A 3-player game theoretic model of a choice between two queueing systems with strategic managerial decision making

Michalis Panayides

EURO 2021 Athens

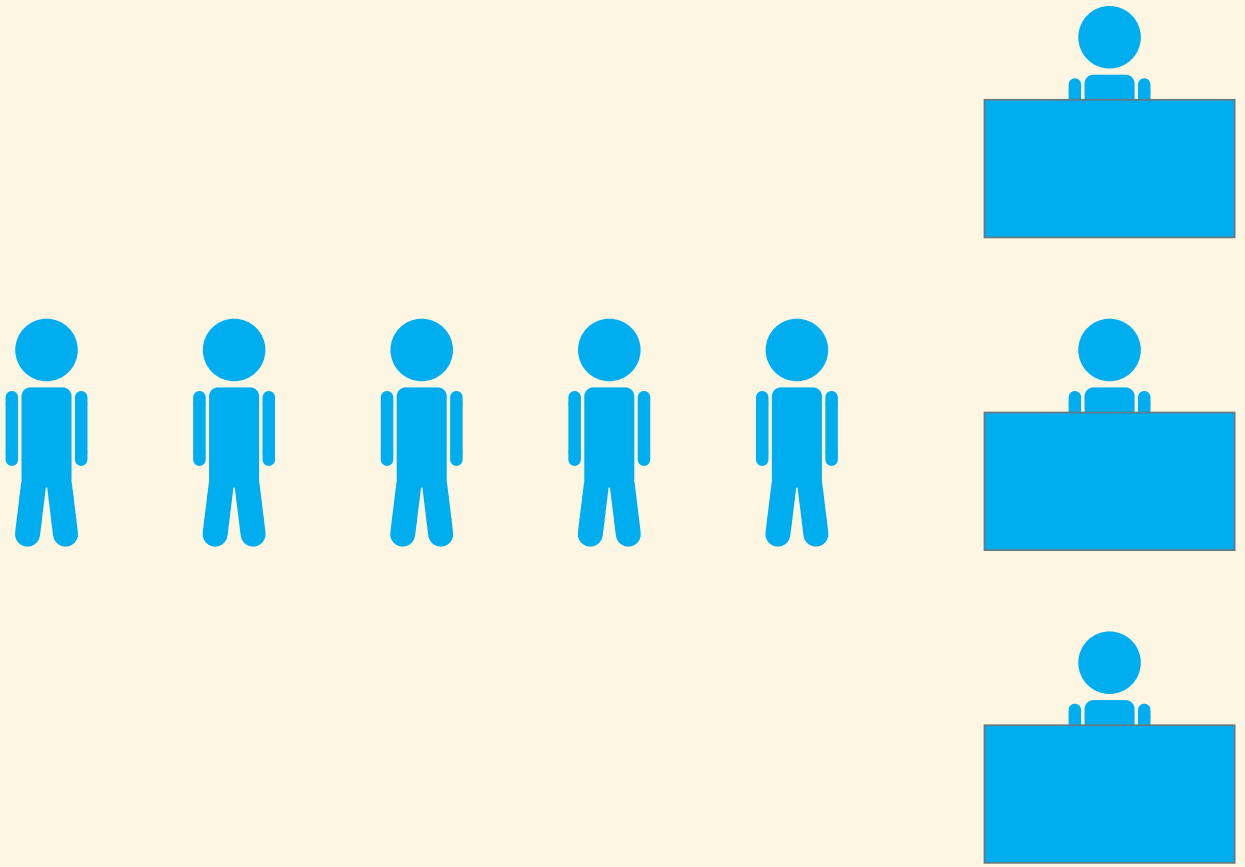
About me

About me

THIS.

