Adaptive Resource Allocation for Improving HIV Testing Processes

Harvard CS Colloquium
Feb 5, 2026

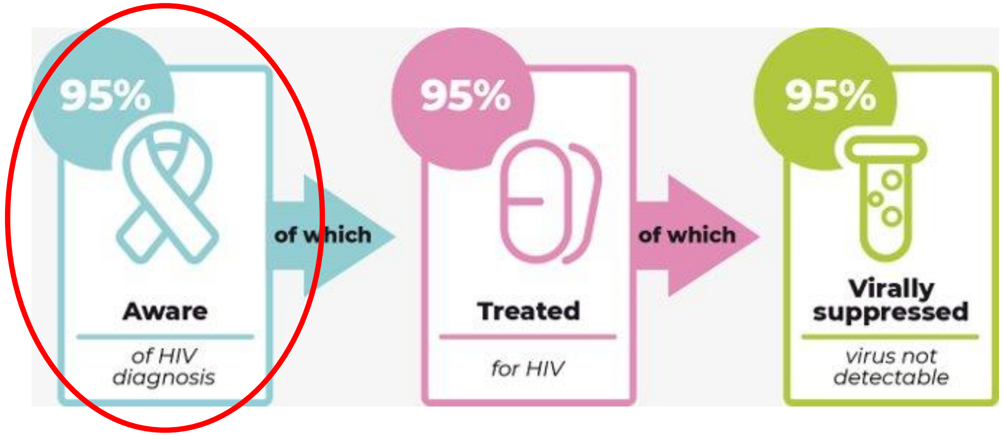
Davin Choo

Postdoctoral Fellow @ Harvard SEAS

Targeting the first 95% in UN’s 95-95-95 initiative

Global initiative by UNAIDS to control the HIV epidemic

- HIV has caused over 40 million deaths to date
- WHO estimates that 1 in 7 HIV positive individuals do not know they are infected
- Undetectable = Untransmittable (U = U)
- UN Sustainable Development Goal 3.3
- The faster we detect positive cases, the faster we can start treatment and change behavior
- Problem made harder with resource limitations due to funding cuts, e.g., to USAID and WHO
- WHO recommends network-based testing



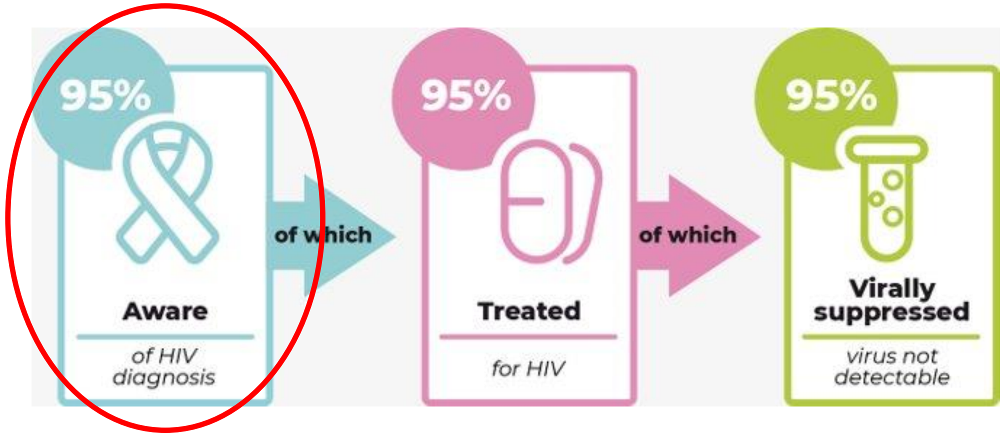
Nursing staff supporting HIV testing in the field

https://www.who.int/news-room/fact-sheets/detail/hiv-aids
https://www.cdc.gov/global-hiv-tb/php/our-approach/undetectable-untransmittable.html
https://sdgs.un.org/goals/goal3 - targets_and_indicators
https://www.unaids.org/en/resources/documents/2024/global-aids-update-2024
https://www.unaids.org/en/impact-US-funding-cuts
https://www.who.int/publications/i/item/9789240096394
Image credit: https://awarehiv.com/en/about-aware-hiv/our-goals; our collaborator Alastair

Modeling the problem

Desired goal and considerations

- *Maximize efficiency* of testing resources in detecting HIV+ cases *as quickly as possible*
  - Resource can be number of test kits, or where to focus efforts of human workers
- Due to budget uncertainty, would be nice if model can enable “anytime” decisions
- Method should exploit underlying transmission graph $\mathcal{G}$
- Preferable to test individuals whose neighbors (in $\mathcal{G}$) have been tested → Frontier exploration

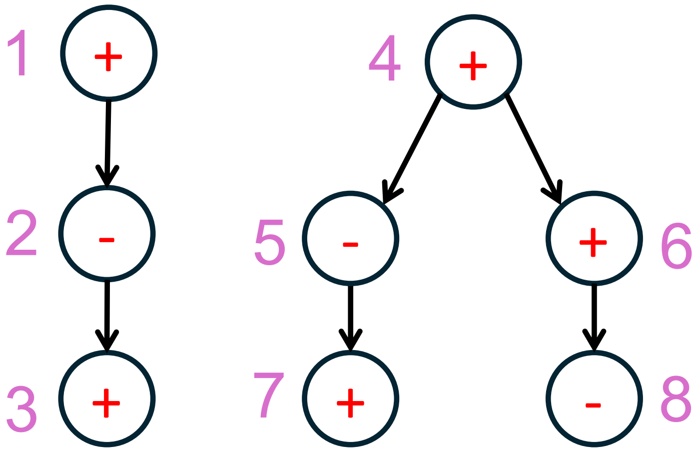


Example on how to compare testing sequences

Let's try to detect positive cases faster, for any fixed amount of testing budget

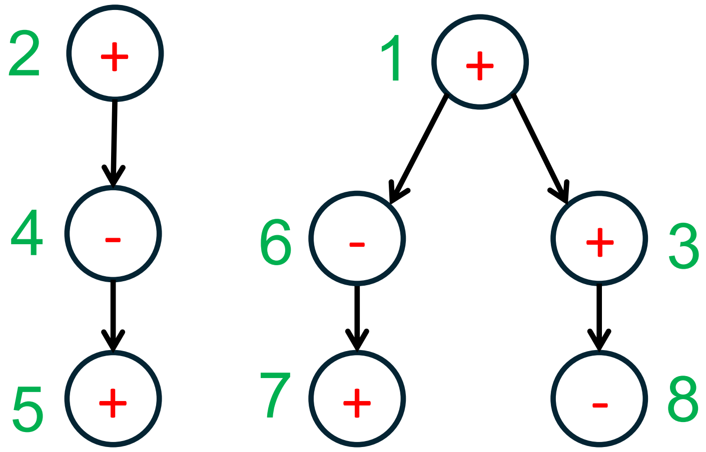
- Why? Early detection → early intervention. There may also be sudden budget cuts.
- *Suppose we know underlying disease statuses*. Which testing sequence is better?

Testing budget	1	2	3	4	5	6	7	8
# positive detected	1	1	2	3	3	4	5	5
	1	2	3	3	4	4	5	5



Green sequence on the right is "better"

For any testing budget, it discovers same or more positive cases



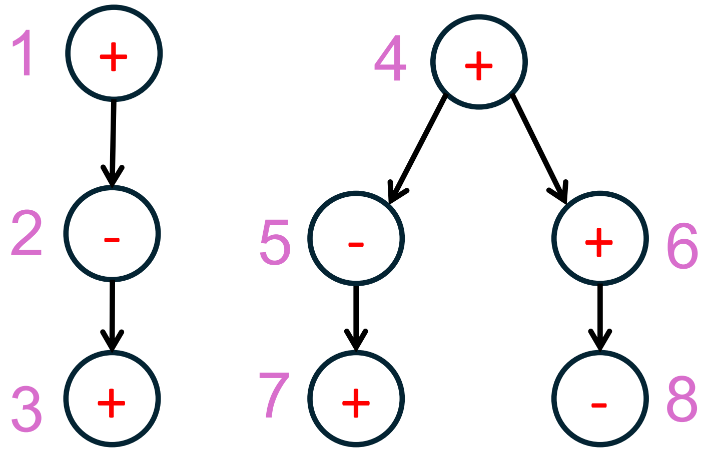
Example on how to compare testing sequences

Let's try to detect positive cases faster, for any fixed amount of testing budget

- Why? Early detection → early intervention. There may also be sudden budget cuts.
- *Suppose we know underlying disease statuses*. Which testing sequence is better?

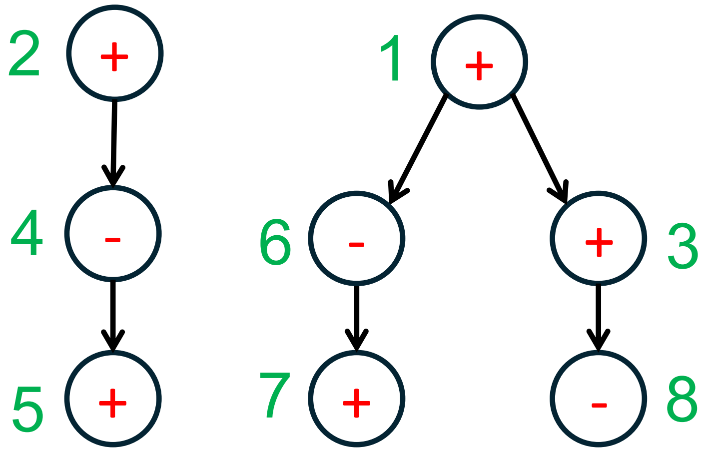
Designing a reward metric with discount factor $\beta \in (0, 1)$ to favor early HIV+ discovery

- Pink sequence reward: $\beta^0 * 1 + \beta^1 * 0 + \beta^2 * 1 + \beta^3 * 1 + \beta^4 * 0 + \beta^5 * 1 + \beta^6 * 1 + \beta^7 * 0$
- Green sequence reward: $\beta^0 * 1 + \beta^1 * 1 + \beta^2 * 1 + \beta^3 * 0 + \beta^4 * 1 + \beta^5 * 0 + \beta^6 * 1 + \beta^7 * 0$
- Intuition: A sequence that optimizes this reward helps detect positive cases faster



Green sequence on the right is "better"

For any testing budget, it discovers same or more positive cases



Our AFEG model [CPW+25]

Davin
Choo

Yuqi
Pan

Tonghan
Wang

Milind
Tambe

Alastair
van Heerden

Cheryl
Johnson

The abstract model

- Interaction graph $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ where $n = |\boldsymbol{V}|$
  - Each node is a person with hidden label from $\boldsymbol{\Omega} = \{0, 1\}$
  - Edges represent possible person-to-person infection
- Joint distribution $\mathcal{P}$ over $2^{|\boldsymbol{V}|}$ status outcomes
- Discount factor $\beta \in (0,1)$

[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.

