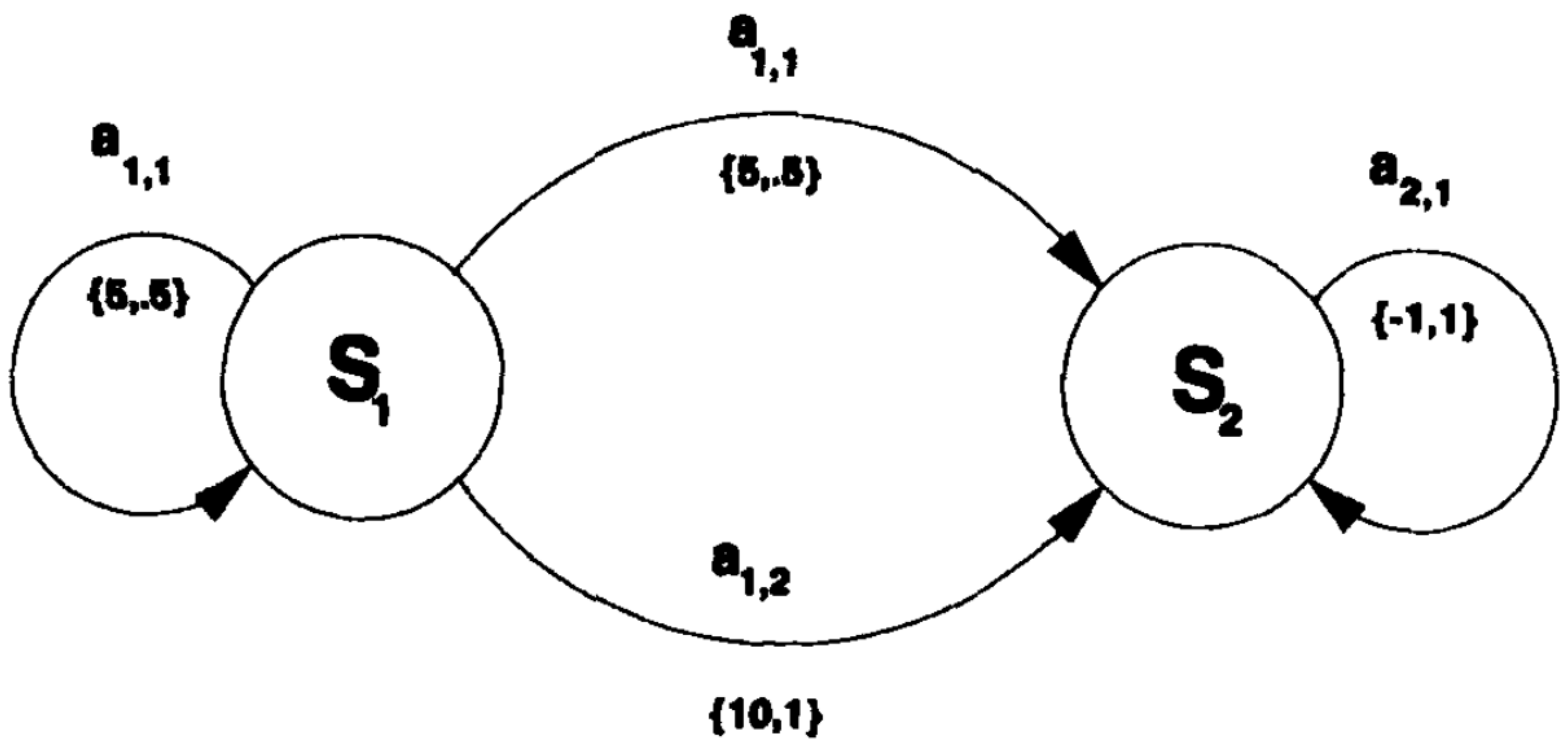
- States: S
- Actions: A



H = 3

Formalism

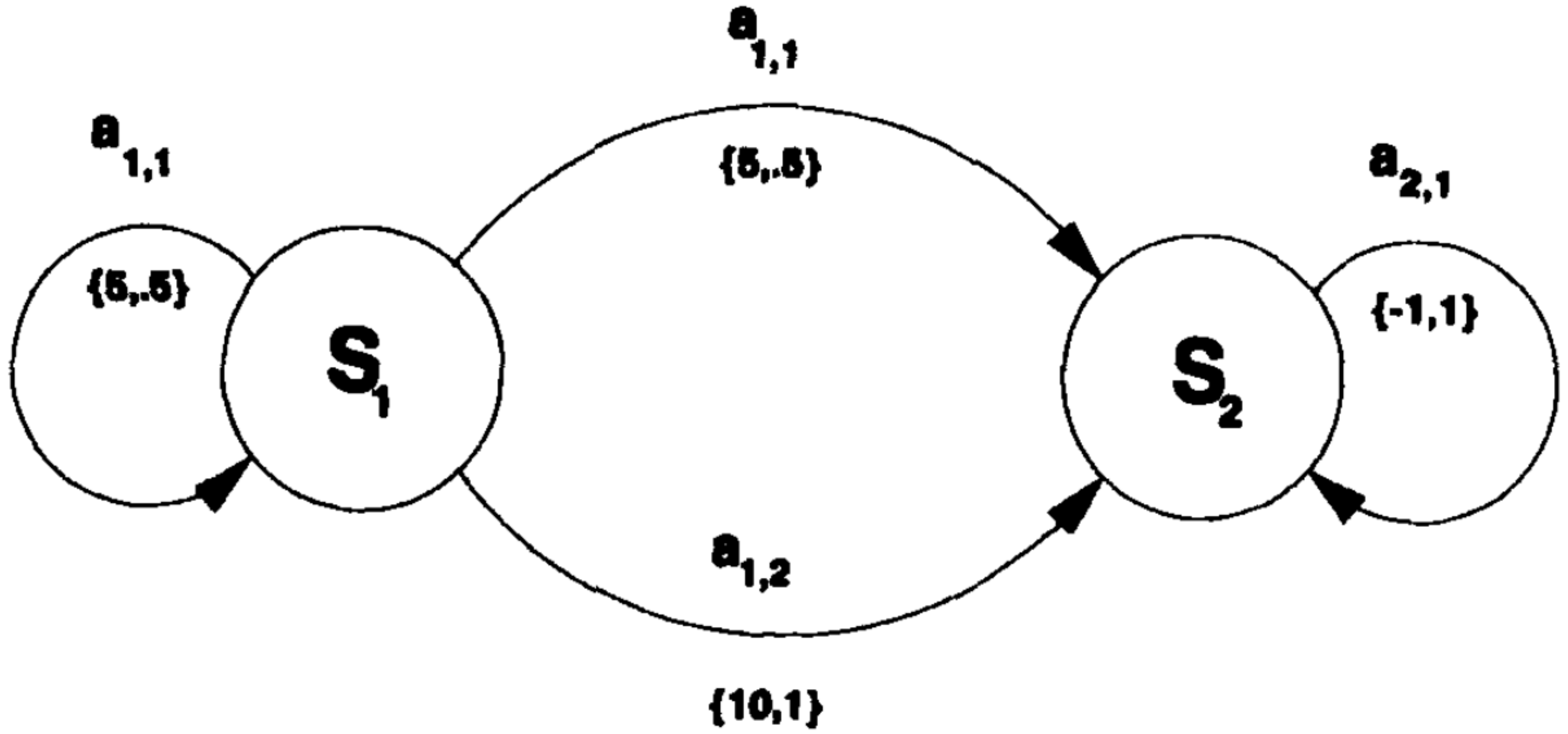
- States: S
- Actions: A
- Rewards: $r_h(s, a)$



H = 3

Formalism

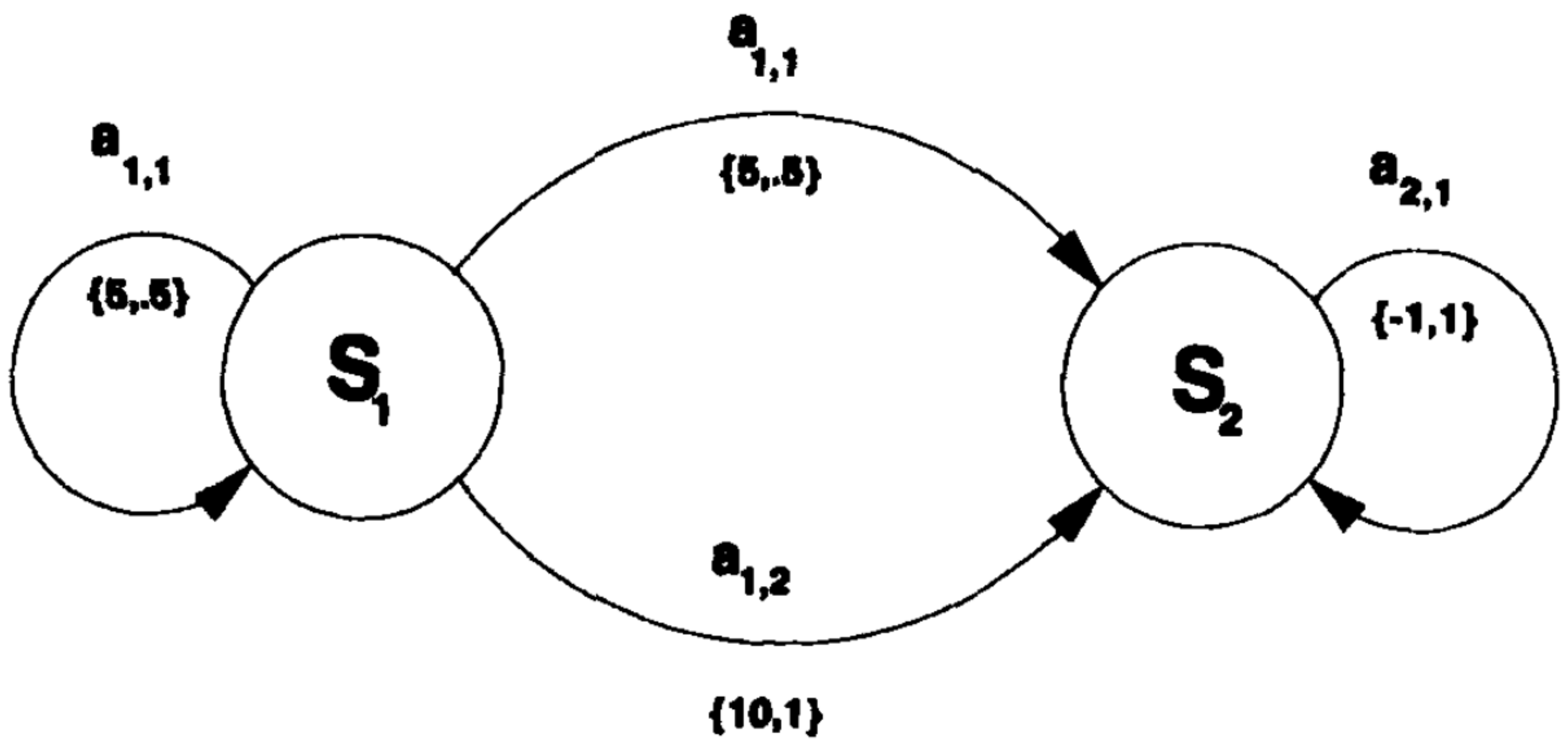
- States: S
- Actions: A
- Rewards: $r_h(s, a)$
- Costs: $c_h(s, a)$



