Learning to Maximize Welfare with a Reusable Resource

Speaker: Matthew Faw

Authors: **F**, Orestis Papadigenopoulos, Constantine Caramanis, and Sanjay Shakkottai
The University of Texas at Austin

[SIGMETRICS/ IFIP Performance 2022]

Motivation

- Gig-economy workers:
  - Ride-sharing drivers
  - Delivery drivers
  - Crowdsourcing workers
- Decide whether to accept a task and collect a reward, or skip on it
- In case of acceptance, worker becomes busy for a short time period
- In many applications, busy periods have fixed duration (hour, day, week)

- **Problem:** Decide whether to accept or reject a given task

Problem Setting

- Sequence of n IID rewards (requests of fixed duration)
- Decision-maker observes realized reward X_t of each round and decides
  - Accept the reward and become busy for the subsequent d rounds or
  - Reject the reward and remain available for the next round
- Delay/busy time d is known, but time horizon n is unknown
- Reward distribution D is known (or can be learned)

- **Goal:** Maximize the total expected reward collected
  - Compete against the expected reward collected by a Prophet
  - Prophet knows all the realizations a priori and has infinite computational power

Known Reward Distribution

- Reward distribution D is **known** to the decision-maker.
- Assuming large (or infinite) time horizon, a **fixed-threshold policy** should apply
  - Compute a threshold τ as a function of D and d
  - At round t, if the resource is available and $X_t \geq \tau$ accept, otherwise reject
- We need to guarantee that the expected reward collected is “close” to that of a Prophet
  - How to define τ?
  - Can we characterize the Prophet’s expected reward?

Known Reward Distribution

Infinite-dimensional LP relaxation:

$$
\begin{aligned}
\textbf{maximize:} \quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x)\ \mathrm{d}x \\
\textbf{s.t.:} \quad & \int_{x=0}^{\infty} q(x)\ \mathrm{d}x \ \leq\ \frac{1}{d+1} \\
& 0 \ \leq\ q(x) \ \leq\ f(x), \qquad \forall x \geq 0.
\end{aligned}
$$

$q(x)$: fraction of time reward x is collected

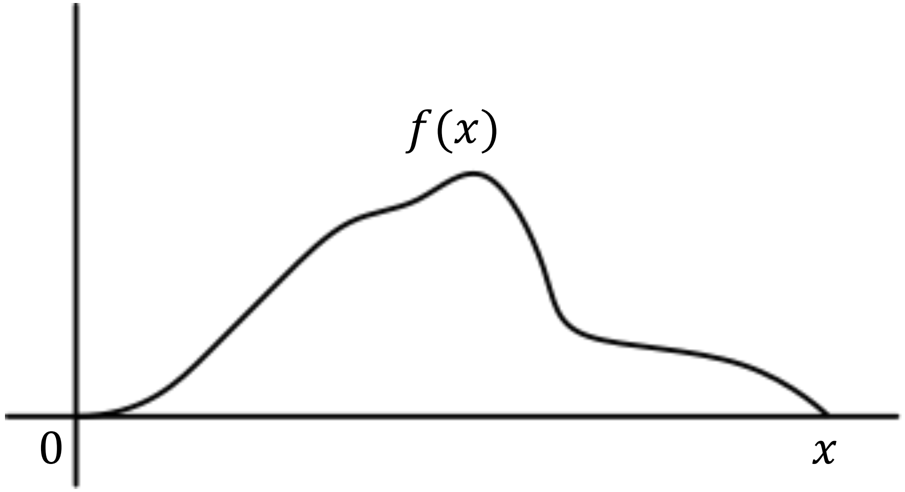
Known Reward Distribution

Infinite-dimensional LP relaxation:

$$\textbf{maximize:} \quad n \cdot \int_{x=0}^{\infty} x \cdot q(x)\ \mathrm{d}x$$
$$\textbf{s.t.:} \int_{x=0}^{\infty} q(x)\ \mathrm{d}x \ \leq \ \frac{1}{d+1}$$
$$0 \ \leq \ q(x) \ \leq \ f(x), \qquad \forall x \geq 0.$$

$q(x)$: fraction of time reward x is collected

$f(x)$: p.d.f. of D



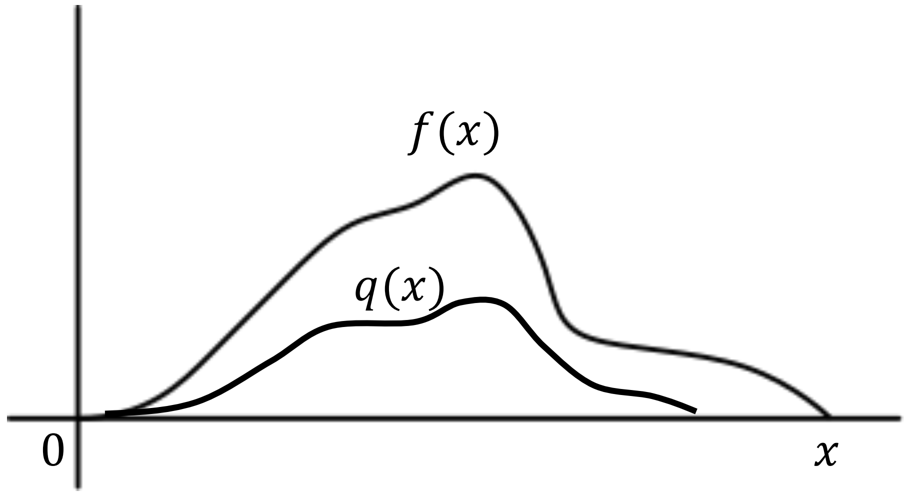
Known Reward Distribution

Infinite-dimensional LP relaxation:

$$
\begin{aligned}
\textbf{maximize:} \quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x) \, \mathrm{d}x \\
\textbf{s.t.:} \quad & \int_{x=0}^{\infty} q(x) \, \mathrm{d}x \;\leq\; \frac{1}{d+1} \\
& 0 \;\leq\; q(x) \;\leq\; f(x), \qquad \forall x \geq 0.
\end{aligned}
$$

$q(x)$: fraction of time reward x is collected

$q(x)$ lies below $f(x)$

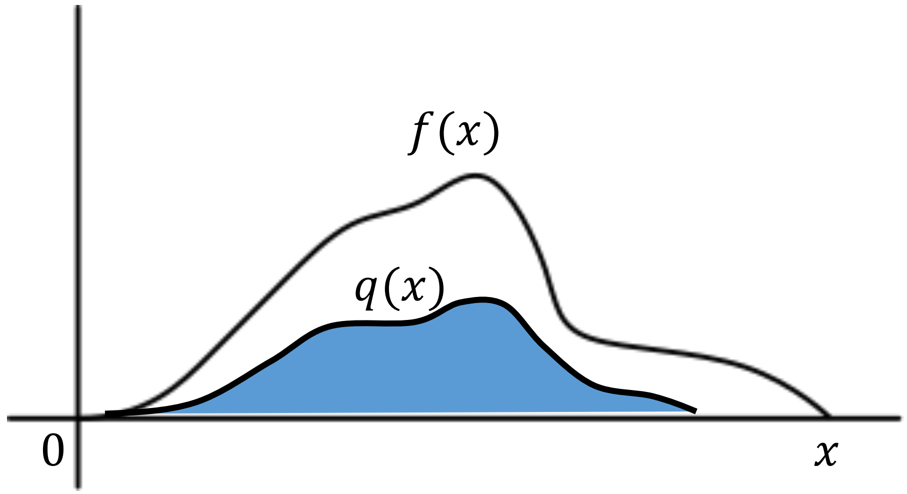


Known Reward Distribution

Infinite-dimensional LP relaxation:

$$\begin{aligned}\textbf{maximize:} \quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x)\ \mathrm{d}x \\ \textbf{s.t.:} \quad & \int_{x=0}^{\infty} q(x)\ \mathrm{d}x \ \le\ \frac{1}{d+1} \\ & 0 \ \le\ q(x)\ \le\ f(x), \qquad \forall x \ge 0.\end{aligned}$$

$\leftarrow$ area of $q(x)$ is at most $\frac{1}{d+1}$



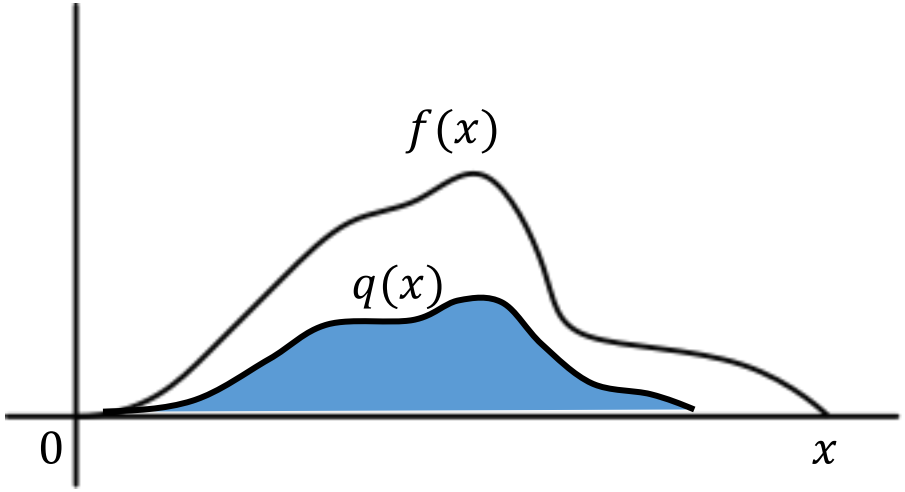
$q(x)$: fraction of time reward x is collected

Known Reward Distribution

Infinite-dimensional LP relaxation:

$$\begin{aligned} \textbf{maximize:} \quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x) \, \mathrm{d}x \\ \textbf{s.t.:} \quad & \int_{x=0}^{\infty} q(x) \, \mathrm{d}x \leq \frac{1}{d+1} \\ & 0 \leq q(x) \leq f(x), \quad \forall x \geq 0. \end{aligned}$$

$\leftarrow$ area of $q(x)$ is at most $\frac{1}{d+1}$



Claim: LP yields upper bound on the Prophet's expected reward:

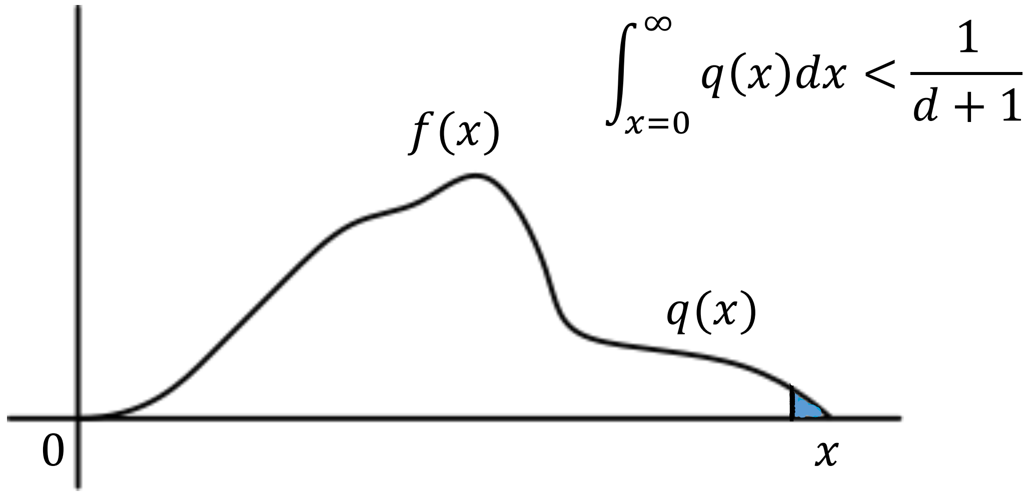
$$n \cdot \int_{x=0}^{\infty} x \cdot q^*(x) \geq OPT \qquad \text{(up to small additive error)}$$

Known Reward Distribution

Infinite-dimensional LP relaxation:

$$
\begin{aligned}
\textbf{maximize:} \quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x)\ \mathrm{d}x \\
\textbf{s.t.:} \quad & \int_{x=0}^{\infty} q(x)\ \mathrm{d}x \ \leq\ \frac{1}{d+1} \\
& 0 \ \leq\ q(x) \ \leq\ f(x), \qquad \forall x \geq 0.
\end{aligned}
$$

Optimal $q^*(x)$ through Greedy Water-filling

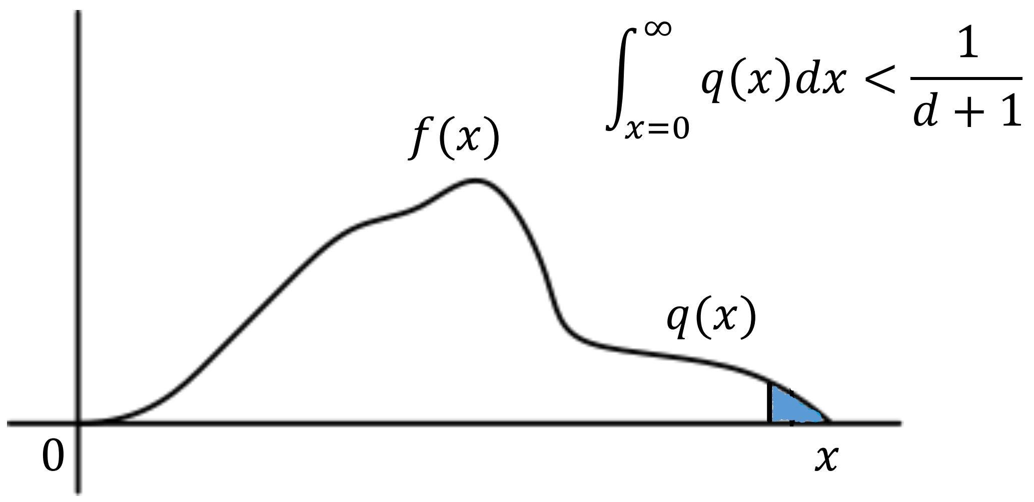


Known Reward Distribution

Infinite-dimensional LP relaxation:

$$\textbf{maximize:}\quad n \cdot \int_{x=0}^{\infty} x \cdot q(x)\ \mathrm{d}x$$
$$\textbf{s.t.:}\int_{x=0}^{\infty} q(x)\ \mathrm{d}x \ \leq\ \frac{1}{d+1}$$
$$0\ \leq\ q(x)\ \leq\ f(x), \qquad \forall x \geq 0.$$

Optimal $q^*(x)$ through Greedy Water-filling

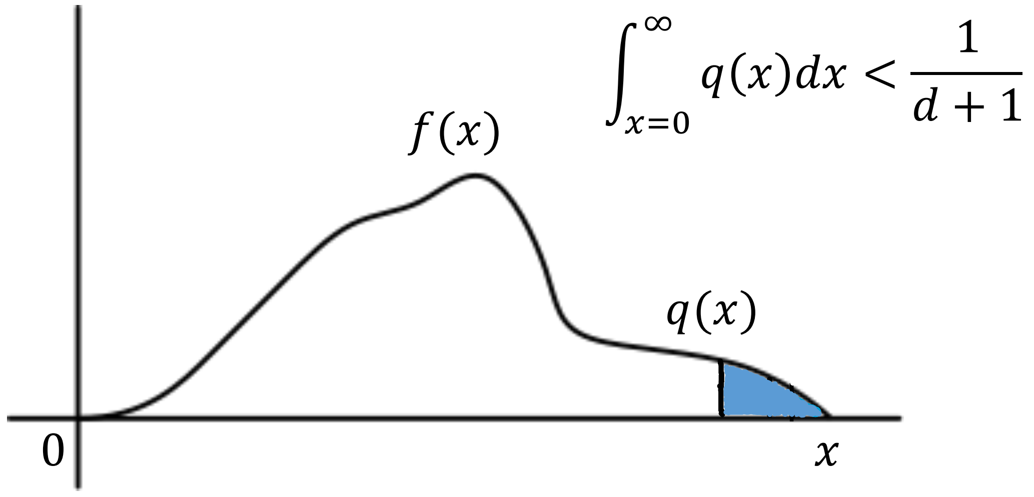


Known Reward Distribution

Infinite-dimensional LP relaxation:

$$
\begin{aligned}
\textbf{maximize:} \quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x)\ \mathrm{d}x \\
\textbf{s.t.:} \quad & \int_{x=0}^{\infty} q(x)\ \mathrm{d}x \ \leq\ \frac{1}{d+1} \\
& 0 \ \leq\ q(x) \ \leq\ f(x), \qquad \forall x \geq 0.
\end{aligned}
$$