Our AFEG model [CPW+25]

Davin
Choo

Yuqi
Pan

Tonghan
Wang

Milind
Tambe

Alastair
van Heerden

Cheryl
Johnson

The abstract model

- Interaction graph $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ where $n = |\boldsymbol{V}|$
  - Each node is a person with hidden label from $\boldsymbol{\Omega} = \{0, 1\}$
  - Edges represent possible person-to-person infection
- Joint distribution $\mathcal{P}$ over $2^{|\boldsymbol{V}|}$ status outcomes
- Discount factor $\beta \in (0,1)$

State and action

- For $t \in [n]$, state $\mathcal{S}_t$ includes revealed labels thus far, as well as some untested frontier nodes
  - Equivalently, frontier nodes are the subset of untested nodes in $\mathcal{S}_t$
  - The initial set $\mathcal{S}_0$ given to us is just a subset of untested individuals in $\mathcal{G}$
- Action space: Given $\mathcal{S}_t$, pick an individual to test next
  - Testing an individual reveals their underlying status, and adds their neighbors to the frontier
  - Get a reward if detect a HIV-positive case

[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.

Our AFEG model [CPW+25]


Davin **Choo**


Yuqi **Pan**


Tonghan **Wang**


Milind **Tambe**


Alastair **van Heerden**


Cheryl **Johnson**

The abstract model

- Interaction graph $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ where $n = |\boldsymbol{V}|$
  - Each node is a person with hidden label from $\boldsymbol{\Omega} = \{0, 1\}$
  - Edges represent possible person-to-person infection
- Joint distribution $\mathcal{P}$ over $2^{|\boldsymbol{V}|}$ status outcomes
- Discount factor $\beta \in (0,1)$
- Policy π: Given state $\mathcal{S}_{t-1}$, pick next person to test

"detecting HIV+ cases *as quickly as possible*"

$$\pi^* = \arg\max_\pi \mathbb{E}_{\mathcal{P}} \left[\sum_{t=1}^{n} \overbrace{\beta^{t-1}} \cdot \underbrace{r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)} \right]$$

Reward for revealing status $v \in \{0, 1\}$ when policy decides to test individual $\pi(\mathcal{S}_{t-1})$

[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.

Our AFEG model [CPW+25]

Davin
Choo

Yuqi
Pan

Tonghan
Wang

Milind
Tambe

Alastair
van Heerden

Cheryl
Johnson

The abstract model

- Interaction graph $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ where $n = |\boldsymbol{V}|$
  - Each node is a person with hidden label from $\boldsymbol{\Omega} = \{0, 1\}$
  - Edges represent possible person-to-person infection
- Joint distribution $\mathcal{P}$ over $2^{|\boldsymbol{V}|}$ status outcomes
- Discount factor $\beta \in (0,1)$
- Policy π: Given state $\mathcal{S}_{t-1}$, pick next person to test

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \overbrace{\beta^{t-1}}^{\text{“detecting HIV+ cases \textit{as quickly as possible}”}} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(\underbrace{X_{\pi(\mathcal{S}_{t-1})} = v}_{\substack{\text{Individual } \pi(\mathcal{S}_{t-1}) \text{ chosen by policy} \\ \text{reveals status } v \in \{0,1\} \text{ upon testing}}} \mid \mathcal{S}_{t-1}\right) \cdot \overbrace{r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)}^{\substack{\text{Reward for revealing person} \\ \pi(\mathcal{S}_{t-1}) \text{ having status } v}}$$

[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.

Our AFEG model [CPW+25]

Davin **Choo**

Yuqi **Pan**

Tonghan **Wang**

Milind **Tambe**

Alastair **van Heerden**

Cheryl **Johnson**

The abstract model

- Interaction graph $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ where $n = |\boldsymbol{V}|$
  - Each node is a person with hidden label from $\boldsymbol{\Omega} = \{0, 1\}$
  - Edges represent possible person-to-person infection
- Joint distribution $\mathcal{P}$ over $2^{|\boldsymbol{V}|}$ status outcomes
- Discount factor $\beta \in (0,1)$
- Policy π: Given state $\mathcal{S}_{t-1}$, pick next person to test

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \overbrace{\beta^{t-1}}^{\text{“detecting HIV+ cases } \textit{as quickly as possible}\text{”}} \cdot \sum_{\boxed{v \in \{0,1\}}} \mathcal{P}\Big(\underbrace{X_{\pi(\mathcal{S}_{t-1})} = v}_{\substack{\text{Individual } \pi(\mathcal{S}_{t-1}) \text{ chosen by policy} \\ \text{reveals status } v \in \{0,1\} \text{ upon testing}}} \mid \mathcal{S}_{t-1}\Big) \cdot \overbrace{\boxed{\text{Indicator for v being HIV+}}}^{\substack{\text{Reward for revealing person} \\ \pi(\mathcal{S}_{t-1}) \text{ having status } v}}$$

[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.

Finding a good AFEG policy

Optimal policy

- Can be solved via dynamic programming, but becomes intractable quickly

Adaptive submodularity [GK11]

- Idea of submodularity: Diminishing returns
- Classic example: Set cover problem
- If adaptive submodular, then a natural greedy policy will yield $\left(1 - \frac{1}{e}\right)$-approximation
- Unfortunately, AFEG is not adaptive submodular
    - Observing an infected neighbor can *increase* marginal benefit for testing that person

Question: What other properties of our problem can we exploit?

[GK11] Daniel Golovin and Andreas Krause. Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization. Journal of Artificial Intelligence Research (JAIR), 42:427–486, 2011.

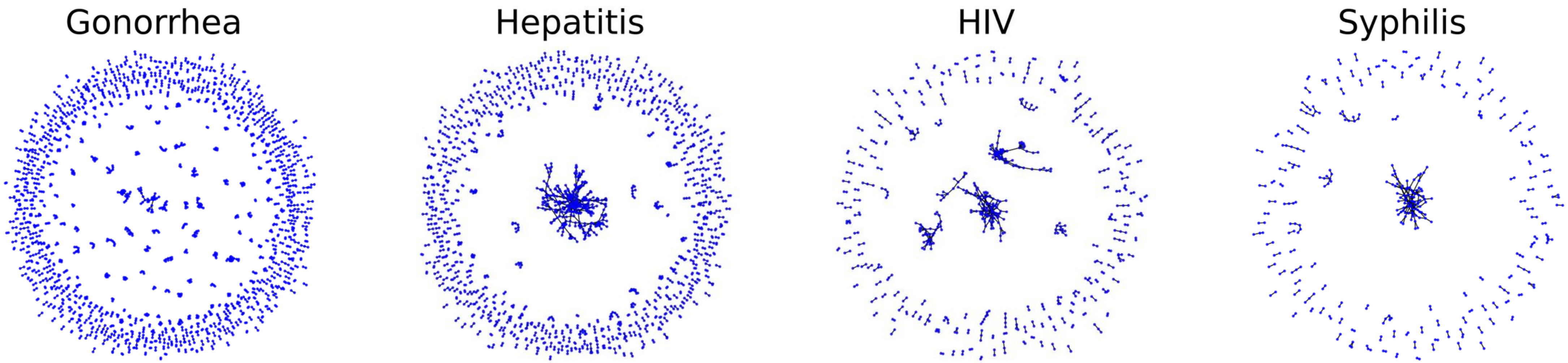
Properties of our problem

Markov property of transmission graphs

- $\mathcal{P}$(person = + | revealed statuses) = $\mathcal{P}$(person = + | revealed statuses of neighbors)

Real-world interaction graphs are sparse and tree-like

- Sexually transmitted diseases do not spread like flu
- Remark: We only ever observe a subset of the true graph



https://www.icpsr.umich.edu/web/ICPSR/studies/22140/summary

Properties of our problem

Markov property of transmission graphs

- $\mathcal{P}$(person = + | revealed statuses) = $\mathcal{P}$(person = + | revealed statuses of neighbors)

Real-world interaction graphs are sparse and tree-like

- Sexually transmitted diseases do not spread like flu
- Remark: We only ever observe a subset of the true graph

Solution approach

- Let's first restrict to the case of *forest* graphs, then adapt it to the real-world graphs later
- Under the frontier testing constraint, we can root each connected component

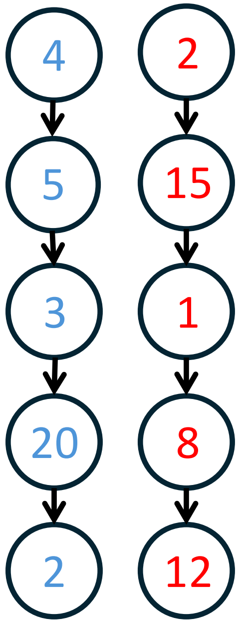
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1} \right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: 2 + 15β + 4β^2 + 5β^3 + 3β^4 + 20β^5 + 1β^6 + 8β^7 + 12β^8 + 2β^9



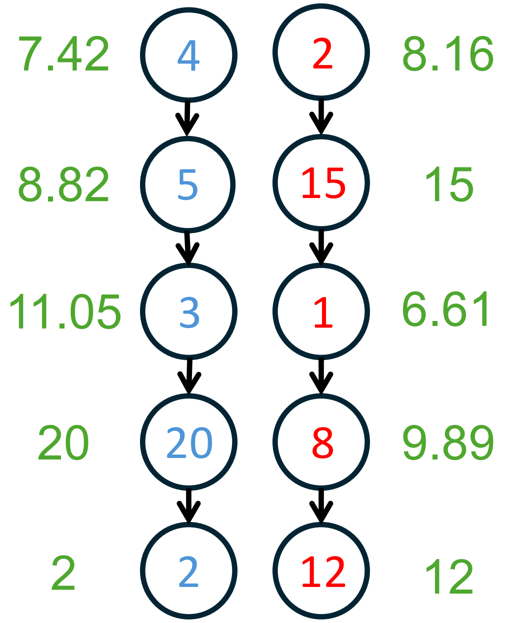
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1}\right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: $2 + 15\beta + 4\beta^2 + 5\beta^3 + 3\beta^4 + 20\beta^5 + 1\beta^6 + 8\beta^7 + 12\beta^8 + 2\beta^9$
- Classic Gittins index policy works here to obtain optimal sequence
  - Compute some Gittins score for each node*
  - Intuition of scores: “total discounted value” / “total discounted time”
  - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



$\beta = 0.9$

* Remark: See backup slides for example computation of two nodes’ values

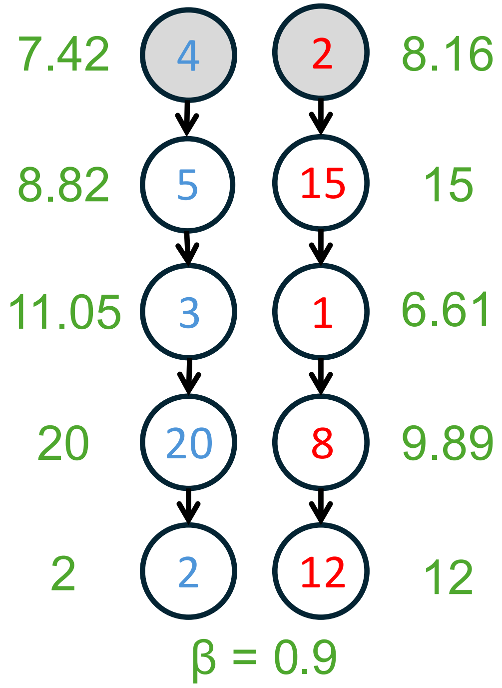
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1}\cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1}\cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1}\right)\cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: 2 + 15β + 4β^2 + 5β^3 + 3β^4 + 20β^5 + 1β^6 + 8β^7 + 12β^8 + 2β^9
- Classic Gittins index policy works here to obtain optimal sequence
  - Compute some Gittins score for each node*
  - Intuition of scores: "total discounted value" / "total discounted time"
  - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