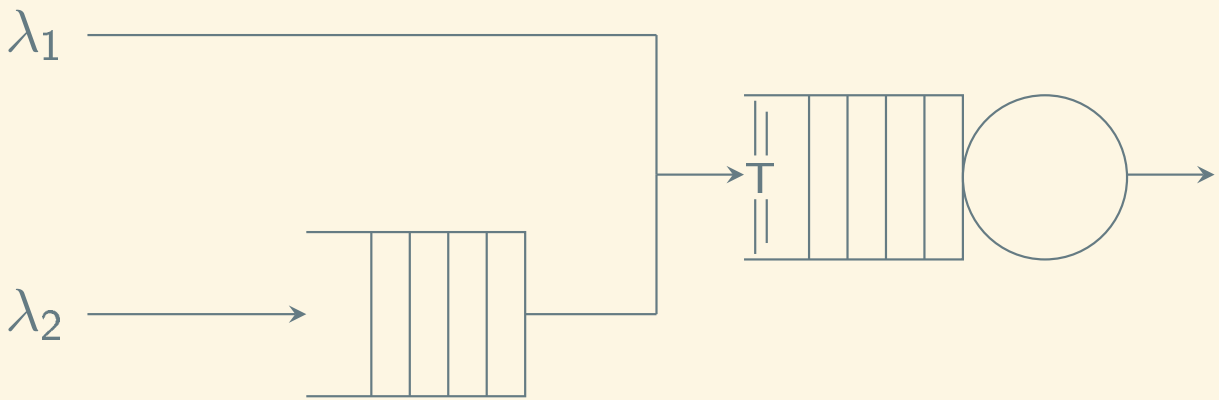
Queues - Examples

Queues - Examples

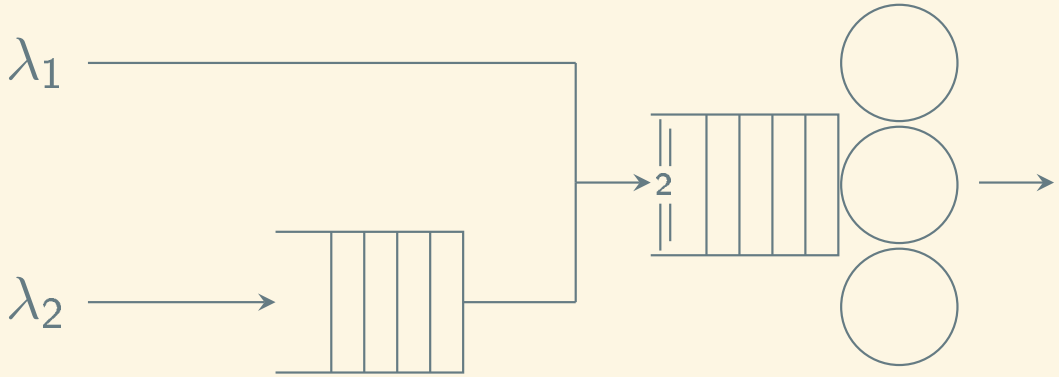


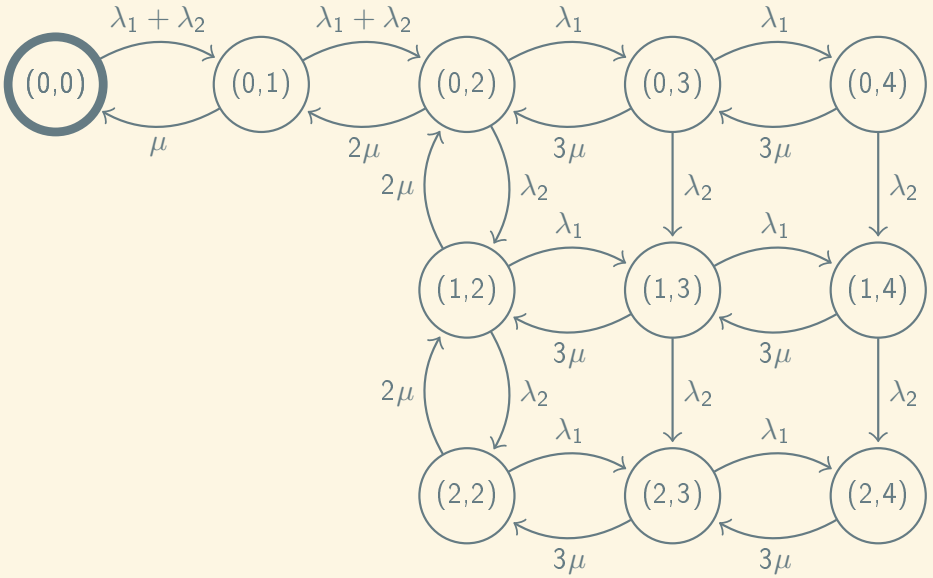
Queues - Examples

Queueing network structure

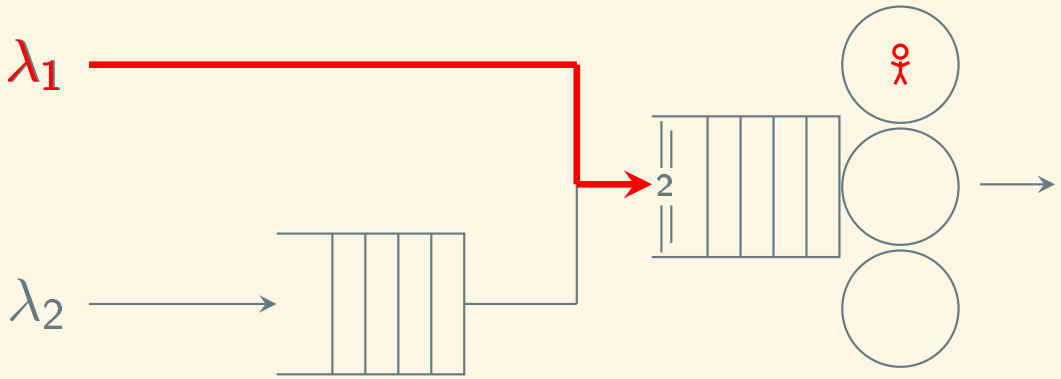