Optimal $q^*(x)$ through Greedy Water-filling

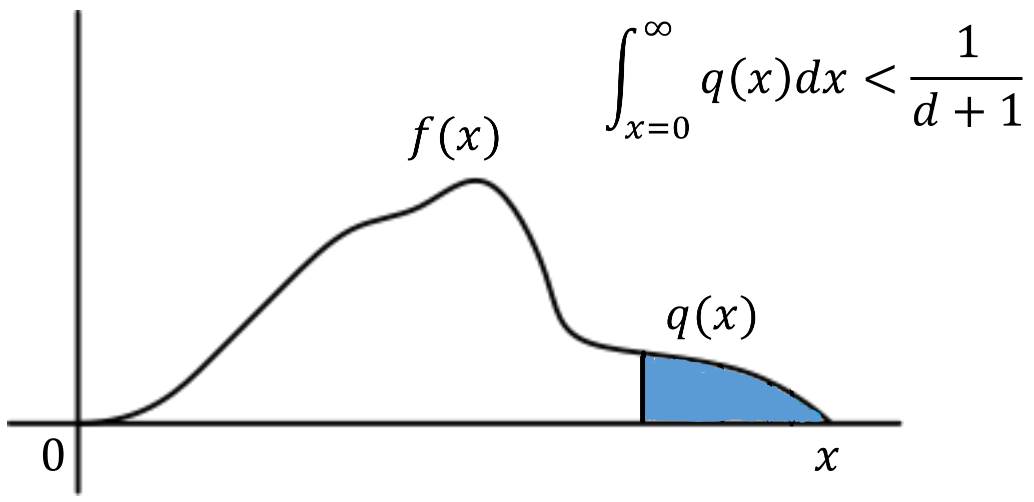


Known Reward Distribution

Infinite-dimensional LP relaxation:

$$
\begin{aligned}
\textbf{maximize:} \quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x) \, \mathrm{d}x \\
\textbf{s.t.:} \quad & \int_{x=0}^{\infty} q(x) \, \mathrm{d}x \leq \frac{1}{d+1} \\
& 0 \leq q(x) \leq f(x), \qquad \forall x \geq 0.
\end{aligned}
$$

Optimal $q^*(x)$ through Greedy Water-filling

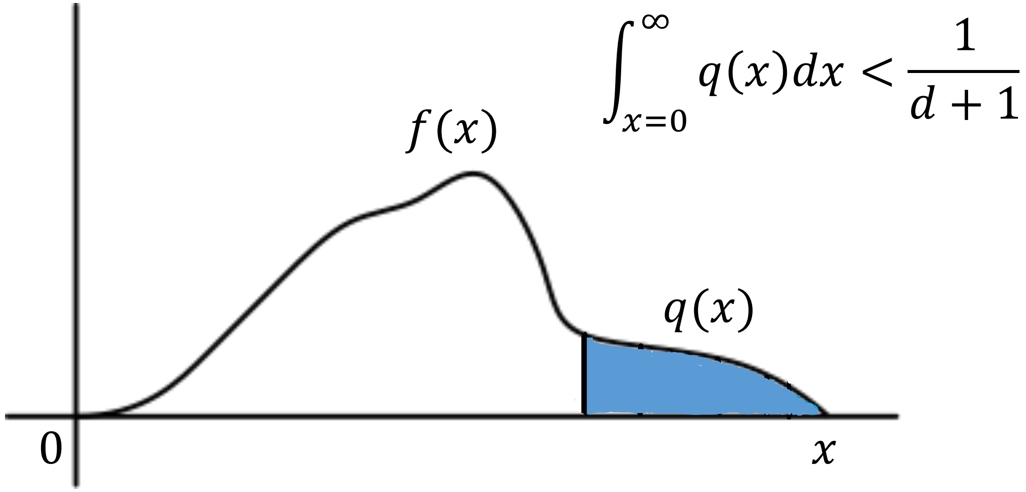


Known Reward Distribution

Infinite-dimensional LP relaxation:

$$\begin{aligned}
\textbf{maximize:} \quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x)\ \mathrm{d}x \\
\textbf{s.t.:} \quad & \int_{x=0}^{\infty} q(x)\ \mathrm{d}x \ \leq\ \frac{1}{d+1} \\
& 0 \ \leq\ q(x) \ \leq\ f(x), \qquad \forall x \geq 0.
\end{aligned}$$

Optimal $q^*(x)$ through Greedy Water-filling

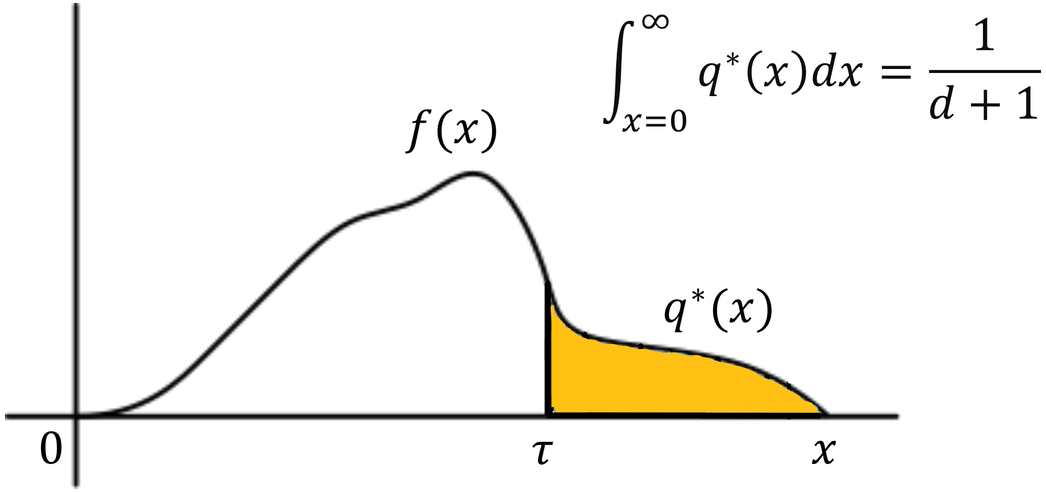


Known Reward Distribution

Infinite-dimensional LP relaxation:

$$\begin{aligned}
\textbf{maximize:} \quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x)\ \mathrm{d}x \\
\textbf{s.t.:} \quad & \int_{x=0}^{\infty} q(x)\ \mathrm{d}x \ \le\ \frac{1}{d+1} \\
& 0 \ \le\ q(x) \ \le\ f(x), \qquad \forall x \ge 0.
\end{aligned}$$

Optimal $q^*(x)$ through Greedy Water-filling



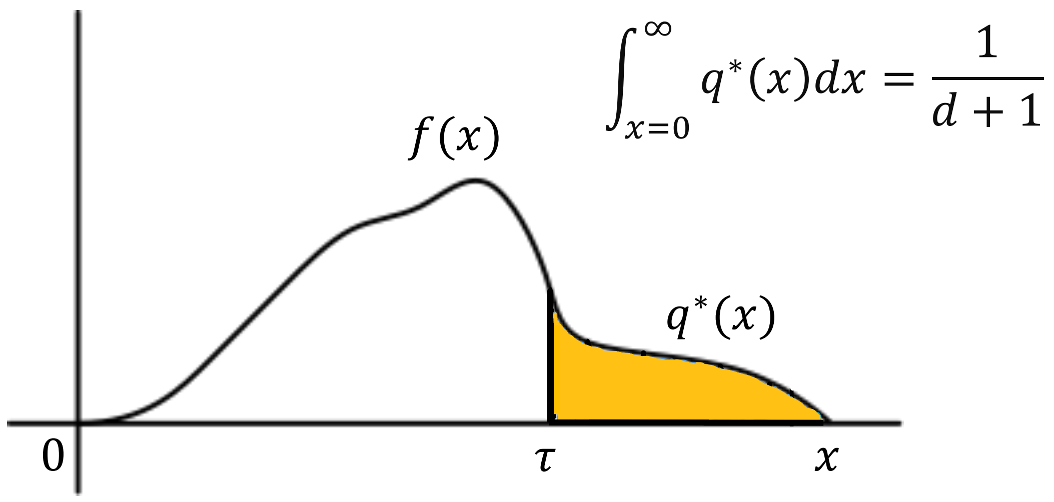
Optimal solution: $q^*(x) = \begin{cases} f(x) & \text{for } x \ge \tau \\ 0 & \text{otherwise} \end{cases}$ where $\tau = F^{-1}(1 - \frac{1}{d+1})$

Known Reward Distribution

Infinite-dimensional LP relaxation:

$$\begin{aligned}\textbf{maximize:}\quad & n \cdot \int_{x=0}^{\infty} x \cdot q(x)\ \mathrm{d}x \\ \textbf{s.t.:}\quad & \int_{x=0}^{\infty} q(x)\ \mathrm{d}x \ \le\ \frac{1}{d+1} \\ & 0 \ \le\ q(x)\ \le\ f(x), \qquad \forall x \ge 0.\end{aligned}$$

Optimal $q^*(x)$ through Greedy Water-filling



Optimal solution: $q^*(x) = \begin{cases} f(x) & \text{for } x \ \ge \tau \\ 0 & \text{otherwise} \end{cases}$ where $\tau = F^{-1}(1 - \frac{1}{d+1})$

Optimal value: $n \cdot \int_{x=0}^{\infty} x \cdot q^*(x)dx = n \cdot \mathbb{E}[X \cdot \mathbb{I}[X \ge \tau]]$

Known Reward Distribution

Set threshold $\tau \leftarrow F^{-1}\left(1 - \frac{1}{d+1}\right)$

for $t = 1, 2, \ldots$ **do**

  Observe reward X_t

  if $X_t \geq \tau$ **and** *resource is available* **then**

    Collect X_t and make resource unavailable for rounds $t+1, \ldots, t+d$

  else

    Skip on X_t

  end

end

Known Reward Distribution

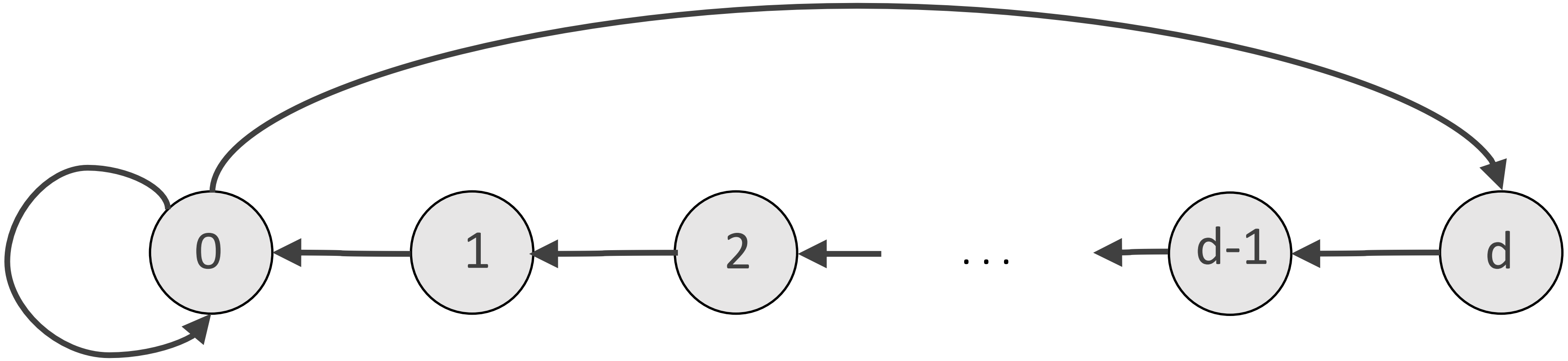
At any round t, Algorithm 1 collects in expectation:

$$\begin{aligned}\mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \text{ is collected}\right\}\right] &= \mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \geq \tau \text{ and } \mathsf{free}_{\mathcal{A}}(t)\right\}\right] \\ &= \mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \geq \tau\right\}\right] \cdot \mathbf{Pr}\left[\mathsf{free}_{\mathcal{A}}(t)\right]\end{aligned}$$

Known Reward Distribution

Markov Reward Process:

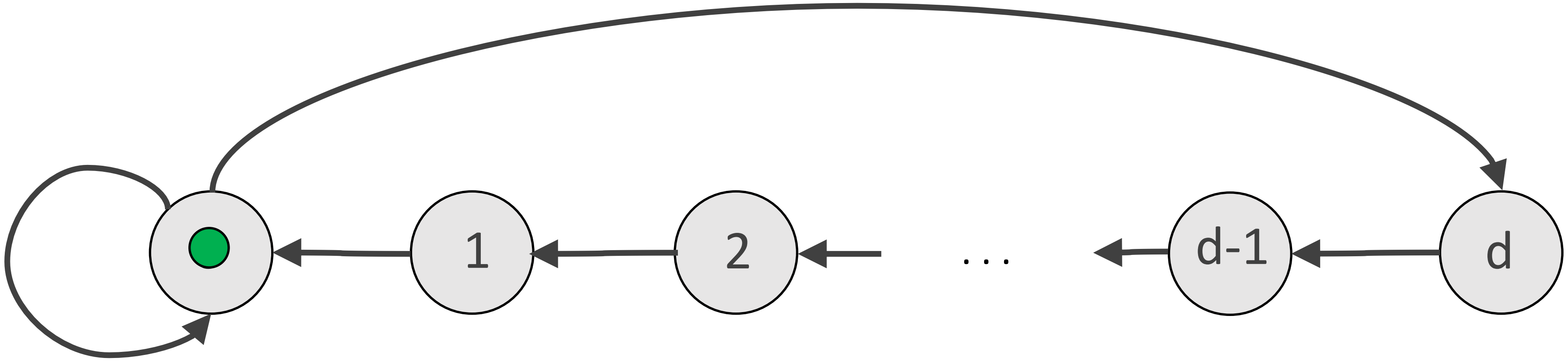
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

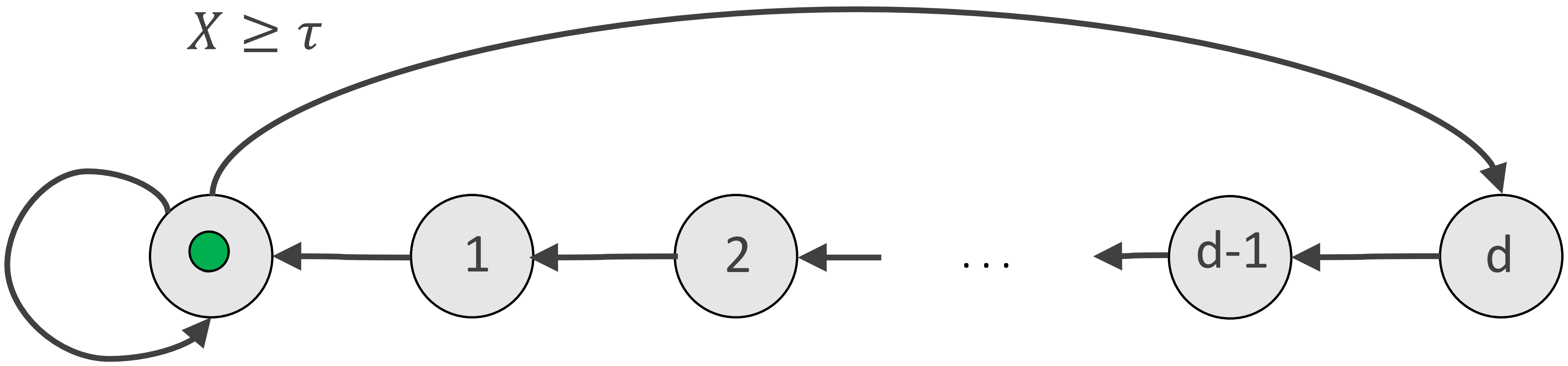
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