* Remark: See backup slides for example computation of two nodes' values

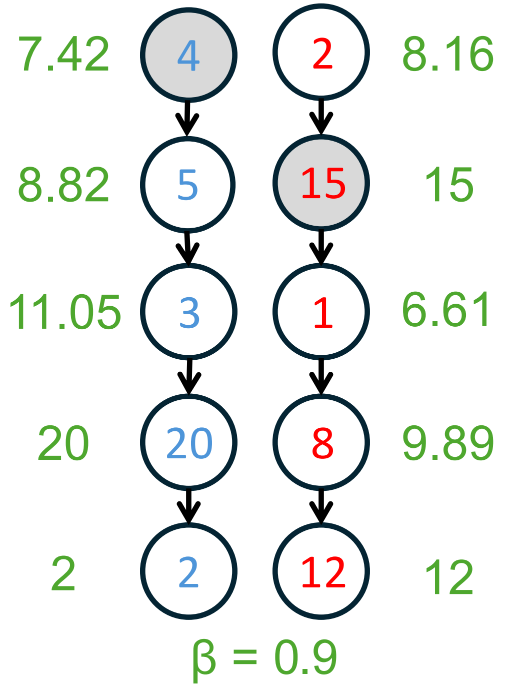
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n}\beta^{t-1}\cdot r\left(X_{\pi(\mathcal{S}_{t-1})},v\right)\right] = \beta^{t-1}\cdot \sum_{v\in\boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})}=v \mid \mathcal{S}_{t-1}\right)\cdot r\left(X_{\pi(\mathcal{S}_{t-1})},v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: $2 + 15\beta + 4\beta^2 + 5\beta^3 + 3\beta^4 + 20\beta^5 + 1\beta^6 + 8\beta^7 + 12\beta^8 + 2\beta^9$
- Classic Gittins index policy works here to obtain optimal sequence
  - Compute some Gittins score for each node*
  - Intuition of scores: “total discounted value” / “total discounted time”
  - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



* Remark: See backup slides for example computation of two nodes’ values

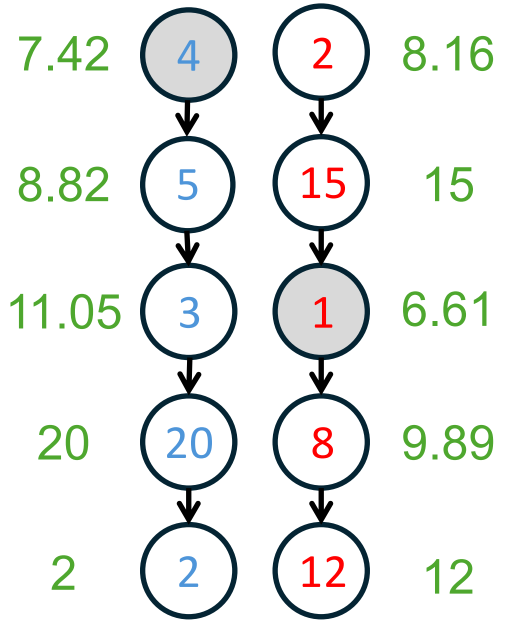
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1} \right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: $2 + 15\beta + \underline{4}\beta^2 + 5\beta^3 + 3\beta^4 + 20\beta^5 + 1\beta^6 + 8\beta^7 + 12\beta^8 + 2\beta^9$
- Classic Gittins index policy works here to obtain optimal sequence
  - Compute some Gittins score for each node*
  - Intuition of scores: “total discounted value” / “total discounted time”
  - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



$\beta = 0.9$

* Remark: See backup slides for example computation of two nodes’ values

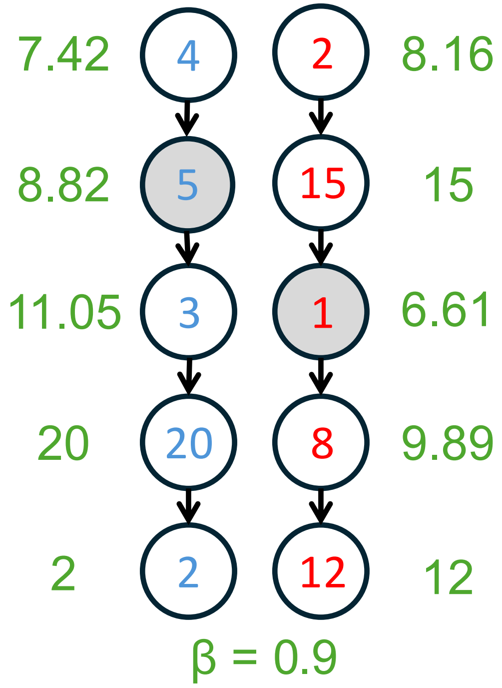
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1}\right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: 2 + 15β + 4β^2 + 5β^3 + 3β^4 + 20β^5 + 1β^6 + 8β^7 + 12β^8 + 2β^9
- Classic Gittins index policy works here to obtain optimal sequence
  - Compute some Gittins score for each node*
  - Intuition of scores: "total discounted value" / "total discounted time"
  - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



* Remark: See backup slides for example computation of two nodes' values

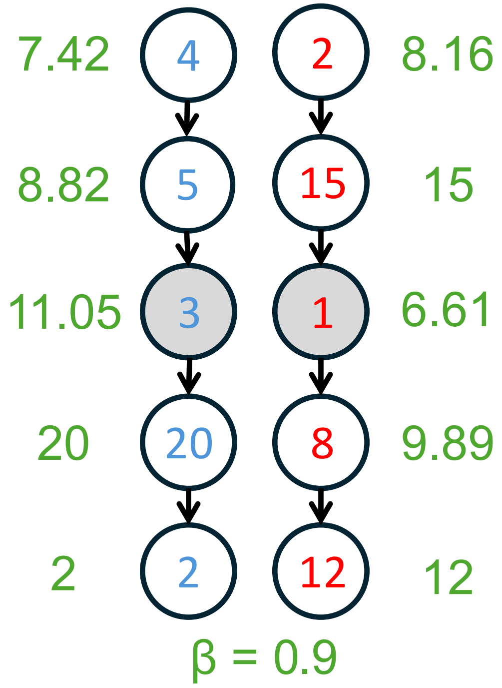
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \,\in\, \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1}\right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: $2 + 15\beta + 4\beta^2 + 5\beta^3 + 3\beta^4 + 20\beta^5 + 1\beta^6 + 8\beta^7 + 12\beta^8 + 2\beta^9$
- Classic Gittins index policy works here to obtain optimal sequence
  - Compute some Gittins score for each node*
  - Intuition of scores: "total discounted value" / "total discounted time"
  - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



* Remark: See backup slides for example computation of two nodes' values

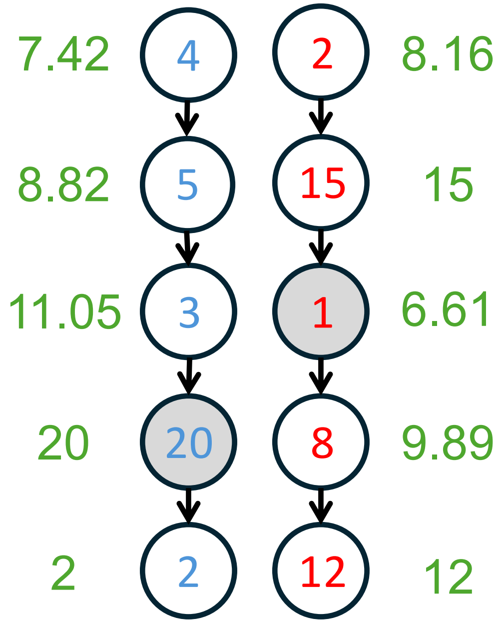
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1}\right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: $2 + 15\beta + 4\beta^2 + 5\beta^3 + 3\beta^4 + \underline{20}\beta^5 + 1\beta^6 + 8\beta^7 + 12\beta^8 + 2\beta^9$
- Classic Gittins index policy works here to obtain optimal sequence
  - Compute some Gittins score for each node*
  - Intuition of scores: “total discounted value” / “total discounted time”
  - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



β = 0.9

* Remark: See backup slides for example computation of two nodes’ values

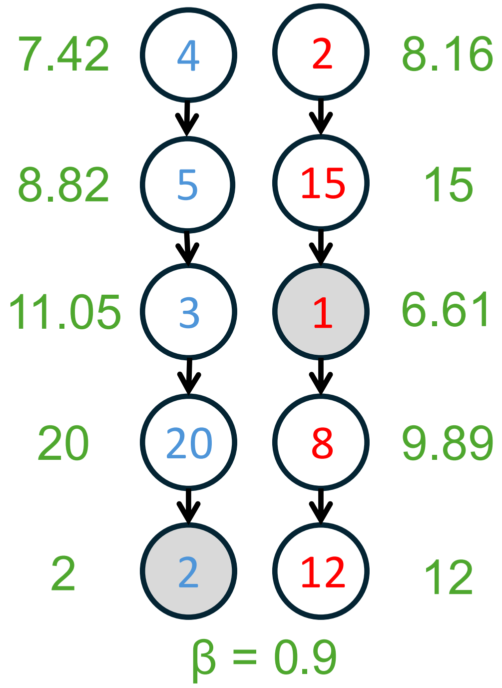
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1}\right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: $2 + 15\beta + 4\beta^2 + 5\beta^3 + 3\beta^4 + 20\beta^5 + \underline{1}\beta^6 + 8\beta^7 + 12\beta^8 + 2\beta^9$
- Classic Gittins index policy works here to obtain optimal sequence
    - Compute some Gittins score for each node*
    - Intuition of scores: "total discounted value" / "total discounted time"
    - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



* Remark: See backup slides for example computation of two nodes' values

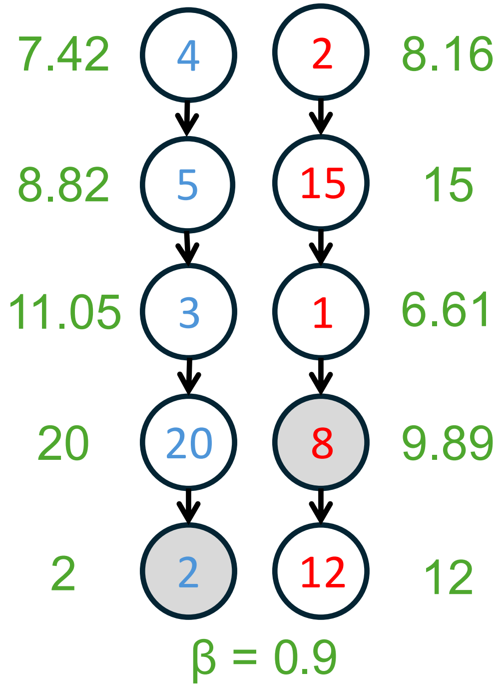
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1}\right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: $2 + 15\beta + 4\beta^2 + 5\beta^3 + 3\beta^4 + 20\beta^5 + 1\beta^6 + \underline{8}\beta^7 + 12\beta^8 + 2\beta^9$
- Classic Gittins index policy works here to obtain optimal sequence
    - Compute some Gittins score for each node*
    - Intuition of scores: “total discounted value” / “total discounted time”
    - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