H = 3

Formalism

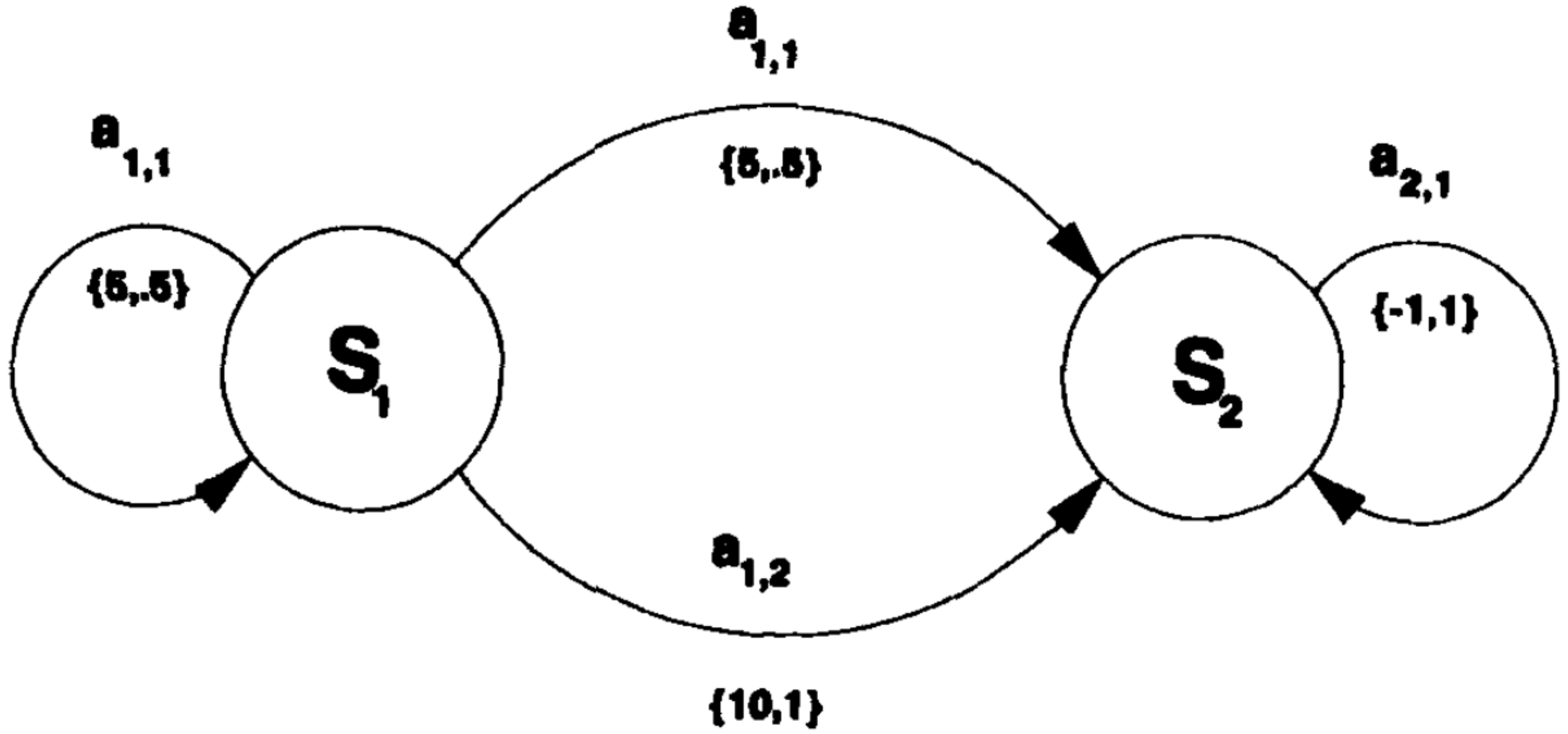
- States: S
- Actions: A
- Rewards: $r_h(s, a)$
- Costs: $c_h(s, a)$
- Transition Probabilities: $P_h(s' \mid s, a)$



H = 3

Formalism

- States: S
- Actions: A
- Rewards: $r_h(s, a)$
- Costs: $c_h(s, a)$
- Transition Probabilities: $P_h(s' \mid s, a)$
- Time Horizon: H



H = 3

Policies

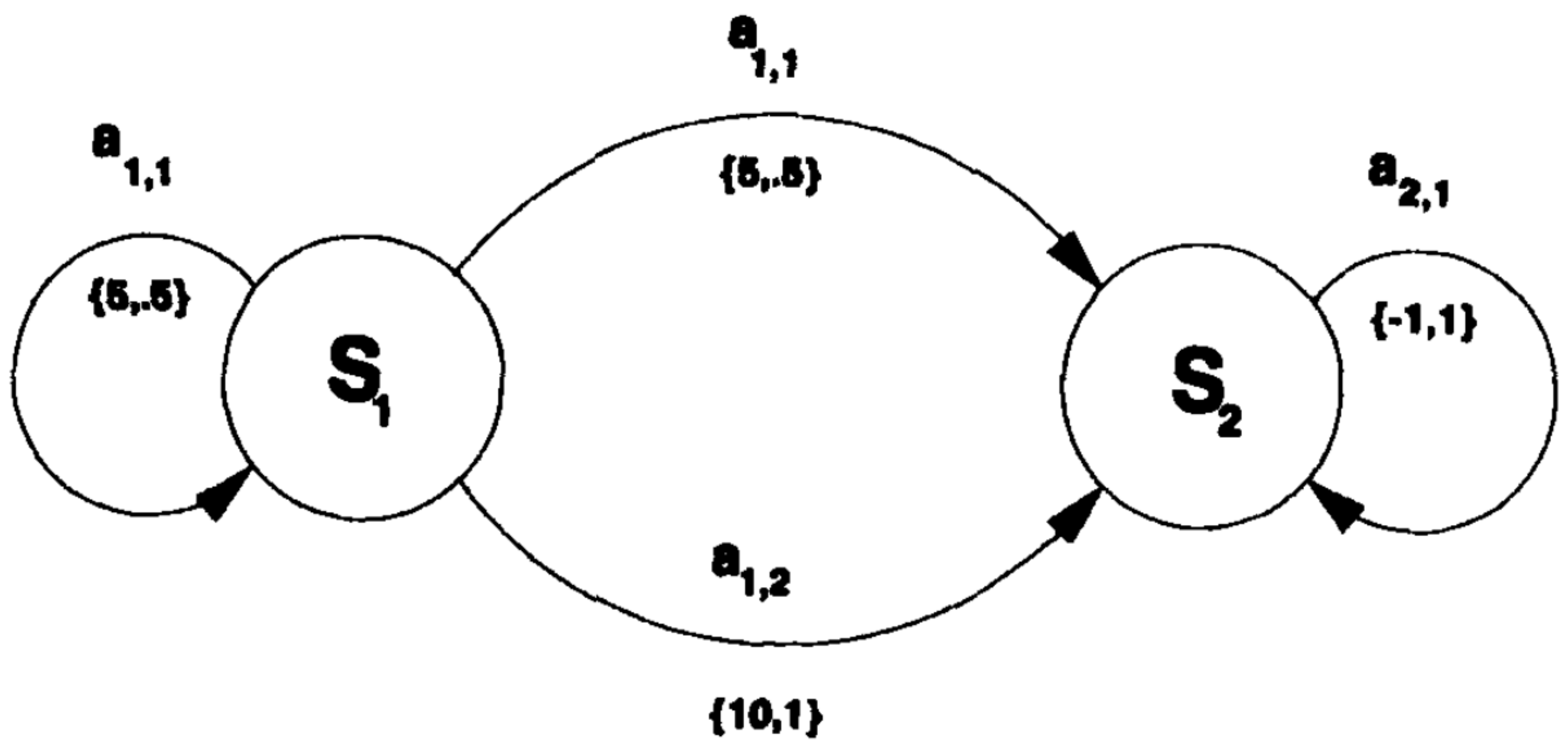
Policies

A **policy** is a plan of what action to take (usually) in each state.

Policies

A **policy** is a plan of what action to take (usually) in each state.