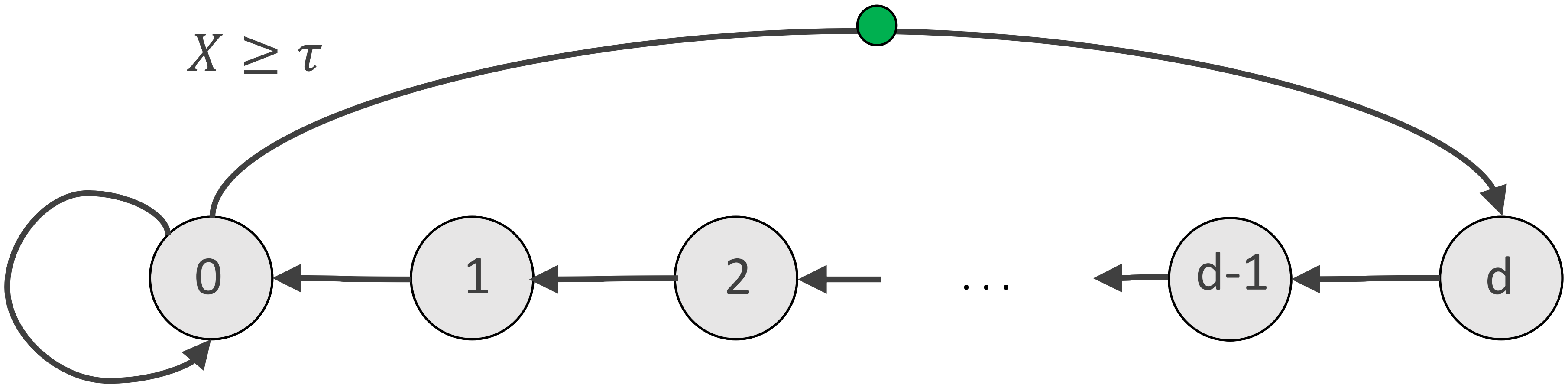
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

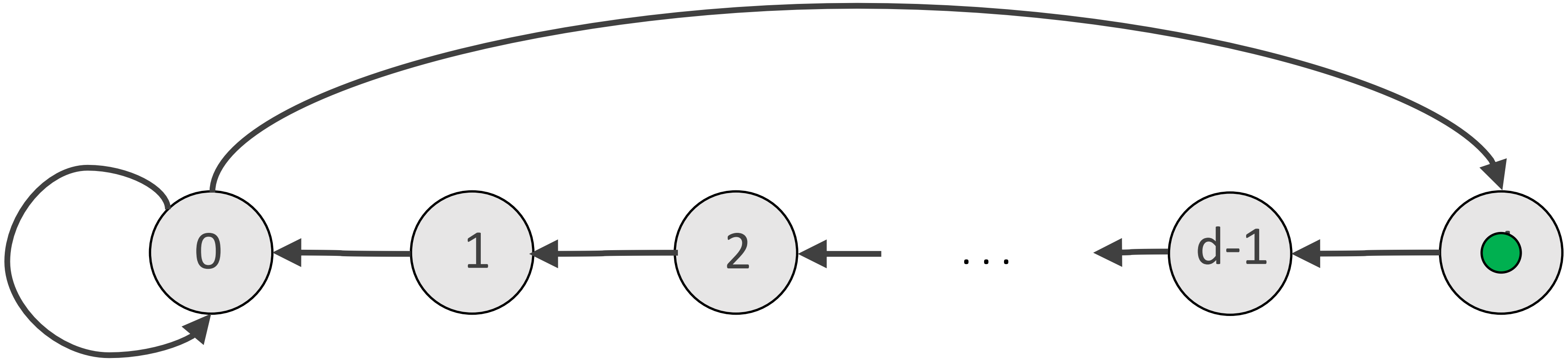
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

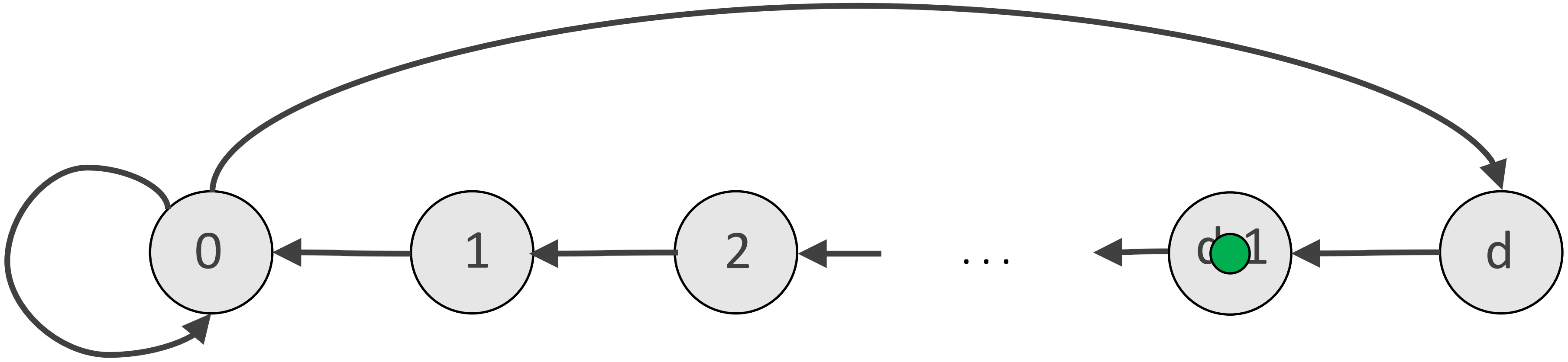
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

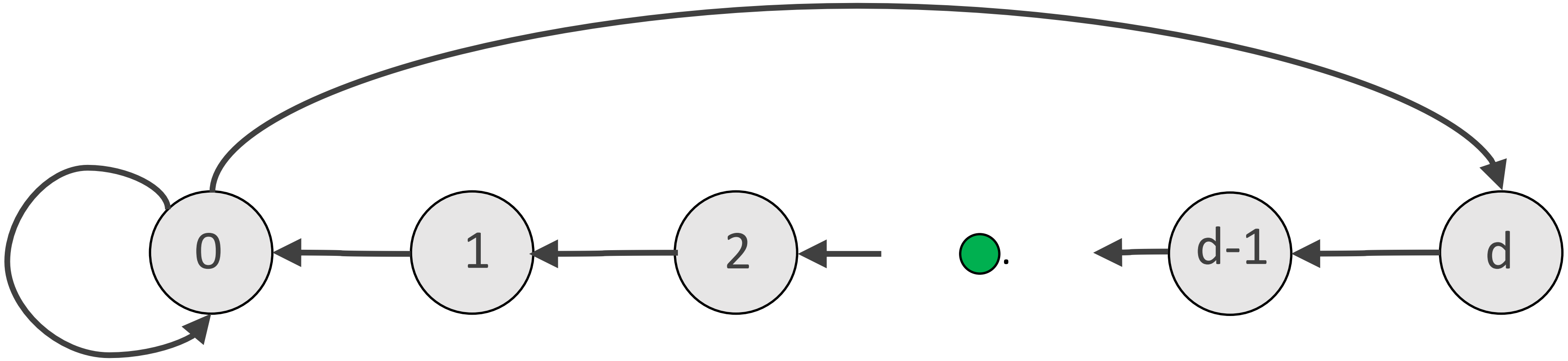
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

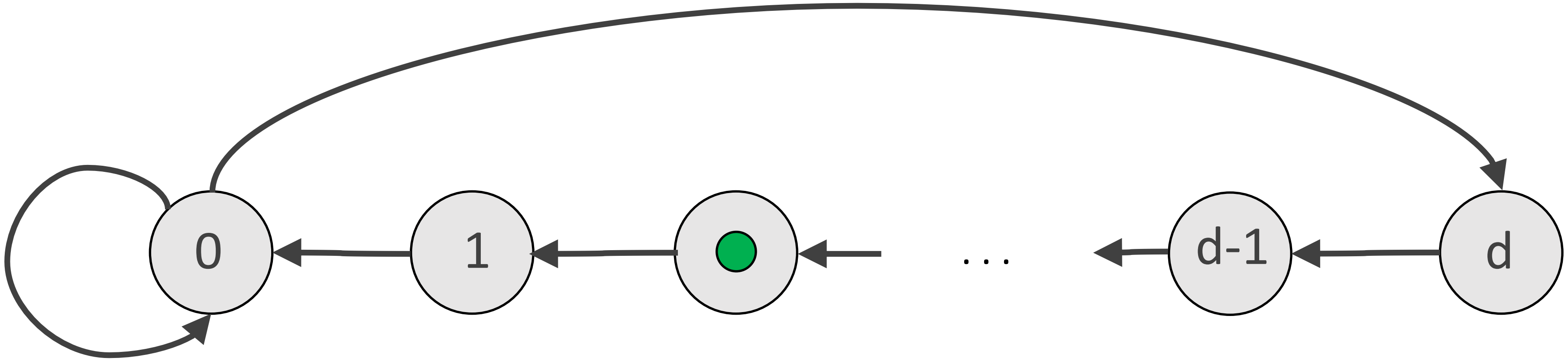
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

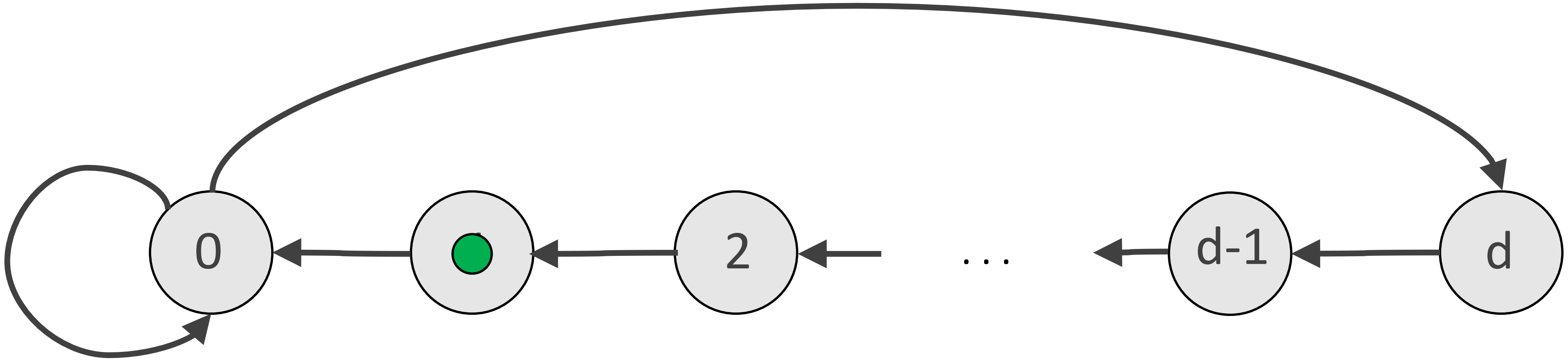
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

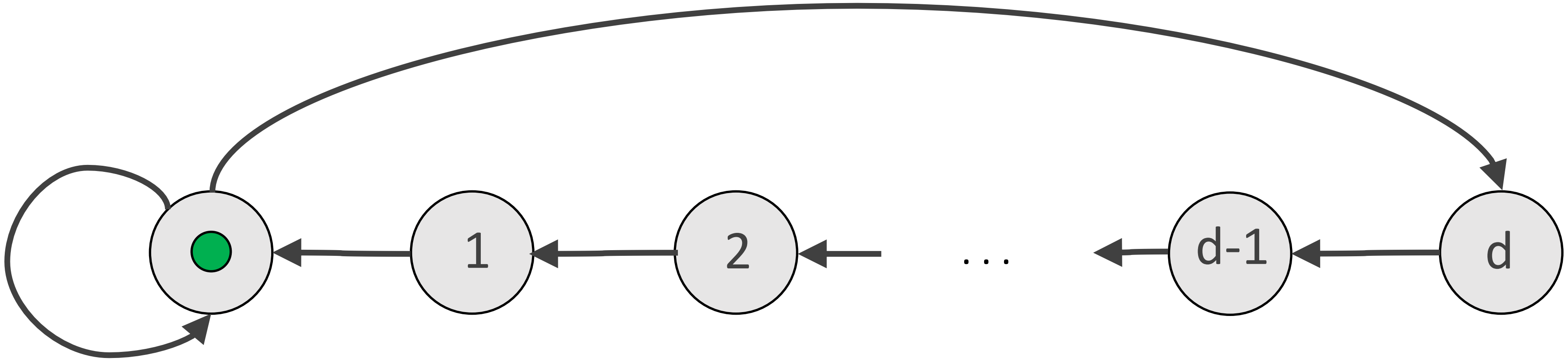
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

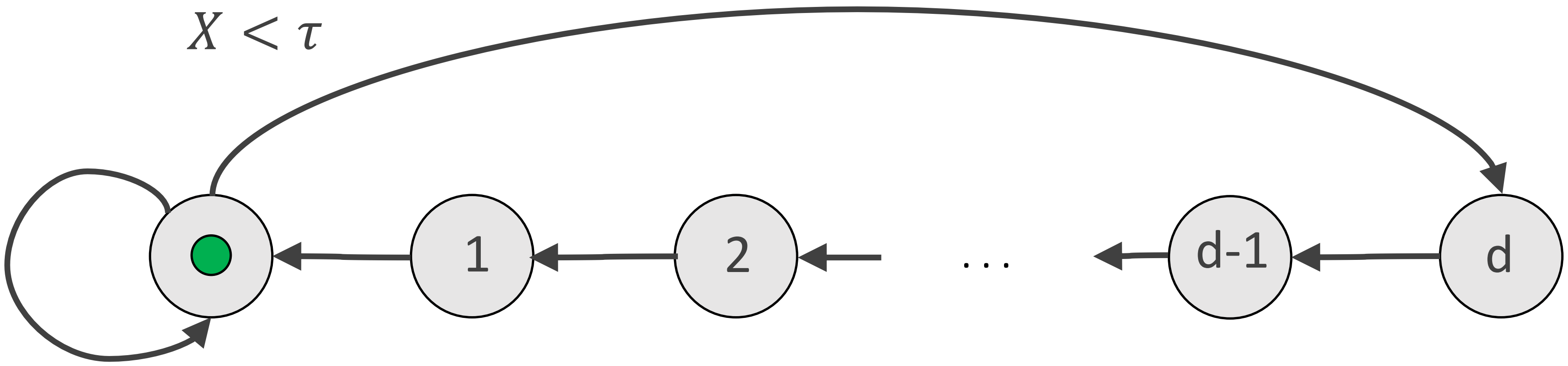
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

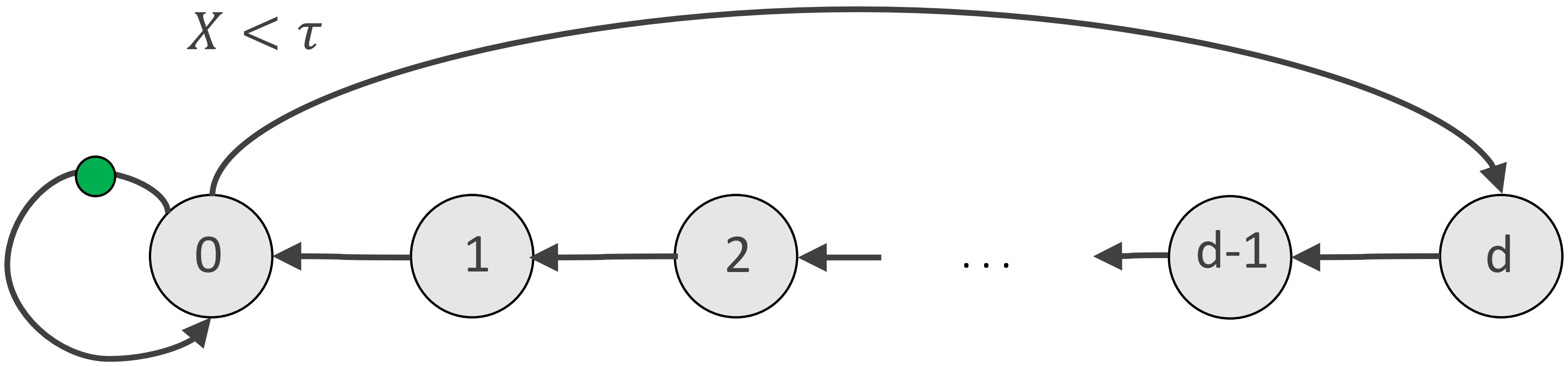
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

