

Online Stochastic Matching Under the First-Come-First-Matched Policy

Céline Comte

TU/e



LAAS
CNRS

IMT – Séminaire de Probabilités – November 14, 2023

- **Stochastic networks** \simeq when queueing theory meets graph theory
 - Product-form queueing systems (e.g., Jackson network \simeq zero-range process)
 - Multi-server systems
 - Matching systems

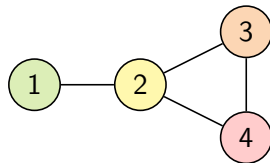
- **Stochastic networks** \simeq when queueing theory meets graph theory
 - Product-form queueing systems (e.g., Jackson network \simeq zero-range process)
 - Multi-server systems
 - Matching systems
- **Markov chains**
 - Explicit “product-form” stationary measures \simeq exponential families
 - Closed-form expression for the normalization constant = partition function
 - Balance equations, reversibility, and insensitivity

- **Stochastic networks** \simeq when queueing theory meets graph theory
 - Product-form queueing systems (e.g., Jackson network \simeq zero-range process)
 - Multi-server systems
 - Matching systems
- **Markov chains**
 - Explicit “product-form” stationary measures \simeq exponential families
 - Closed-form expression for the normalization constant = partition function
 - Balance equations, reversibility, and insensitivity
- **Reinforcement learning**
 - Policy-gradient methods \simeq stochastic gradient descent

Compatibility graph

Graph G undirected, connected, without self-loop, non-bipartite.

- Set \mathcal{V} of nodes \rightarrow item classes.
- Set \mathcal{V}_i of neighbors of node $i \rightarrow$ possible matches.

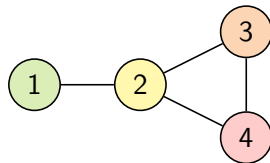


Compatibility graph

Graph G undirected, connected, without self-loop, non-bipartite.

- Set \mathcal{V} of nodes \rightarrow item classes.
- Set \mathcal{V}_i of neighbors of node $i \rightarrow$ possible matches.

Class- i items arrive as a **Poisson process** with rate μ_i , for each $i \in \mathcal{V}$.



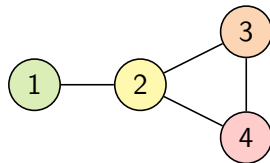
Compatibility graph

Graph G undirected, connected, without self-loop, non-bipartite.

- Set \mathcal{V} of nodes \rightarrow item classes.
- Set \mathcal{V}_i of neighbors of node $i \rightarrow$ possible matches.

Class- i items arrive as a **Poisson process** with rate μ_i , for each $i \in \mathcal{V}$.

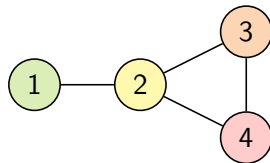
Items are matched according to the **first-come-first-matched (FCFM)** policy.



Compatibility graph

Graph G undirected, connected, without self-loop, non-bipartite.

- Set \mathcal{V} of nodes \rightarrow item classes.
- Set \mathcal{V}_i of neighbors of node $i \rightarrow$ possible matches.



Class- i items arrive as a **Poisson process** with rate μ_i , for each $i \in \mathcal{V}$.

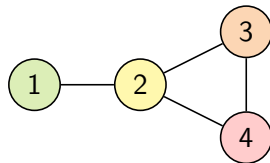
Items are matched according to the **first-come-first-matched (FCFM)** policy.



Compatibility graph

Graph G undirected, connected, without self-loop, non-bipartite.

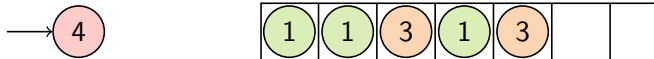
- Set \mathcal{V} of nodes \rightarrow item classes.
- Set \mathcal{V}_i of neighbors of node $i \rightarrow$ possible matches.



Class- i items arrive as a **Poisson process** with rate μ_i , for each $i \in \mathcal{V}$.

Items are matched according to the **first-come-first-matched (FCFM)** policy.

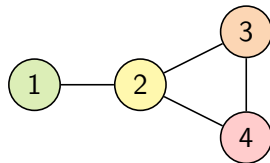
New item



Compatibility graph

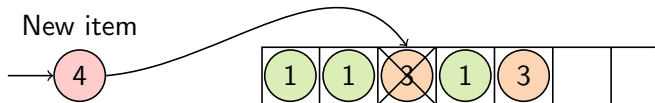
Graph G undirected, connected, without self-loop, non-bipartite.

- Set \mathcal{V} of nodes \rightarrow item classes.
- Set \mathcal{V}_i of neighbors of node $i \rightarrow$ possible matches.



Class- i items arrive as a **Poisson process** with rate μ_i , for each $i \in \mathcal{V}$.

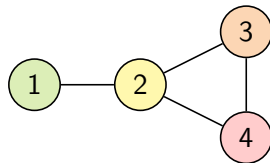
Items are matched according to the **first-come-first-matched (FCFM)** policy.



Compatibility graph

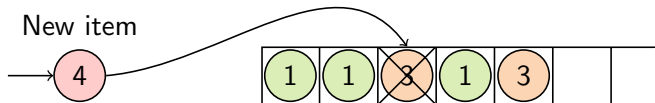
Graph G undirected, connected, without self-loop, non-bipartite.

- Set \mathcal{V} of nodes \rightarrow item classes.
- Set \mathcal{V}_i of neighbors of node $i \rightarrow$ possible matches.



Class- i items arrive as a **Poisson process** with rate μ_i , for each $i \in \mathcal{V}$.

Items are matched according to the **first-come-first-matched (FCFM)** policy.

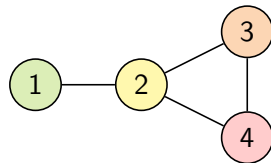


The evolution of the sequence of unmatched item classes defines a **Markov process** whose transition rates depend on the graph G and the arrival rates μ_i , $i \in \mathcal{V}$.

Neighbors and independent sets

Set of neighbors

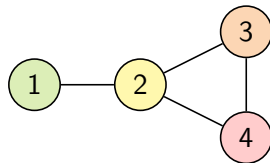
- The neighbor set of a class $i \in \mathcal{V}$ is \mathcal{V}_i .



Neighbors and independent sets

Set of neighbors

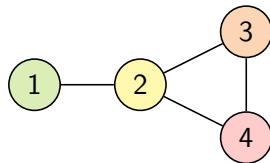
- The neighbor set of a class $i \in \mathcal{V}$ is \mathcal{V}_i .
- The neighbor set of a set $\mathcal{I} \subseteq \mathcal{V}$ is $\mathcal{V}(\mathcal{I}) = \bigcup_{i \in \mathcal{I}} \mathcal{V}_i$.



Neighbors and independent sets

Set of neighbors

- The neighbor set of a class $i \in \mathcal{V}$ is \mathcal{V}_i .
- The neighbor set of a set $\mathcal{I} \subseteq \mathcal{V}$ is $\mathcal{V}(\mathcal{I}) = \bigcup_{i \in \mathcal{I}} \mathcal{V}_i$.



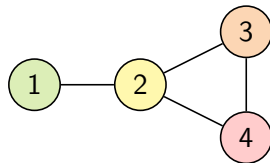
Independent sets

- A node set $\mathcal{I} \subseteq \mathcal{V}$ is an **independent set** if its elements are (pairwise) nonadjacent.

Neighbors and independent sets

Set of neighbors

- The neighbor set of a class $i \in \mathcal{V}$ is \mathcal{V}_i .
- The neighbor set of a set $\mathcal{I} \subseteq \mathcal{V}$ is $\mathcal{V}(\mathcal{I}) = \bigcup_{i \in \mathcal{I}} \mathcal{V}_i$.



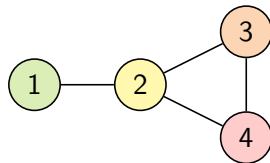
Independent sets

- A node set $\mathcal{I} \subseteq \mathcal{V}$ is an **independent set** if its elements are (pairwise) nonadjacent. Equivalently, a node set $\mathcal{I} \subseteq \mathcal{V}$ is an independent set if $\mathcal{I} \cap \mathcal{V}(\mathcal{I}) = \emptyset$.

Neighbors and independent sets

Set of neighbors

- The neighbor set of a class $i \in \mathcal{V}$ is \mathcal{V}_i .
- The neighbor set of a set $\mathcal{I} \subseteq \mathcal{V}$ is $\mathcal{V}(\mathcal{I}) = \bigcup_{i \in \mathcal{I}} \mathcal{V}_i$.



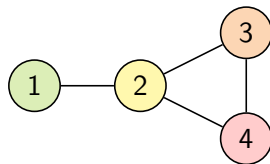
Independent sets

- A node set $\mathcal{I} \subseteq \mathcal{V}$ is an **independent set** if its elements are (pairwise) nonadjacent. Equivalently, a node set $\mathcal{I} \subseteq \mathcal{V}$ is an independent set if $\mathcal{I} \cap \mathcal{V}(\mathcal{I}) = \emptyset$.
- The family of independent sets of the graph G is denoted by \mathbb{I} .

Neighbors and independent sets

Set of neighbors

- The neighbor set of a class $i \in \mathcal{V}$ is \mathcal{V}_i .
- The neighbor set of a set $\mathcal{I} \subseteq \mathcal{V}$ is $\mathcal{V}(\mathcal{I}) = \bigcup_{i \in \mathcal{I}} \mathcal{V}_i$.