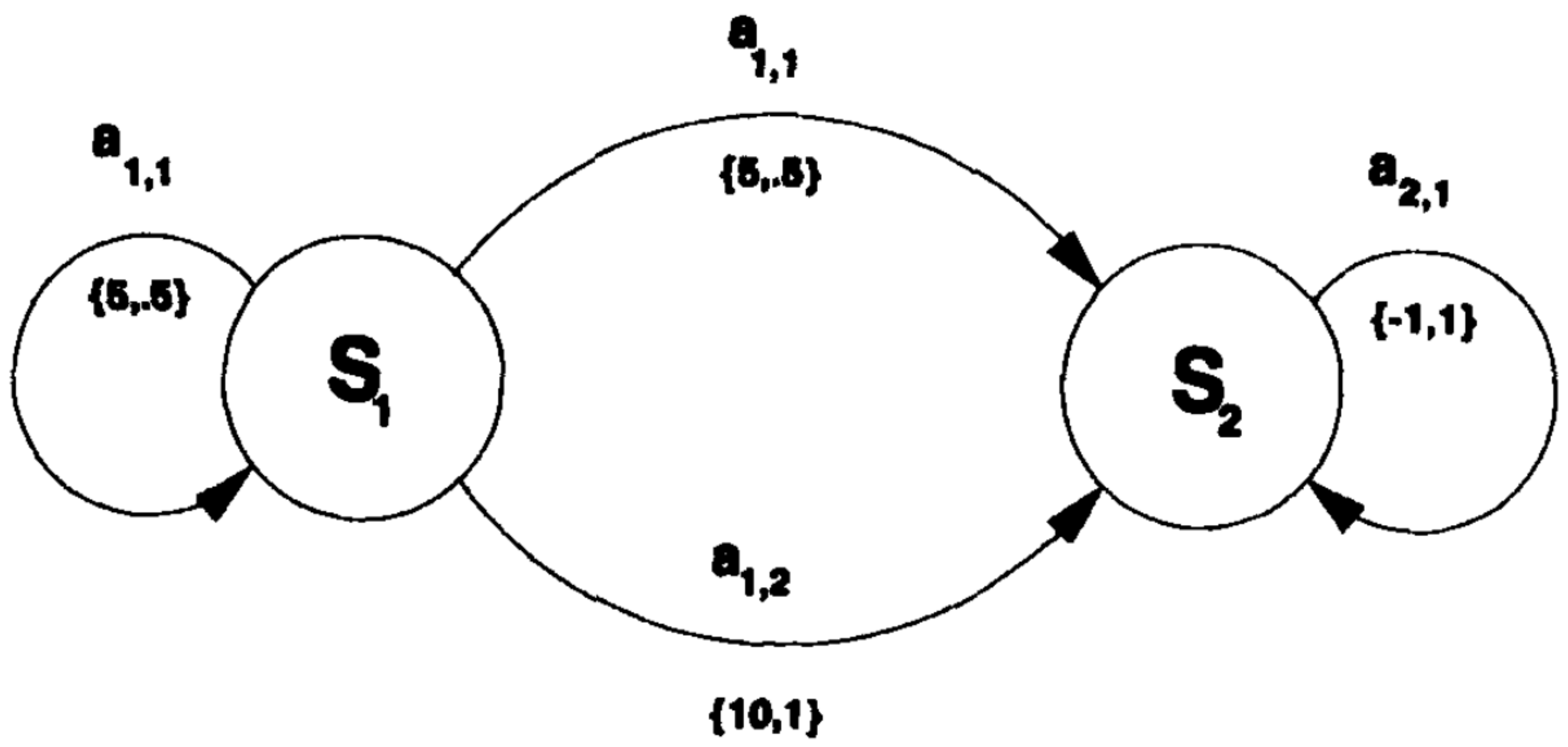
$$\pi(s_1) = a_{1,2} \quad \pi(s_2) = a_{2,1}$$



Policies

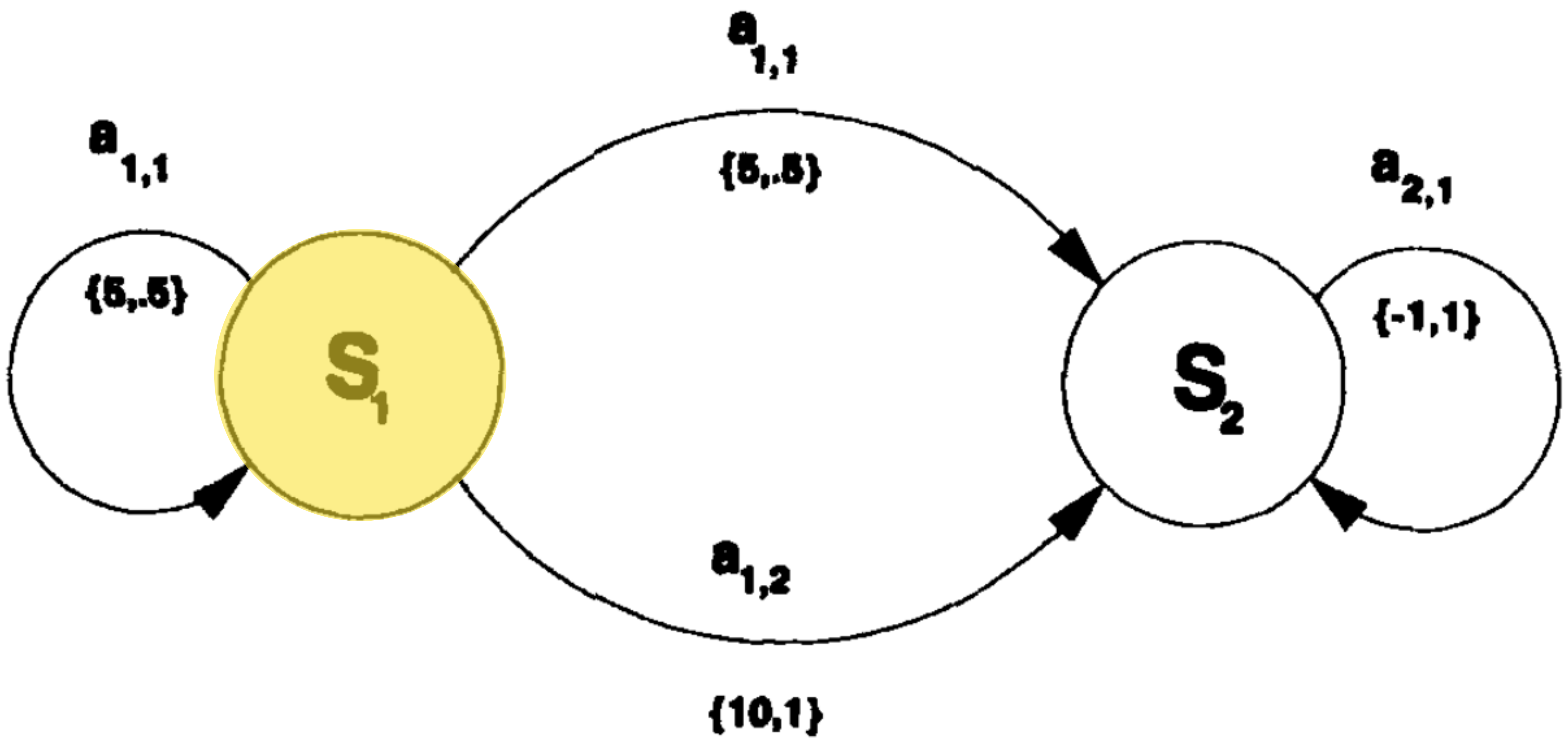
A **policy** is a plan of what action to take (usually) in each state.