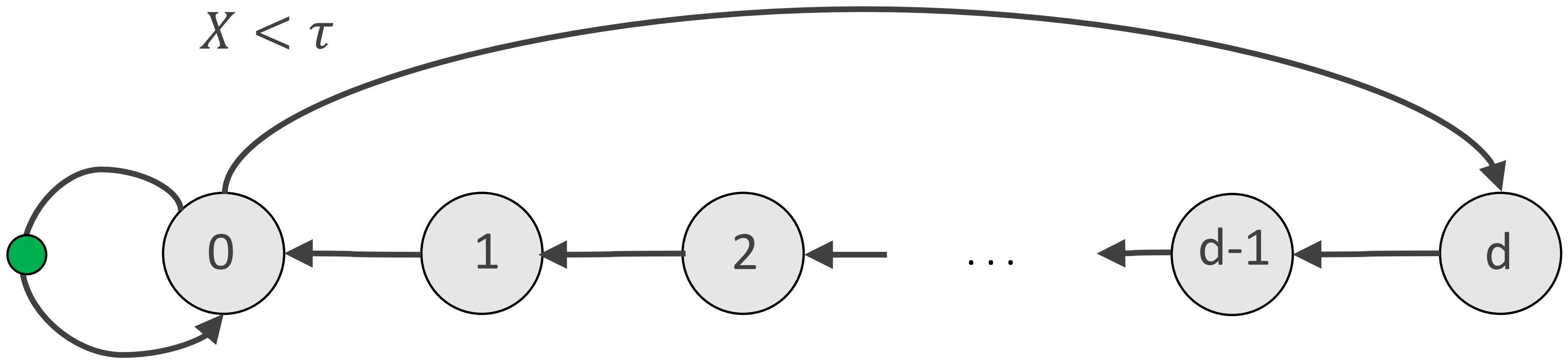
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

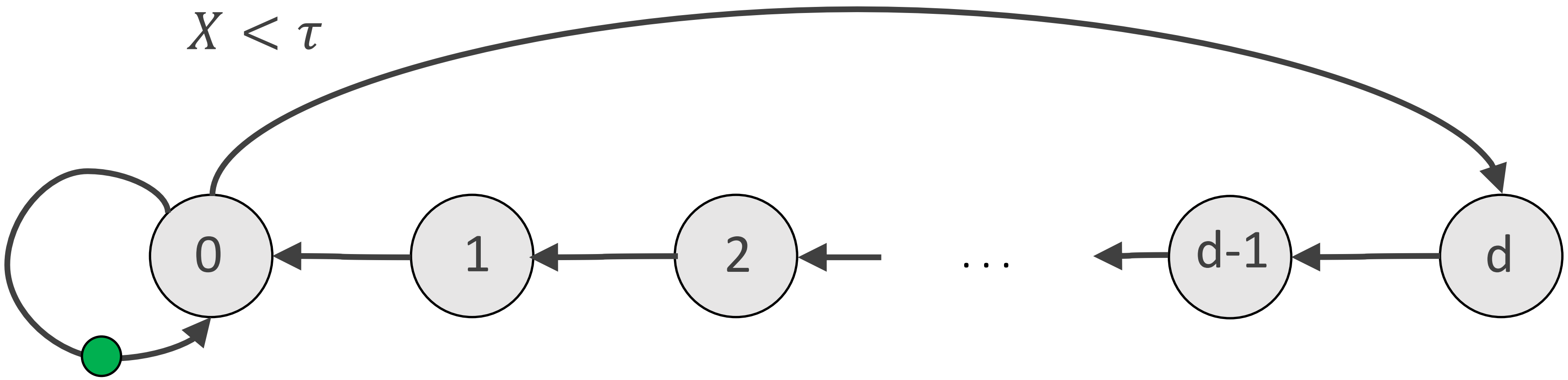
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

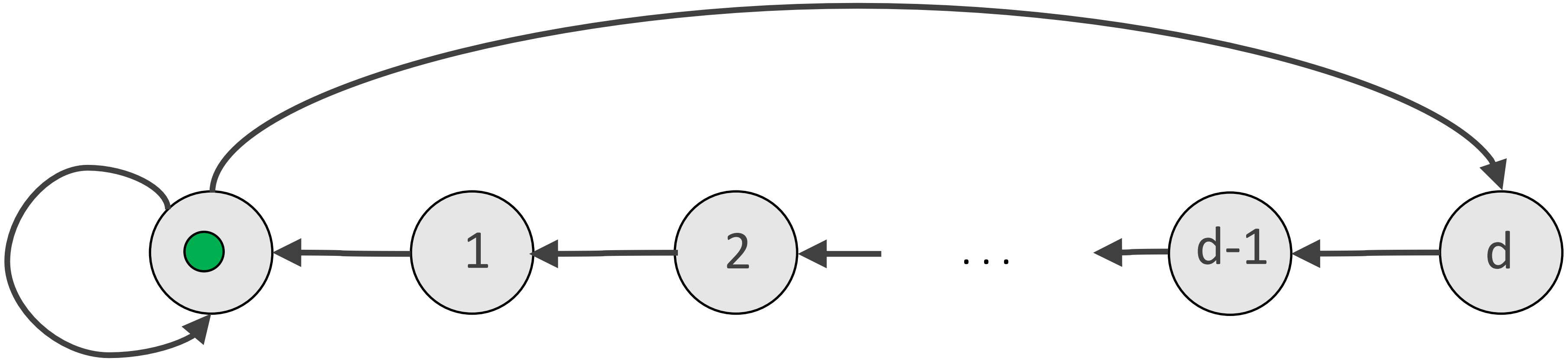
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

State: #rounds until available

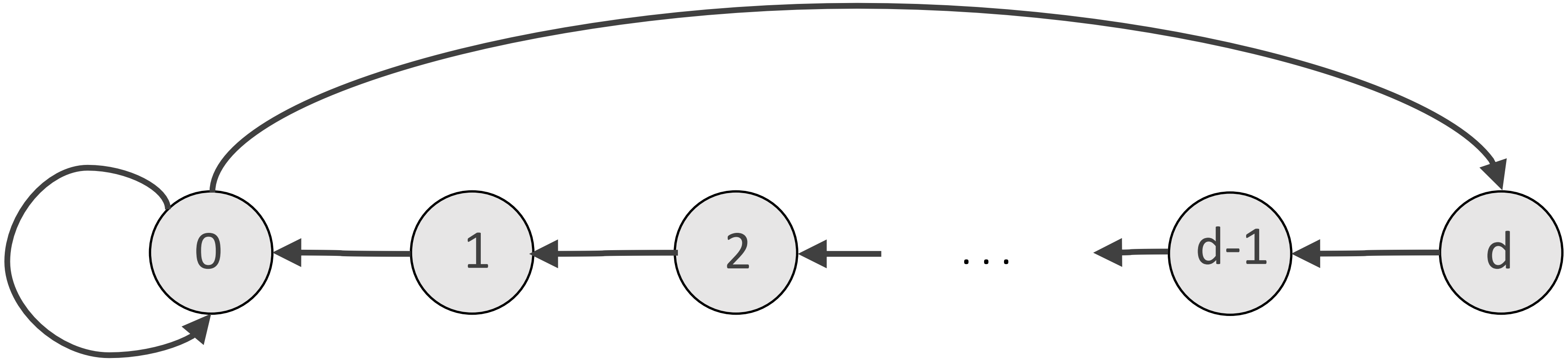


Known Reward Distribution

Markov Reward Process:

State: #rounds until available

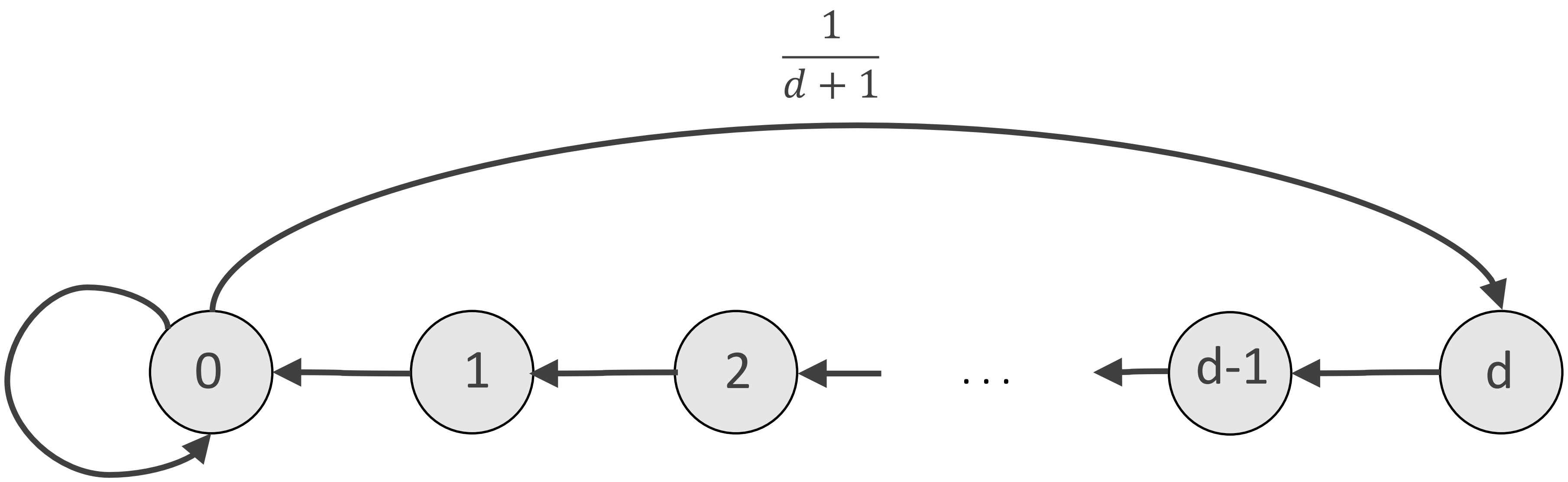
$$\textbf{Pr}[X \geq \tau] = 1 - F(\tau)$$
$$= 1 - F\left(F^{-1}\left(1 - \frac{1}{d+1}\right)\right) = \frac{1}{d+1}$$



Known Reward Distribution

Markov Reward Process:

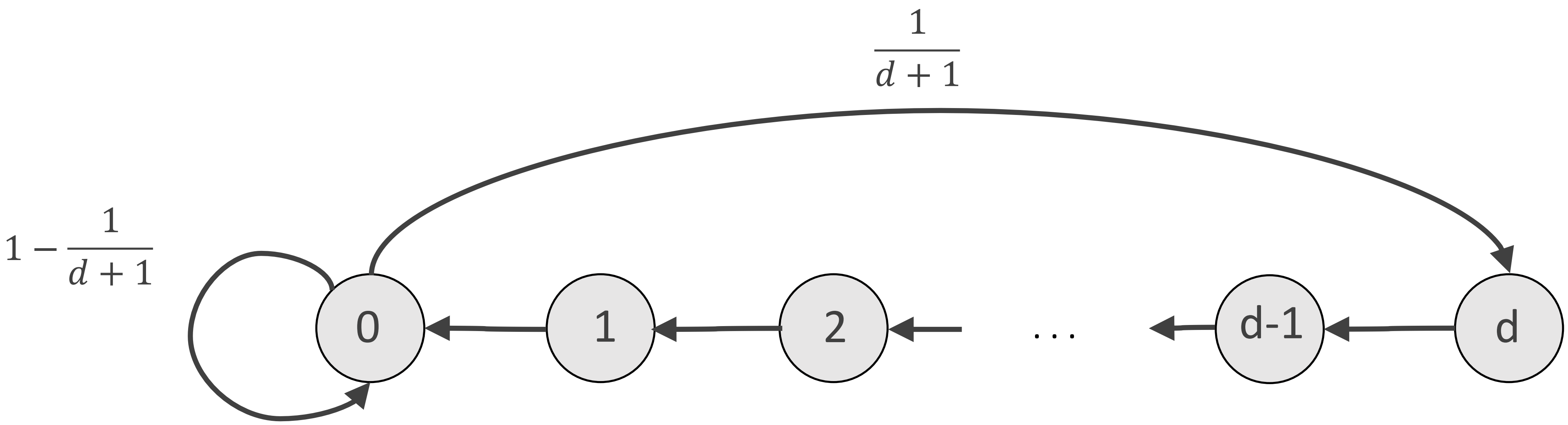
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

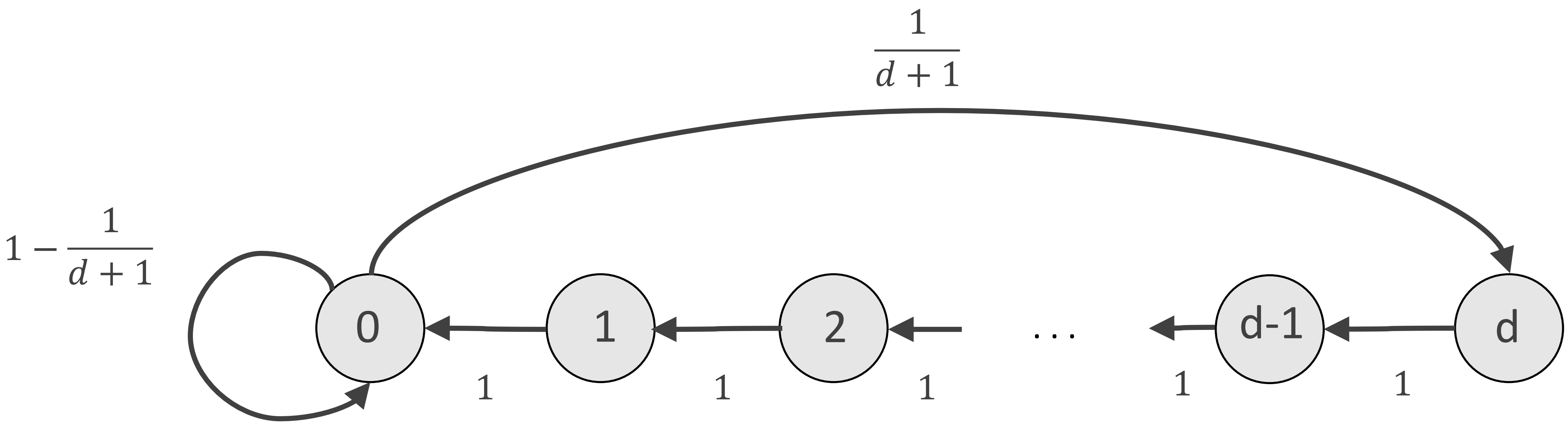
State: #rounds until available



Known Reward Distribution

Markov Reward Process:

State: #rounds until available

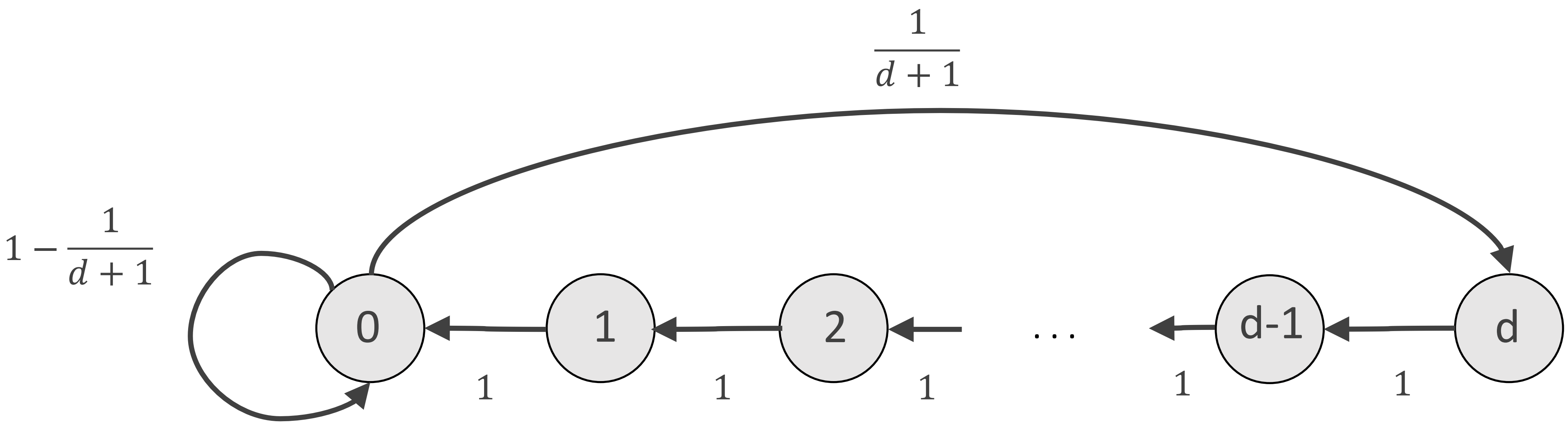


Known Reward Distribution

Markov Reward Process:

State: #rounds until available

Stationary Distribution: $\pi^*(0) = \frac{d+1}{2d+1} = \rho(d)$ and $\pi^*(\omega) = \frac{1}{2d+1}$ for each $\omega > 0$



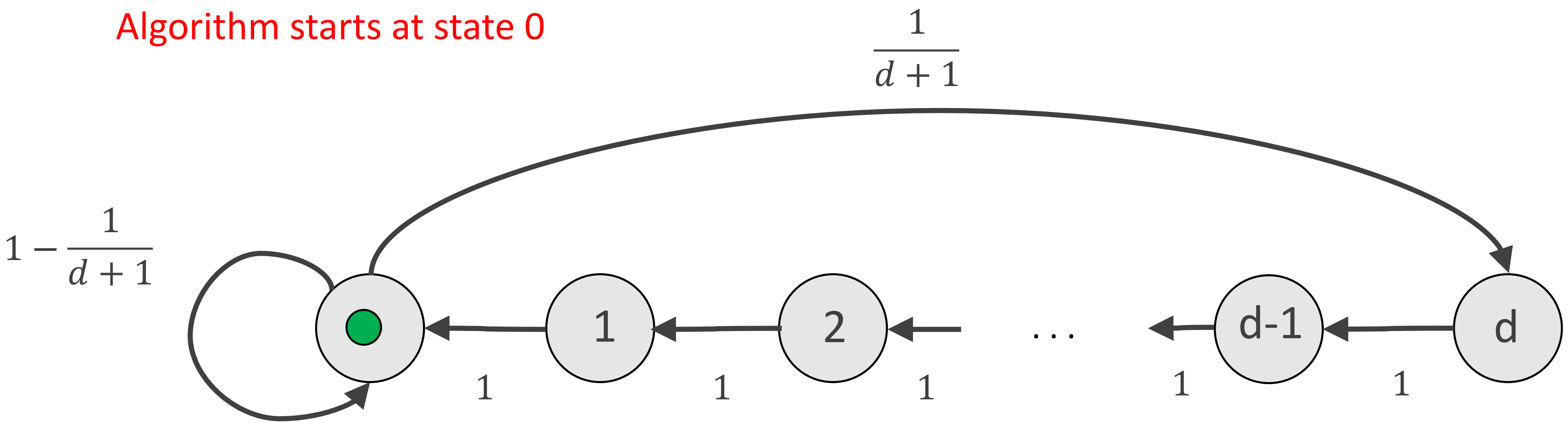
Known Reward Distribution

Markov Reward Process:

State: #rounds until available

Stationary Distribution: $\pi^*(0) = \frac{d+1}{2d+1} = \rho(d)$ and $\pi^*(\omega) = \frac{1}{2d+1}$ for each $\omega > 0$

Algorithm starts at state 0



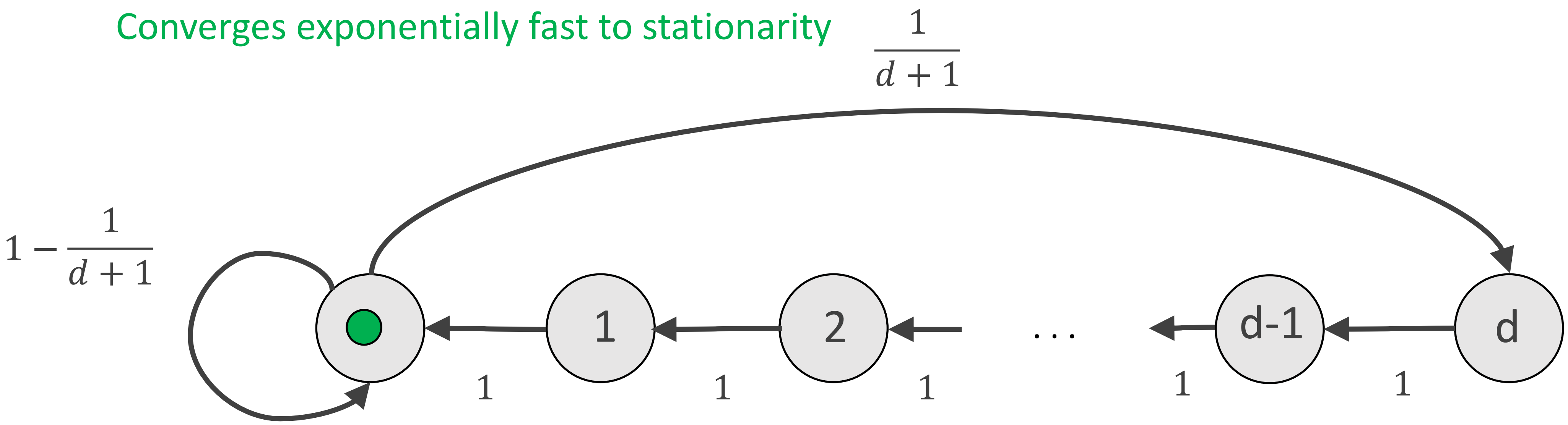
Known Reward Distribution

Markov Reward Process:

State: #rounds until available

Stationary Distribution: $\pi^*(0) = \frac{d+1}{2d+1} = \rho(d)$ and $\pi^*(\omega) = \frac{1}{2d+1}$ for each $\omega > 0$

Converges exponentially fast to stationarity



Known Reward Distribution

At any round t, Algorithm 1 collects in expectation:

$$\begin{aligned}\mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \text{ is collected}\right\}\right] &= \mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \geq \tau \text{ and } \mathsf{free}_{\mathcal{A}}(t)\right\}\right] \\ &= \mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \geq \tau\right\}\right] \cdot \mathbf{Pr}\left[\mathsf{free}_{\mathcal{A}}(t)\right]\end{aligned}$$

- Probability of being available: $\mathbf{Pr}\left[\mathsf{free}_{\mathcal{A}}(t)\right] = \frac{d+1}{2d+1} \approx \frac{1}{2}$ (up to vanishing in t additive error)

Known Reward Distribution

At any round t, Algorithm 1 collects in expectation:

$$\begin{aligned}\mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \text{ is collected}\right\}\right] &= \mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \geq \tau \text{ and } \mathsf{free}_{\mathcal{A}}(t)\right\}\right] \\ &= \mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \geq \tau\right\}\right] \cdot \mathbf{Pr}\left[\mathsf{free}_{\mathcal{A}}(t)\right]\end{aligned}$$

- Probability of being available: $\mathbf{Pr}\left[\mathsf{free}_{\mathcal{A}}(t)\right] = \frac{d+1}{2d+1} \approx \frac{1}{2}$ (up to vanishing in t additive error)
- Also, $\mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \geq \tau\right\}\right] = \int_{x=0}^{\infty} x \cdot q^*(x)dx$

Hence, $\mathbb{E}\left[X_t \cdot \mathbb{1}\left\{X_t \text{ is collected}\right\}\right] \geq \frac{OPT}{n} \cdot \frac{1}{2}$ (up to vanishing in t additive error)

Known Reward Distribution

By summing over n rounds, we get:

Theorem *Let* $\mathbb{E}[\mathsf{OPT}]$ *be the prophet's expected reward. For* $\rho(d) = \frac{d+1}{2d+1}$*, the expected reward of Algorithm 1 satisfies*

$$\mathbb{E}[\mathsf{ALG}] \geq \underbrace{\rho(d) \cdot \mathbb{E}[\mathsf{OPT}]}_{\text{competitive guarantee}} - \underbrace{\rho(d) \cdot (d+1) \cdot \mathbb{E}[X]}_{\text{loss due to LP upper bound}} - \underbrace{e \cdot d \cdot \mathbb{E}[X]}_{\text{loss due to mixing}} .$$

- We show that $\rho(d) \approx \frac{1}{2}$ is the best possible guarantee asymptotically

Unknown Reward Distribution

- Reward distribution is initially unknown
- Busy time d is known, but time horizon n is unknown
- Distribution has bounded support in [0,1]
- Rewards are <u>not observed</u> while blocked

Unknown Reward Distribution

Natural extension from the Bayesian case:

- Threshold-based algorithm as before
- At each round t, use as threshold

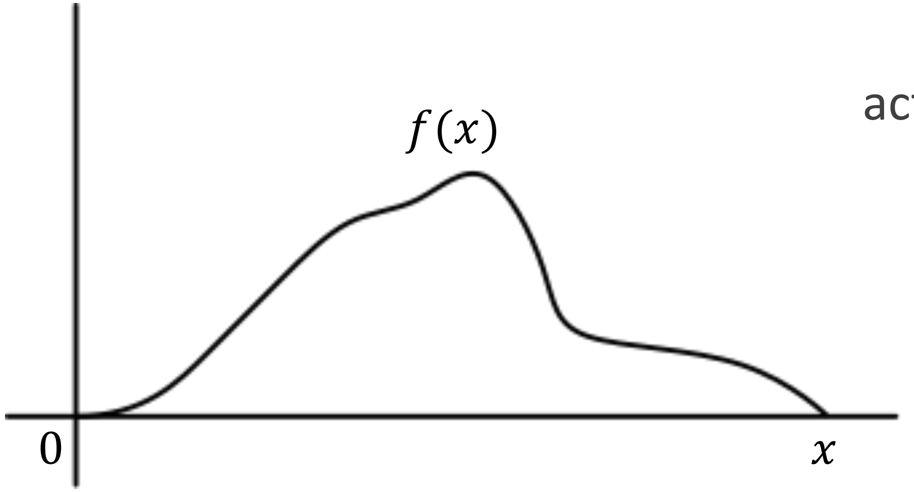
$\hat{\tau}(\mathrm{t})$: $(1 - \frac{1}{d+1})$-quantile of empirical distribution from past samples

Unknown Reward Distribution

Natural extension from the Bayesian case:

- Threshold-based algorithm as before
- At each round t, use as threshold

$\hat{\tau}(\mathrm{t})$: $(1-\frac{1}{d+1})$-quantile of empirical distribution from past samples



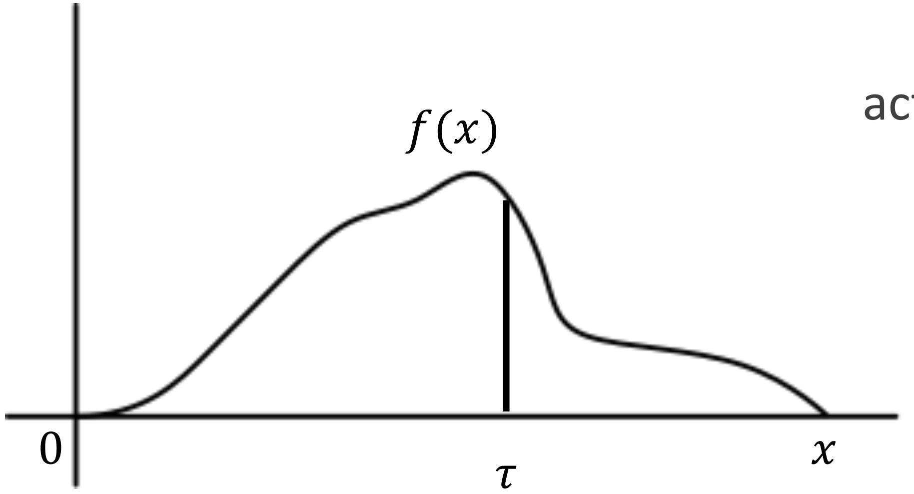
actual distribution

Unknown Reward Distribution

Natural extension from the Bayesian case:

- Threshold-based algorithm as before
- At each round t, use as threshold

$\hat{\tau}(t)$: $(1-\frac{1}{d+1})$-quantile of empirical distribution from past samples



actual distribution

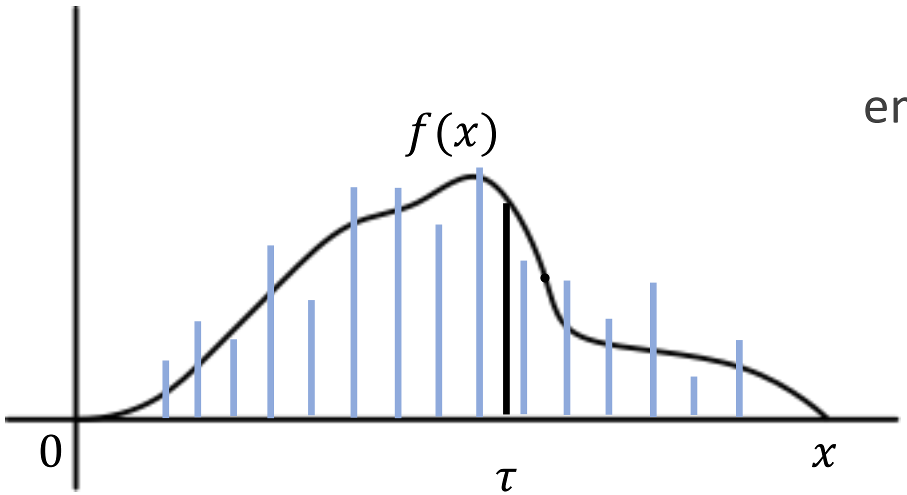
$$\tau = F^{-1}(1-\frac{1}{d+1})$$

Unknown Reward Distribution

Natural extension from the Bayesian case:

- Threshold-based algorithm as before
- At each round t, use as threshold

$\hat{\tau}(\mathrm{t})$: $(1 - \frac{1}{d+1})$-quantile of empirical distribution from past samples



empirical distribution

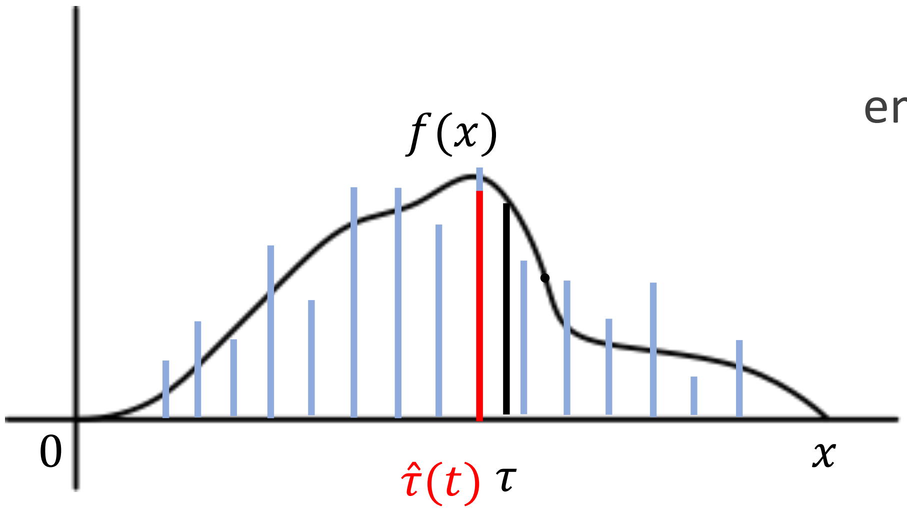
$$\tau = F^{-1}(1 - \frac{1}{d+1})$$

Unknown Reward Distribution

Natural extension from the Bayesian case:

- Threshold-based algorithm as before
- At each round t, use as threshold

$\hat{\tau}(\mathrm{t})$: $(1 - \frac{1}{d+1})$-quantile of empirical distribution from past samples



empirical distribution

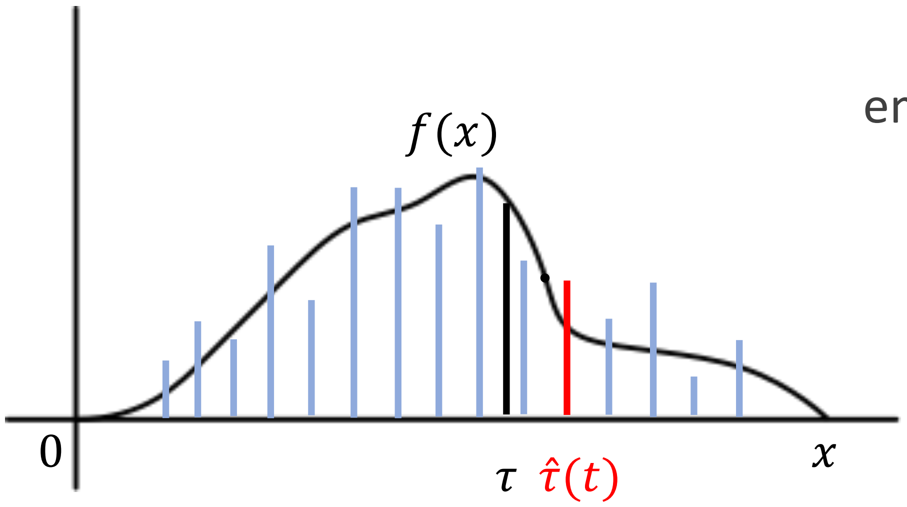
$$\tau = F^{-1}(1 - \frac{1}{d+1})$$

Unknown Reward Distribution

Natural extension from the Bayesian case:

- Threshold-based algorithm as before
- At each round t, use as threshold

$$\hat{\tau}(\mathrm{t}):\ (1-\frac{1}{d+1})\text{-quantile of empirical distribution from past samples}$$



empirical distribution

$$\tau = F^{-1}(1-\frac{1}{d+1})$$

Unknown Reward Distribution

Goal: Bound the regret against the Bayesian policy for n rounds

$$\text{Regret}(n) = \mathbb{E}[A(n)] - \mathbb{E}[L(n)]$$

$A(n)$: reward of (Bayesian) Algorithm for n rounds
$L(n)$: reward of Learning algorithm for n rounds

Unknown Reward Distribution

Step 1: Reducing regret to estimation error of each round

- Compensated coupling technique [Vera & Banerjee, 2020]
- Define (fictitious) policy P_i, which
  - Follows the decisions of L for the first i steps (including i)
  - Follows the decisions of A for rounds $i + 1$ to n
  - Note $A \equiv P_0$ and $L \equiv P_n$

Unknown Reward Distribution

Step 1: Reducing regret to estimation error of each round

$$\text{Regret}(n) = \mathbb{E}[A(n)] - \mathbb{E}[L(n)] = \sum_{i=1}^{n} \mathbb{E}\left[P_{i-1}(n) - P_i(n)\right]$$

- Since rewards are in [0,1] and d is fixed, for any round i:

$$\mathbb{E}[P_{i-1}(n) - P_i(n)] \leq \mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$$

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Fact:** At round i the learning algorithm collects at least $\approx i/(d+1)$ samples.