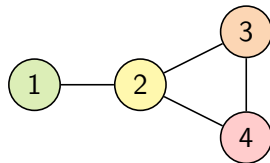
Independent sets

- A node set $\mathcal{I} \subseteq \mathcal{V}$ is an **independent set** if its elements are (pairwise) nonadjacent. Equivalently, a node set $\mathcal{I} \subseteq \mathcal{V}$ is an independent set if $\mathcal{I} \cap \mathcal{V}(\mathcal{I}) = \emptyset$.
- The family of independent sets of the graph G is denoted by \mathbb{I} . In our toy example, $\mathbb{I} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 3\}, \{1, 4\}\}$.

Neighbors and independent sets

Set of neighbors

- The neighbor set of a class $i \in \mathcal{V}$ is \mathcal{V}_i .
- The neighbor set of a set $\mathcal{I} \subseteq \mathcal{V}$ is $\mathcal{V}(\mathcal{I}) = \bigcup_{i \in \mathcal{I}} \mathcal{V}_i$.



Independent sets

- A node set $\mathcal{I} \subseteq \mathcal{V}$ is an **independent set** if its elements are (pairwise) nonadjacent. Equivalently, a node set $\mathcal{I} \subseteq \mathcal{V}$ is an independent set if $\mathcal{I} \cap \mathcal{V}(\mathcal{I}) = \emptyset$.
- The family of independent sets of the graph G is denoted by \mathbb{I} . In our toy example, $\mathbb{I} = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 3\}, \{1, 4\}\}$.
- We let $\mathbb{I}^+ = \mathbb{I} \setminus \{\emptyset\}$ denote the family of nonempty independent sets.

- 1 Literature review
- 2 Basic results
 - Stability condition
 - Definition of the discrete-time Markov chain
 - Product-form stationary distribution and partial balance
- 3 Performance analysis
 - Normalization constant
 - Performance metrics
 - Heavy-traffic analysis
- 4 Concluding remarks

- 1 Literature review
- 2 Basic results
 - Stability condition
 - Definition of the discrete-time Markov chain
 - Product-form stationary distribution and partial balance
- 3 Performance analysis
 - Normalization constant
 - Performance metrics
 - Heavy-traffic analysis
- 4 Concluding remarks

Early works

- René Caldentey, Edward H. Kaplan, and Gideon Weiss. “FCFS Infinite Bipartite Matching of sServers and Customers”. *Advances in Applied Probability* 41.3 (Sept. 2009), pp. 695–730. DOI: [10.1239/aap/1253281061](https://doi.org/10.1239/aap/1253281061)
- Ivo Adan and Gideon Weiss. “Exact FCFS Matching Rates for Two Infinite Multitype Sequences”. *Operations Research* 60.2 (Apr. 1, 2012), pp. 475–489. DOI: [10.1287/opre.1110.1027](https://doi.org/10.1287/opre.1110.1027)

Early works

- René Caldentey, Edward H. Kaplan, and Gideon Weiss. “FCFS Infinite Bipartite Matching of sServers and Customers”. *Advances in Applied Probability* 41.3 (Sept. 2009), pp. 695–730. DOI: [10.1239/aap/1253281061](https://doi.org/10.1239/aap/1253281061)
- Ivo Adan and Gideon Weiss. “Exact FCFS Matching Rates for Two Infinite Multitype Sequences”. *Operations Research* 60.2 (Apr. 1, 2012), pp. 475–489. DOI: [10.1287/opre.1110.1027](https://doi.org/10.1287/opre.1110.1027)

Stability (i.e., positive recurrence) condition

- Ana Bušić, Varun Gupta, and Jean Mairesse. “Stability of the Bipartite Matching Model”. *Advances in Applied Probability* 45.2 (June 2013), pp. 351–378. DOI: [10.1239/aap/1370870122](https://doi.org/10.1239/aap/1370870122)

Performance evaluation

- Ivo Adan et al. “Reversibility and Further Properties of FCFS Infinite Bipartite Matching”. *Mathematics of Operations Research* 43.2 (Dec. 12, 2017), pp. 598–621. DOI: [10.1287/moor.2017.0874](https://doi.org/10.1287/moor.2017.0874)
- Céline Comte and Jan-Pieter Dorsman. “Performance Evaluation of Stochastic Bipartite Matching Models”. *Performance Engineering and Stochastic Modeling*. Ed. by Paolo Ballarini et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 425–440. DOI: [10.1007/978-3-030-91825-5_26](https://doi.org/10.1007/978-3-030-91825-5_26)

Stability

- Jean Mairesse and Pascal Moyal. “Stability of the Stochastic Matching Model”. *Journal of Applied Probability* 53.4 (Dec. 2016), pp. 1064–1077. DOI: [10.1017/jpr.2016.65](https://doi.org/10.1017/jpr.2016.65)

Stability

- Jean Mairesse and Pascal Moyal. “Stability of the Stochastic Matching Model”. *Journal of Applied Probability* 53.4 (Dec. 2016), pp. 1064–1077. DOI: [10.1017/jpr.2016.65](https://doi.org/10.1017/jpr.2016.65)

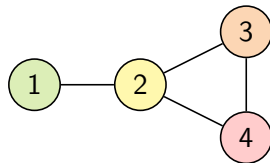
Performance evaluation

- Pascal Moyal, Ana Bušić, and Jean Mairesse. “A Product Form for the General Stochastic Matching Model”. *Journal of Applied Probability* 58.2 (June 2021), pp. 449–468. DOI: [10.1017/jpr.2020.100](https://doi.org/10.1017/jpr.2020.100)
- Céline Comte. “Stochastic Non-Bipartite Matching Models and Order-Independent Loss Queues”. *Stochastic Models* 38.1 (Jan. 2, 2022), pp. 1–36. DOI: [10.1080/15326349.2021.1962352](https://doi.org/10.1080/15326349.2021.1962352)

- 1 Literature review
- 2 Basic results
 - Stability condition
 - Definition of the discrete-time Markov chain
 - Product-form stationary distribution and partial balance
- 3 Performance analysis
 - Normalization constant
 - Performance metrics
 - Heavy-traffic analysis
- 4 Concluding remarks

Stability condition

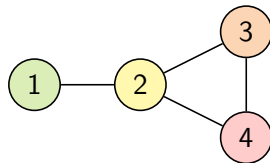
- Studied in [BGM13], [MM16], and [MBM21]



Stability condition

- Studied in [BGM13], [MM16], and [MBM21]
- The Markov process is positive recurrent if and only if

$$\rho(\mathcal{I}) \triangleq \frac{\sum_{i \in \mathcal{I}} \mu_i}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i} < 1, \quad \mathcal{I} \in \mathbb{I}^+.$$



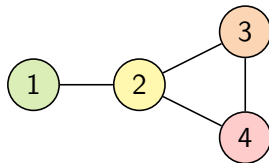
Stability condition

- Studied in [BGM13], [MM16], and [MBM21]
- The Markov process is positive recurrent if and only if

$$\rho(\mathcal{I}) \triangleq \frac{\sum_{i \in \mathcal{I}} \mu_i}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i} < 1, \quad \mathcal{I} \in \mathbb{I}^+.$$

In our toy example:

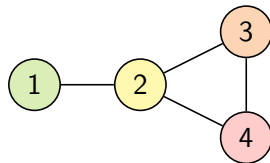
$$\begin{cases} \rho(\{1\}) = \frac{\mu_1}{\mu_2}, & \rho(\{2\}) = \frac{\mu_2}{\mu_1 + \mu_3 + \mu_4}, & \rho(\{3\}) = \frac{\mu_3}{\mu_2 + \mu_4}, \\ \rho(\{4\}) = \frac{\mu_4}{\mu_2 + \mu_3}, & \rho(\{1, 3\}) = \frac{\mu_1 + \mu_3}{\mu_2 + \mu_4}, & \rho(\{1, 4\}) = \frac{\mu_1 + \mu_4}{\mu_2 + \mu_3}. \end{cases}$$



Stability condition

- Studied in [BGM13], [MM16], and [MBM21]
- The Markov process is positive recurrent if and only if

$$\rho(\mathcal{I}) \triangleq \frac{\sum_{i \in \mathcal{I}} \mu_i}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i} < 1, \quad \mathcal{I} \in \mathbb{I}^+.$$



In our toy example:

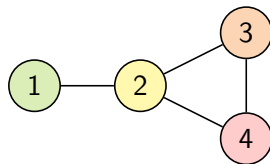
$$\begin{cases} \rho(\{1\}) = \frac{\mu_1}{\mu_2}, & \rho(\{2\}) = \frac{\mu_2}{\mu_1 + \mu_3 + \mu_4}, & \rho(\{3\}) = \frac{\mu_3}{\mu_2 + \mu_4}, \\ \rho(\{4\}) = \frac{\mu_4}{\mu_2 + \mu_3}, & \rho(\{1, 3\}) = \frac{\mu_1 + \mu_3}{\mu_2 + \mu_4}, & \rho(\{1, 4\}) = \frac{\mu_1 + \mu_4}{\mu_2 + \mu_3}. \end{cases}$$

- (This condition can be satisfied only if the graph is non-bipartite.)

- **Sequence of unmatched items ordered by arrival times:**

$$c = (c_1, c_2, \dots, c_\ell) \in \mathcal{V}^*,$$

where c_1 is the class of the oldest unmatched item.



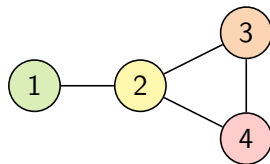
State and state space

- **Sequence of unmatched items ordered by arrival times:**

$$c = (c_1, c_2, \dots, c_\ell) \in \mathcal{V}^*,$$

where c_1 is the class of the oldest unmatched item.