$$\pi(s_1) = a_{1,2} \quad \pi(s_2) = a_{2,1}$$



$$V^{\pi}(s) = E_{\pi}\left[\sum_{h=1}^{H} r_h(s,a) \mid s_0 = s\right]$$

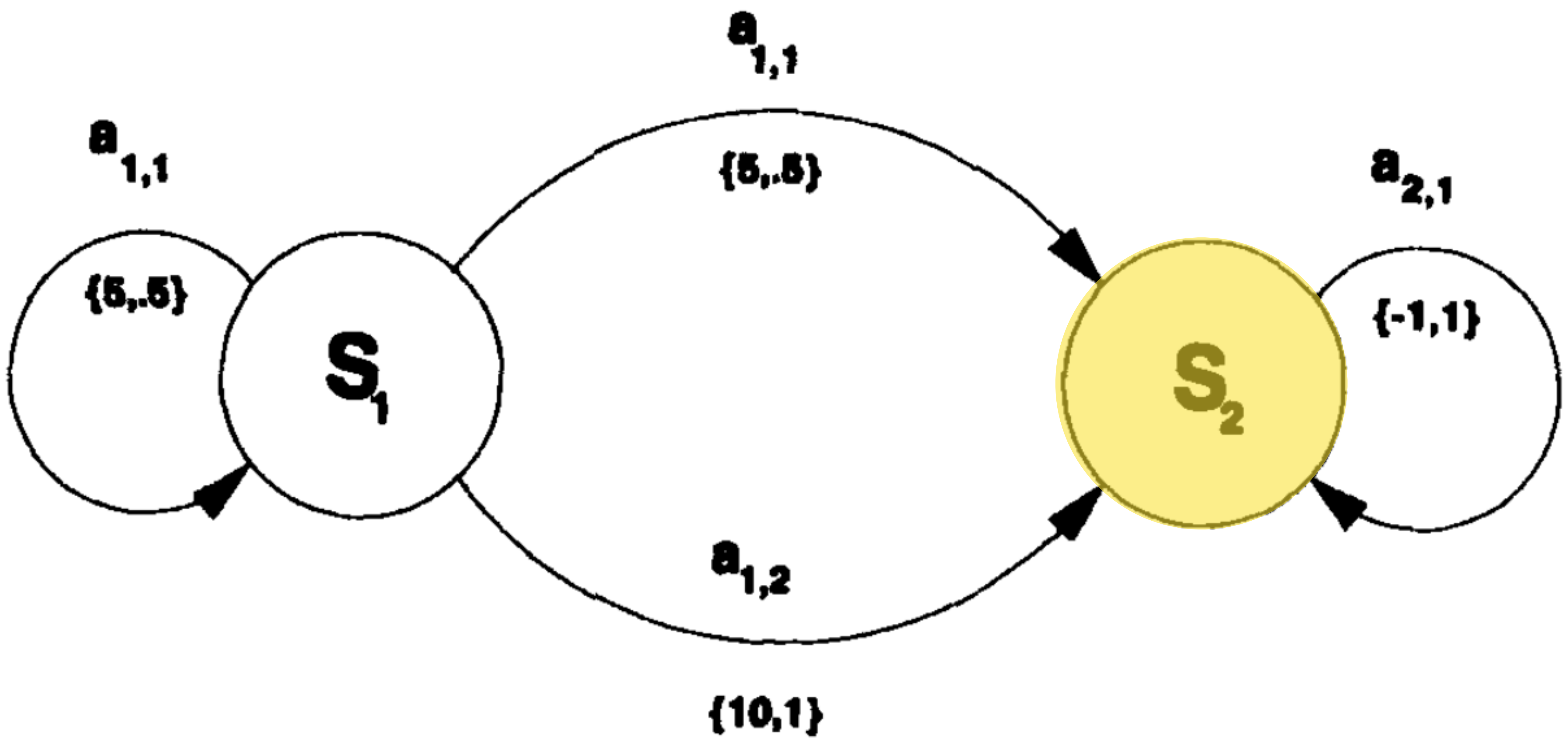
Value



$\pi(s_1) = a_{1,2}$

Reward = 10

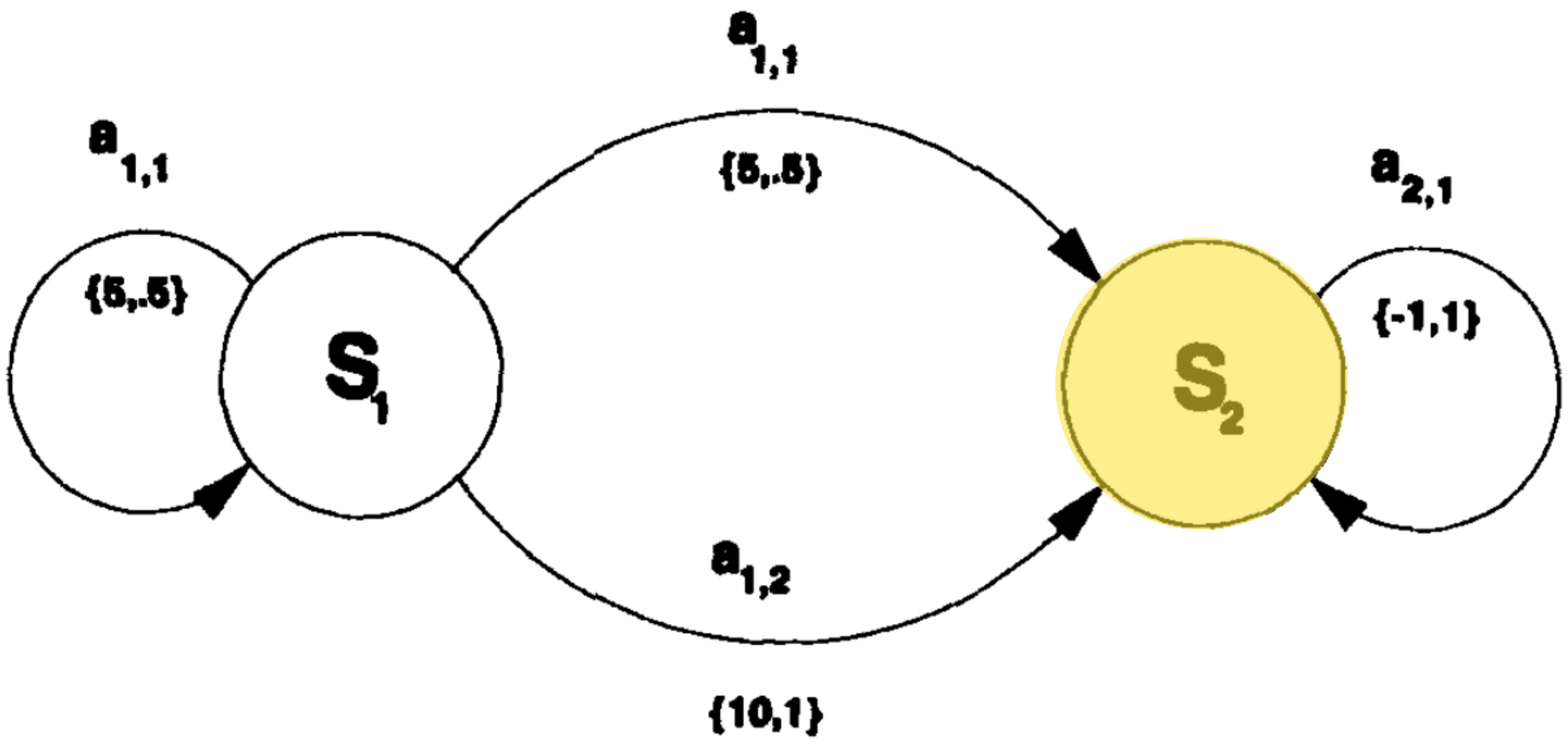
Value



$\pi(s_2) = a_{2,1}$

Reward = -1

Value



$\pi(s_2) = a_{2,1}$

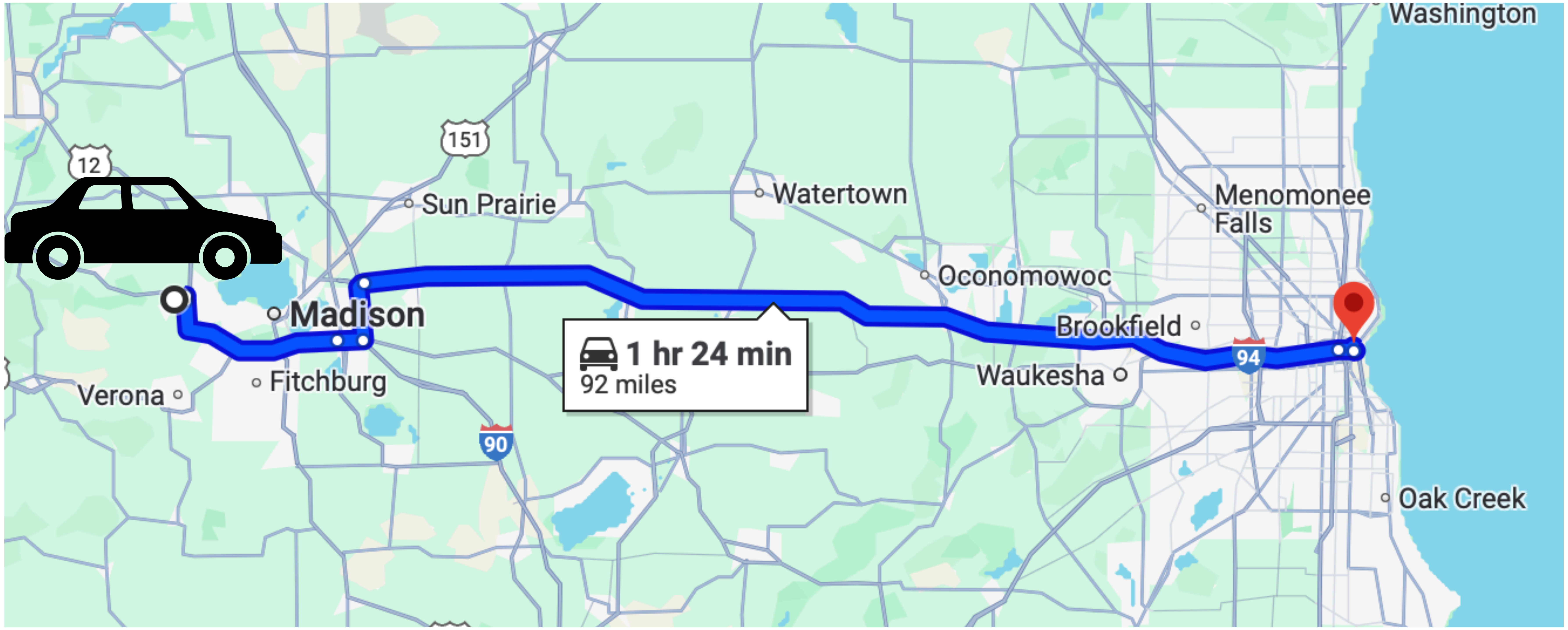
Reward = -1

Value

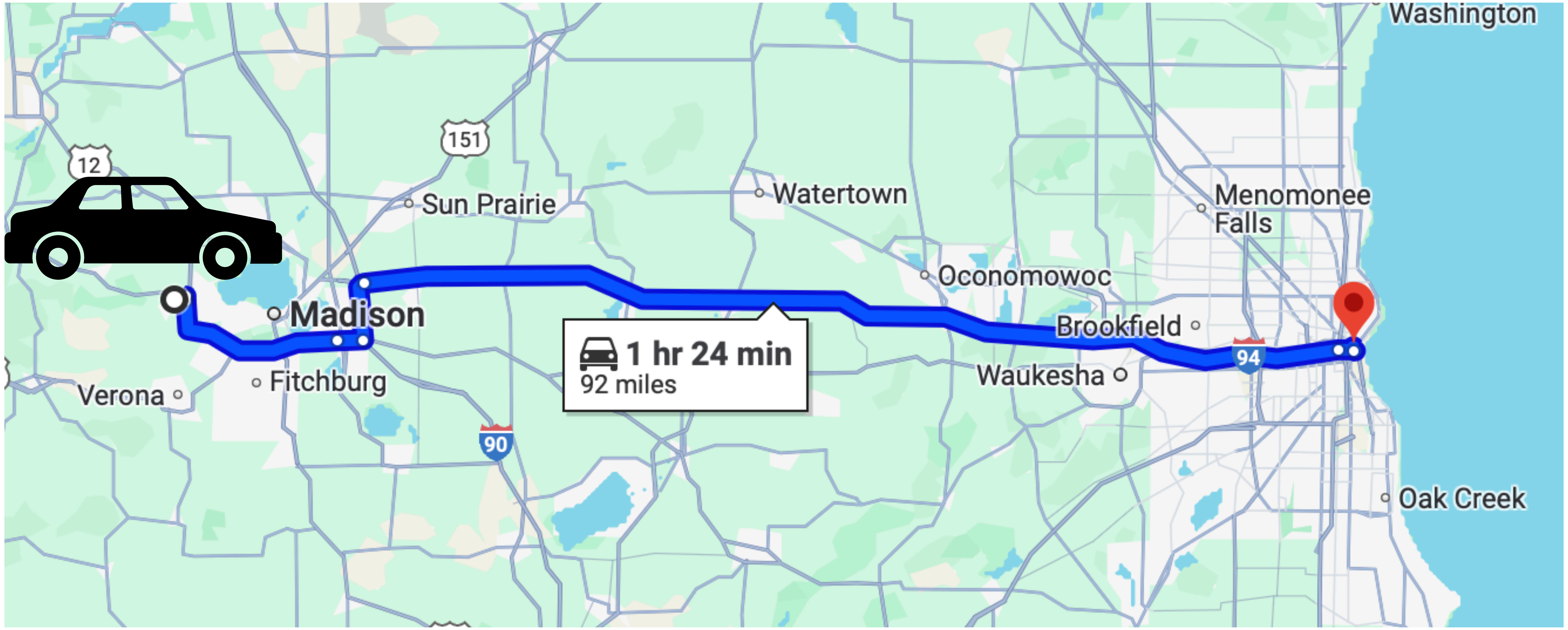
$$V^{\pi}(s_1) = 10 - 1 - 1 = 8$$

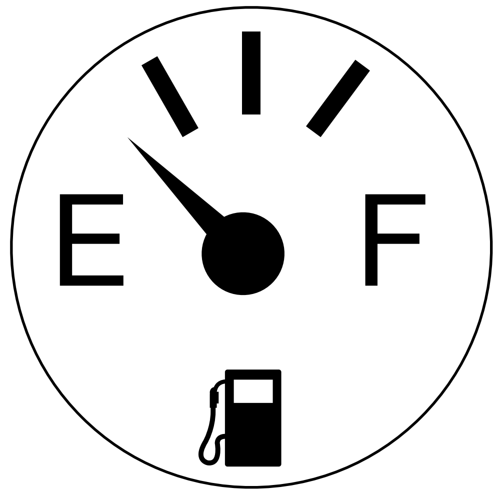
Constrained RL

Constrained RL

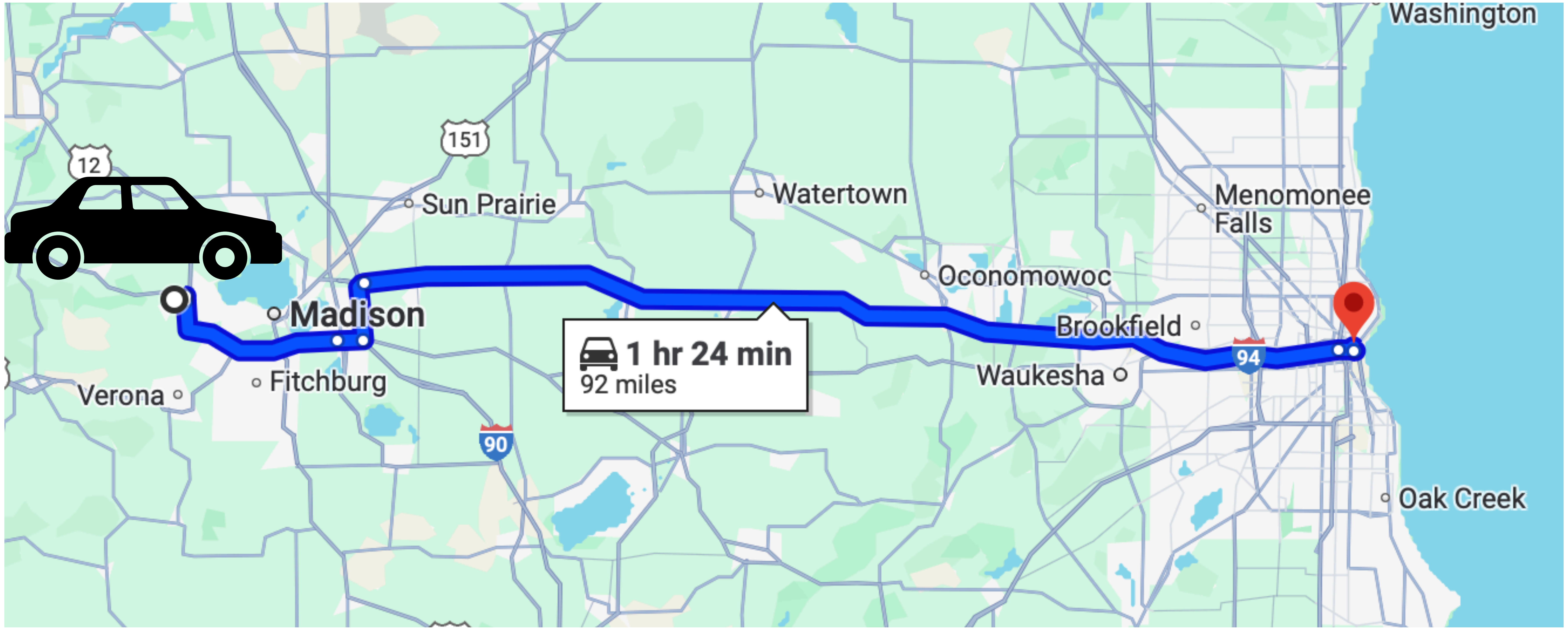


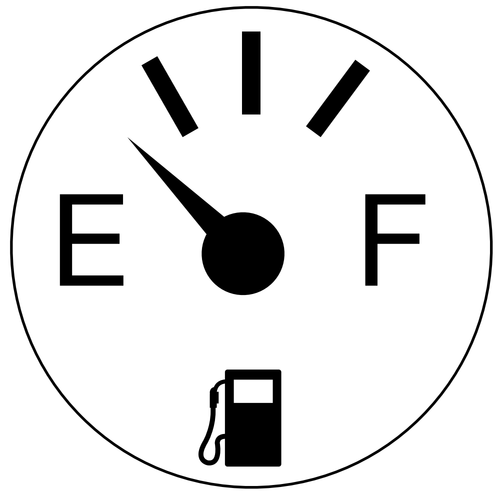
Constrained RL

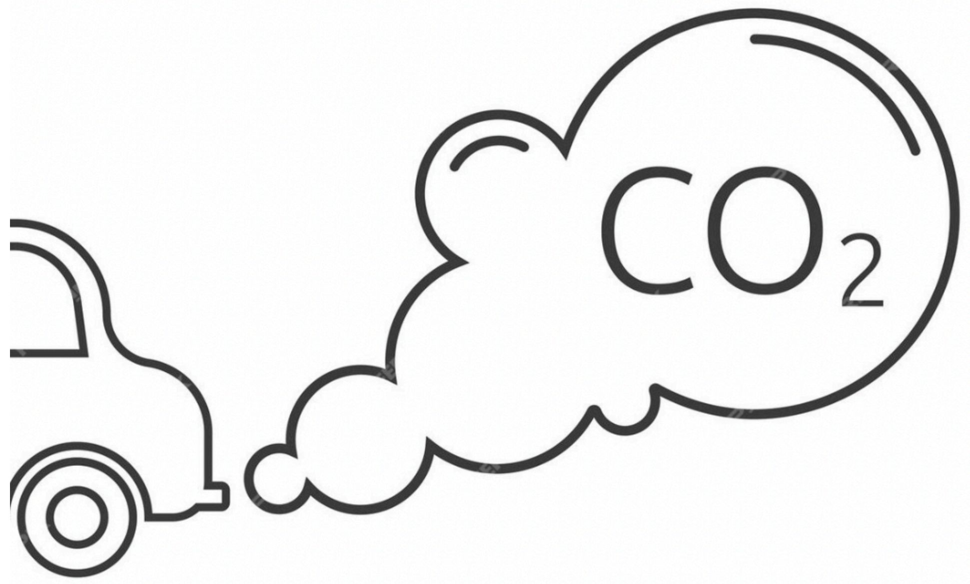




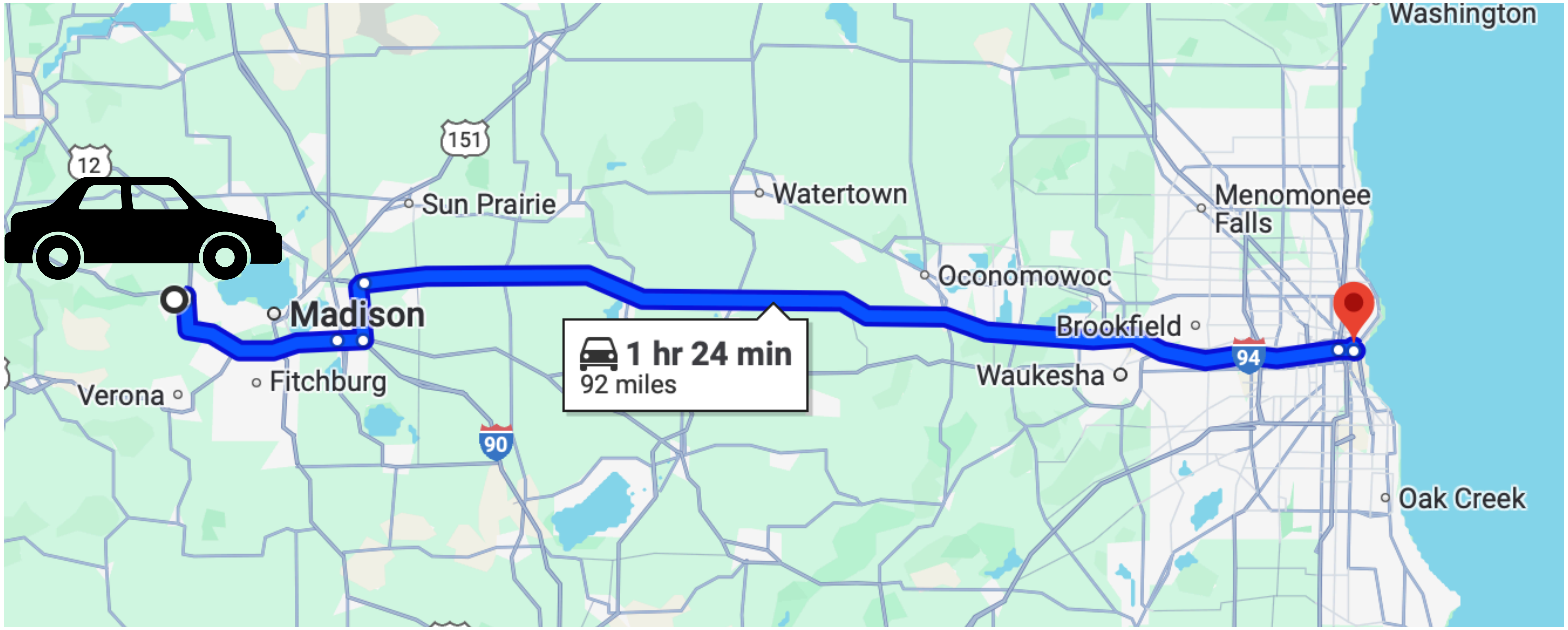
Constrained RL

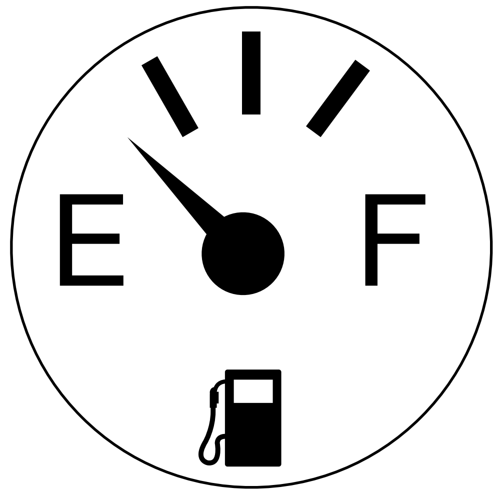


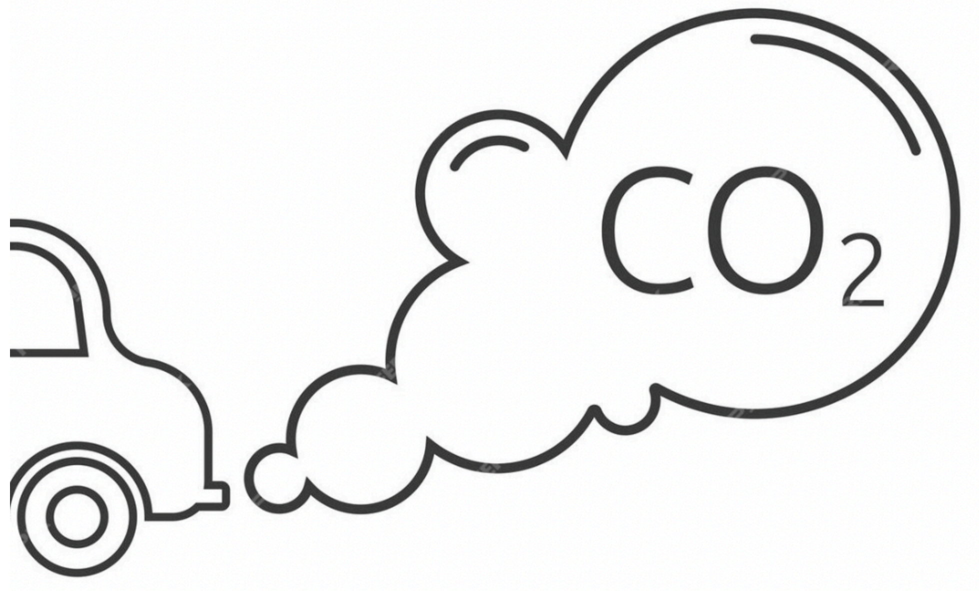




Constrained RL







Why Deterministic Policies?

Why Deterministic Policies?

- Cheap [1]

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]
  - Predictable

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]
  - Predictable

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]
  - Predictable
- Optimal for modern constraints [4]

Modern Constraints

Modern Constraints

Expectation

Modern Constraints

$$\mathbb{E}_M^\pi\left[\sum_{h=1}^{H} c_h\right] \leq B$$

Expectation

Modern Constraints

$$\mathbb{E}_M^{\pi}\left[\sum_{h=1}^{H} c_h\right] \leq B$$

Expectation

⬇

Chance

Modern Constraints

$$\mathbb{E}_M^\pi\left[\sum_{h=1}^H c_h\right] \leq B$$

Expectation

⬇

$$\mathbb{P}_M^\pi\left[\sum_{h=1}^H c_h > B\right] \leq \delta$$

Chance

Modern Constraints

$$\mathbb{E}^{\pi}_{M}\left[\sum_{h=1}^{H} c_h\right] \leq B$$

Expectation

⬇

$$\mathbb{P}^{\pi}_{M}\left[\sum_{h=1}^{H} c_h > B\right] \leq \delta$$

Chance

⬇

Almost Sure

Modern Constraints

$$\mathbb{E}_M^\pi \left[\sum_{h=1}^{H} c_h \right] \leq B$$

Expectation

↓

$$\mathbb{P}_M^\pi \left[\sum_{h=1}^{H} c_h > B \right] \leq \delta$$

Chance

↓

$$\mathbb{P}_M^\pi \left[\sum_{h=1}^{H} c_h \leq B \right] = 1$$

Almost Sure

Modern Constraints