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Fact:** At round i the learning algorithm collects at least $\approx i/(d+1)$ samples.
- Using standard concentration results (Dvoretzky-Kiefer-Wolfowitz inq.), we show that

$$\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)] \lesssim \sqrt{\frac{d}{i} \log(i)}$$

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Fact:** At round i the learning algorithm collects at least $\approx i/(d+1)$ samples.
- Using standard concentration results (Dvoretzky-Kiefer-Wolfowitz inq.), we show that

$$\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)] \lesssim \sqrt{\frac{d}{i} \log(i)}$$

and, hence,

$$\text{Regret}(n) \lesssim \sqrt{n\, d \log(\text{n})}$$

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Fact:** At round i the learning algorithm collects at least $\approx i/(d+1)$ samples.
- Using standard concentration results (Dvoretzky-Kiefer-Wolfowitz inq.), we show that

$$\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)] \lesssim \sqrt{\frac{d}{i} \log(i)}$$

and, hence,

$$\text{Regret}(n) \lesssim \sqrt{n\, d \log(\text{n})}$$

Dependence on d is not necessary

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Fact:** At round i the learning algorithm collects at least $\approx i/(d+1)$ samples.
- Using standard concentration results (Dvoretzky-Kiefer-Wolfowitz inq.), we show that

$$\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)] \lesssim \sqrt{\frac{d}{i} \log(i)}$$

and, hence,

$$\mathrm{Regret}(n) \lesssim \sqrt{n\, d \log(\mathrm{n})}$$

Collecting $\approx i/(d+1)$ samples by round i is very pessimistic

Unknown Reward Distribution

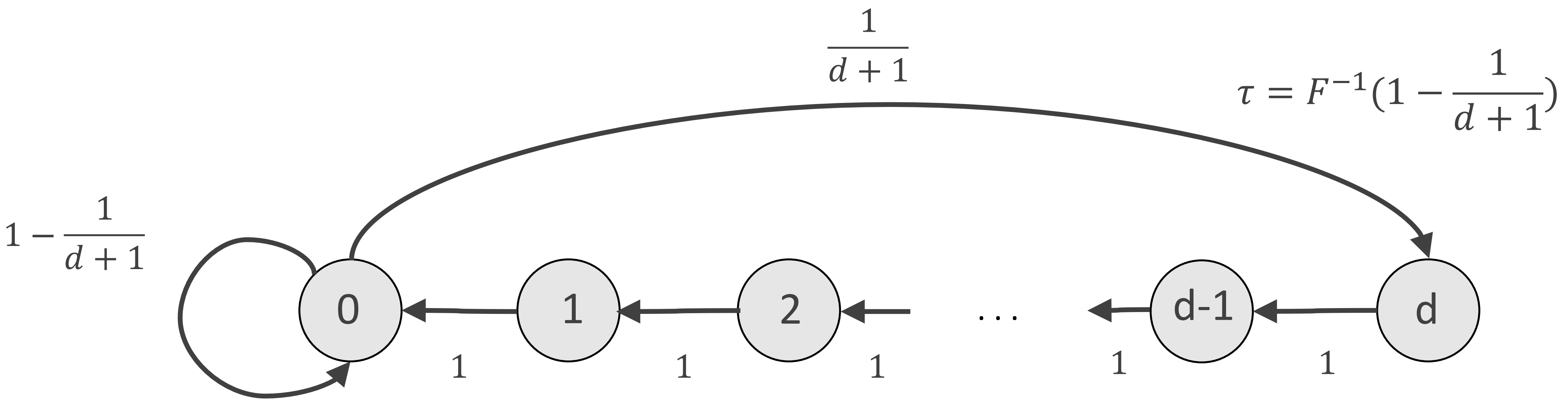
Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- Recall that for algorithm A (Bayesian), the resource is available $\approx 1/2$-fraction of rounds (in expectation)

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

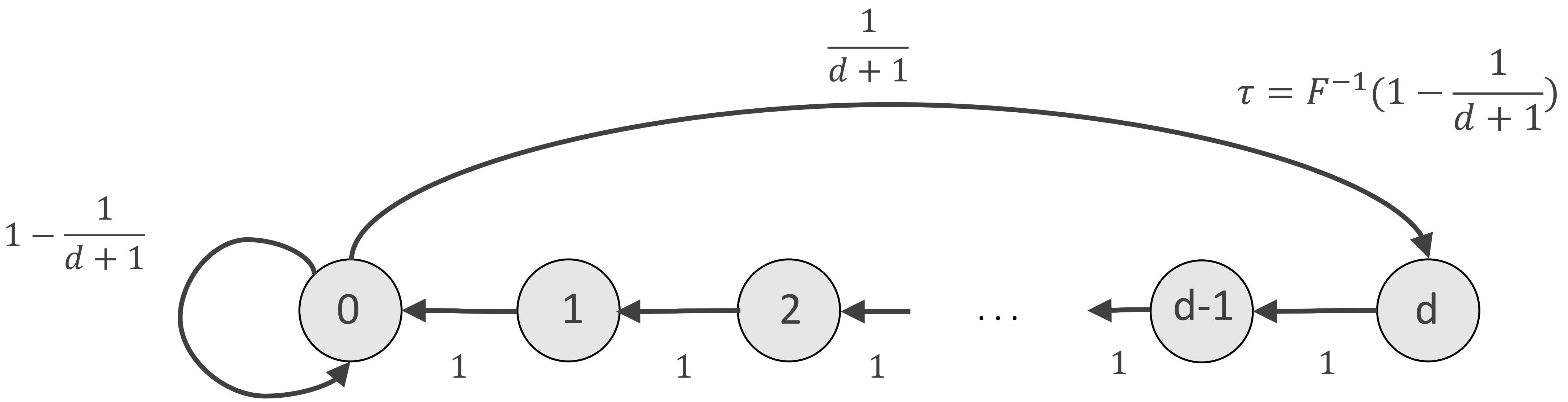
- Recall that for algorithm A (Bayesian), the resource is available $\approx 1/2$-fraction of rounds (in expectation)



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

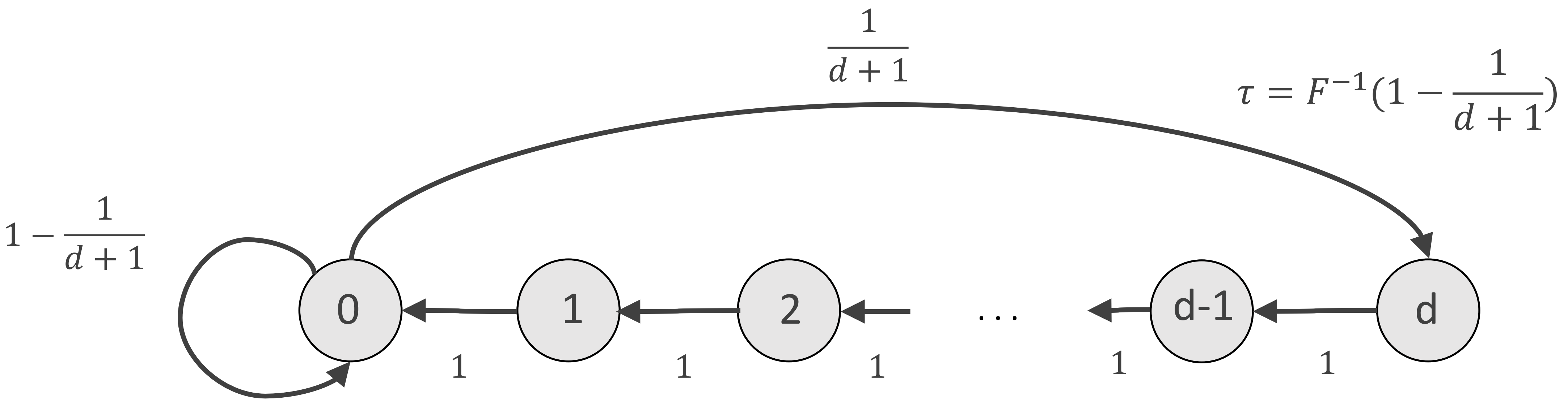
- Recall that for algorithm A (Bayesian), the resource is available $\approx 1/2$-fraction of rounds (in expectation)
- This holds even if we slightly perturb the threshold



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

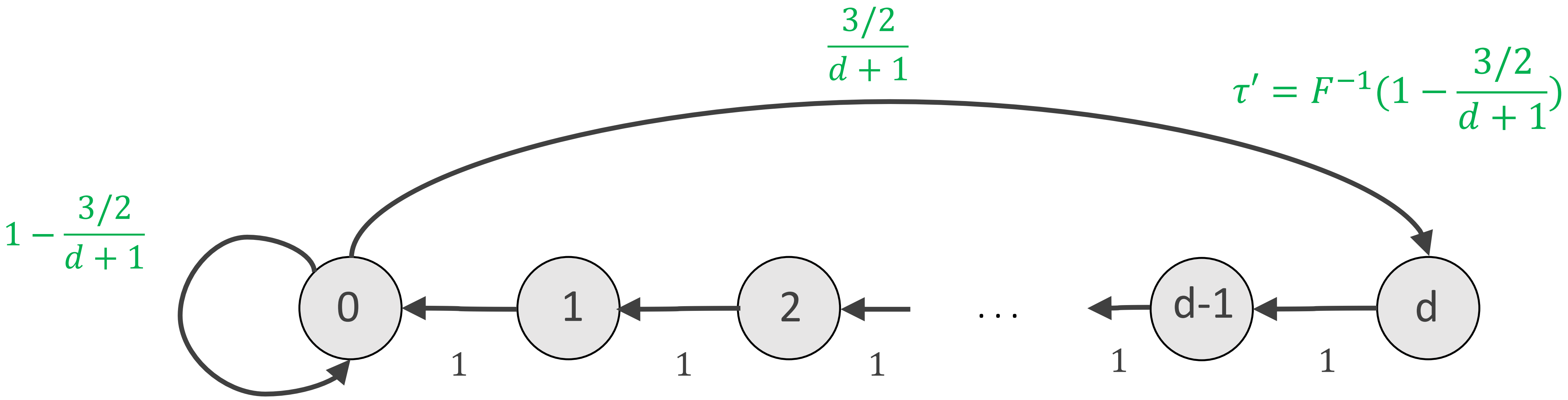
- Recall that for algorithm A (Bayesian), the resource is available $\approx 1/2$-fraction of rounds (in expectation)
- Let B be an **eager** version of algorithm A, with threshold $\tau' = F^{-1}(1 - \frac{3/2}{d+1})$



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- Recall that for algorithm A (Bayesian), the resource is available $\approx 1/2$-fraction of rounds (in expectation)
- Let B be an **eager** version of algorithm A, with threshold $\tau' = F^{-1}(1 - \frac{3/2}{d+1})$



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- Recall that for algorithm A (Bayesian), the resource is available $\approx 1/2$-fraction of rounds (in expectation)
- Let B be an **eager** version of algorithm A, with threshold $\tau' = F^{-1}(1 - \frac{3/2}{d+1})$

Properties:

- B has smaller threshold than A

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- Recall that for algorithm A (Bayesian), the resource is available $\approx 1/2$-fraction of rounds (in expectation)
- Let B be an **eager** version of algorithm A, with threshold $\tau' = F^{-1}(1 - \frac{3/2}{d+1})$

Properties:

- B has smaller threshold than A
- Thus, B observes less samples than A in expectation

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- Recall that for algorithm A (Bayesian), the resource is available $\approx 1/2$-fraction of rounds (in expectation)
- Let B be an **eager** version of algorithm A, with threshold $\tau' = F^{-1}(1 - \frac{3/2}{d+1})$

Properties:

- B has smaller threshold than A
- Thus, B observes less samples than A in expectation
- Still, B observes $O(i)$ (independent of d) number of samples by round i w.h.p.

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.

Unknown Reward Distribution

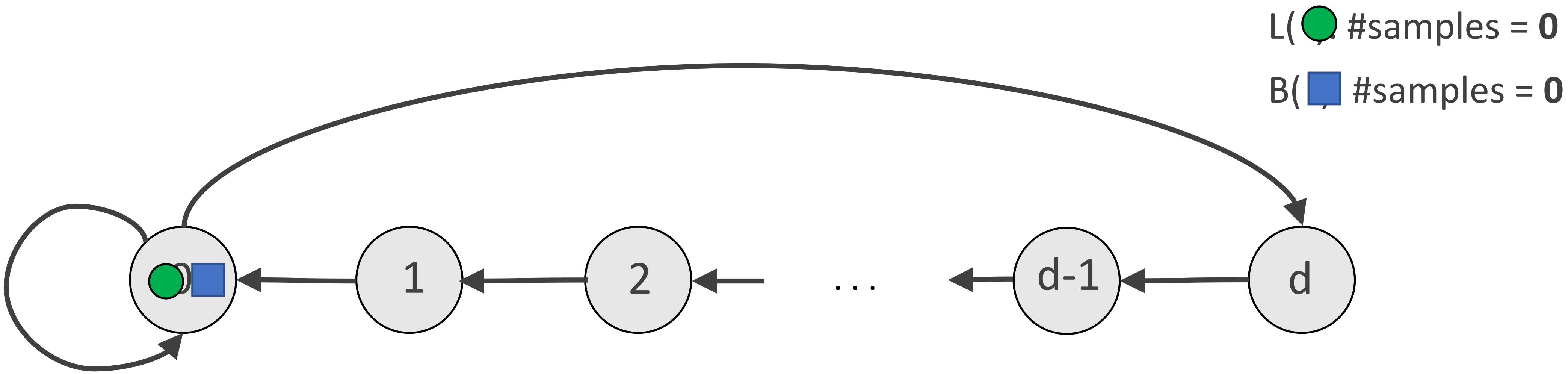
Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3\log(\mathrm{n}))$ w.h.p.
- Thus, L also observes $O(i)$ (independent of d) number of samples by round i w.h.p.

Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.