* Remark: See backup slides for example computation of two nodes’ values

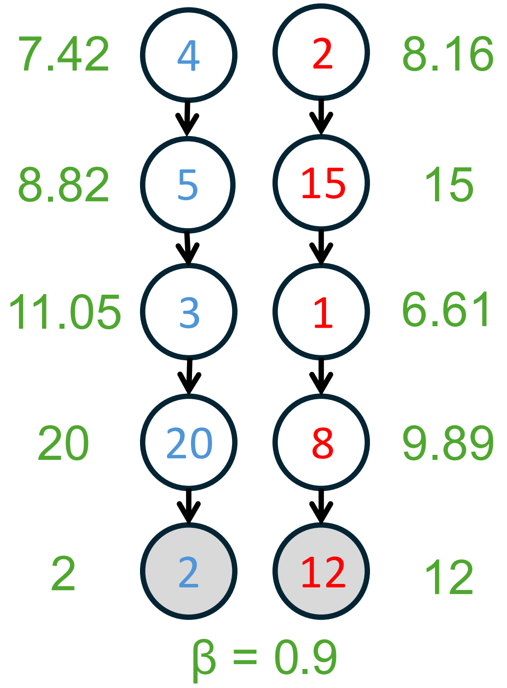
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1}\right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: $2 + 15\beta + 4\beta^2 + 5\beta^3 + 3\beta^4 + 20\beta^5 + 1\beta^6 + 8\beta^7 + \underline{12}\beta^8 + 2\beta^9$
- Classic Gittins index policy works here to obtain optimal sequence
  - Compute some Gittins score for each node*
  - Intuition of scores: “total discounted value” / “total discounted time”
  - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent



* Remark: See backup slides for example computation of two nodes’ values

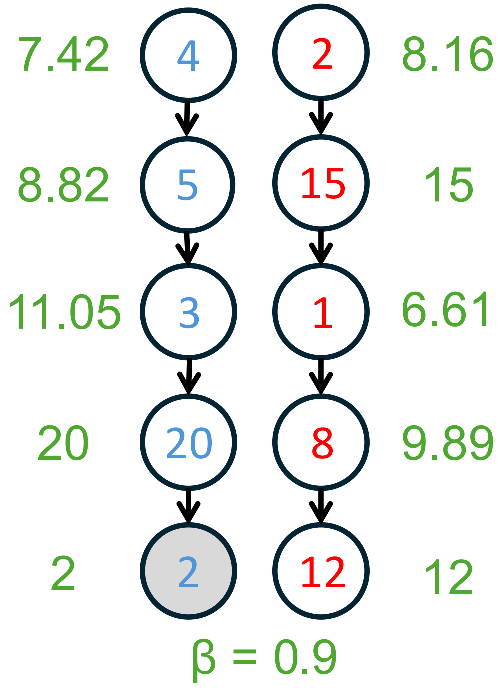
When input graph is just chains, classic bandit reduction works and Gittins index is optimal

Recall our objective function to maximize:

$$\mathbb{E}_{\mathcal{P}}\left[\sum_{t=1}^{n} \beta^{t-1} \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)\right] = \beta^{t-1} \cdot \sum_{v \in \boldsymbol{\Omega}} \mathcal{P}\left(X_{\pi(\mathcal{S}_{t-1})} = v \mid \mathcal{S}_{t-1}\right) \cdot r\left(X_{\pi(\mathcal{S}_{t-1})}, v\right)$$

In special case of chain graph input

- Sequence: 2, 15, 4, 5, 3, 20, 1, 8, 12, 2
- Reward: $2 + 15\beta + 4\beta^2 + 5\beta^3 + 3\beta^4 + 20\beta^5 + 1\beta^6 + 8\beta^7 + 12\beta^8 + 2\beta^9$
- Classic Gittins index policy works here to obtain optimal sequence
  - Compute some Gittins score for each node*
  - Intuition of scores: "total discounted value" / "total discounted time"
  - Repeatedly pick the argmax at the frontier
- Methods work even when rewards are probabilistic and depend on parent

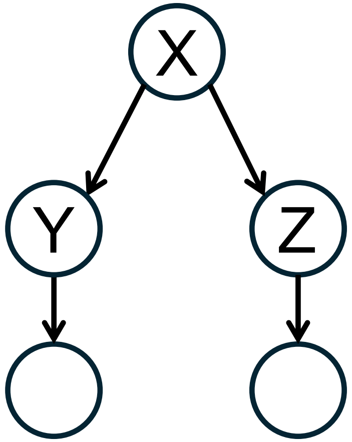


* Remark: See backup slides for example computation of two nodes' values

Branching bandits apply for the more general case when input graph is a rooted forest

For more general forest graph input

- Testing a person can create more than 1 branch, so classic bandit reduction *breaks*
- Under assumption that branches are independent (true due to our Markov assumption), there is a generalization called branching bandits* that we can map AFEG with rooted forests into



Computation of X's value is as straightforward as before:
Optimal sequence from X may require us to switch between the Y and Z branches,

* Remark: There is another branching bandit formulation of [Wei88] that is often studied in queuing theory. Unfortunately, it is not suitable for our AFEG model, and we use the formulation of [KO03] instead.

[Wei88] Gideon Weiss. Branching Bandit Processes. Probability in the Engineering and Informational Sciences, 2(3):269–278, 1988.

[KO03] Godfrey Keller and Alison Oldale. Branching bandits: a sequential search process with correlated pay-offs. Journal of Economic Theory, 113(2):302–315, 2003.

Branching bandits apply for the more general case when input graph is a rooted forest

For more general forest graph input

- Testing a person can create more than 1 branch, so classic bandit reduction *breaks*
- Under assumption that branches are independent (true due to our Markov assumption), there is a generalization called branching bandits* that we can map AFEG with rooted forests into

Branching bandits [KO03]

- Conceptually, the solution concept is same idea as before
  - Compute some Gittins scores from leaves towards the root. Then, repeatedly pick the argmax
- Optimality was proven by [KO03] via recursive formulas involving terms like ϕ and Φ
  - Pretty complicated: involves integral of product of partial differentials; Happy to discuss offline
- We provide the first efficient polynomial time dynamic programming (DP) method, with working Python code, in the 20 years since [KO03] for computing ϕ and Φ for *discrete labels*
  - Prove and exploit piecewise linearity of ϕ and Φ, enabling efficient representation of ϕ and Φ in the DP
  - Number of pieces scales well with the number of nodes and number of labels

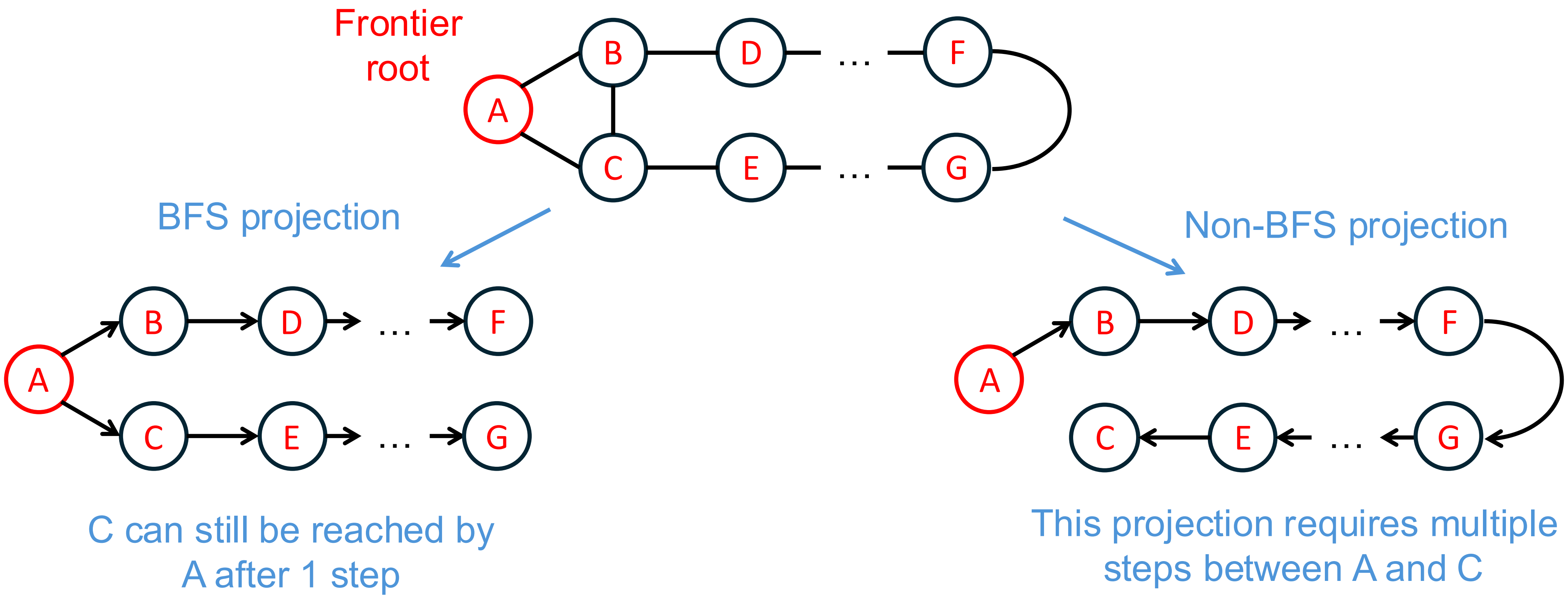
* Remark: There is another branching bandit formulation of [Wei88] that is often studied in queuing theory. Unfortunately, it is not suitable for our AFEG model, and we use the formulation of [KO03] instead.

[Wei88] Gideon Weiss. Branching Bandit Processes. Probability in the Engineering and Informational Sciences, 2(3):269–278, 1988.
[KO03] Godfrey Keller and Alison Oldale. Branching bandits: a sequential search process with correlated pay-offs. Journal of Economic Theory, 113(2):302–315, 2003.