$$\mathbb{E}_M^\pi\left[\sum_{h=1}^H c_h\right] \le B$$ Expectation

⬇

$$\mathbb{P}_M^\pi\left[\sum_{h=1}^H c_h > B\right] \le \delta$$ Chance

⬇

$$\mathbb{P}_M^\pi\left[\sum_{h=1}^H c_h \le B\right] = 1$$ Almost Sure

⬇

Anytime

Modern Constraints

$$\mathbb{E}_M^\pi\left[\sum_{h=1}^H c_h\right] \leq B$$

Expectation

⬇

$$\mathbb{P}_M^\pi\left[\sum_{h=1}^H c_h > B\right] \leq \delta$$

Chance

⬇

$$\mathbb{P}_M^\pi\left[\sum_{h=1}^H c_h \leq B\right] = 1$$

Almost Sure

⬇

$$\mathbb{P}_M^\pi\left[\forall t \in [H],\ \sum_{h=1}^t c_h \leq B\right] = 1$$

Anytime

Cost Functions

Cost Functions

Expectation → $$C_M^\pi := \mathbb{E}_M^\pi\left[\sum_{h=1}^{H} c_h\right]$$

Cost Functions

Expectation → $$C_M^\pi := \mathbb{E}_M^\pi\left[\sum_{h=1}^{H} c_h\right]$$

Chance → $$C_M^\pi := \mathbb{P}_M^\pi\left[\sum_{h=1}^{H} c_h > B\right]$$

Cost Functions

Expectation → $C^\pi_M := \mathbb{E}^\pi_M \left[\sum_{h=1}^{H} c_h \right]$

Chance → $C^\pi_M := \mathbb{P}^\pi_M \left[\sum_{h=1}^{H} c_h > B \right]$

Almost Sure → $C^\pi_M := \max_{\tau_{H+1}} \sum_{t=1}^{H} c_t$

Cost Functions

Expectation → $$C_M^\pi := \mathbb{E}_M^\pi\left[\sum_{h=1}^{H} c_h\right]$$

Chance → $$C_M^\pi := \mathbb{P}_M^\pi\left[\sum_{h=1}^{H} c_h > B\right]$$

Almost Sure → $$C_M^\pi := \max_{\tau_{H+1}} \sum_{t=1}^{H} c_t$$

Anytime → $$C_M^\pi := \max_{h} \max_{\tau_h} \sum_{t=1}^{h-1} c_t$$

Problem

Problem

$$\max_{\pi \in \Pi} \mathbb{E}_M^\pi \left[\sum_{h=1}^{H} r_h(s_h, a_h) \right] \quad \text{s.t.} \quad \begin{cases} C_M^\pi \leq B \\ \pi \text{ deterministic} \end{cases}$$

Problem

$$\max_{\pi \in \Pi} \mathbb{E}_M^\pi \left[\sum_{h=1}^{H} r_h(s_h, a_h) \right] \quad \text{s.t.} \quad \begin{cases} C_M^\pi \leq B \\ \pi \text{ deterministic} \end{cases}$$

C is a general cost criteria

Can near-optimal deterministic policies be computed efficiently?

Challenges

Challenges

- Problem is NP-hard

Challenges

- Problem is NP-hard
- Feasibility is NP-hard for > 1 constraint

Challenges

- Problem is NP-hard
- Feasibility is NP-hard for $>$ 1 constraint