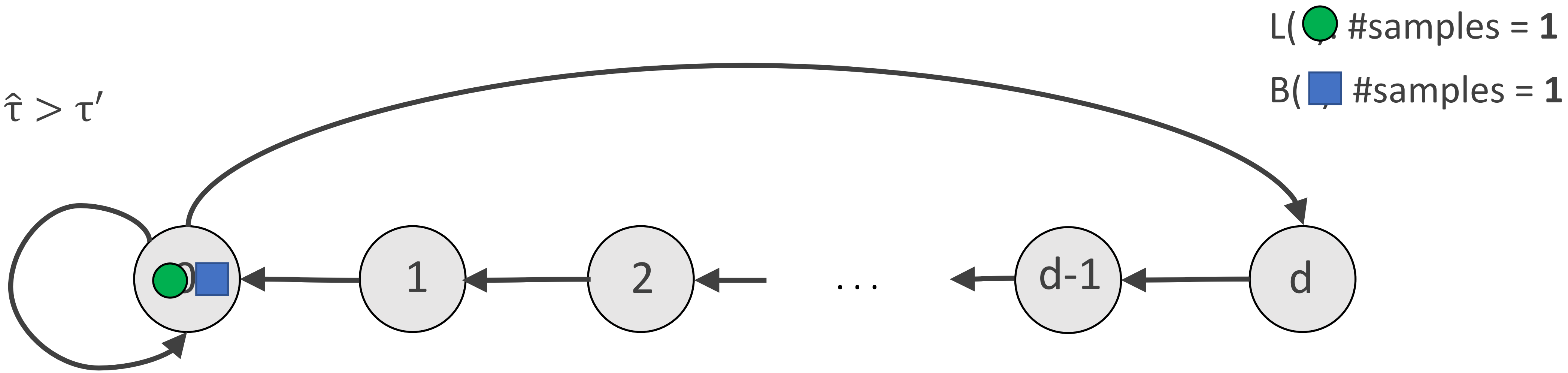


Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3\log(\mathrm{n}))$ w.h.p.

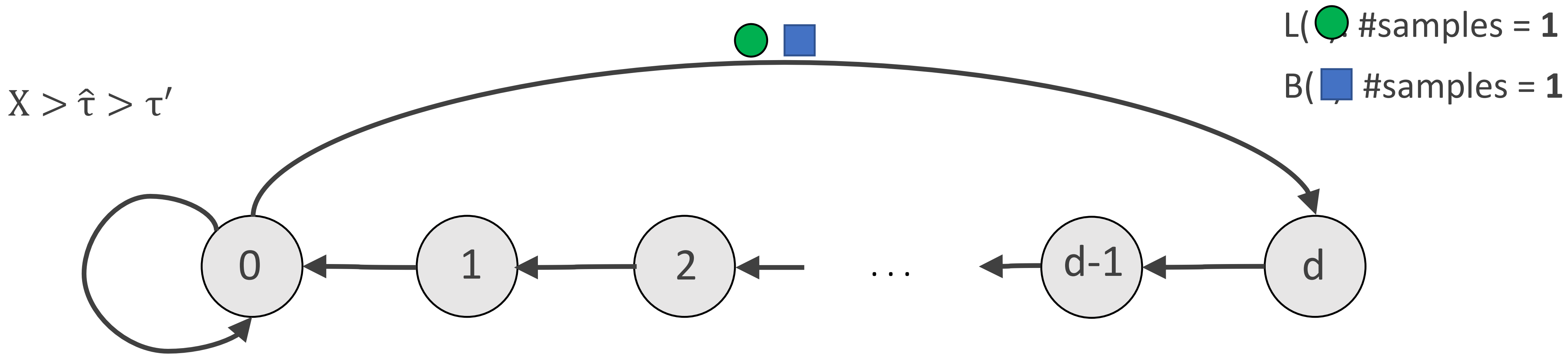
$\mathrm{X} > \hat{\tau} > \tau'$



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

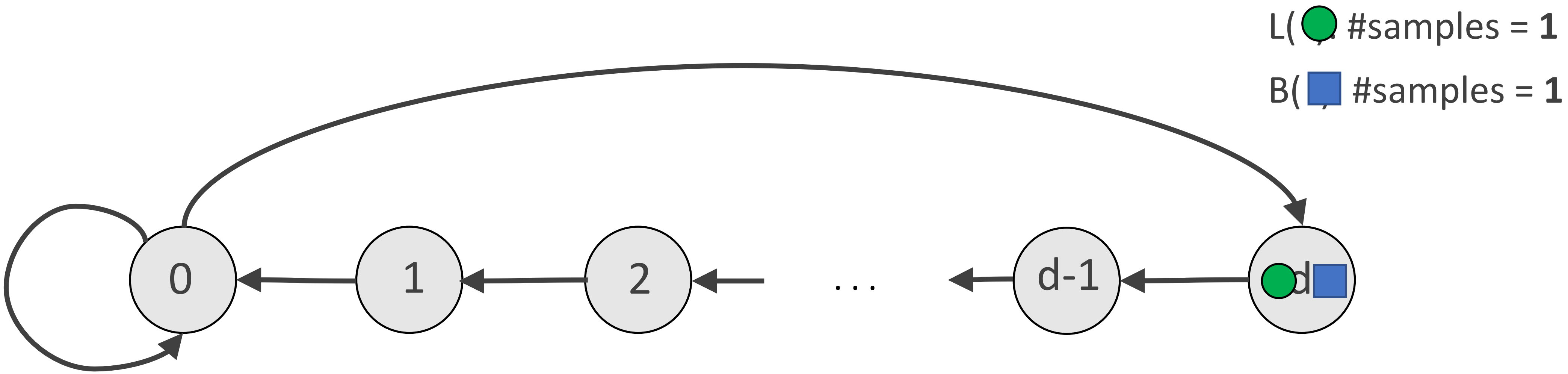
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

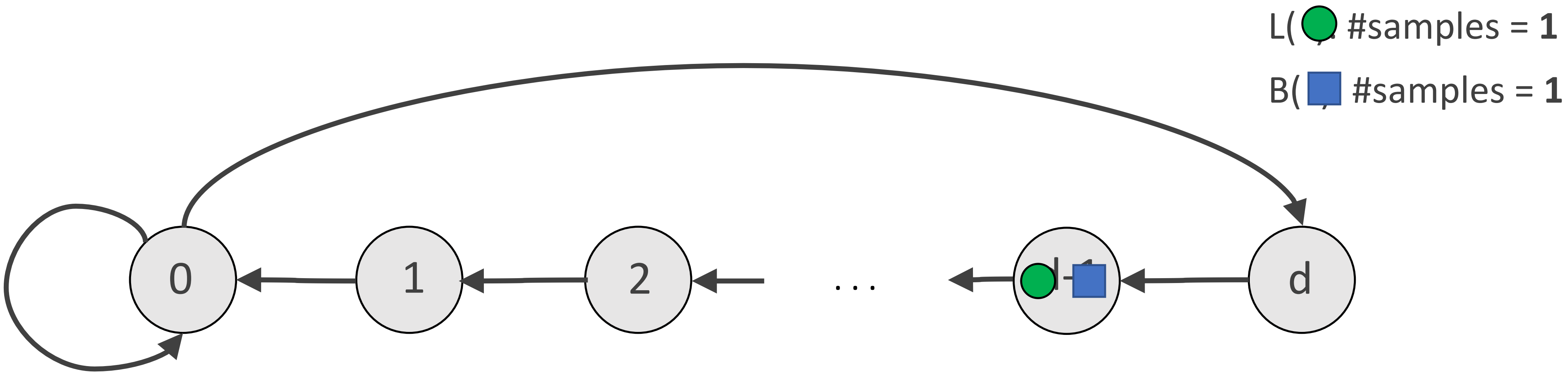
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3\log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

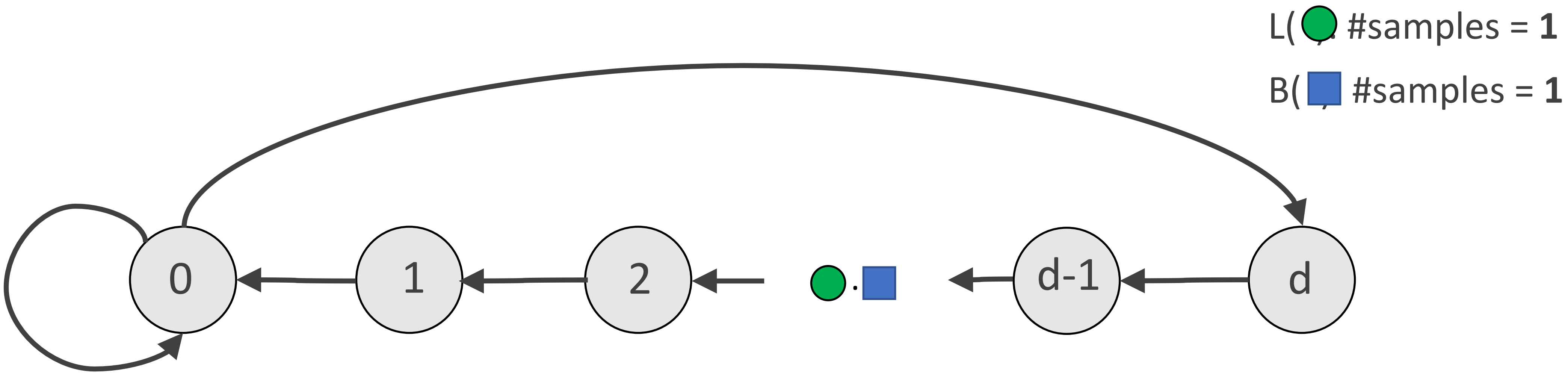
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

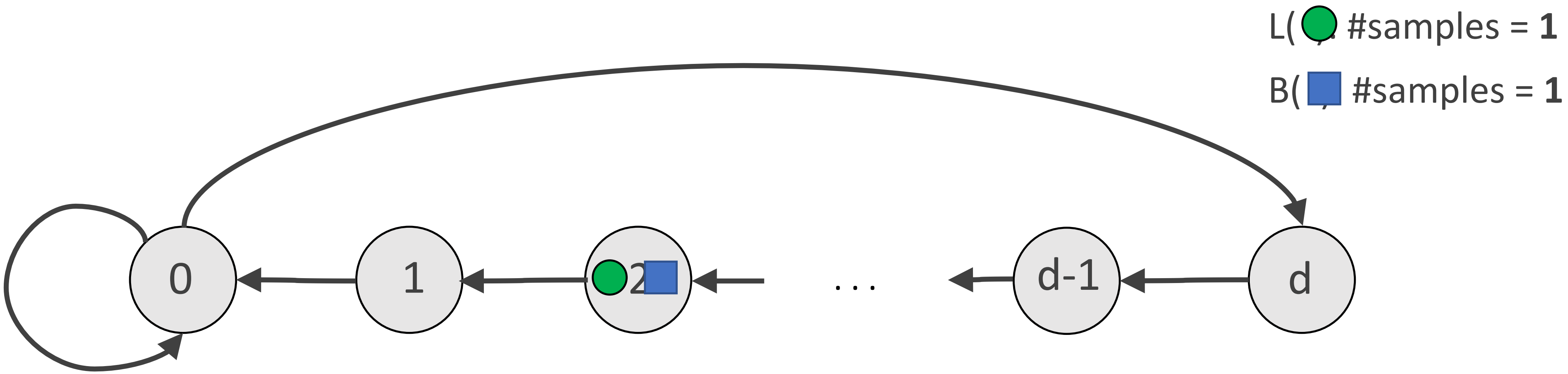
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3\log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

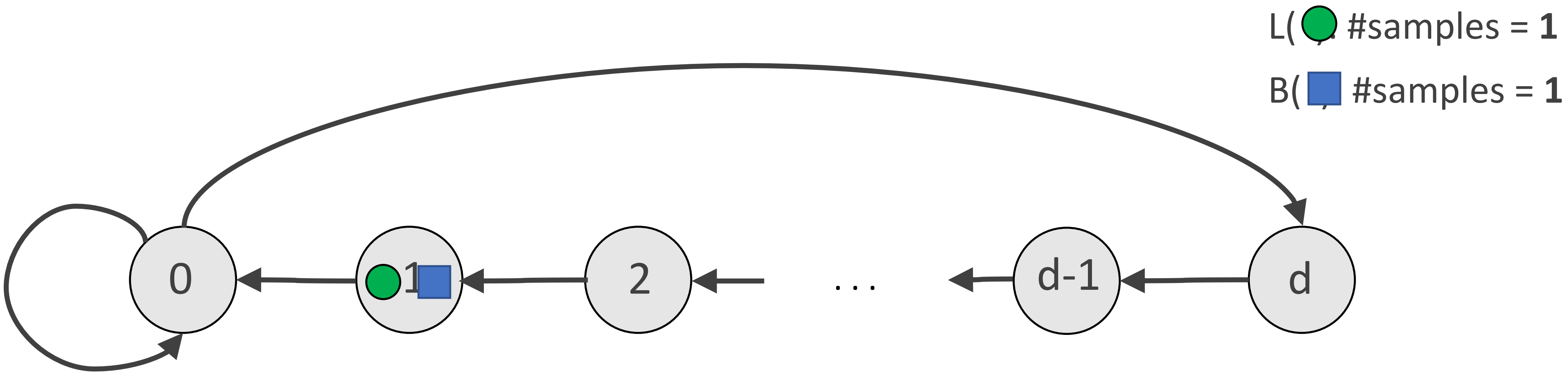
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

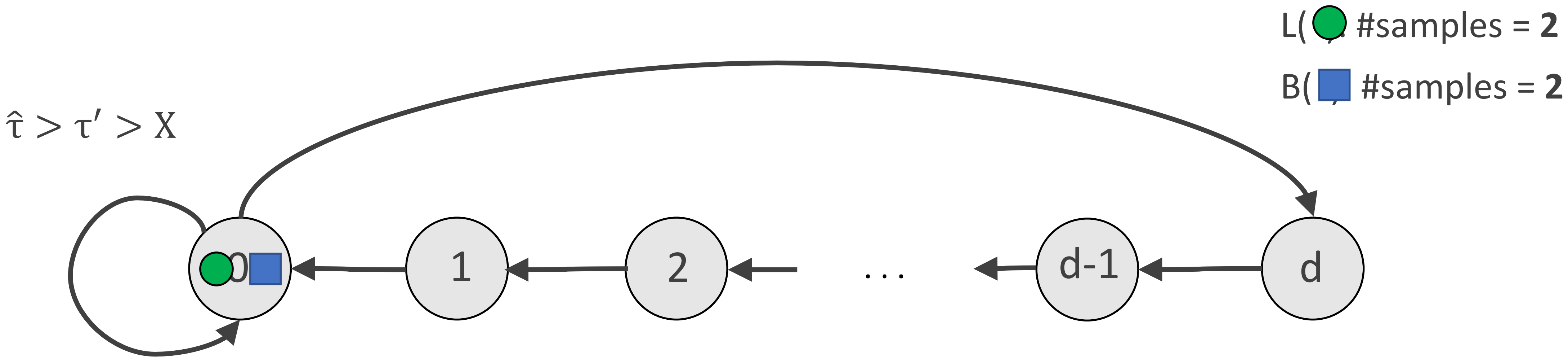
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

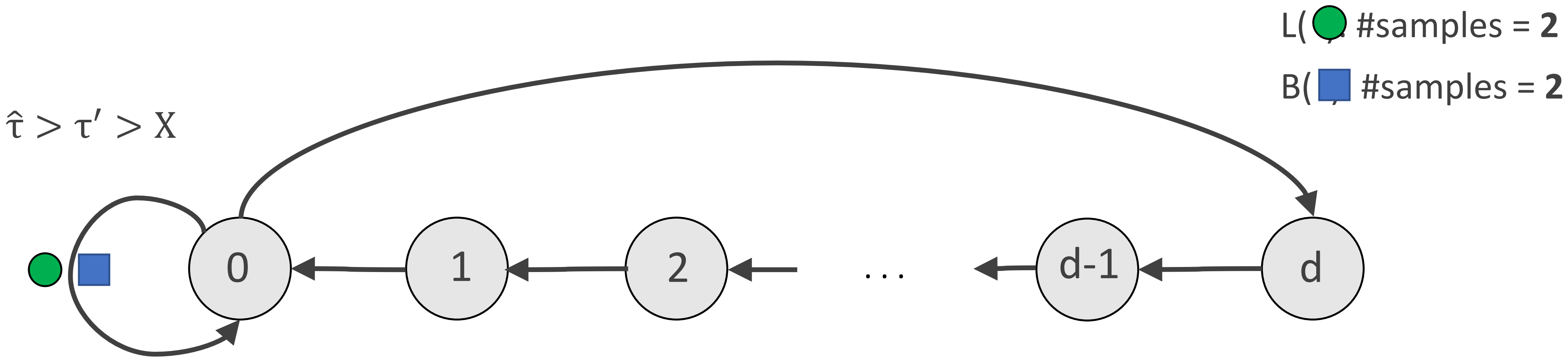
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