- The empty state, with $\ell = 0$, is denoted by \emptyset .

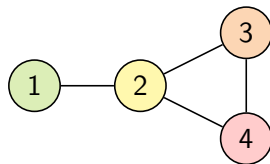


- **Sequence of unmatched items ordered by arrival times:**

$$c = (c_1, c_2, \dots, c_\ell) \in \mathcal{V}^*,$$

where c_1 is the class of the oldest unmatched item.

- The empty state, with $\ell = 0$, is denoted by \emptyset .
- The set of unmatched items is an **independent set**



- **Sequence of unmatched items ordered by arrival times:**

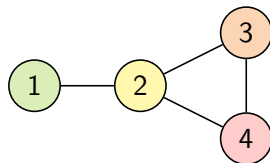
$$c = (c_1, c_2, \dots, c_\ell) \in \mathcal{V}^*,$$

where c_1 is the class of the oldest unmatched item.

- The empty state, with $\ell = 0$, is denoted by \emptyset .
- The set of unmatched items is an **independent set**, meaning that the state space is

$$\mathcal{C} = \bigcup_{\mathcal{I} \in \mathbb{I}} \mathcal{C}_{\mathcal{I}},$$

where $\mathcal{C}_{\mathcal{I}}$ is the set of sequences $c \in \mathcal{V}^*$ where the set of classes is \mathcal{I} , for each $\mathcal{I} \in \mathbb{I}$



- **Sequence of unmatched items ordered by arrival times:**

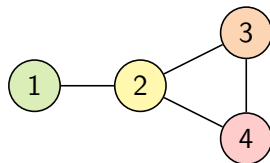
$$c = (c_1, c_2, \dots, c_\ell) \in \mathcal{V}^*,$$

where c_1 is the class of the oldest unmatched item.

- The empty state, with $\ell = 0$, is denoted by \emptyset .
- The set of unmatched items is an **independent set**, meaning that the state space is

$$\mathcal{C} = \bigcup_{\mathcal{I} \in \mathbb{I}} \mathcal{C}_{\mathcal{I}},$$

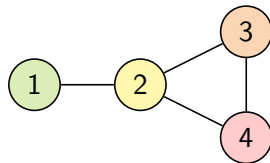
where $\mathcal{C}_{\mathcal{I}}$ is the set of sequences $c \in \mathcal{V}^*$ where the set of classes is \mathcal{I} , for each $\mathcal{I} \in \mathbb{I}$ (with the convention that $\mathcal{C}_{\emptyset} = \{\emptyset\}$).



Product-form stationary distribution

- Stationary distribution [MM16]: for $(c_1, c_2, \dots, c_\ell) \in \mathcal{C} \setminus \{\emptyset\}$,

$$\pi(c_1, c_2, \dots, c_\ell) = \pi(\emptyset) \prod_{p=1}^{\ell} \frac{\mu_{c_p}}{\sum_{i \in \mathcal{V}(\{c_1, c_2, \dots, c_p\})} \mu_i}.$$



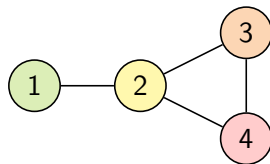
Product-form stationary distribution

- Stationary distribution [MM16]: for $(c_1, c_2, \dots, c_\ell) \in \mathcal{C} \setminus \{\emptyset\}$,

$$\pi(c_1, c_2, \dots, c_\ell) = \pi(\emptyset) \prod_{p=1}^{\ell} \frac{\mu_{c_p}}{\sum_{i \in \mathcal{V}(\{c_1, c_2, \dots, c_p\})} \mu_i}.$$

- In our toy example:

$$\pi(\textcircled{1}, \textcircled{1}, \textcircled{3}, \textcircled{1}, \textcircled{3})$$



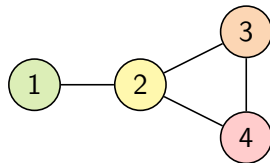
Product-form stationary distribution

- Stationary distribution [MM16]: for $(c_1, c_2, \dots, c_\ell) \in \mathcal{C} \setminus \{\emptyset\}$,

$$\pi(c_1, c_2, \dots, c_\ell) = \pi(\emptyset) \prod_{p=1}^{\ell} \frac{\mu_{c_p}}{\sum_{i \in \mathcal{V}(\{c_1, c_2, \dots, c_p\})} \mu_i}.$$

- In our toy example:

$$\pi(\textcircled{1}, \textcircled{1}, \textcircled{3}, \textcircled{1}, \textcircled{3}) = \pi(\emptyset)$$



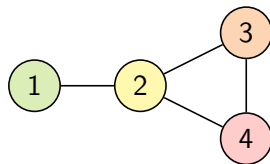
Product-form stationary distribution

- Stationary distribution [MM16]: for $(c_1, c_2, \dots, c_\ell) \in \mathcal{C} \setminus \{\emptyset\}$,

$$\pi(c_1, c_2, \dots, c_\ell) = \pi(\emptyset) \prod_{p=1}^{\ell} \frac{\mu_{c_p}}{\sum_{i \in \mathcal{V}(\{c_1, c_2, \dots, c_p\})} \mu_i}.$$

- In our toy example:

$$\pi(\textcircled{1}, \textcircled{1}, \textcircled{3}, \textcircled{1}, \textcircled{3}) = \pi(\emptyset) \frac{\mu_1}{\mu_2}$$



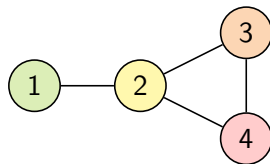
Product-form stationary distribution

- Stationary distribution [MM16]: for $(c_1, c_2, \dots, c_\ell) \in \mathcal{C} \setminus \{\emptyset\}$,

$$\pi(c_1, c_2, \dots, c_\ell) = \pi(\emptyset) \prod_{p=1}^{\ell} \frac{\mu_{c_p}}{\sum_{i \in \mathcal{V}(\{c_1, c_2, \dots, c_p\})} \mu_i}.$$

- In our toy example:

$$\pi(\textcircled{1}, \textcircled{1}, \textcircled{3}, \textcircled{1}, \textcircled{3}) = \pi(\emptyset) \frac{\mu_1}{\mu_2} \frac{\mu_1}{\mu_2}$$



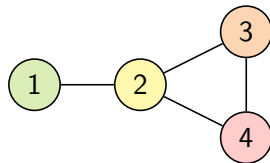
Product-form stationary distribution

- Stationary distribution [MM16]: for $(c_1, c_2, \dots, c_\ell) \in \mathcal{C} \setminus \{\emptyset\}$,

$$\pi(c_1, c_2, \dots, c_\ell) = \pi(\emptyset) \prod_{p=1}^{\ell} \frac{\mu_{c_p}}{\sum_{i \in \mathcal{V}(\{c_1, c_2, \dots, c_p\})} \mu_i}.$$

- In our toy example:

$$\pi(\textcircled{1}, \textcircled{1}, \textcircled{3}, \textcircled{1}, \textcircled{3}) = \pi(\emptyset) \frac{\mu_1}{\mu_2} \frac{\mu_1}{\mu_2} \frac{\mu_3}{\mu_2 + \mu_4}$$



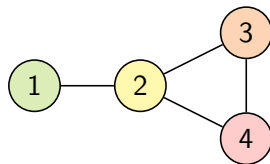
Product-form stationary distribution

- Stationary distribution [MM16]: for $(c_1, c_2, \dots, c_\ell) \in \mathcal{C} \setminus \{\emptyset\}$,

$$\pi(c_1, c_2, \dots, c_\ell) = \pi(\emptyset) \prod_{p=1}^{\ell} \frac{\mu_{c_p}}{\sum_{i \in \mathcal{V}(\{c_1, c_2, \dots, c_p\})} \mu_i}.$$

- In our toy example:

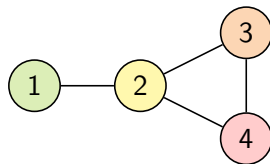
$$\pi(\textcircled{1}, \textcircled{1}, \textcircled{3}, \textcircled{1}, \textcircled{3}) = \pi(\emptyset) \frac{\mu_1}{\mu_2} \frac{\mu_1}{\mu_2} \frac{\mu_3}{\mu_2 + \mu_4} \frac{\mu_1}{\mu_2 + \mu_4}$$



Product-form stationary distribution

- Stationary distribution [MM16]: for $(c_1, c_2, \dots, c_\ell) \in \mathcal{C} \setminus \{\emptyset\}$,

$$\pi(c_1, c_2, \dots, c_\ell) = \pi(\emptyset) \prod_{p=1}^{\ell} \frac{\mu_{c_p}}{\sum_{i \in \mathcal{V}(\{c_1, c_2, \dots, c_p\})} \mu_i}.$$



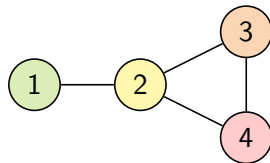
- In our toy example:

$$\pi(\textcircled{1}, \textcircled{1}, \textcircled{3}, \textcircled{1}, \textcircled{3}) = \pi(\emptyset) \frac{\mu_1}{\mu_2} \frac{\mu_1}{\mu_2} \frac{\mu_3}{\mu_2 + \mu_4} \frac{\mu_1}{\mu_2 + \mu_4} \frac{\mu_3}{\mu_2 + \mu_4}.$$