- Approximate Feasibility NP-hard when $d \geq S$

Challenges

- Problem is NP-hard
- Feasibility is NP-hard for >1 constraint
- Approximate Feasibility NP-hard when $d \geq S$
- Problem is not continuous

Challenges

- Problem is NP-hard
- Feasibility is NP-hard for $>$ 1 constraint
- Approximate Feasibility NP-hard when $d \geq S$
- Problem is not continuous
- Dynamic programming fails

Results

Results

Answer: **Yes**!

Results

Answer: **Yes**!

We design an additive and relative **FPTAS** for general cost criteria, including **expectation**, **almost-sure**, and **anytime**.

Results

Answer: **Yes**!

We design an additive and relative **FPTAS** for general cost criteria, including **expectation**, **almost-sure**, and **anytime**.

**We only exclude chance constraints which are provably inapproximable*

Key: Feasibility Computation

Key: Feasibility Computation

Sufficient for efficient feasibility checking: efficient *policy evaluation*

Key: Feasibility Computation

Sufficient for efficient feasibility checking: efficient *policy evaluation*

Assumption [time-space recursive]: *the cost of a policy is computable recursively over both* ***time*** *and state* ***space***

Key: Feasibility Computation

Sufficient for efficient feasibility checking: efficient *policy evaluation*

Assumption [time-space recursive]: *the cost of a policy is computable recursively over both **time** and state **space***

**holds for expectation, almost sure, and anytime constraints*

Definition 1 (TSR)**.** We call a cost criterion C *time-recursive* (TR) if for any cMDP M and policy $\pi \in \Pi^D$, π's cost decomposes recursively into $C_M^\pi = C_1^\pi(s_0)$. Here, $C_{H+1}^\pi(\cdot) = \mathbf{0}$ and for any $h \in [H]$ and $\tau_h \in \mathcal{H}_h$,

$$C_h^\pi(\tau_h) = c_h(s,a) + f\left(\left(P_h(s' \mid s,a), C_{h+1}^\pi(\tau_h, a, s')\right)_{s' \in P_h(s,a)}\right), \tag{TR}$$

where $s = s_h(\tau_h)$, $a = \pi_h(\tau_h)$, and f is a non-decreasing function[1] computable in $O(S)$ time. For technical reasons, we also require that $f(x) = \infty$ whenever $\infty \in x$.