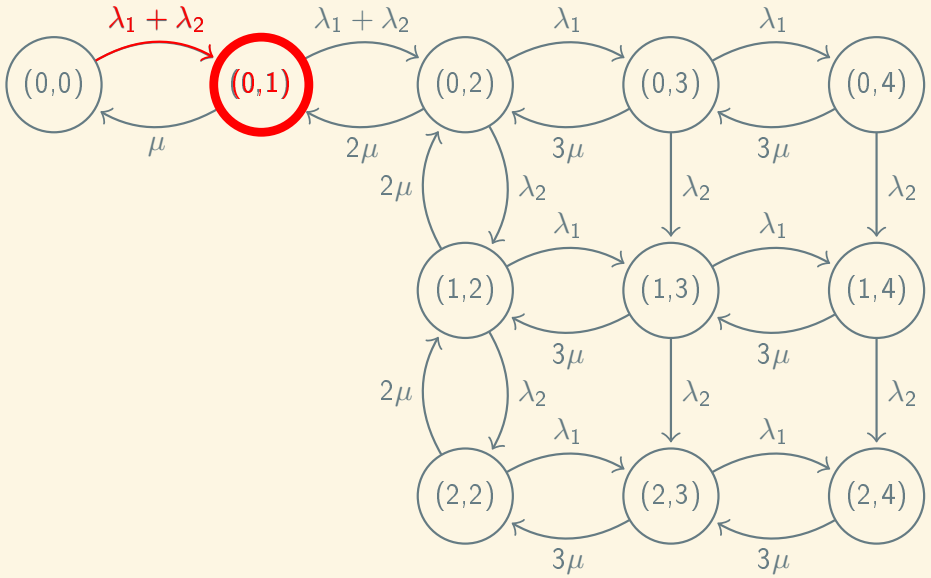
Markov Chain - Custom network



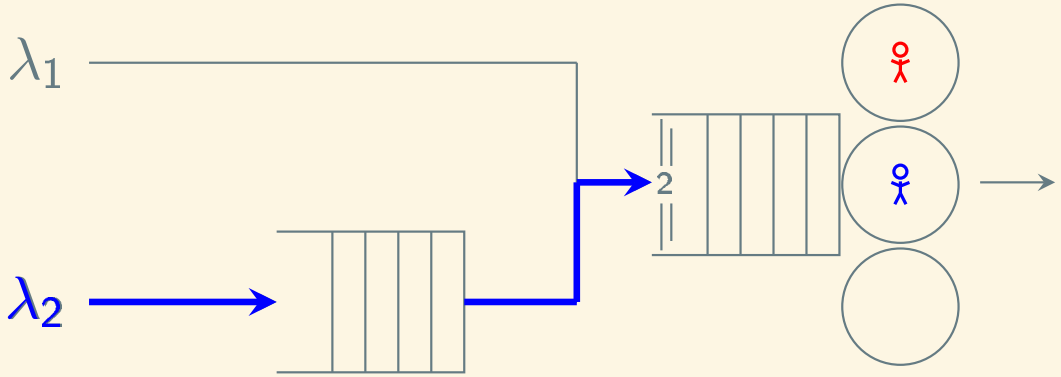


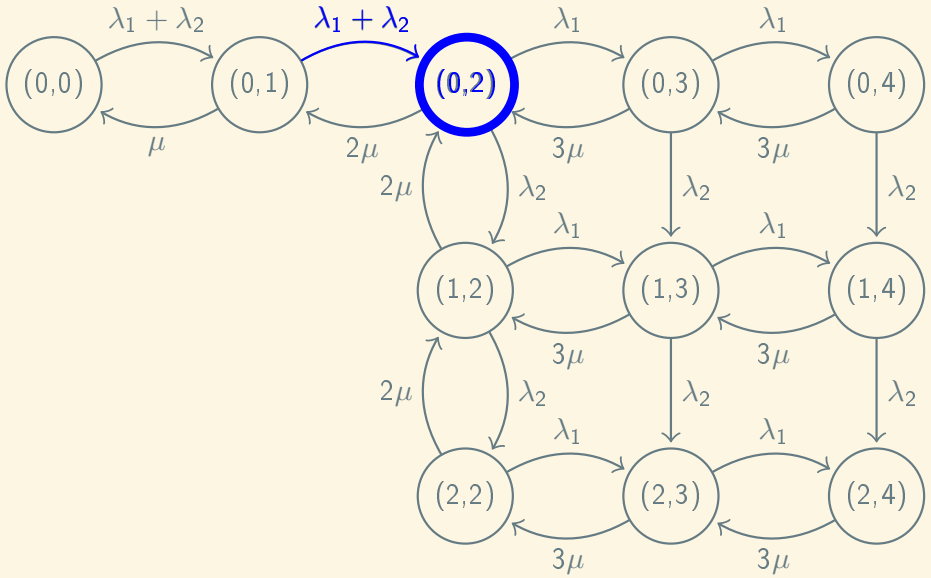
Markov Chain - Custom network



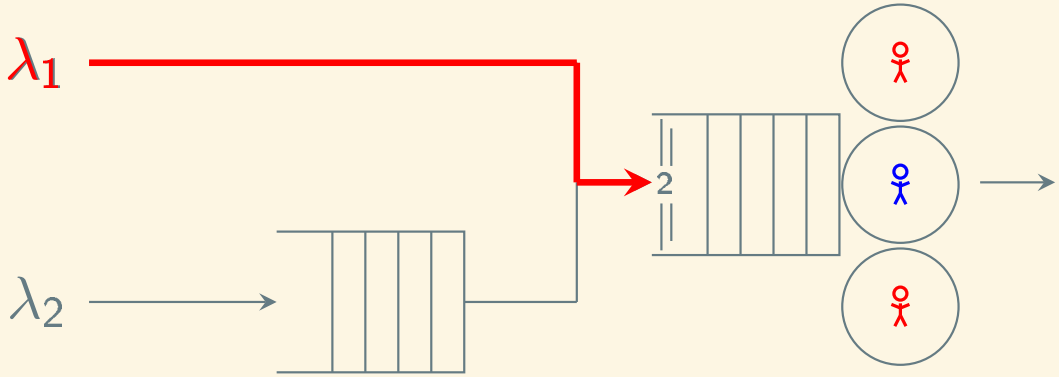