Product-form stationary distribution

- Stationary distribution [MM16]: for $(c_1, c_2, \dots, c_\ell) \in \mathcal{C} \setminus \{\emptyset\}$,

$$\pi(c_1, c_2, \dots, c_\ell) = \pi(\emptyset) \prod_{p=1}^{\ell} \frac{\mu_{c_p}}{\sum_{i \in \mathcal{V}(\{c_1, c_2, \dots, c_p\})} \mu_i}.$$



- In our toy example:

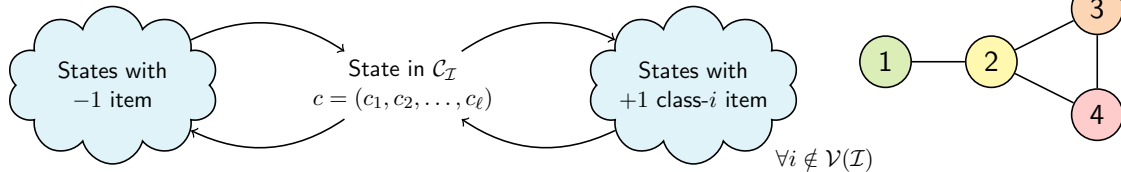
$$\pi(\textcircled{1}, \textcircled{1}, \textcircled{3}, \textcircled{1}, \textcircled{3}) = \pi(\emptyset) \frac{\mu_1}{\mu_2} \frac{\mu_1}{\mu_2} \frac{\mu_3}{\mu_2 + \mu_4} \frac{\mu_1}{\mu_2 + \mu_4} \frac{\mu_3}{\mu_2 + \mu_4}.$$

- In other words, for each $\mathcal{I} \in \mathbb{I}^+$ and $(c_1, c_2, \dots, c_{\ell-1}, c_\ell) \in \mathcal{C}_{\mathcal{I}}$,

$$\pi(c_1, c_2, \dots, c_{\ell-1}, c_\ell) = \pi(c_1, c_2, \dots, c_{\ell-1}) \frac{\mu_{c_\ell}}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i}.$$

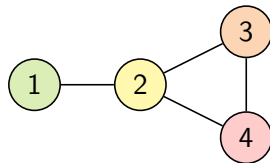
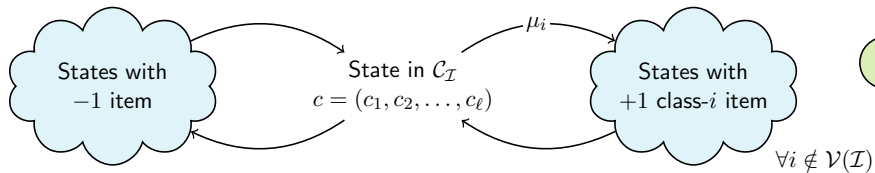
Direct proof using partial balance

Transitions out of and into a state



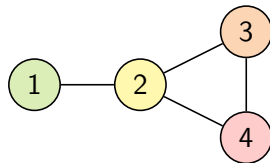
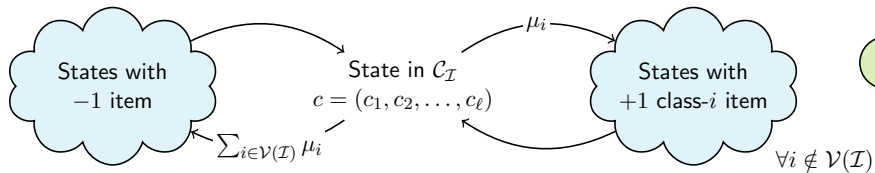
Direct proof using partial balance

Transitions out of and into a state



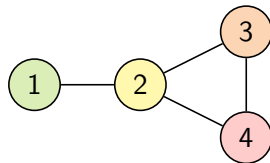
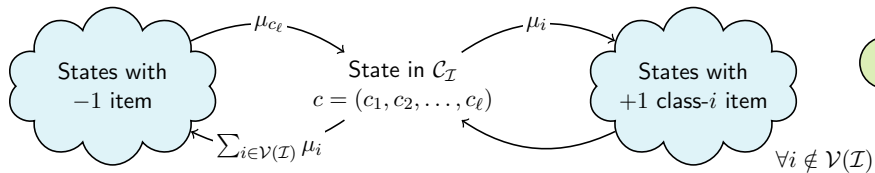
Direct proof using partial balance

Transitions out of and into a state



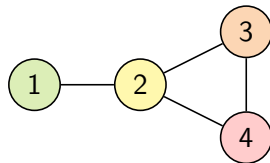
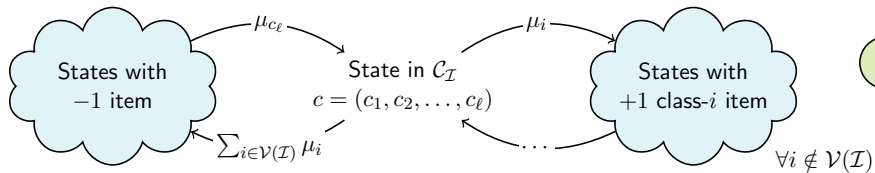
Direct proof using partial balance

Transitions out of and into a state



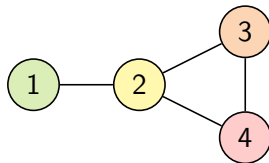
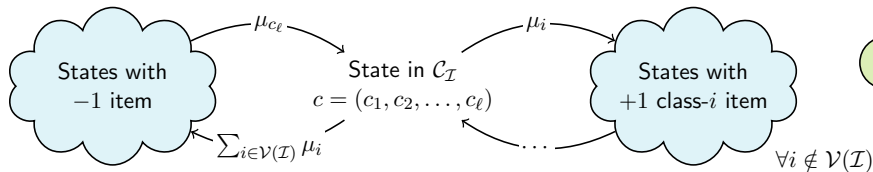
Direct proof using partial balance

Transitions out of and into a state



Direct proof using partial balance

Transitions out of and into a state



“Partial” balance equations

- Balance between state c and states with -1 item:

$$\pi(c_1, \dots, c_\ell) \left(\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i \right) = \pi(c_1, \dots, c_{\ell-1}) \mu_{c_\ell},$$

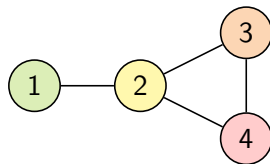
- Balance between state c and states with $+1$ class- i item:

$$\pi(c_1, \dots, c_\ell) \mu_i = \sum_{p=1}^{\ell+1} \pi(c_1, \dots, c_{p-1}, i, c_p, \dots, c_\ell) \left(\sum_{j \in \mathcal{V}_i \setminus \mathcal{V}(\{c_1, \dots, c_{p-1}\})} \mu_j \right), \quad i \notin \mathcal{V}(\mathcal{I}).$$

- 1 Literature review
- 2 Basic results
 - Stability condition
 - Definition of the discrete-time Markov chain
 - Product-form stationary distribution and partial balance
- 3 Performance analysis**
 - Normalization constant
 - Performance metrics
 - Heavy-traffic analysis
- 4 Concluding remarks

- Stationary distribution of the **set of unmatched item classes**:

$$\pi(\mathcal{I}) = \sum_{c \in \mathcal{C}_{\mathcal{I}}} \pi(c), \quad \mathcal{I} \in \mathbb{I}.$$



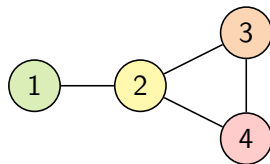
Normalization constant

- Stationary distribution of the **set of unmatched item classes**:

$$\pi(\mathcal{I}) = \sum_{c \in \mathcal{C}_{\mathcal{I}}} \pi(c), \quad \mathcal{I} \in \mathbb{I}.$$

- We prove that this distribution satisfies the recursion

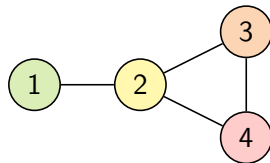
$$\pi(\mathcal{I}) = \frac{\rho(\mathcal{I})}{1 - \rho(\mathcal{I})} \left(\sum_{i \in \mathcal{I}} \frac{\mu_i}{\sum_{j \in \mathcal{I}} \mu_j} \pi(\mathcal{I} \setminus \{i\}) \right), \quad \mathcal{I} \in \mathbb{I}^+.$$



Normalization constant

- Stationary distribution of the **set of unmatched item classes**:

$$\pi(\mathcal{I}) = \sum_{c \in \mathcal{C}_{\mathcal{I}}} \pi(c), \quad \mathcal{I} \in \mathbb{I}.$$



- We prove that this distribution satisfies the recursion

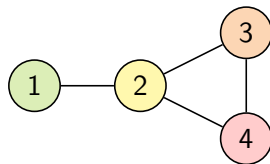
$$\pi(\mathcal{I}) = \frac{\rho(\mathcal{I})}{1 - \rho(\mathcal{I})} \left(\sum_{i \in \mathcal{I}} \frac{\mu_i}{\sum_{j \in \mathcal{I}} \mu_j} \pi(\mathcal{I} \setminus \{i\}) \right), \quad \mathcal{I} \in \mathbb{I}^+.$$

The normalization constant $\pi(\emptyset)$ follows from the normalization equation $\sum_{\mathcal{I} \in \mathbb{I}} \pi(\mathcal{I}) = 1$.