In practice, use BFS to project non-trees into trees before applying Gittins computation

Interaction graphs $\mathcal{G}$ may not be forests in general

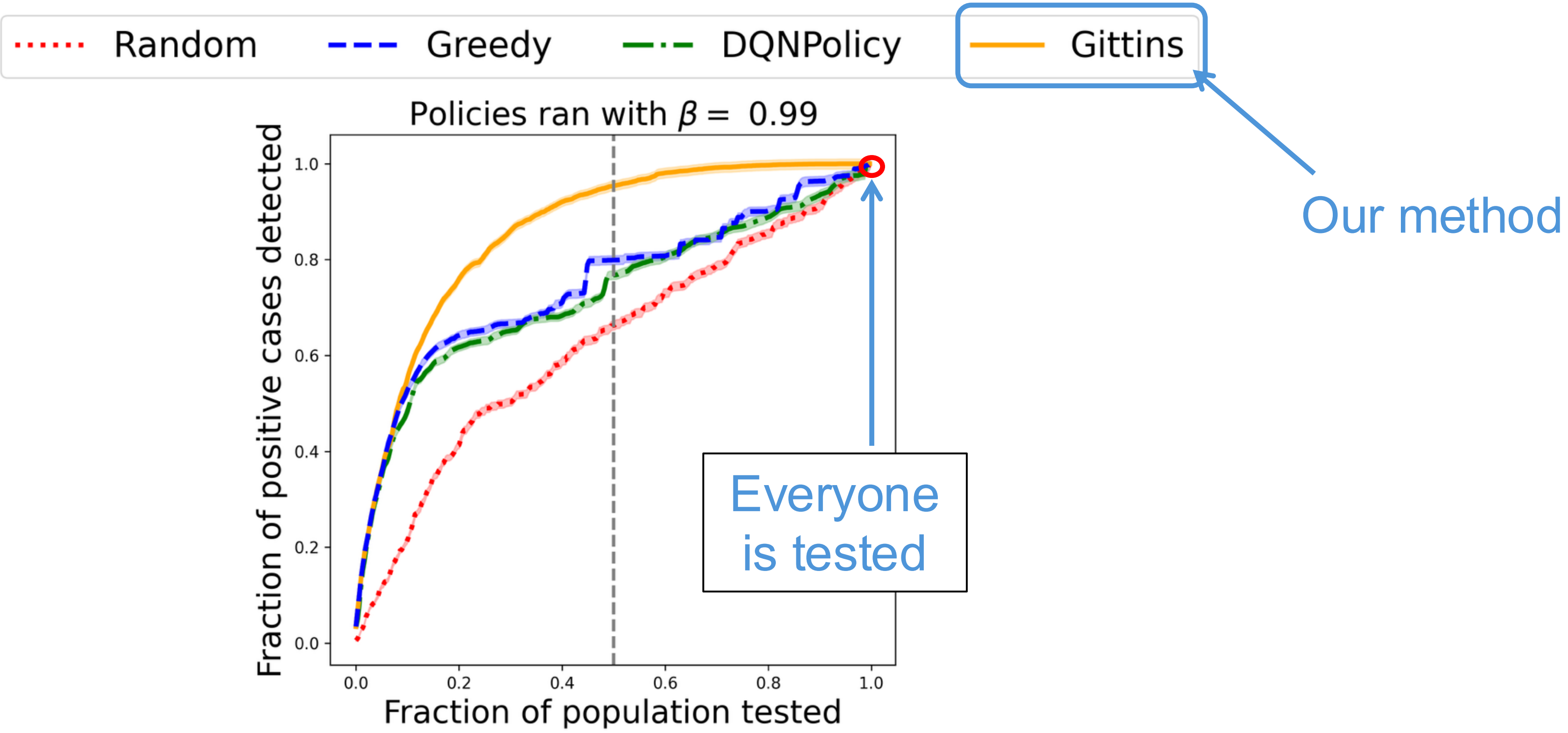
- Run breadth-first search (BFS) on $\mathcal{G}$ from the frontier roots in each component
- This minimizes height to root, reducing artificial frontier constraint due to projection



Empirical evaluation* on HIV interaction graph

With only budget to test half the population, Gittins detects almost all positive cases in expectation while the other methods still miss about 20% of the positive cases

- BFS, then reduction to branching bandits, with provable optimality on tree-structured instances

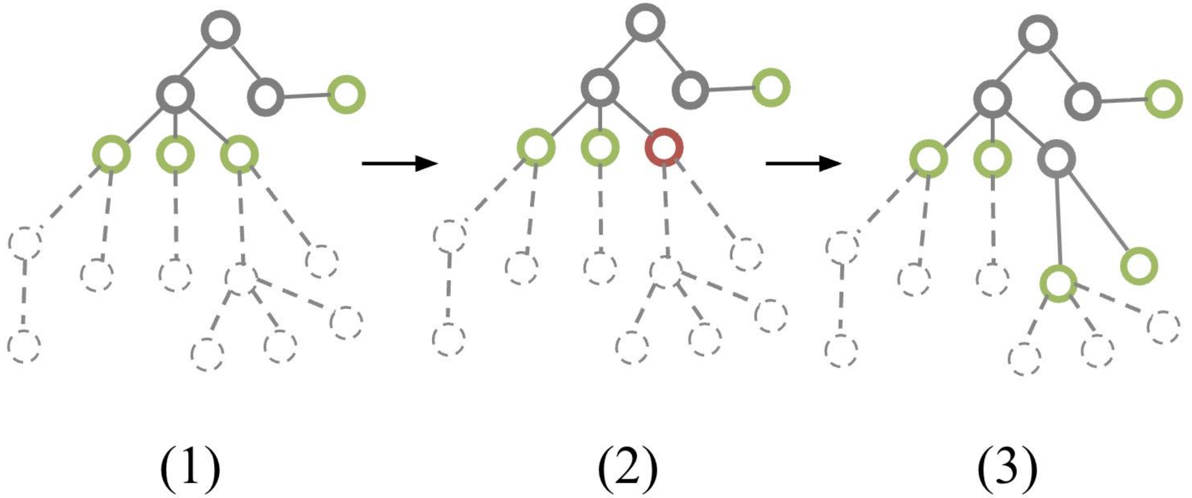


* We also have experiments on settings where policies only know a noisy version of the underlying distribution $\mathcal{P}$

In many real-world settings, the interaction graph is revealed incrementally as we act on it

The method of [CPW+25] assumes the interaction graph $\mathcal{G}$ is given as input

- In many real-world settings, this graph is only revealed incrementally over time
- More realistic: As we test people in the frontier, their neighbors get revealed



[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.

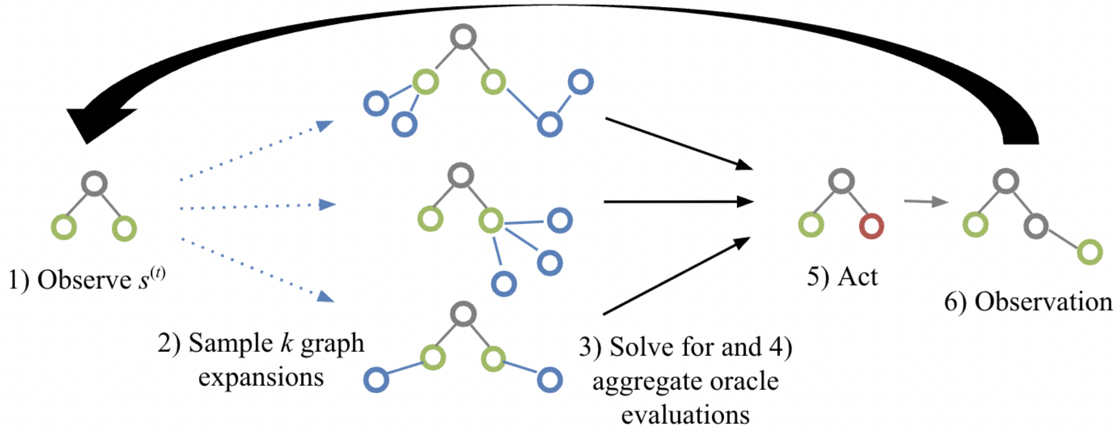
In many real-world settings, the interaction graph is revealed incrementally as we act on it

The method of [CPW+25] assumes the interaction graph $\mathcal{G}$ is given as input

- In many real-world settings, this graph is only revealed incrementally over time
- More realistic: As we test people in the frontier, their neighbors get revealed

[KCK+26] removes this assumption

- Idea: generate graph expansions, run Gittins, aggregate




Akseli **Kangaslahti**


Davin **Choo**


Lingkai **Kong**


Milind **Tambe**


Alastair **van Heerden**


Cheryl **Johnson**

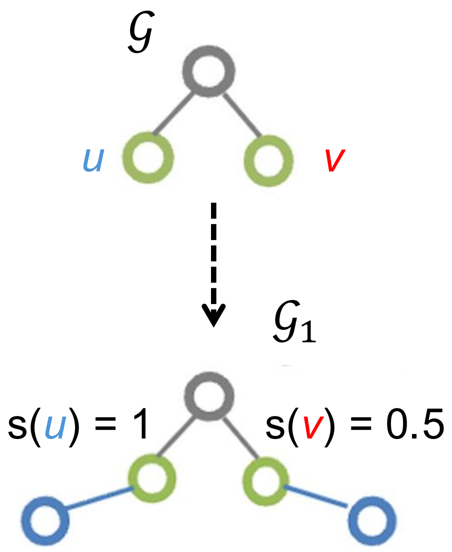
[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.
[KCK+26] Akseli Kangaslahti, Davin Choo, Lingkai Kong, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Policy-Embedded Graph Expansion: Networked HIV Testing with Diffusion-Driven Network Samples. Under submission, 2026.

* Note: Gittin scores s(.) are arbitrary numbers in this example

A glimpse of [KCK+26]

How to “generate graph expansions, run Gittins, aggregate”?

- Idea: Generate “most likely graph”, then compute Gittins via [CPW+25] and pick from frontier



$\Pr(\mathcal{G}_1 \mid \mathcal{G}) = 0.51$

Based on “most likely graph”, we should test node u next

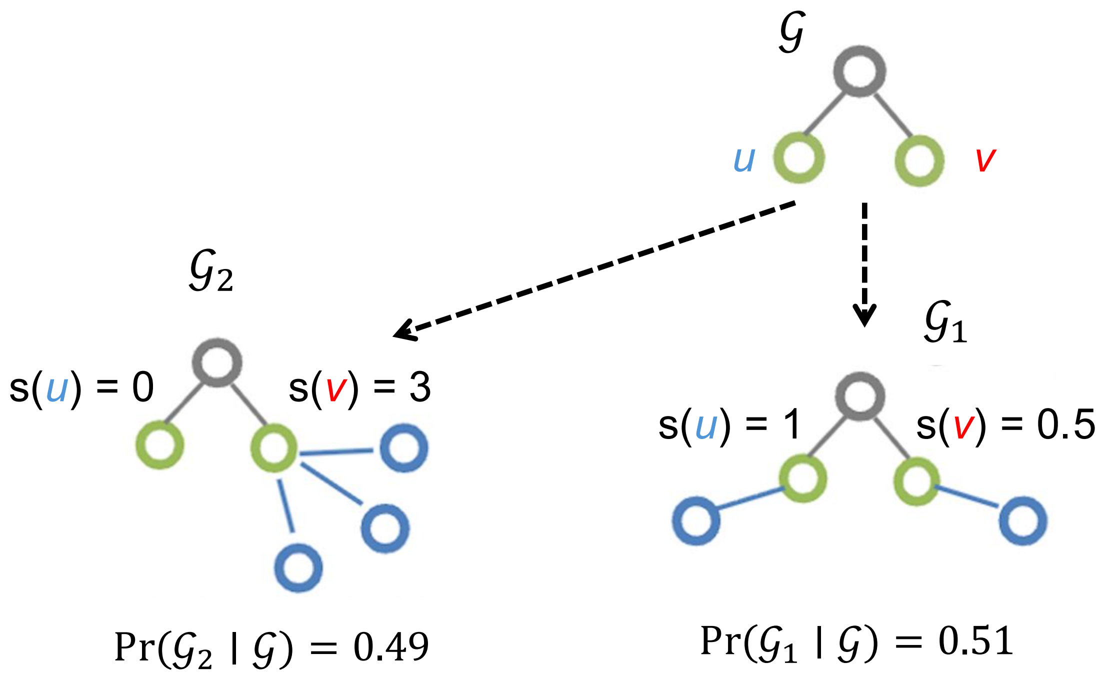
[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.
[KCK+26] Akseli Kangaslahti, Davin Choo, Lingkai Kong, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Policy-Embedded Graph Expansion: Networked HIV Testing with Diffusion-Driven Network Samples. Under submission, 2026.