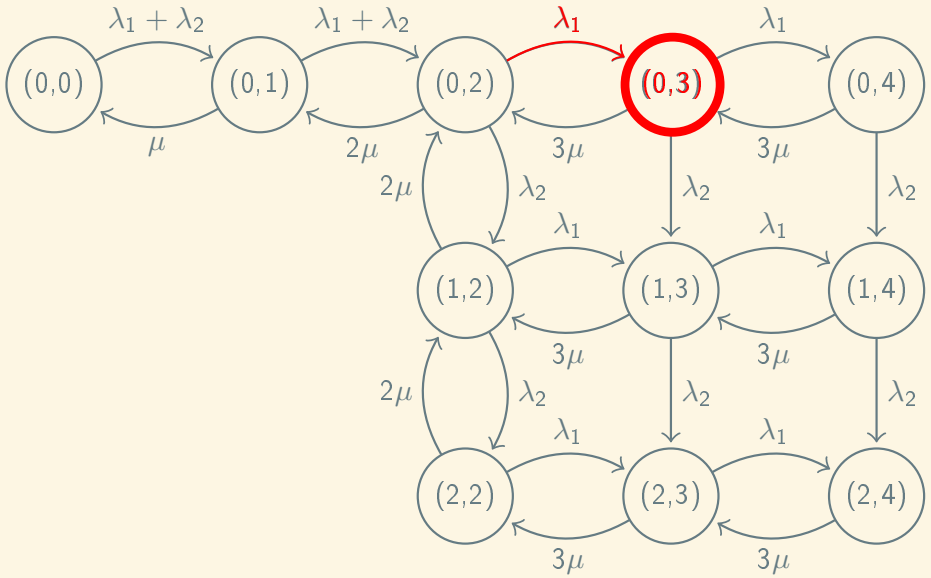
Markov Chain - Custom network



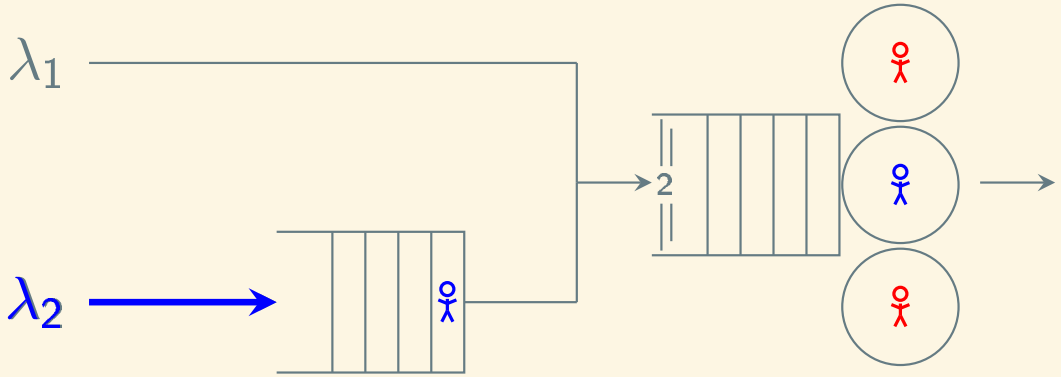


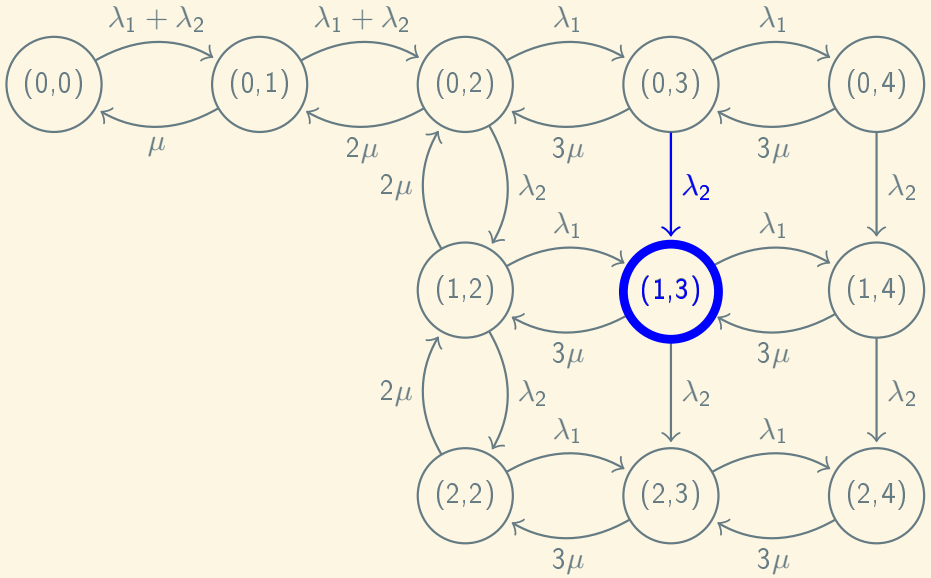
Markov Chain - Custom network



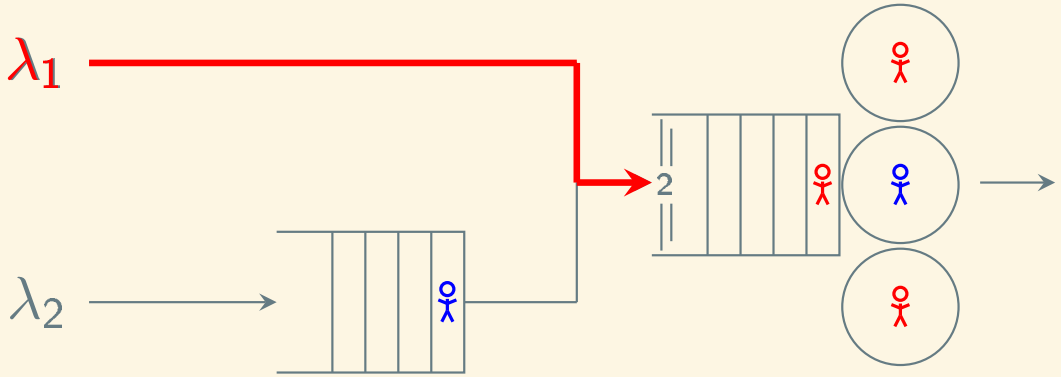