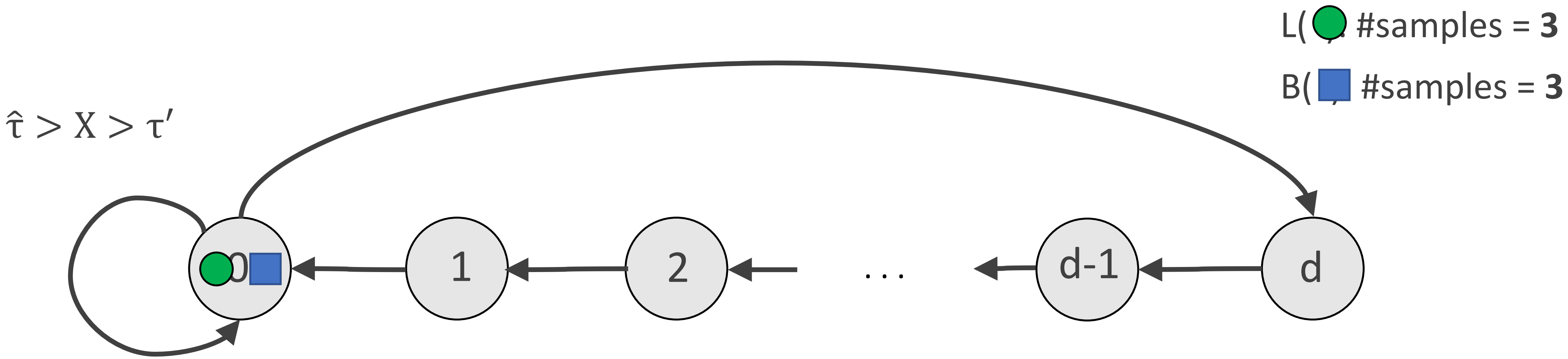
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3\log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

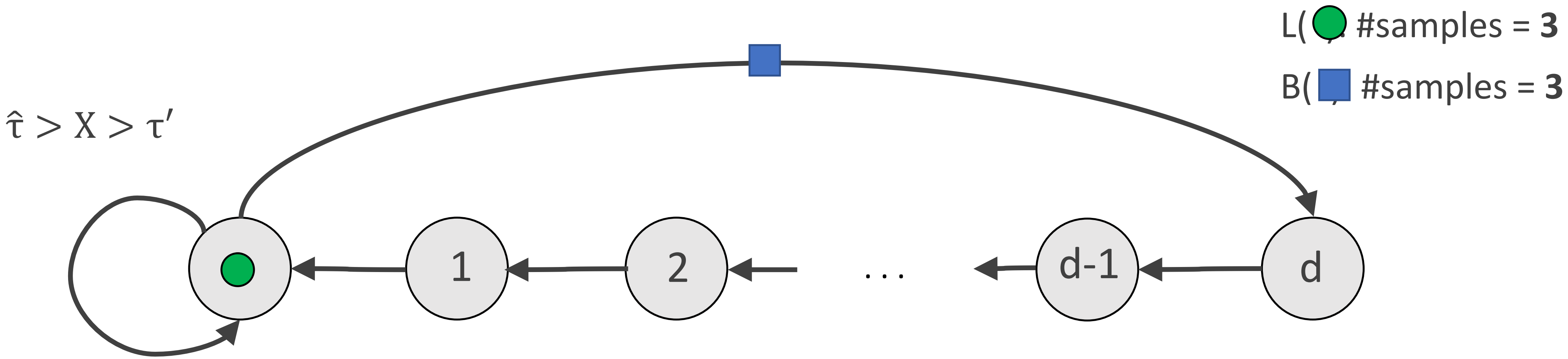
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3\log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

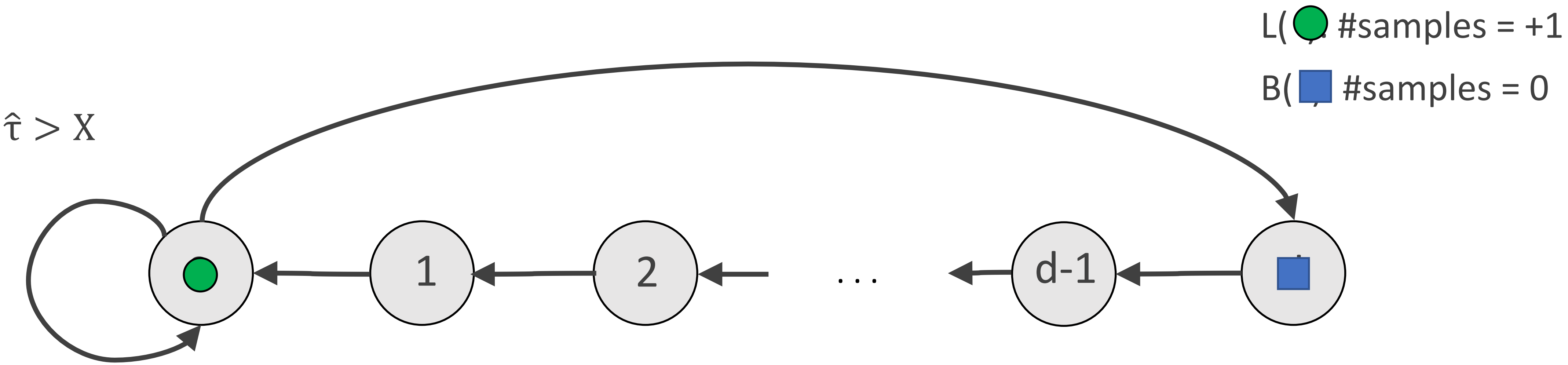
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3\log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

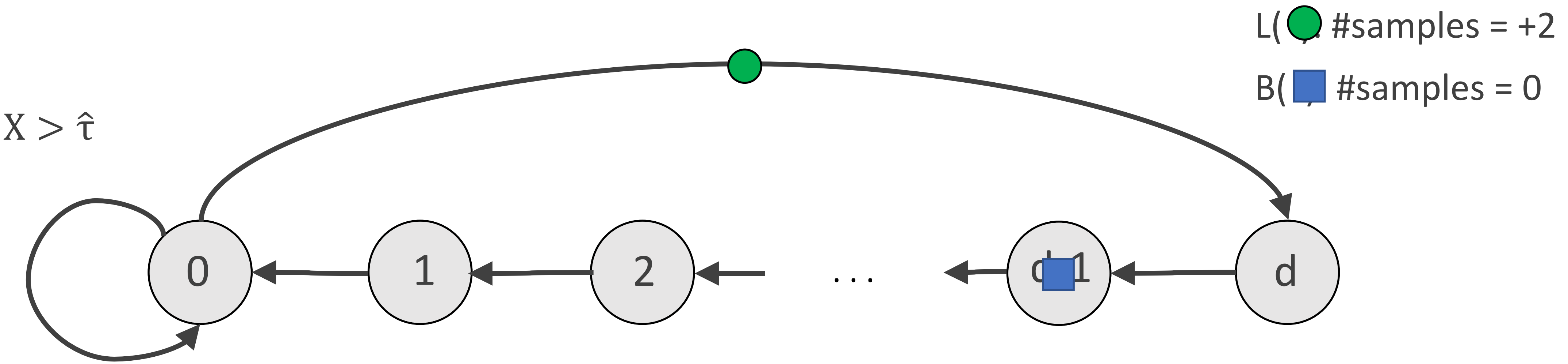
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3\log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

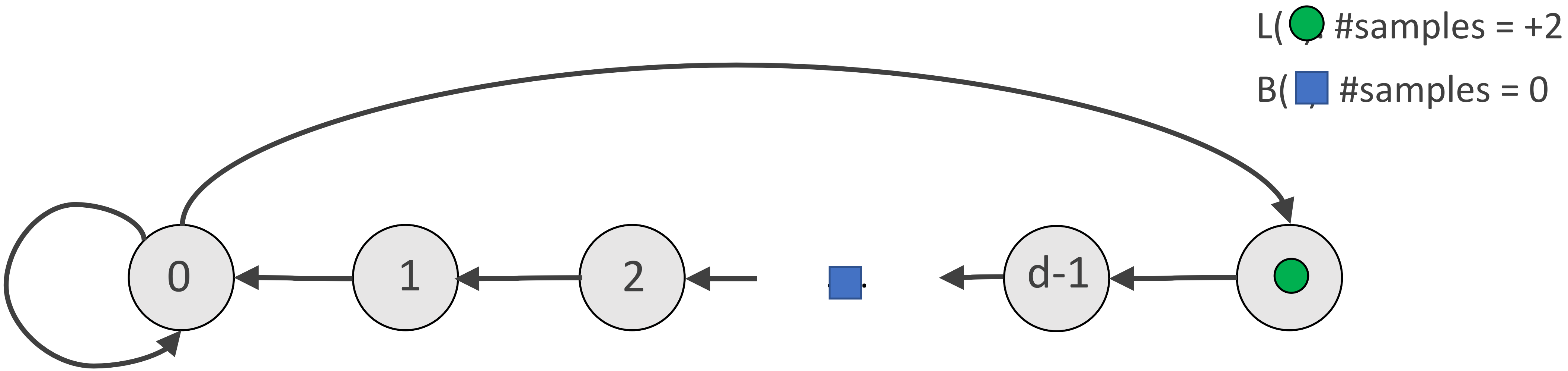
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

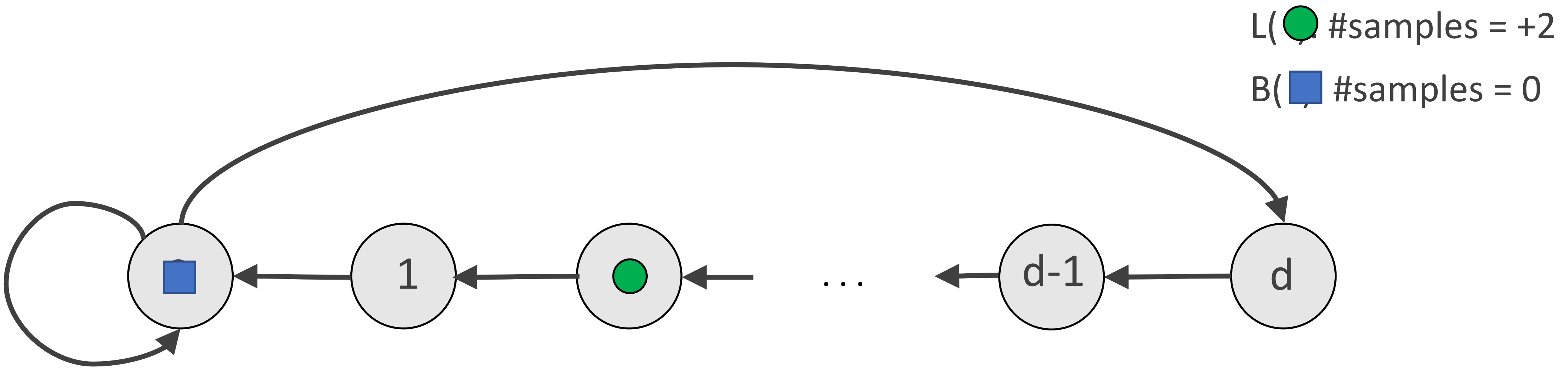
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3\log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3\log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

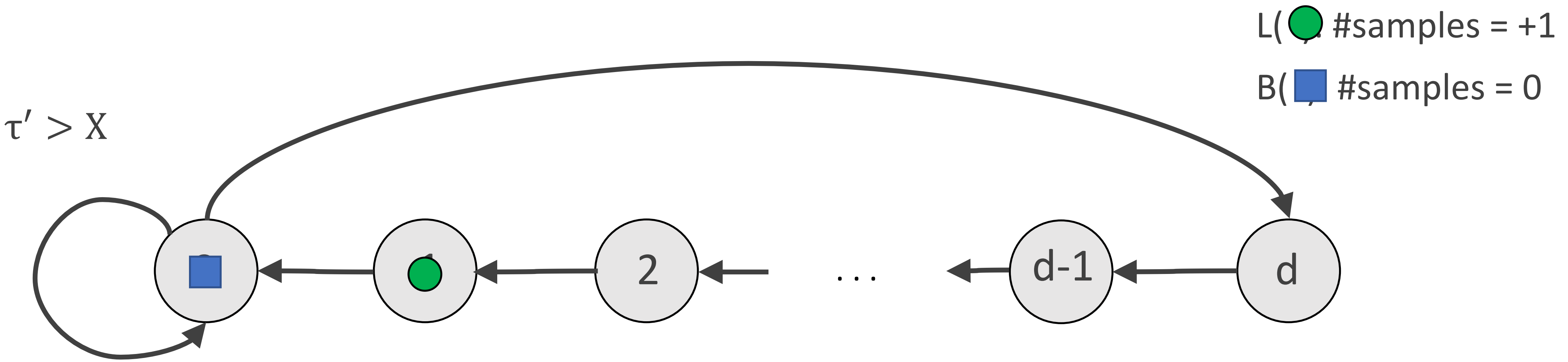
- **Key-insight:** After $\mathrm{O(d^3 log(n))}$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O(d^3 log(n))}$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(n))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

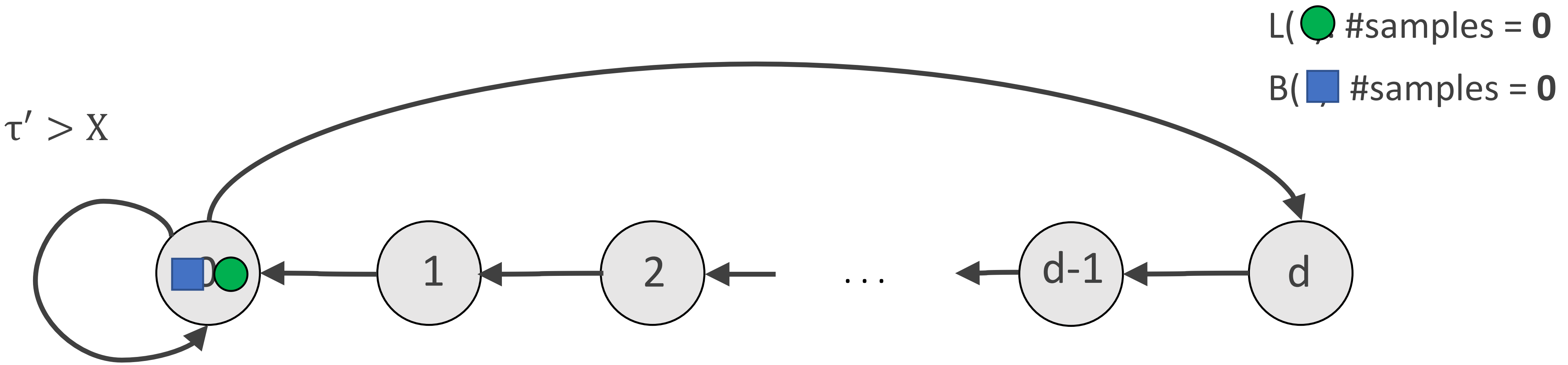
- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- **Key-insight:** After $\mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ rounds, the threshold of learning algorithm L will be greater than that of B for all $i \geq \mathrm{O}(\mathrm{d}^3 \log(\mathrm{n}))$ w.h.p.
- In this case, we show via coupling that L observes **more** samples than B for $i \geq O(d^3 \log(\mathrm{n}))$ w.h.p.



Unknown Reward Distribution

Step 2: Control $\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)]$

- Thus, at round i the learning algorithm collects $\mathrm{O}(i)$ samples w.h.p.
- Using that fact

$$\mathbf{Pr}[X_i \in (\tau, \hat{\tau}(i)) \cup (\hat{\tau}(i), \tau)] \lesssim \sqrt{\frac{\log(i)}{i}}$$

and, hence,

$$\mathrm{Regret}(n) \lesssim \sqrt{n \log(\mathrm{n})}$$

Regret Lower Bound

By reducing the problem to a two-armed bandit problem, we prove the following lower bound:

THEOREM (REGRET LOWER BOUND). *For any learning policy and any $d \geq 1$, there exists an environment with delay d such that the regret of that policy is at least $\Omega\left(\sqrt{n}/d^{3/2}\right)$.*

- The regret of our algorithm is optimal up to polylog factors and dependence on d

Thank you for your attention