* Note: Gittin scores s(.) are arbitrary numbers in this example

A glimpse of [KCK+26]

How to “generate graph expansions, run Gittins, aggregate”?

- Idea: Generate “most likely graph”, then compute Gittins via [CPW+25] and pick from frontier
- Problem: Relying on MLE graph expansion itself is suboptimal



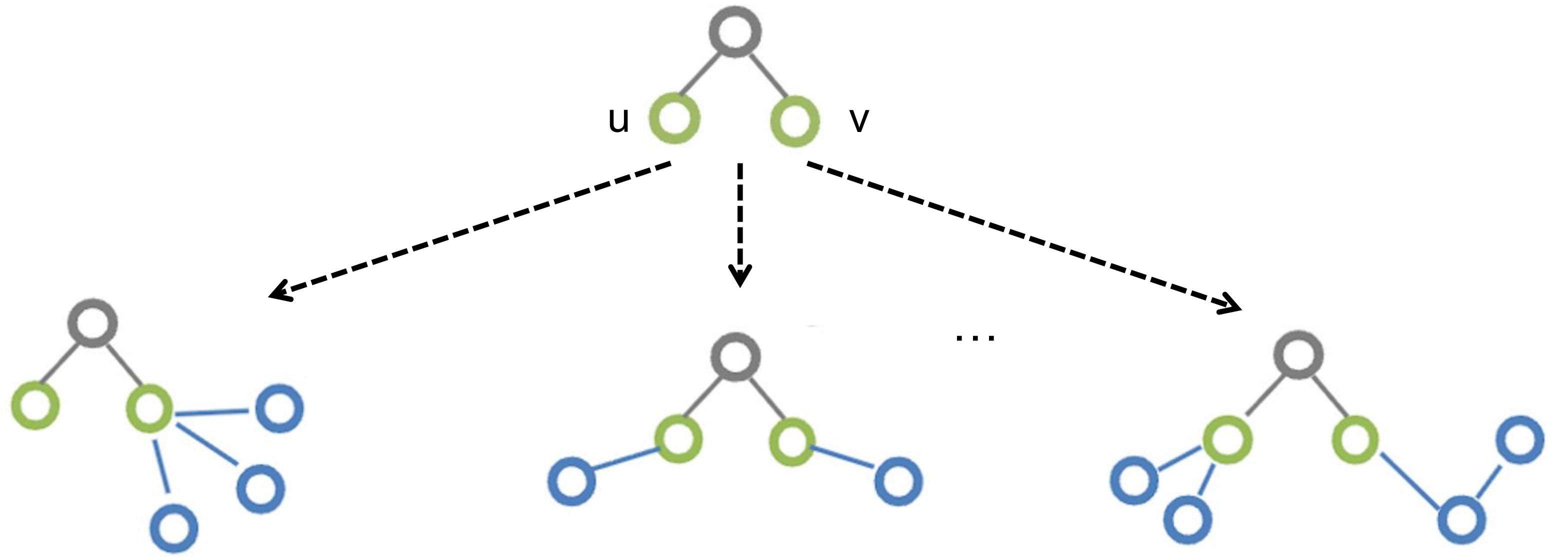
In expectation, over all graph completions, we should test v instead!

[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.
[KCK+26] Akseli Kangaslahti, Davin Choo, Lingkai Kong, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Policy-Embedded Graph Expansion: Networked HIV Testing with Diffusion-Driven Network Samples. Under submission, 2026.

A glimpse of [KCK+26]

How to “generate graph expansions, run Gittins, aggregate”?

- Idea: Generate “most likely graph”, then compute Gittins via [CPW+25] and pick from frontier
- Problem: Relying on MLE graph expansion itself is suboptimal
  - Fix: Learn a *distribution via diffusion models*, then use Monte Carlo for expected Gittins value

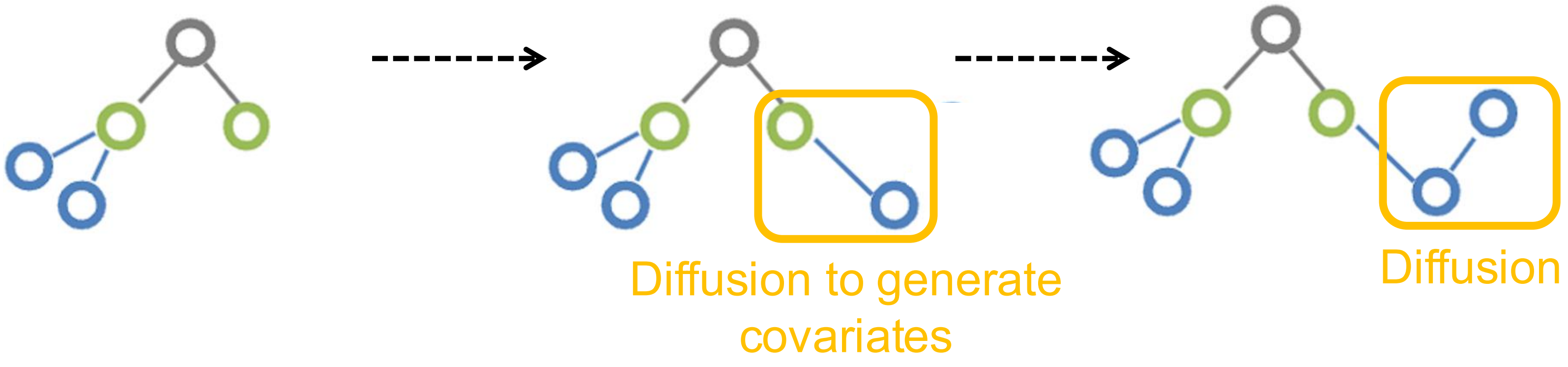


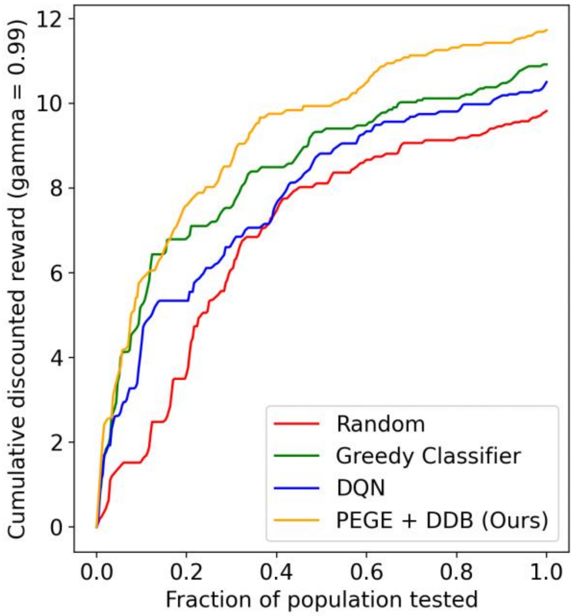
[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.
[KCK+26] Akseli Kangaslahti, Davin Choo, Lingkai Kong, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Policy-Embedded Graph Expansion: Networked HIV Testing with Diffusion-Driven Network Samples. Under submission, 2026.

A glimpse of [KCK+26]

How to "generate graph expansions, run Gittins, aggregate"?

- Idea: Generate "most likely graph", then compute Gittins via [CPW+25] and pick from frontier
- Problem: Relying on MLE graph expansion itself is suboptimal
  - Fix: Learn a *distribution via diffusion models*, then use Monte Carlo for expected Gittins value
- Problem: Limited training data in our problem setting
  - Fix: Exploit referral structure, and use edgewise auto-regressive diffusion





[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.
[KCK+26] Akseli Kangaslahti, Davin Choo, Lingkai Kong, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Policy-Embedded Graph Expansion: Networked HIV Testing with Diffusion-Driven Network Samples. Under submission, 2026.

One more thing: Improving referral schemes to improve engagement with underlying population

Existing resource allocation schemes are typically "static"

- A fixed number of resources per individual is decided ahead of time, independent of actual data
- Example 1: Allocating self-test kits
  - Give each person 5 self-test kits to distribute to contacts
  - People can come into clinic after performing self-tests for follow-up

One more thing: Improving referral schemes to improve engagement with underlying population

Existing resource allocation schemes are typically "static"

- A fixed number of resources per individual is decided ahead of time, independent of actual data
- Example 1: Allocating self-test kits
  - Give each person 5 self-test kits to distribute to contacts
  - People can come into clinic after performing self-tests for follow-up
- Example 2: Allocating vouchers
  - Give each person 5 ID-tagged vouchers to refer their contacts
  - Promise to pay $1 per successful referral
  - $5 is "locked in" and shouldn't be re-used elsewhere, even if less than 5 referrals end up coming in
- Example 3: Allocating LLM tokens
  - There is an ongoing trial in KwaZulu-Natal, South Africa, where they have a health chatbot
  - They are investigating how to distribute LLM tokens to encourage users to sign up and refer others

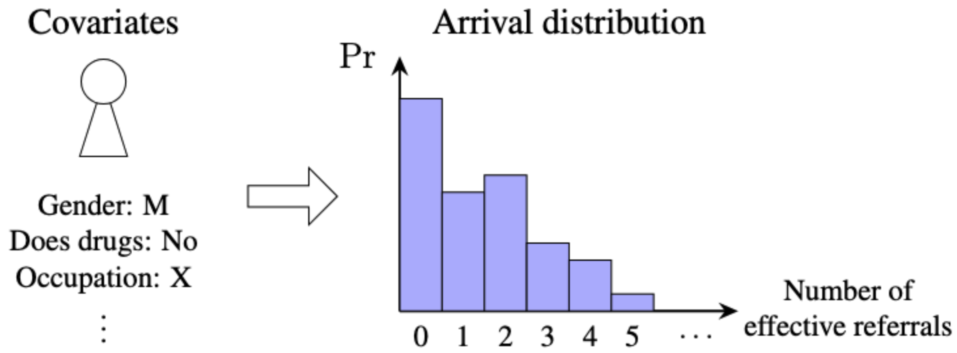
In all these examples: once resource is given out, we cannot take them back

- Current operational guidelines: everyone receives same constant number of resources

Distributional information about effective referrals can improve efficiency of adaptive referral schemes

[PCW+26] uses distributional information to improve resource allocation for referrals

- Infer and exploit a distribution for everyone using their covariates and past data
- Similar setup to prophet inequality settings
  - Make decisions based on observed distributions
  - Receive reward based on actual realization



Yuqi **Pan**

Davin **Choo**

Haichuan **Wang**

Milind **Tambe**

Alastair **van Heerden**

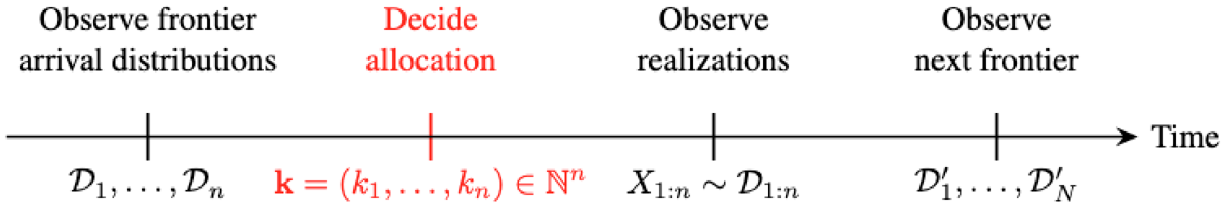
Cheryl **Johnson**

[PCW+26] Yuqi Pan*, Davin Choo*, Haichuan Wang, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Adaptive Multi-Round Allocation with Stochastic Arrivals. Under submission, 2026.