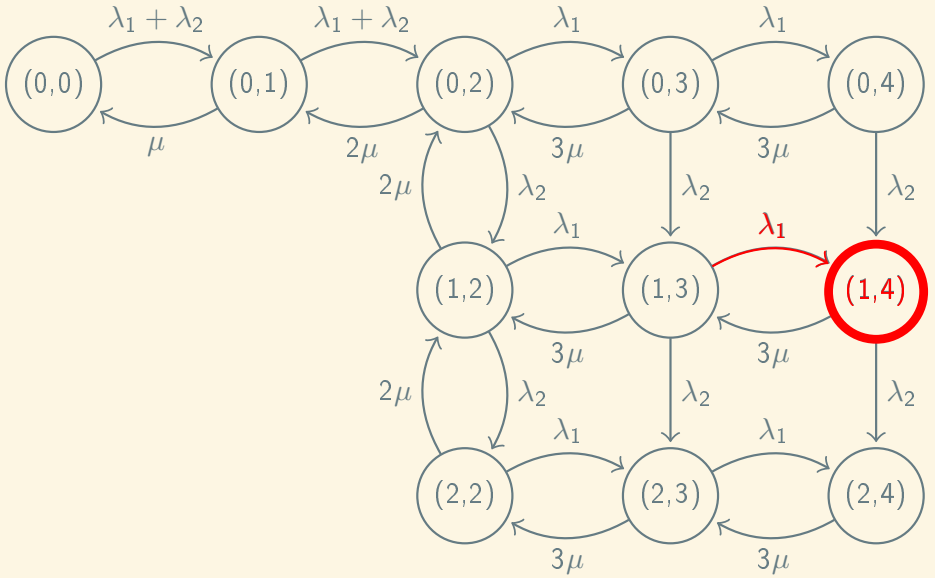
Markov Chain - Custom network



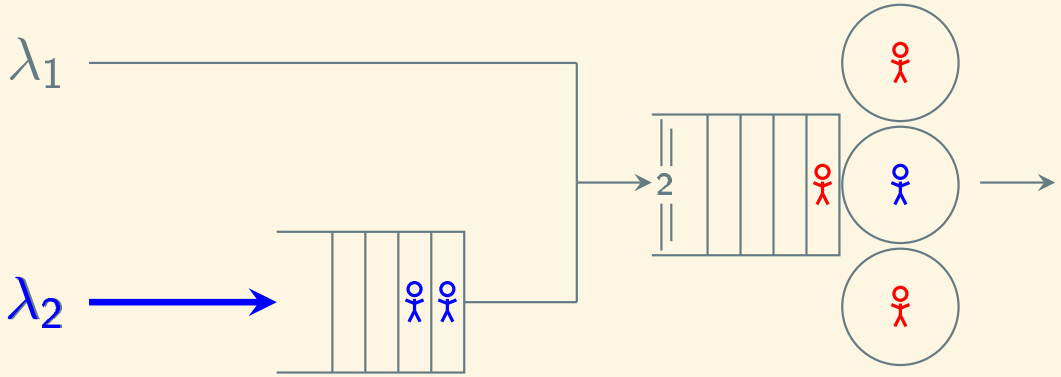


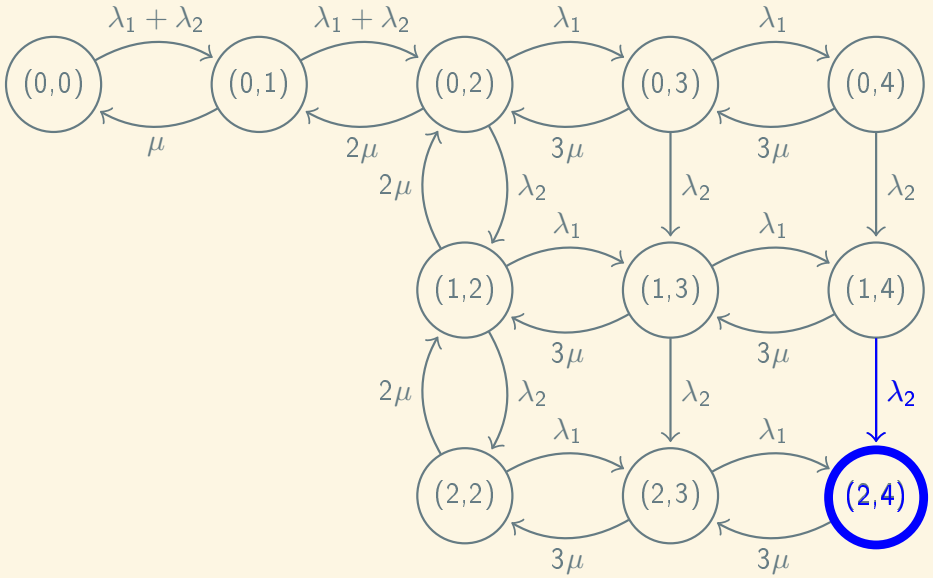
Markov Chain - Custom network



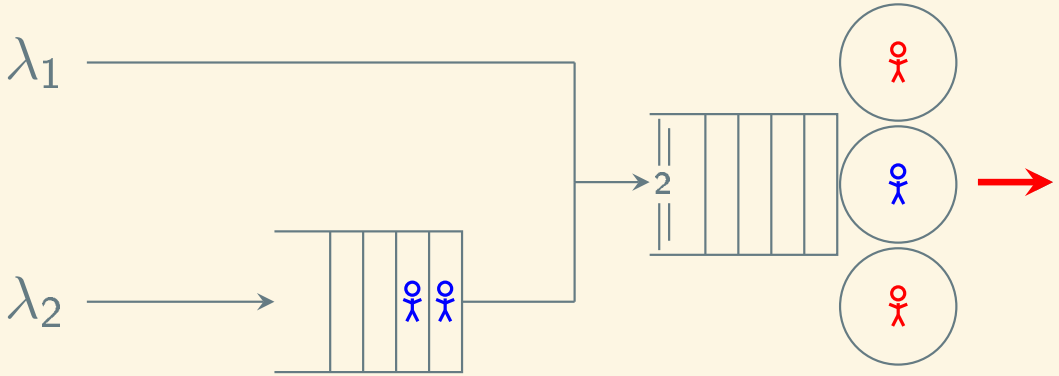