Normalization constant

- Stationary distribution of the **set of unmatched item classes**:

$$\pi(\mathcal{I}) = \sum_{c \in \mathcal{C}_{\mathcal{I}}} \pi(c), \quad \mathcal{I} \in \mathbb{I}.$$



- We prove that this distribution satisfies the recursion

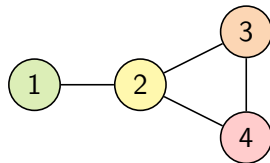
$$\pi(\mathcal{I}) = \frac{\rho(\mathcal{I})}{1 - \rho(\mathcal{I})} \left(\sum_{i \in \mathcal{I}} \frac{\mu_i}{\sum_{j \in \mathcal{I}} \mu_j} \pi(\mathcal{I} \setminus \{i\}) \right), \quad \mathcal{I} \in \mathbb{I}^+.$$

The normalization constant $\pi(\emptyset)$ follows from the normalization equation $\sum_{\mathcal{I} \in \mathbb{I}} \pi(\mathcal{I}) = 1$.

- Key step: $\pi(c_1, \dots, c_\ell, i) = \pi(c_1, \dots, c_\ell) \frac{\mu_i}{\sum_{j \in \mathcal{V}(\mathcal{I})} \mu_j}$, $\mathcal{I} \in \mathbb{I}^+$, $i \in \mathcal{I}$, $(c_1, \dots, c_\ell) \in \mathcal{C}_{\mathcal{I}} \cup \mathcal{C}_{\mathcal{I} \setminus \{i\}}$.

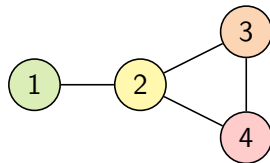
- **Waiting probability** of class i :

$$\omega_i = \sum_{\mathcal{I} \in \mathbb{I}: i \notin \mathcal{V}(\mathcal{I})} \pi(\mathcal{I}),$$



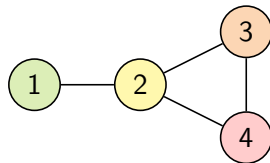
- **Waiting probability** of class i :

$$\omega_i = \sum_{\mathcal{I} \in \mathbb{I}: i \notin \mathcal{V}(\mathcal{I})} \pi(\mathcal{I}), \quad \text{which implies} \quad \frac{\sum_{i \in \mathcal{V}} \mu_i \omega_i}{\sum_{i \in \mathcal{V}} \mu_i} = \frac{1}{2}.$$



- **Waiting probability** of class i :

$$\omega_i = \sum_{\mathcal{I} \in \mathbb{I}: i \notin \mathcal{V}(\mathcal{I})} \pi(\mathcal{I}), \quad \text{which implies} \quad \frac{\sum_{i \in \mathcal{V}} \mu_i \omega_i}{\sum_{i \in \mathcal{V}} \mu_i} = \frac{1}{2}.$$

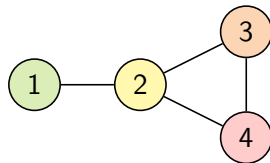


- **Mean queue length**:

$$L = \sum_{\mathcal{I} \in \mathbb{I}^+} \ell(\mathcal{I}), \quad \text{where} \quad \ell(\mathcal{I}) = \frac{\pi(\mathcal{I})}{1 - \rho(\mathcal{I})} + \frac{\rho(\mathcal{I})}{1 - \rho(\mathcal{I})} \left(\sum_{i \in \mathcal{I}} \frac{\mu_i}{\sum_{j \in \mathcal{I}} \mu_j} \ell(\mathcal{I} \setminus \{i\}) \right).$$

- **Waiting probability** of class i :

$$\omega_i = \sum_{\mathcal{I} \in \mathbb{I}: i \notin \mathcal{V}(\mathcal{I})} \pi(\mathcal{I}), \quad \text{which implies} \quad \frac{\sum_{i \in \mathcal{V}} \mu_i \omega_i}{\sum_{i \in \mathcal{V}} \mu_i} = \frac{1}{2}.$$



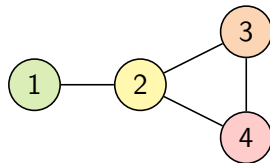
- **Mean queue length**:

$$L = \sum_{\mathcal{I} \in \mathbb{I}^+} \ell(\mathcal{I}), \quad \text{where} \quad \ell(\mathcal{I}) = \frac{\pi(\mathcal{I})}{1 - \rho(\mathcal{I})} + \frac{\rho(\mathcal{I})}{1 - \rho(\mathcal{I})} \left(\sum_{i \in \mathcal{I}} \frac{\mu_i}{\sum_{j \in \mathcal{I}} \mu_j} \ell(\mathcal{I} \setminus \{i\}) \right).$$

Similar formulas for the per-class performance.

- **Waiting probability** of class i :

$$\omega_i = \sum_{\mathcal{I} \in \mathbb{I}: i \notin \mathcal{V}(\mathcal{I})} \pi(\mathcal{I}), \quad \text{which implies} \quad \frac{\sum_{i \in \mathcal{V}} \mu_i \omega_i}{\sum_{i \in \mathcal{V}} \mu_i} = \frac{1}{2}.$$



- **Mean queue length**:

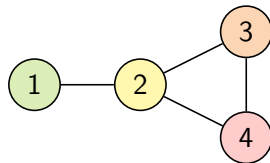
$$L = \sum_{\mathcal{I} \in \mathbb{I}^+} \ell(\mathcal{I}), \quad \text{where} \quad \ell(\mathcal{I}) = \frac{\pi(\mathcal{I})}{1 - \rho(\mathcal{I})} + \frac{\rho(\mathcal{I})}{1 - \rho(\mathcal{I})} \left(\sum_{i \in \mathcal{I}} \frac{\mu_i}{\sum_{j \in \mathcal{I}} \mu_j} \ell(\mathcal{I} \setminus \{i\}) \right).$$

Similar formulas for the per-class performance.

- The **mean waiting time** follows by applying Little's law.

- **Waiting probability** of class i :

$$\omega_i = \sum_{\mathcal{I} \in \mathbb{I}: i \notin \mathcal{V}(\mathcal{I})} \pi(\mathcal{I}), \quad \text{which implies} \quad \frac{\sum_{i \in \mathcal{V}} \mu_i \omega_i}{\sum_{i \in \mathcal{V}} \mu_i} = \frac{1}{2}.$$



- **Mean queue length**:

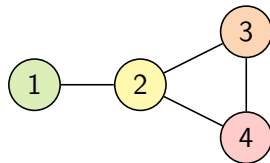
$$L = \sum_{\mathcal{I} \in \mathbb{I}^+} \ell(\mathcal{I}), \quad \text{where} \quad \ell(\mathcal{I}) = \frac{\pi(\mathcal{I})}{1 - \rho(\mathcal{I})} + \frac{\rho(\mathcal{I})}{1 - \rho(\mathcal{I})} \left(\sum_{i \in \mathcal{I}} \frac{\mu_i}{\sum_{j \in \mathcal{I}} \mu_j} \ell(\mathcal{I} \setminus \{i\}) \right).$$

Similar formulas for the per-class performance.

- The **mean waiting time** follows by applying Little's law.
- Similar results for the bipartite variant of the model [CD21].

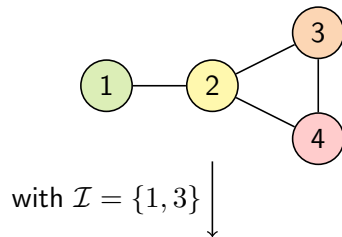
Heavy-traffic analysis

- Independent set $\mathcal{I} \in \mathbb{I}$ that is **maximal** for the inclusion, meaning that $\mathcal{V}(\mathcal{I}) = \mathcal{V} \setminus \mathcal{I}$.



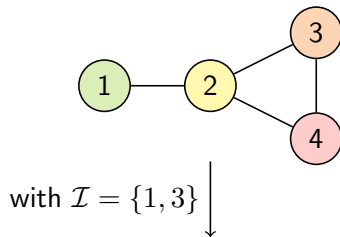
Heavy-traffic analysis

- Independent set $\mathcal{I} \in \mathbb{I}$ that is **maximal** for the inclusion, meaning that $\mathcal{V}(\mathcal{I}) = \mathcal{V} \setminus \mathcal{I}$.



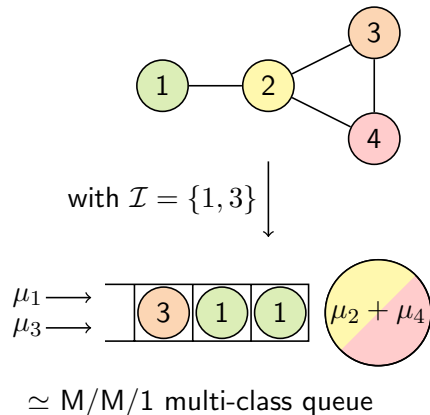
Heavy-traffic analysis

- Independent set $\mathcal{I} \in \mathbb{I}$ that is **maximal** for the inclusion, meaning that $\mathcal{V}(\mathcal{I}) = \mathcal{V} \setminus \mathcal{I}$.
- When the load $\rho(\mathcal{I}) \triangleq \frac{\sum_{i \in \mathcal{I}} \mu_i}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i}$ tends to 1



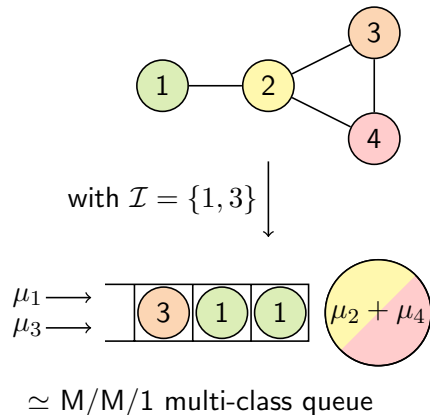
Heavy-traffic analysis

- Independent set $\mathcal{I} \in \mathbb{I}$ that is **maximal** for the inclusion, meaning that $\mathcal{V}(\mathcal{I}) = \mathcal{V} \setminus \mathcal{I}$.
- When the load $\rho(\mathcal{I}) \triangleq \frac{\sum_{i \in \mathcal{I}} \mu_i}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i}$ tends to 1