Distributional information about effective referrals can improve efficiency of adaptive referral schemes

[PCW+26] uses distributional information to improve resource allocation for referrals

- Infer and exploit a distribution for everyone using their covariates and past data
- Similar setup to prophet inequality settings
  - Make decisions based on observed distributions
  - Receive reward based on actual realization
- Additionally, our problem setting is multi-round
  - "Allocation + realization" directly impacts number of distributions that appear in the next round
  - Need to decide how much budget to use this round and how to allocate budgets within round



$$N(\mathbf{k}; X_{1:n}) = \sum_{i=1}^{n} \min\{k_i, X_i\}$$


Yuqi **Pan**


Davin **Choo**


Haichuan **Wang**


Milind **Tambe**


Alastair **van Heerden**

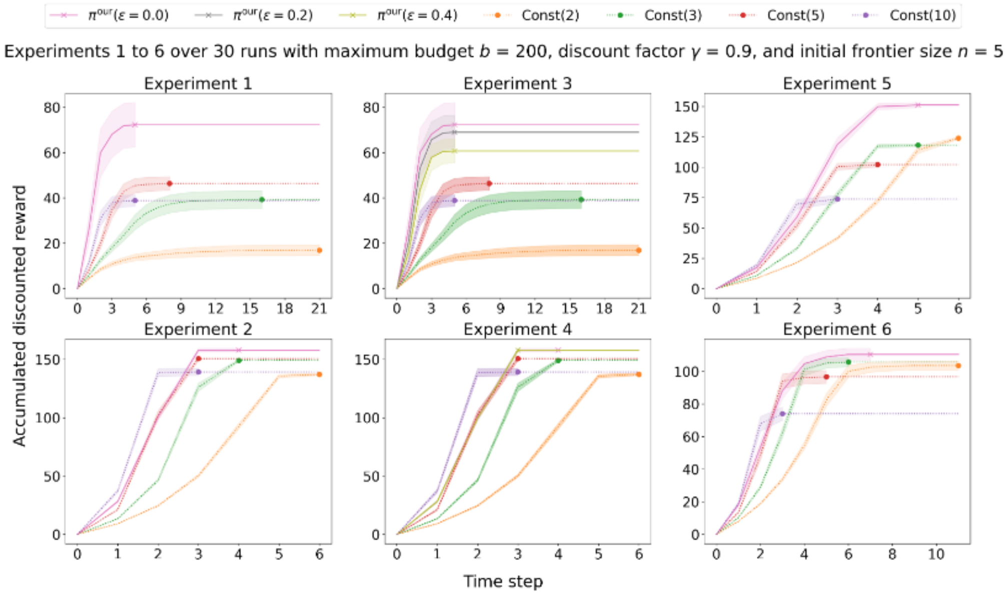

Cheryl **Johnson**

[PCW+26] Yuqi Pan*, Davin Choo*, Haichuan Wang, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Adaptive Multi-Round Allocation with Stochastic Arrivals. Under submission, 2026.

A glimpse of [PCW+26]

Our solution concept

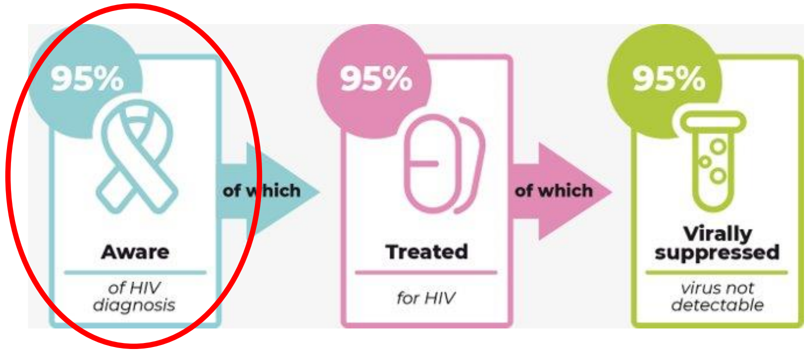
- For any single-round budget, greedy over the survival probabilities is optimal
  - Repeatedly assign resource to $\mathrm{argmax}_i \Pr_{X_i \sim \mathcal{D}_i}(X_i \geq \ell + 1)$, where ℓ is currently allocated amount
- Solving the full multi-round Bellman is intractable, but there is a poly-time proxy we can solve
  - Our derived proxy-based policy outperforms “constant” policies, even when there is noise in the estimation of the underlying distributions $\mathcal{D}$ and $\mathcal{P}$



See backup slides for details

[PCW+26] Yuqi Pan*, Davin Choo*, Haichuan Wang, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Adaptive Multi-Round Allocation with Stochastic Arrivals. Under submission, 2026.
ICPSR dataset: https://www.icpsr.umich.edu/web/ICPSR/studies/22140/summary

Adaptive Resource Allocation for Improving HIV Testing Processes



Real-world objective

- Improve resource efficiency
- Identify as many HIV+ cases as fast as possible

Our efforts

- [CPW+25] defines the AFEG model and design a Gittins index-based method
- [KCK+26] tackles setting where interaction graph $\mathcal{G}$ is revealed incrementally as we test
- [PCW+26] exploits distributional information to improve resource allocation for referral schemes
- Working with collaborators on an ongoing trial in KwaZulu-Natal, South Africa

Thank you for your kind attention!

[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.
[KCK+26] Akseli Kangaslahti, Davin Choo, Lingkai Kong, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Policy-Embedded Graph Expansion: Networked HIV Testing with Diffusion-Driven Network Samples. Under submission, 2026.
[PCW+26] Yuqi Pan*, Davin Choo*, Haichuan Wang, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Adaptive Multi-Round Allocation with Stochastic Arrivals. Under submission, 2026.

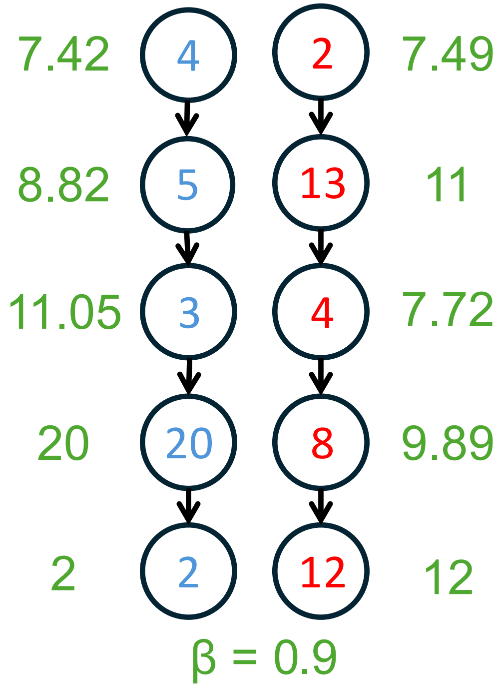
Back up slides

Gittins calculation example

Gittins index formula: $g = \max_{\tau \geq 1} A_\tau = \max_{\tau \geq 1} \frac{\sum_{t=0}^{\tau} \beta^t \cdot r_t}{\sum_{t=0}^{\tau} \beta^t}$

Consider chain 4 → 5 → 3 → 20 → 2

τ	A_τ
1	$\frac{4}{1} = 4$
2	$\frac{4+0.9\cdot 5}{1+0.9} \approx 4.47$
3	$\frac{4+0.9\cdot 5+0.9^2\cdot 3}{1+0.9+0.9^2} \approx 4.03$
4	$\frac{4+0.9\cdot 5+0.9^2\cdot 3+0.9^3\cdot 20}{1+0.9+0.9^2+0.9^3} \approx$ 7.42
5	$\frac{4+0.9\cdot 5+0.9^2\cdot 3+0.9^3\cdot 20+0.9^4\cdot 2}{1+0.9+0.9^2+0.9^3+0.9^4} \approx 6.54$

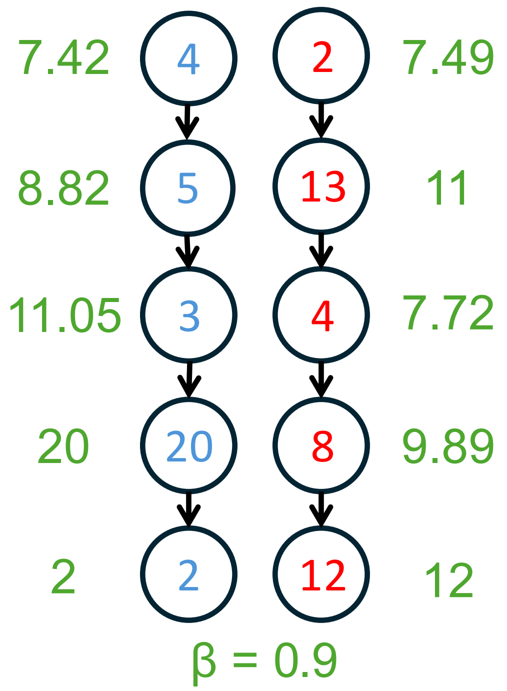


Gittins calculation example

Gittins index formula: $g = \max_{\tau \geq 1} A_\tau = \max_{\tau \geq 1} \frac{\sum_{t=0}^{\tau} \beta^t \cdot r_t}{\sum_{t=0}^{\tau} \beta^t}$

Consider chain 4 → 5 → 3 → 20 → 2

τ	A_τ
1	$\frac{5}{1} = 5$
2	$\frac{5+0.9\cdot 3}{1+0.9} \approx 4.05$
3	$\frac{5+0.9\cdot 3+0.9^2\cdot 20}{1+0.9+0.9^2} \approx$ **8.82**
4	$\frac{5+0.9\cdot 3+0.9^2\cdot 20+0.9^3\cdot 2}{1+0.9+0.9^2+0.9^3} \approx 7.37$



Recursive formula of [KO03]

To define the Gittins index, let us first define two recursive functions ϕ and Φ, as per [KO03]. For any non-root node $X \in \mathbf{X}$, label $b \in \mathbf{\Omega}$, and value $0 \leq m \leq \frac{\bar{r}}{1-\beta}$,

$$\phi_{X,b}(m) = \max\left\{m, \sum_{v \in \mathbf{\Omega}} \mathcal{P}(X = v \mid \mathrm{Pa}(X) = b) \cdot \left[r(X, v) + \beta \cdot \Phi_{\mathrm{Ch}(X),v}(m)\right]\right\} \tag{1}$$

If X is the root, we define $\phi_{X,\emptyset}(m) = \max\left\{m, \sum_{v \in \mathbf{\Omega}} \mathcal{P}(X = v) \cdot \left[r(X, v) + \beta \cdot \Phi_{\mathrm{Ch}(X),v}(m)\right]\right\}$. For any subset of nodes $\mathbf{S} \in \mathbf{X}$, label $v \in \mathbf{\Omega}$, and value $0 \leq m \leq \frac{\bar{r}}{1-\beta}$,

$$\Phi_{\mathbf{S},v}(m) = \begin{cases} \frac{\bar{r}}{1-\beta} - \int_m^{\frac{\bar{r}}{1-\beta}} \prod_{Y \in \mathbf{S}} \frac{\partial \phi_{Y,v}(k)}{\partial k}\, dk & \text{if } \mathbf{S} \neq \emptyset \\ m & \text{if } \mathbf{S} = \emptyset \end{cases} \tag{2}$$

Gittins index for node X when parent's realization is b

$$g(X, b) = \min\left\{m \in \left[0, \frac{\bar{r}}{1-\beta}\right] : \phi_{X,b}(m) \geq m\right\}$$

The "min ≥" captures the intuition of "fair value"

[KO03] Godfrey Keller and Alison Oldale. Branching bandits: a sequential search process with correlated pay-offs. Journal of Economic Theory, 113(2):302–315, 2003.