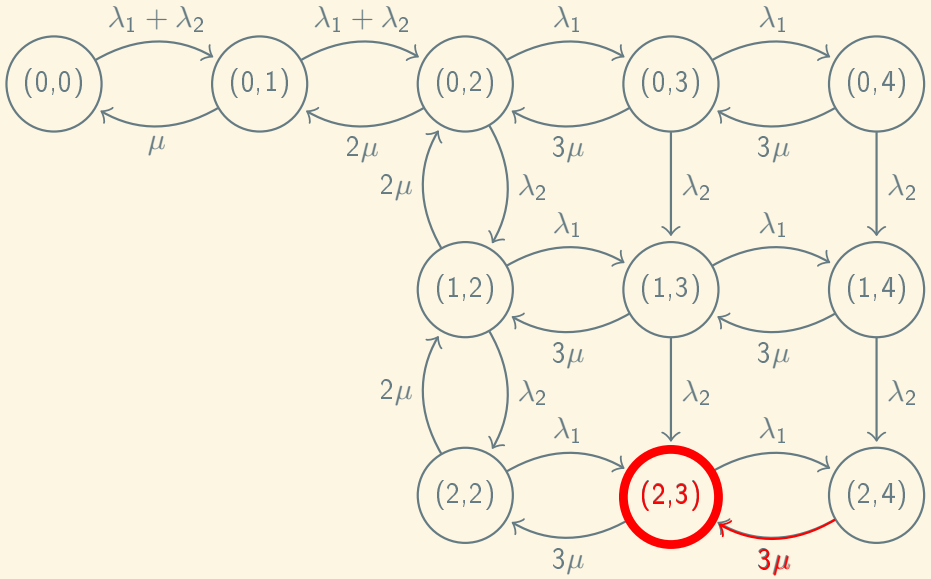
Markov Chain - Custom network



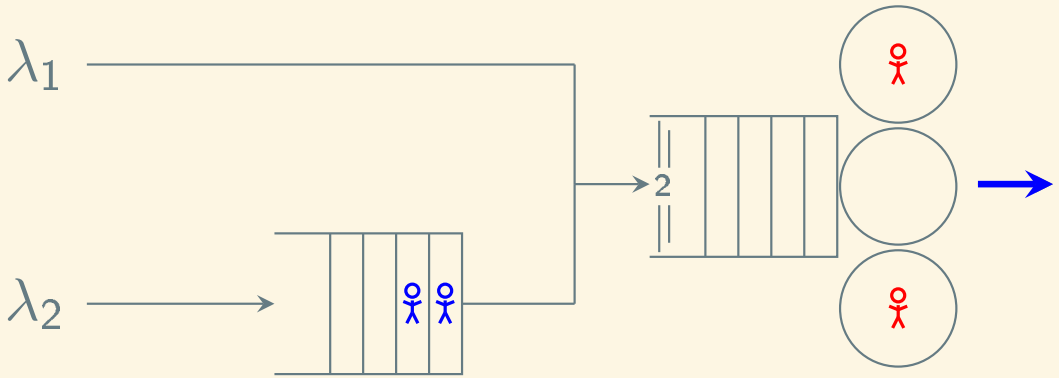


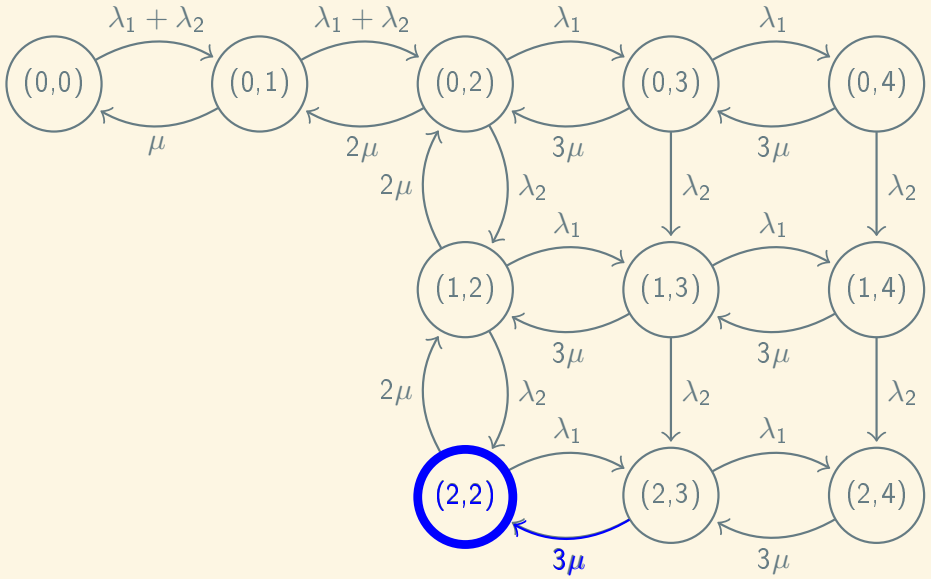
Markov Chain - Custom network



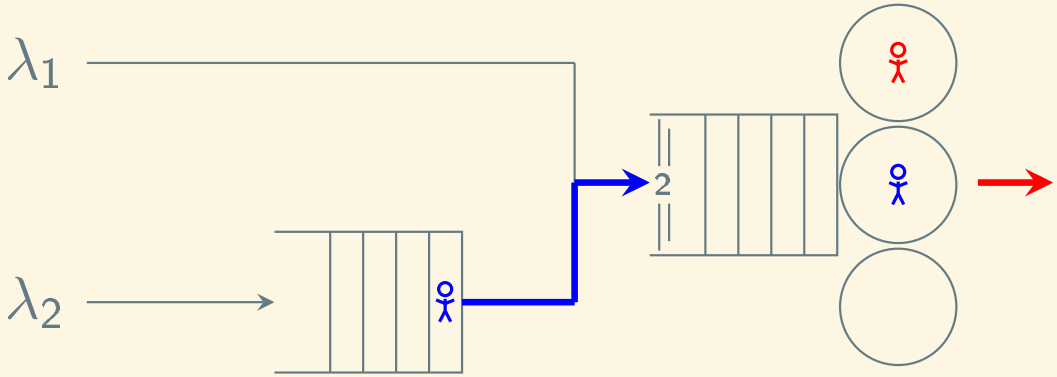