We further say C is *time-space-recursive* (TSR) if the f term above is equal to $g_h^{\tau_h,a}(1)$. Here, $g_h^{\tau_h,a}(S+1) = 0$ and for any $t \leq S$,

$$g_h^{\tau_h,a}(t) = \alpha\left(\beta\left(P_h(t \mid s,a), C_{h+1}^\pi(\tau_h, a, t)\right), g_h^{\tau_h,a}(t+1)\right), \tag{SR}$$

where α is a non-decreasing function, and both α, β are computable in $O(1)$ time. We also assume that $\alpha(\cdot, \infty) = \infty$, and β satisfies $\alpha(\beta(0,\cdot), x) = x$ to match f's condition.

Reduction

Reduction

Packing (Primal)

$$\max_{\pi \in \Pi^D} \quad V_M^\pi$$
$$\text{s.t.} \quad C_M^\pi \leq B$$

Reduction

Packing (Primal)

$$\max_{\pi \in \Pi^D} \quad V_M^\pi$$
$$\text{s.t.} \quad C_M^\pi \leq B$$

→

Covering (Dual)

$$\min_{\pi \in \Pi^D} \quad C_M^\pi$$
$$\text{s.t.} \quad V_M^\pi \geq V^*$$

Knapsack Algorithms

Knapsack Algorithms

Budget: $K(i, b) := \max(v_i + K(i+1, b - w_i), K(i+1, b))$

Knapsack Algorithms

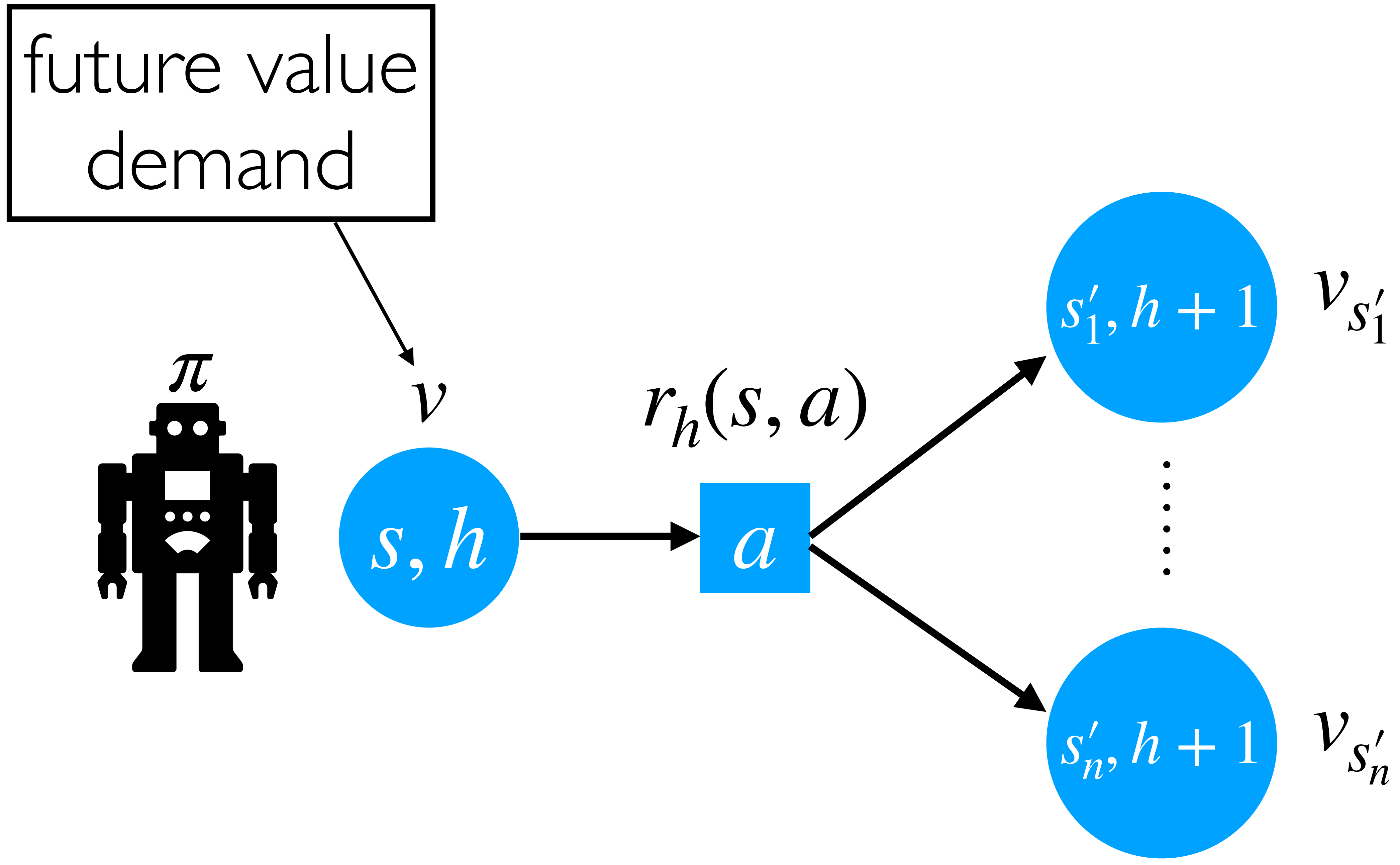
Budget: $K(i, b) := \max(v_i + K(i+1, b - w_i), K(i+1, b))$

⬇

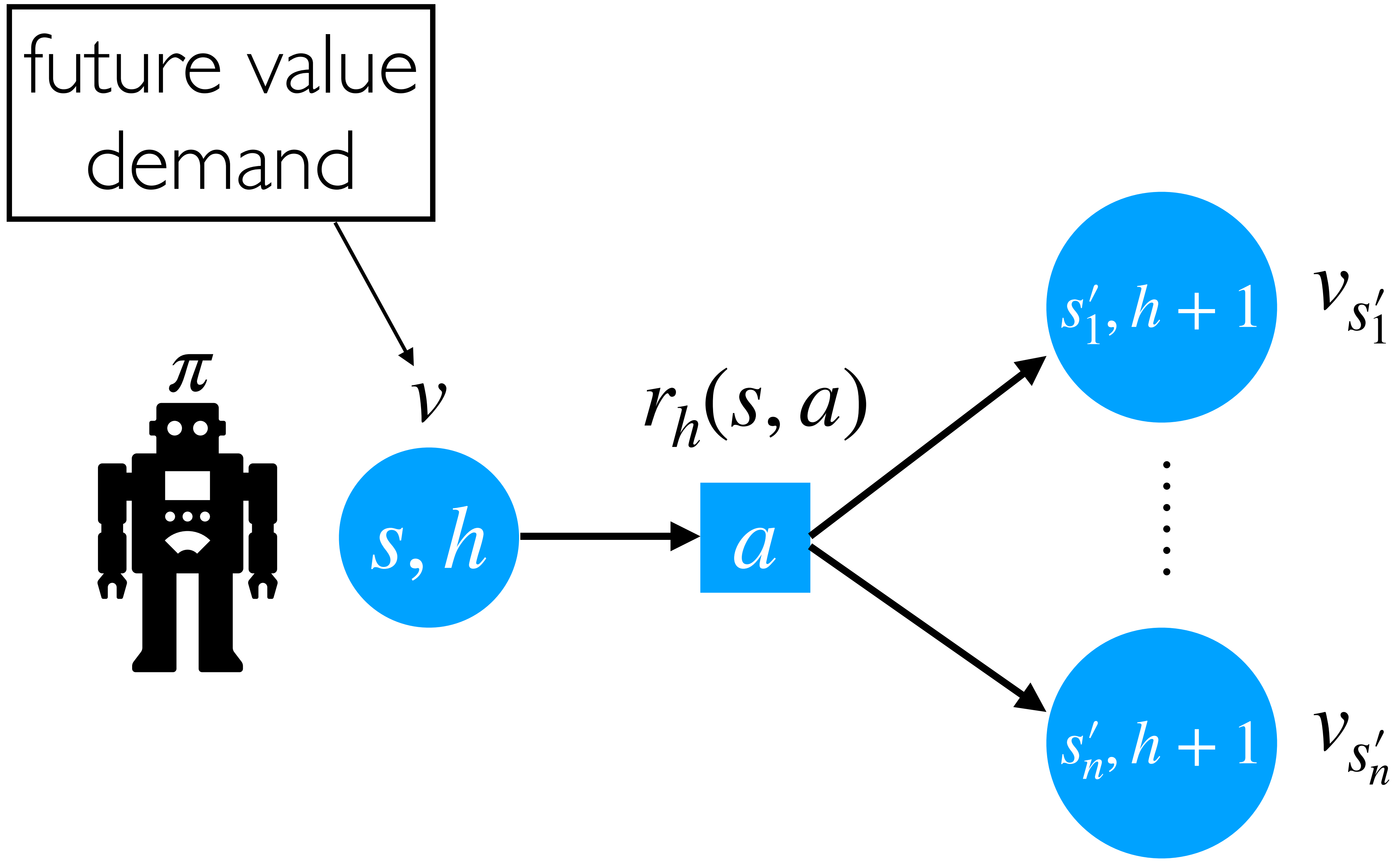
Demand: $K(i, d) := \min(w_i + K(i+1, d - v_i), K(i+1, d))$

State Augmentation

State Augmentation



State Augmentation



Want: $$C_h^*(s, v) = \min_{\pi \in \Pi^D} \quad C_h^\pi(\tau_h)$$
$$\text{s.t.} \quad V_h^\pi(\tau_h) \geq v$$

Action Augmentation

Action Augmentation

$$V_h^\pi(s, v) = r_h(s, a) + \sum_{s'} P_h(s' \mid s, a) V_{h+1}^\pi(s', v_{s'}) \geq v$$

Action Augmentation

$$V_h^\pi(s, v) = r_h(s, a) + \sum_{s'} P_h(s' \mid s, a) V_{h+1}^\pi(s', v_{s'}) \geq v$$

How to choose $v_1, \ldots, v_S$?