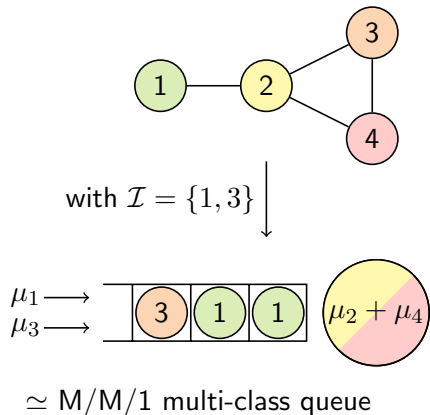
Heavy-traffic analysis

- Independent set $\mathcal{I} \in \mathbb{I}$ that is **maximal** for the inclusion, meaning that $\mathcal{V}(\mathcal{I}) = \mathcal{V} \setminus \mathcal{I}$.
- When the load $\rho(\mathcal{I}) \triangleq \frac{\sum_{i \in \mathcal{I}} \mu_i}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i}$ tends to 1,
 - the set of unmatched classes is \mathcal{I} with probability 1,



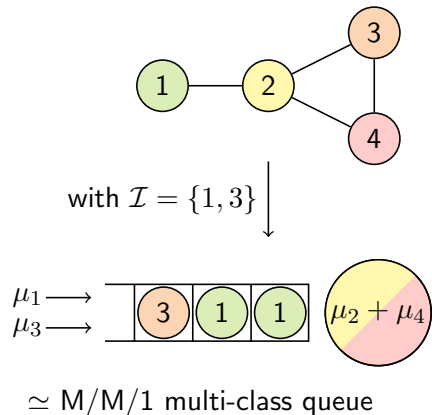
Heavy-traffic analysis

- Independent set $\mathcal{I} \in \mathbb{I}$ that is **maximal** for the inclusion, meaning that $\mathcal{V}(\mathcal{I}) = \mathcal{V} \setminus \mathcal{I}$.
- When the load $\rho(\mathcal{I}) \triangleq \frac{\sum_{i \in \mathcal{I}} \mu_i}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i}$ tends to 1,
 - the set of unmatched classes is \mathcal{I} with probability 1,
 - items of classes in \mathcal{I} wait with probability 1, while items of classes in $\mathcal{V} \setminus \mathcal{I}$ wait with probability 0,



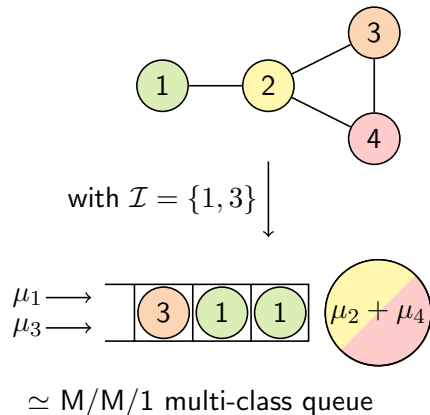
Heavy-traffic analysis

- Independent set $\mathcal{I} \in \mathbb{I}$ that is **maximal** for the inclusion, meaning that $\mathcal{V}(\mathcal{I}) = \mathcal{V} \setminus \mathcal{I}$.
- When the load $\rho(\mathcal{I}) \triangleq \frac{\sum_{i \in \mathcal{I}} \mu_i}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i}$ tends to 1,
 - the set of unmatched classes is \mathcal{I} with probability 1,
 - items of classes in \mathcal{I} wait with probability 1, while items of classes in $\mathcal{V} \setminus \mathcal{I}$ wait with probability 0,
 - the mean number of unmatched items is $\sim \frac{\rho(\mathcal{I})}{1-\rho(\mathcal{I})}$.



Heavy-traffic analysis

- Independent set $\mathcal{I} \in \mathbb{I}$ that is **maximal** for the inclusion, meaning that $\mathcal{V}(\mathcal{I}) = \mathcal{V} \setminus \mathcal{I}$.
- When the load $\rho(\mathcal{I}) \triangleq \frac{\sum_{i \in \mathcal{I}} \mu_i}{\sum_{i \in \mathcal{V}(\mathcal{I})} \mu_i}$ tends to 1,
 - the set of unmatched classes is \mathcal{I} with probability 1,
 - items of classes in \mathcal{I} wait with probability 1, while items of classes in $\mathcal{V} \setminus \mathcal{I}$ wait with probability 0,
 - the mean number of unmatched items is $\sim \frac{\rho(\mathcal{I})}{1-\rho(\mathcal{I})}$.
- Take-away: **minimizing the maximum load** is likely a good heuristic to optimize performance.

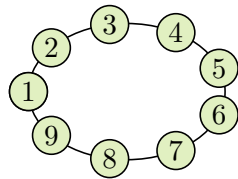


Numerical results: Cycle

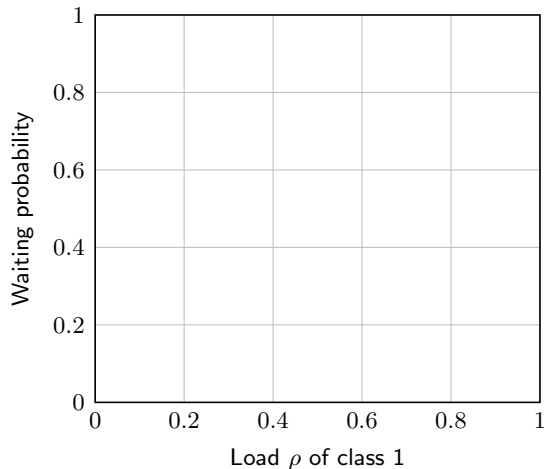
$$\mu_1 \propto \rho$$

$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$



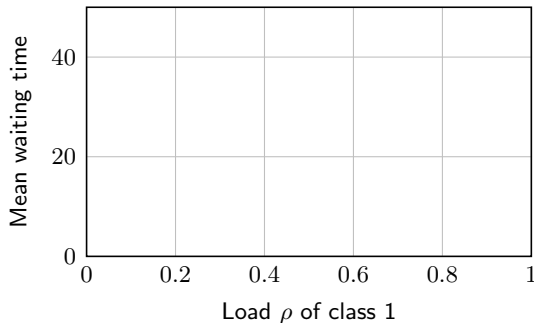
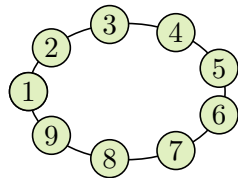
Numerical results: Cycle



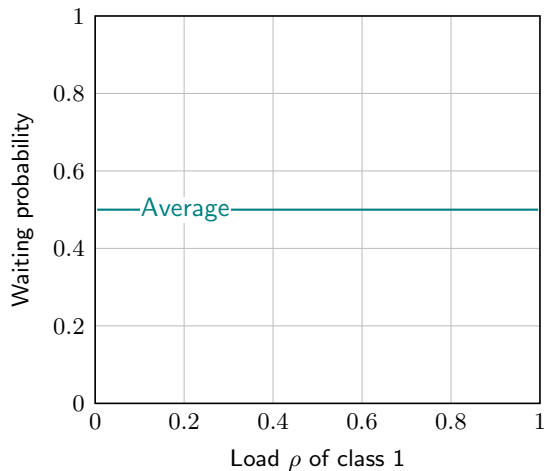
$$\mu_1 \propto \rho$$

$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$



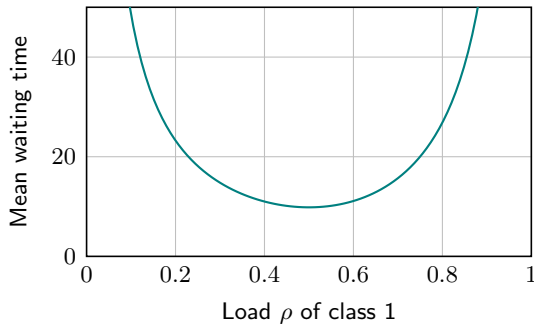
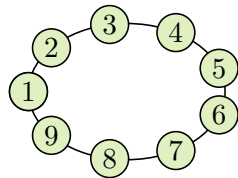
Numerical results: Cycle



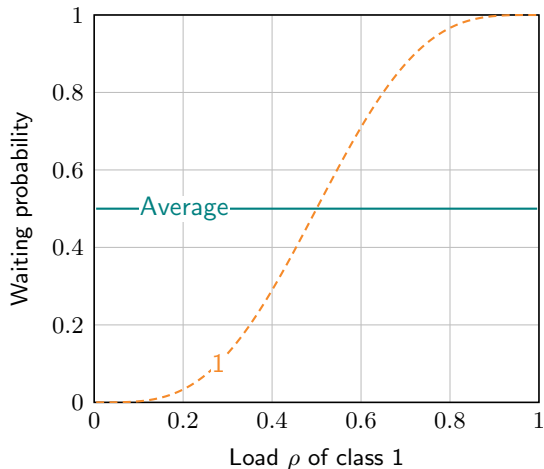
$$\mu_1 \propto \rho$$

$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$



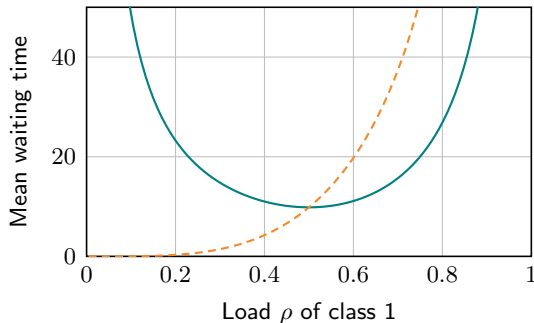
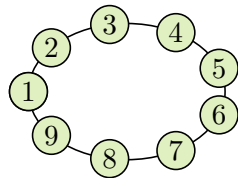
Numerical results: Cycle