Gittins policy of [CPW+25] when given noisy $\mathcal{P}$

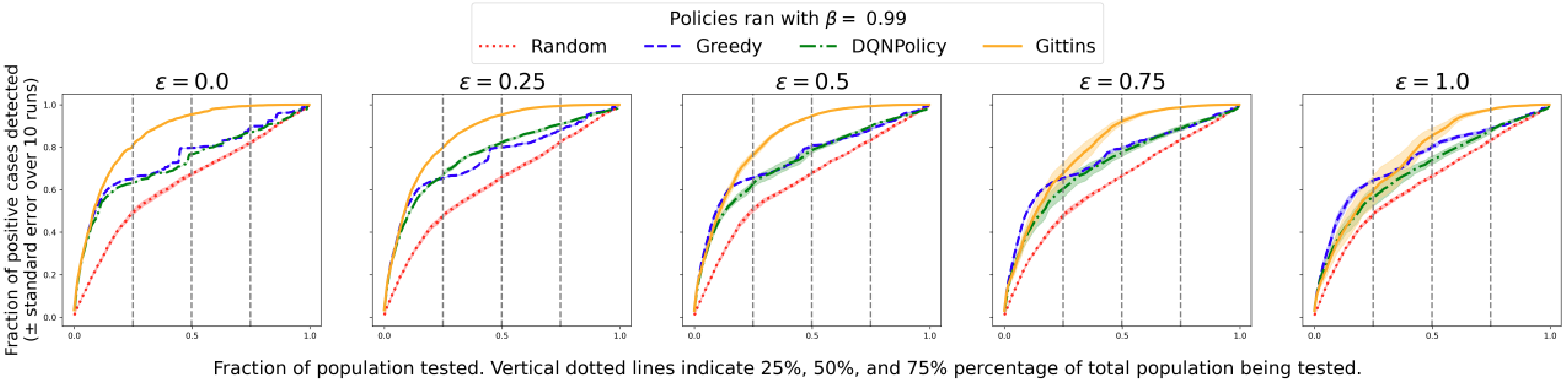


Figure 6: Experiments on the HIV dataset where policies only have access noisy version $\mathcal{Q}_\varepsilon$ of the underlying distribution $\mathcal{P}$ on the HIV dataset. Error bars illustrate the standard error due to 10 random instantiations of $\mathcal{Q}_\varepsilon$ for each corresponding value of ε.

[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.

[CPW+25] versus [KCK+26]

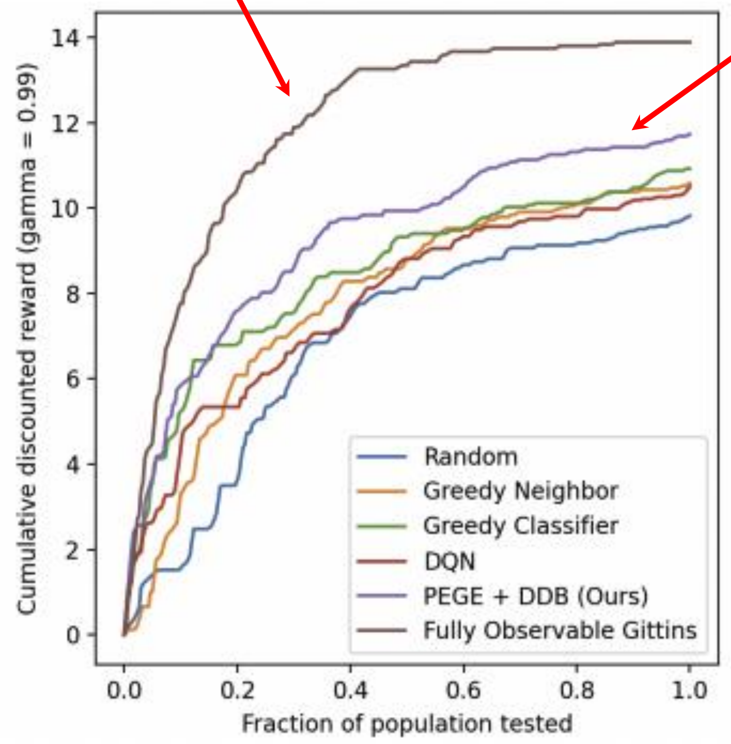


Figure 5: Performance of our method (PEGE + DDB) compared to several baselines and the FOG upper bound, as measured by cumulative discounted reward.

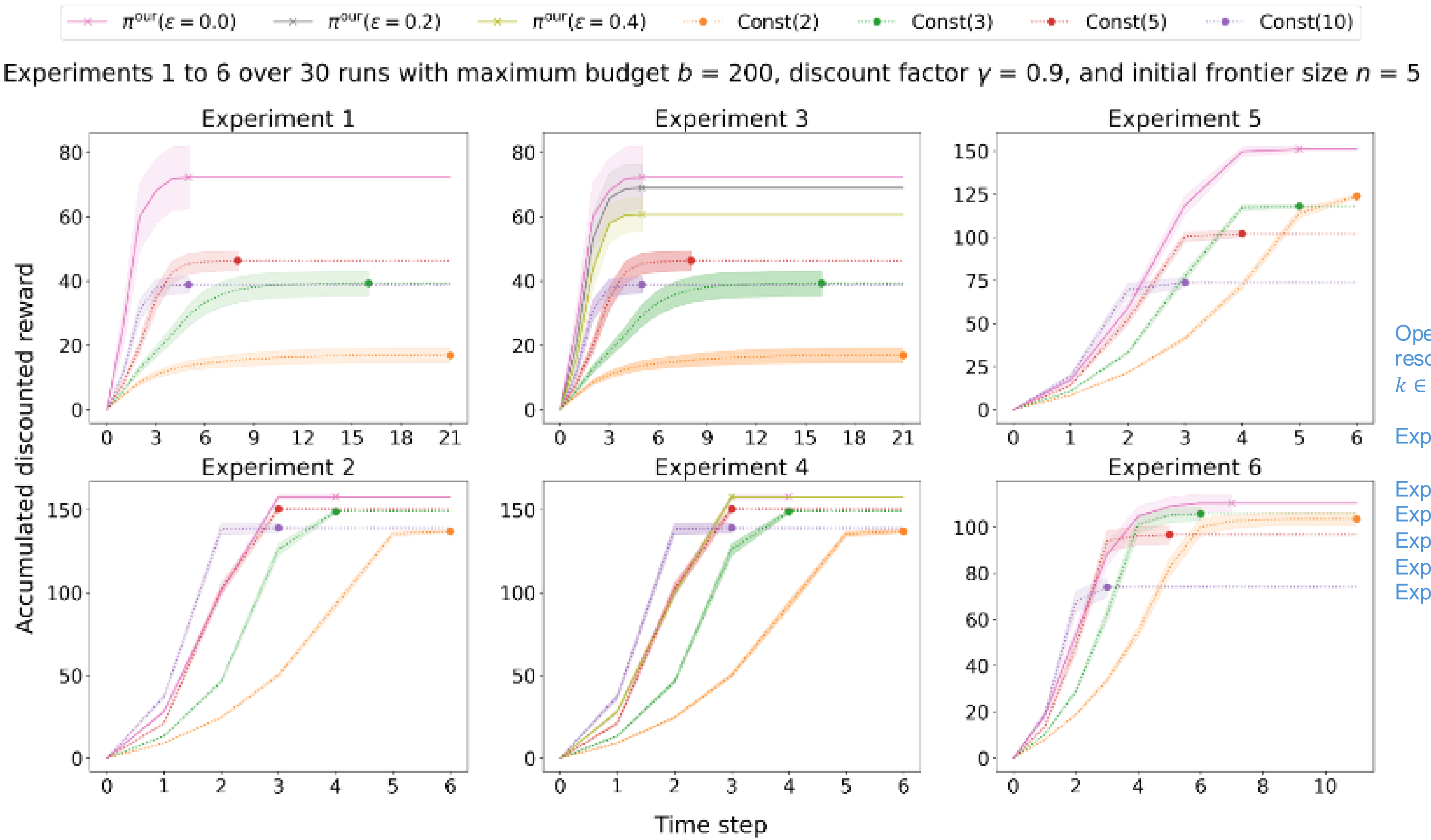
Algorithm	AUC	25%	50%	75%
Random	6.92	5.10	8.11	9.13
Greedy Neighbor	7.86	6.70	8.75	9.91
Greedy Classifier	8.45	7.12	9.32	10.1
DQN	7.78	6.11	8.81	9.75
PEGE + DDB (Ours)	**9.29**	**8.02**	**9.94**	**11.2**
Fully Observable Gittins	*12.0*	*11.1*	*13.3*	*13.7*

Table 1: Total AUC and cumulative discounted reward at different testing budgets (25%, 50%, 75% of population) for each algorithm. In each column, the best is **bolded** and the upper bound is *italicized*.

[CPW+25] Davin Choo*, Yuqi Pan*, Tonghan Wang, Milind Tambe, Alastair van Heerden, and Cheryl Johnson. Adaptive Frontier Exploration on Graphs with Applications to Network-Based Disease Testing. Conference on Neural Information Processing Systems (NeurIPS), 2025.

[KCK+26] Akseli Kangaslahti, Davin Choo, Lingkai Kong, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Policy-Embedded Graph Expansion: Networked HIV Testing with Diffusion-Driven Network Samples. Under submission, 2026.

Experimental plots of [PCW+26]



Operational guidelines typically assign 2 or 3 resources per individual. Here, we test Const(k) for $k \in \{2,3,5,10\}$

Exp 1: Synthetic Lomax referral distribution (i.e., discrete power law distribution)
Exp 2: Synthetic Uniform distribution
Exp 3: Noisy version of Exp 1
Exp 4: Noisy version of Exp 2
Exp 5: Distributions from real-world ICPSR dataset
Exp 6: Using same distributional information from Exp 5, but realize $X_{1:n}$ from actual graph neighborhoods

[PCW+26] Yuqi Pan*, Davin Choo*, Haichuan Wang, Milind Tambe, Alastair Van Heerden, Cheryl Johnson. Adaptive Multi-Round Allocation with Stochastic Arrivals. Under submission, 2026.
ICPSR dataset: https://www.icpsr.umich.edu/web/ICPSR/studies/22140/summary