Action Augmentation

$$V_h^\pi(s, v) = r_h(s, a) + \sum_{s'} P_h(s' \mid s, a) V_{h+1}^\pi(s', v_{s'}) \geq v$$

How to choose $v_1, \ldots, v_S$? **Try them all!**

$$\mathcal{A}_h(s, v) := \left\{ (a, \mathbf{v}) \in \mathcal{A} \times \mathcal{V}^S \mid r_h(s, a) + \sum_{s'} P_h(s' \mid s, a) v_{s'} \geq v \right\}$$

Algorithm

Algorithm

Solve: $$C_h^*(s, v) = \min_{a,\mathbf{v} \in \mathcal{A}_h(s,v)} \; c_h(s, a) + \sum_{s'} P_h(s' \mid s, a) C_{h+1}^*(s', v_{s'})$$

Algorithm

Expectation Constraints

Solve: $$C_h^*(s,v) = \min_{a,\mathbf{v} \in \mathcal{A}_h(s,v)} \; c_h(s,a) + \overbrace{\sum_{s'} P_h(s' \mid s,a) C_{h+1}^*(s', v_{s'})}$$

Algorithm

Expectation Constraints

Solve: $$C_h^*(s, v) = \min_{a, \mathbf{v} \in \mathcal{A}_h(s,v)} \; c_h(s, a) + \sum_{s'} \overbrace{P_h(s' \mid s, a) C_{h+1}^*(s', v_{s'})}$$

Output: $$V_M^* = \max \{v \in \mathcal{V} \mid C_1^*(s_0, v) \leq B\}$$

Issues

Issues

1. Too many states — rounding

Issues

1. Too many states — rounding
2. Too many actions — sub DP

Subproblem DP

Subproblem DP

$$r_h(s, a) + P_h(1 \mid s, a)v_1 + \cdots + P_h(S \mid s, a)v_S$$

Subproblem DP

$$r_h(s,a) + \underbrace{P_h(1 \mid s,a)v_1 + \cdots + P_h(S \mid s,a)v_S}_{\text{Can choose each } v_i \text{ independently}}$$

Subproblem DP

$$r_h(s,a) + P_h(1 \mid s,a)v_1 + \cdots + P_h(S \mid s,a)v_S$$

Can choose each v_i independently

Space Recursion!

Subproblem DP

$$r_h(s,a) + P_h(1 \mid s,a)v_1 + \cdots + P_h(S \mid s,a)v_S$$

Can choose each v_i independently

Space Recursion!

$$g(t,u) = \min_{v_t \in \mathcal{V}} P_h(t \mid s,a)C^*_{h+1}(t,v_t) + g(t+1, u + P_h(t \mid s,a)v_t)$$

Subproblem DP

$$r_h(s,a) + P_h(1 \mid s,a)v_1 + \cdots + P_h(S \mid s,a)v_S$$

Can choose each v_i independently

Space Recursion!

Partial sum

$$g(t,u) = \min_{v_t \in \mathcal{V}} P_h(t \mid s,a)C^*_{h+1}(t,v_t) + g(t+1, u + P_h(t \mid s,a)v_t)$$

Subproblem DP

$$r_h(s,a) + P_h(1 \mid s,a)v_1 + \cdots + P_h(S \mid s,a)v_S$$

Can choose each v_i independently

Space Recursion!

Partial sum

$$g(t,u) = \min_{v_t \in \mathcal{V}} P_h(t \mid s,a)C^*_{h+1}(t,v_t) + g(t+1, u + P_h(t \mid s,a)v_t)$$

Value check at end: $g(S+1,u) := \chi_{\{u \geq v\}}$

Approximation

Approximation

Round values down to the closest in $\tilde{V} = \{0,1,\frac{1}{1-\delta}^{2},\dots,\frac{1}{1-\delta}^{k}\}$

Approximation

Round values down to the closest in $\tilde{V} = \{0, 1, \frac{1}{1-\delta}^2, \ldots, \frac{1}{1-\delta}^k\}$

- Main DP accumulates error over time

Approximation

Round values down to the closest in $\tilde{V} = \{0, 1, \frac{1}{1-\delta}^2, \ldots, \frac{1}{1-\delta}^k\}$

- Main DP accumulates error over time
- Sub DP accumulates error over space

Approximation

Round values down to the closest in $\tilde{V} = \{0, 1, \frac{1}{1-\delta}^{2}, \ldots, \frac{1}{1-\delta}^{k}\}$

- Main DP accumulates error over time
- Sub DP accumulates error over space

Approximation

Round values down to the closest in $\tilde{V} = \{0, 1, \frac{1}{1-\delta}^2, \ldots, \frac{1}{1-\delta}^k\}$

- Main DP accumulates error over time
- Sub DP accumulates error over space

$$V_M^\pi \geq (1-\delta)^{SH} v$$

Approximation

Round values down to the closest in $\tilde{V} = \{0, 1, \frac{1}{1-\delta}^2, \ldots, \frac{1}{1-\delta}^k\}$

- Main DP accumulates error over time
- Sub DP accumulates error over space

$$V_M^\pi \geq (1-\delta)^{SH} v$$

$$\delta = \frac{\epsilon}{SH} \implies V_M^\pi \geq (1-\epsilon)V^*$$

Iterative Rounding

Iterative Rounding

- Must use a different rounding per h since values involve varying products

Iterative Rounding

- Must use a different rounding per h since values involve varying products
- Instead we use one consistent recursive rounding

Guarantees

Guarantees

The guarantees depend on the reward structure:

Guarantees

The guarantees depend on the reward structure:

- **Theorem 1 (Additive)**: If the reward range is *bounded* by $\mathrm{poly}(|M|)$, we get an *additive* FPTAS.

Guarantees

The guarantees depend on the reward structure:

- **Theorem 1 (Additive)**: If the reward range is *bounded* by $\text{poly}(|M|)$, we get an *additive* FPTAS.

- **Theorem 2 (Relative)**: If the rewards are *non-negative*, we get a *relative* FPTAS.

Guarantees

The guarantees depend on the reward structure:

- **Theorem 1 (Additive)**: If the reward range is *bounded* by $\mathrm{poly}(|M|)$, we get an *additive* FPTAS.

- **Theorem 2 (Relative)**: If the rewards are *non-negative*, we get a *relative* FPTAS.

**These assumptions are necessary as well*

Conclusion

Conclusion

Answers **three** long-standing open questions.

Conclusion

Answers **three** long-standing open questions.

Polynomial-time approximability is possible for:

Conclusion

Answers **three** long-standing open questions.

Polynomial-time approximability is possible for:

- *Almost-sure-constrained policies*

Conclusion

Answers **three** long-standing open questions.

Polynomial-time approximability is possible for:

- *Almost-sure-constrained policies*
- *Anytime-constrained policies*

Conclusion

Answers **three** long-standing open questions.

Polynomial-time approximability is possible for:

- *Almost-sure-constrained policies*
- *Anytime-constrained policies*
- *Deterministic, expectation-constrained policies*

Conclusion

Answers **three** long-standing open questions.

Polynomial-time approximability is possible for:

- *Almost-sure-constrained policies*
- *Anytime-constrained policies*
- *Deterministic, expectation-constrained policies*

Open for nearly 25 years!

Future Work

Future Work

Are multiple constraints truly much harder?

Future Work

Are multiple constraints truly much harder?

- Are there special cases for which multiple constraints are solvable?

Future Work

Are multiple constraints truly much harder?

- Are there special cases for which multiple constraints are solvable?
- Like for the simplex method, is the smoothed complexity or average case complexity small?

Thank you!

References

1

MATHEMATICS OF OPERATIONS RESEARCH
Vol. 25, No. 1, February 2000
Printed in U.S.A.

CONSTRAINED DISCOUNTED MARKOV DECISION PROCESSES AND HAMILTONIAN CYCLES

EUGENE A. FEINBERG

2

Towards a formalization of teamwork with resource constraints

Praveen Paruchuri, Milind Tambe, Fernando Ordonez
University of Southern California
Los Angeles, CA 90089
{paruchur,tambe,fordon}@usc.edu

Sarit Kraus
Bar-Ilan University
Ramat-Gan 52900, Israel
sarit@macs.biu.ac.il

3

Stationary Deterministic Policies for Constrained MDPs with Multiple Rewards, Costs, and Discount Factors

Dmitri Dolgov and Edmund Durfee
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109
{ddolgov, durfee}@umich.edu

4

Anytime-Constrained Reinforcement Learning

Jeremy McMahan **Xiaojin Zhu**
University of Wisconsin-Madison

Definition 8 (Relative Approx). Fix $\epsilon > 0$. We define,

$$\lfloor v \rfloor_{\mathcal{G}} \overset{\text{def}}{=} v^{min} \left(\frac{1}{1-\delta}\right)^{\left\lfloor \log_{\frac{1}{1-\delta}} \frac{v}{v^{min}} \right\rfloor} \text{ and } \kappa(v) \overset{\text{def}}{=} v(1-\delta)^{S+1}, \tag{7}$$

where $\delta \overset{\text{def}}{=} \frac{\epsilon}{H(S+1)+1}$, $v_{min} = p_{min}^H r_{min}$, and $v_{max} = H r_{max}$.

Theorem 3 (Relative FPTAS). *For $\epsilon > 0$, Algorithm 5 using Definition 8 given any cMDP M and TSR criteria C either correctly outputs the instance is infeasible, or produces a policy π satisfying $\hat{V}^\pi \geq V_M^*(1-\epsilon)$ in $O(H^7 S^5 A \log\left(r_{max}/r_{min}p_{min}\right)^3 / \epsilon^3)$ time. Thus, it is a relative-FPTAS for the class of cMDPs with non-negative rewards and TSR criteria.*