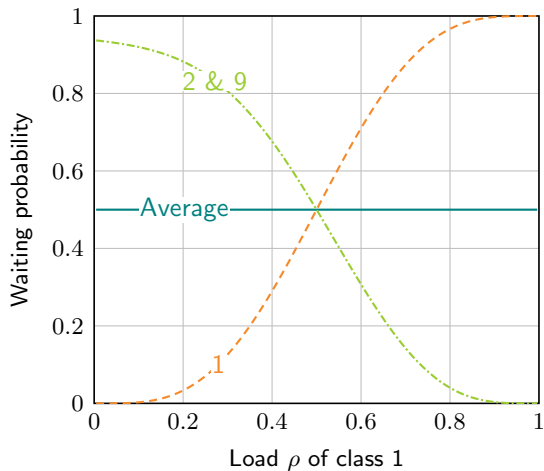
$$\mu_1 \propto \rho$$

$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$



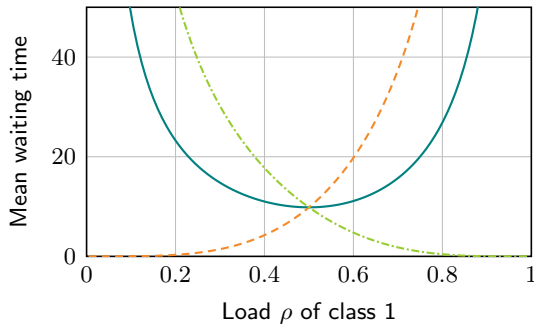
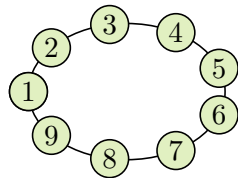
Numerical results: Cycle



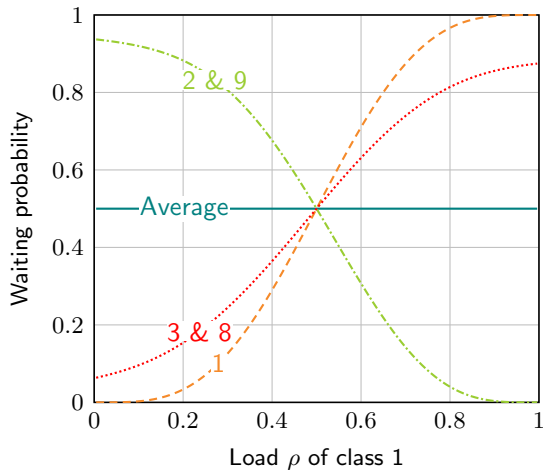
$$\mu_1 \propto \rho$$

$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$



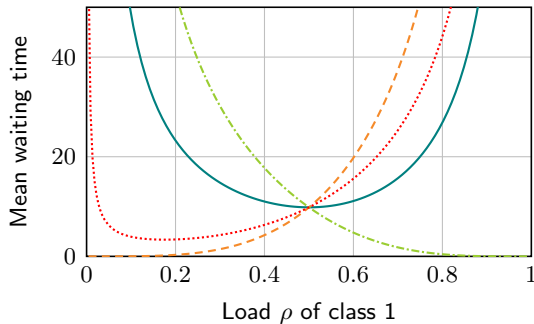
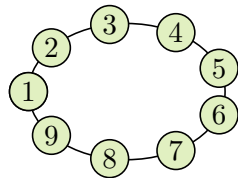
Numerical results: Cycle



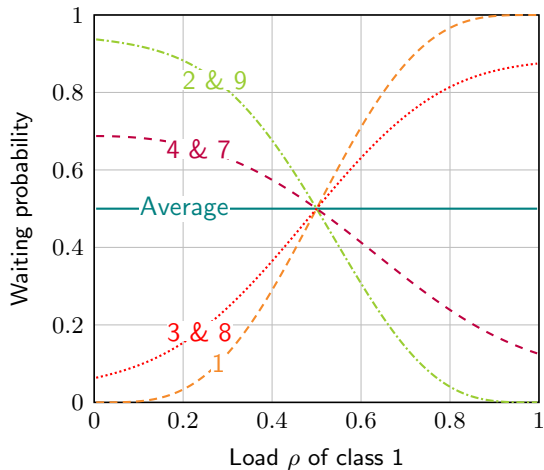
$$\mu_1 \propto \rho$$

$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$



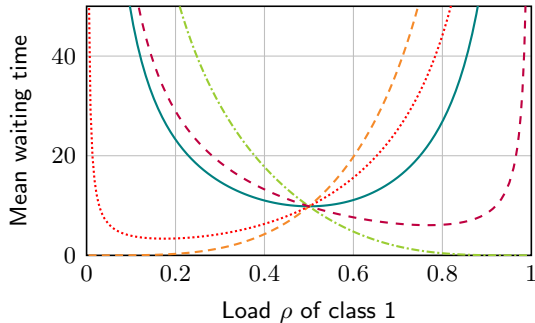
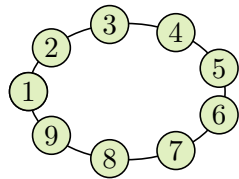
Numerical results: Cycle



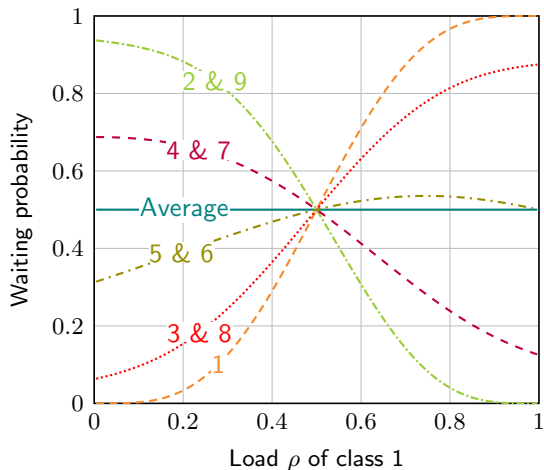
$$\mu_1 \propto \rho$$

$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$



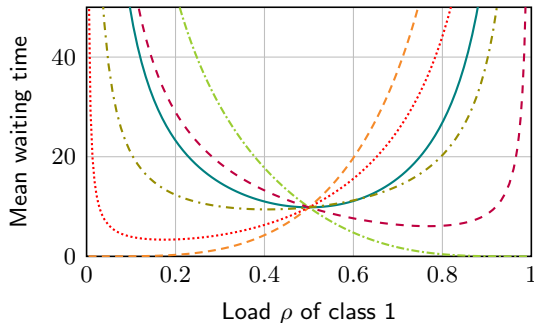
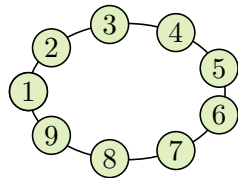
Numerical results: Cycle



$$\mu_1 \propto \rho$$

$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$

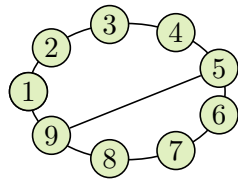


Numerical results: Cycle with a chord

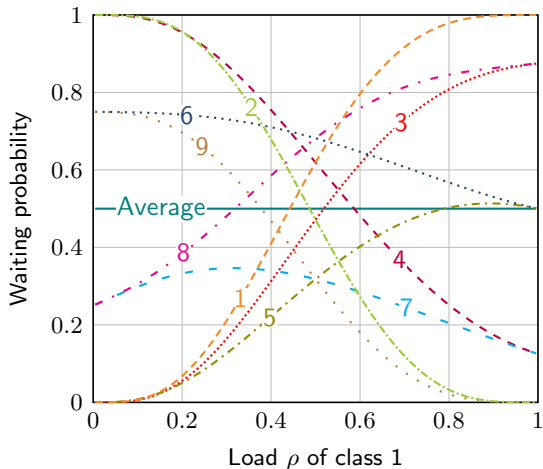
$$\mu_1 \propto \rho$$

$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$



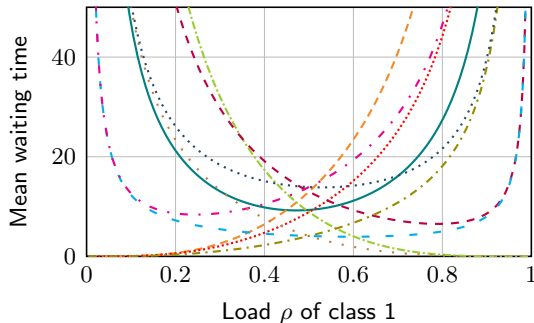
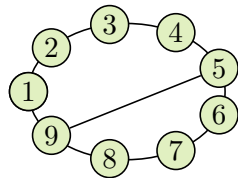
Numerical results: Cycle with a chord



$$\mu_1 \propto \rho$$

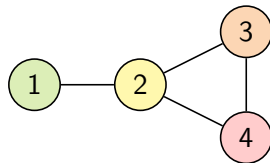
$$\mu_2 = \mu_3 = \dots = \mu_9 \propto \frac{1}{2}$$

$$\rho = \frac{\mu_1}{\mu_2 + \mu_9} = \rho(\{1\})$$



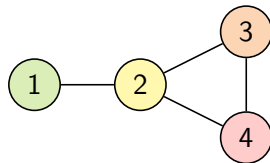
Remark: Augmented state descriptor of [MBM21]

- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.



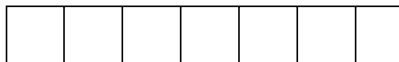
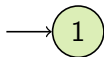
Remark: Augmented state descriptor of [MBM21]

- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:

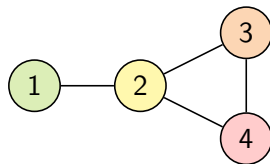


Remark: Augmented state descriptor of [MBM21]

- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:

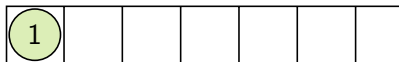


$s = \emptyset$

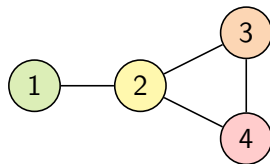


Remark: Augmented state descriptor of [MBM21]

- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:

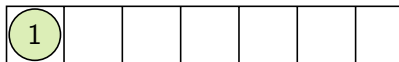
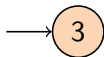


$$s = (\textcircled{1})$$

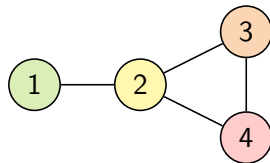


Remark: Augmented state descriptor of [MBM21]

- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:

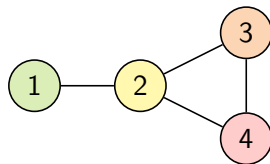


$$s = (\textcircled{1})$$



Remark: Augmented state descriptor of [MBM21]

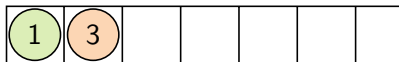
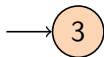
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



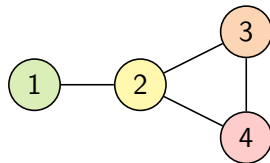
$$s = (\textcircled{1}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:

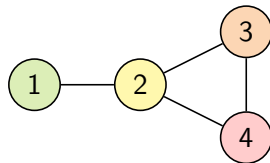
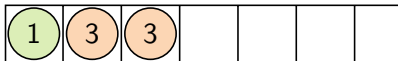


$$s = (\textcircled{1}, \textcircled{3})$$



Remark: Augmented state descriptor of [MBM21]

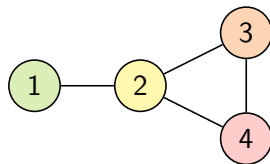
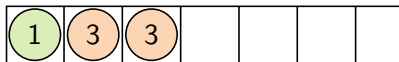
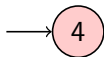
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (1, 3, 3)$$

Remark: Augmented state descriptor of [MBM21]

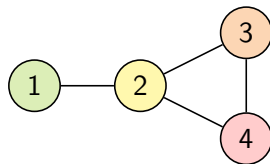
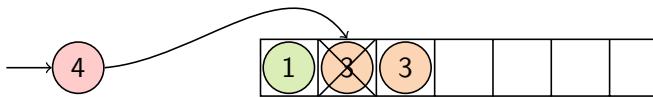
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \textcircled{3}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

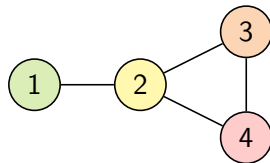
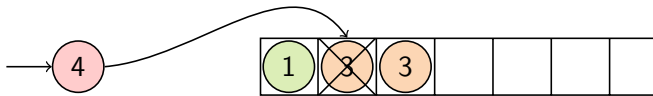
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \textcircled{3}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

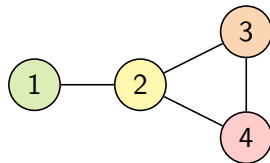
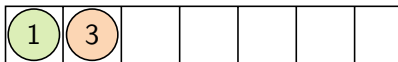
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \bar{\textcircled{4}}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

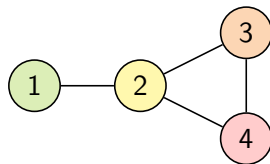
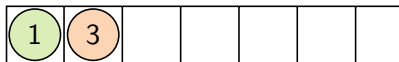
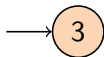
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \bar{\textcircled{4}}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

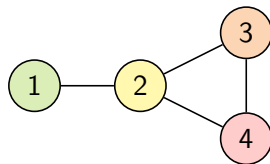
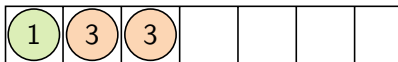
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \bar{\textcircled{4}}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

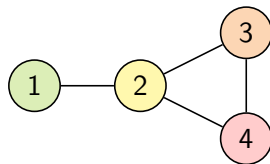
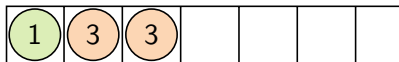
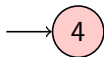
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \bar{\textcircled{4}}, \textcircled{3}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

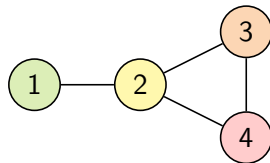
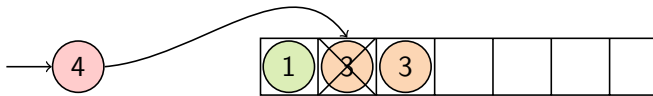
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \bar{\textcircled{4}}, \textcircled{3}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

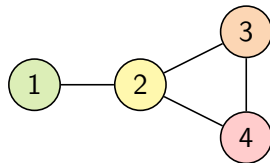
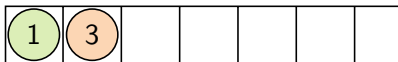
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \bar{\textcircled{4}}, \bar{\textcircled{4}}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

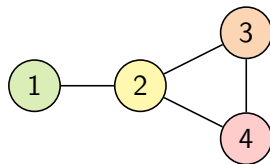
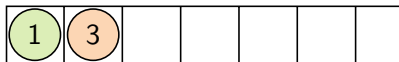
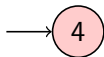
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \bar{\textcircled{4}}, \bar{\textcircled{4}}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

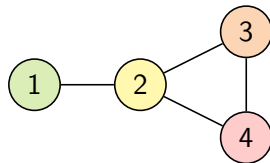
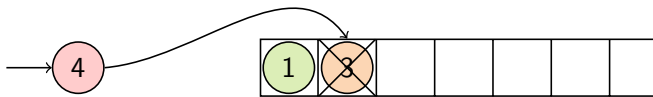
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1}, \bar{\textcircled{4}}, \bar{\textcircled{4}}, \textcircled{3})$$

Remark: Augmented state descriptor of [MBM21]

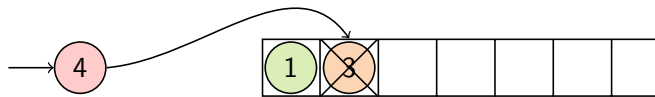
- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



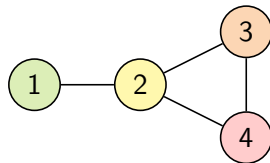
$$s = (\textcircled{1})$$

Remark: Augmented state descriptor of [MBM21]

- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



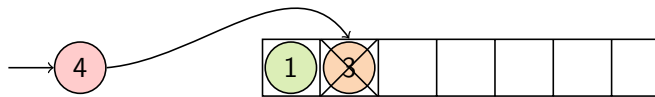
$$s = (\textcircled{1})$$



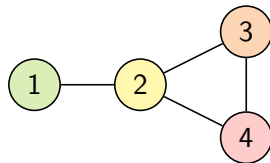
- Stationary distribution [MBM21]: $\pi(s_1, s_2, \dots, s_k) \propto \prod_{p=1}^k \mu_{s_p}$.

Remark: Augmented state descriptor of [MBM21]

- Independent copy $\bar{\mathcal{V}} = \{\bar{i}, i \in \mathcal{V}\}$ of the set of item classes.
- Augmented state descriptor $s = (s_1, s_2, \dots, s_k) \in (\mathcal{V} \cup \bar{\mathcal{V}})^*$:



$$s = (\textcircled{1})$$



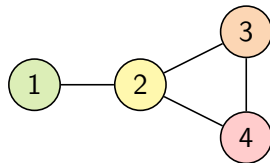
- Stationary distribution [MBM21]: $\pi(s_1, s_2, \dots, s_k) \propto \prod_{p=1}^k \mu_{s_p}$.
- The complexity is “hidden” in the description of the state space and normalization constant.

- 1 Literature review
- 2 Basic results
 - Stability condition
 - Definition of the discrete-time Markov chain
 - Product-form stationary distribution and partial balance
- 3 Performance analysis
 - Normalization constant
 - Performance metrics
 - Heavy-traffic analysis
- 4 Concluding remarks

Concluding remarks

Take-away

- Product-form stationary distribution
- Closed-form expressions for performance metrics
- Heavy-traffic analysis



Take-away

- Product-form stationary distribution
- Closed-form expressions for performance metrics
- Heavy-traffic analysis

Future works

- Extensions to state-dependent arrival rates? hypergraphs? other policies?
- More fundamental relationship between balance, reversibility, and insensitivity?