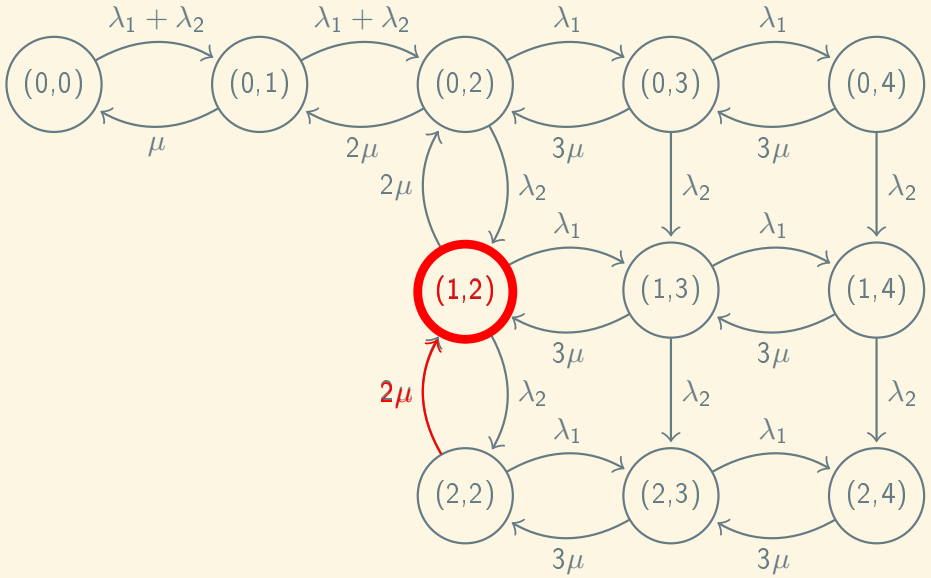
Markov Chain - Custom network





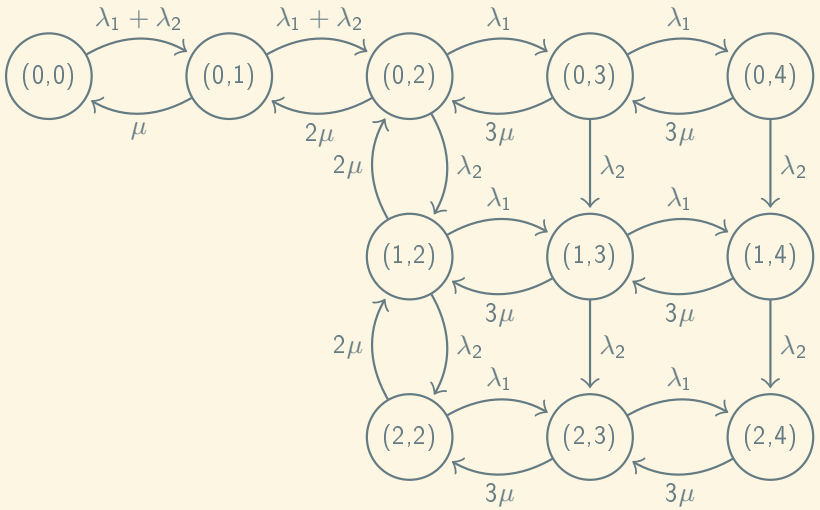
Markov Chain - Custom network





Steady state probabilities - Custom network

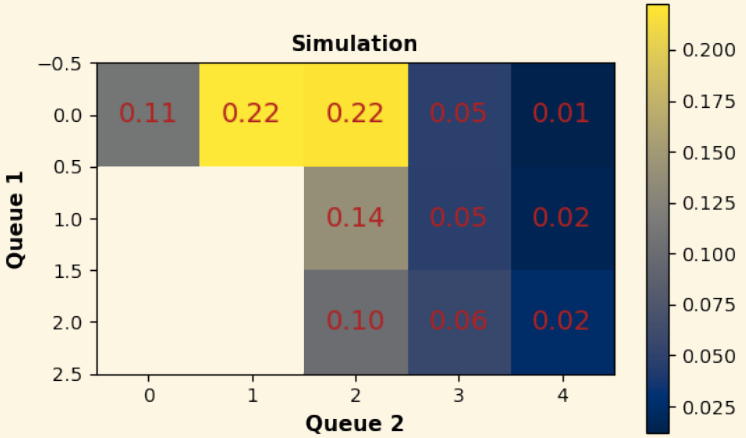
$$
Q = \begin{array}{c} \begin{matrix} (0,0) & (0,1) & (0,2) & \dots & (2,3) & (2,4) \end{matrix} \\ \begin{pmatrix} -\lambda_1 - \lambda_2 & \lambda_1 + \lambda_2 & 0 & \dots & 0 & 0 \\ \mu & -\mu - \lambda_1 - \lambda_2 & \lambda_1 + \lambda_2 & \dots & 0 & 0 \\ 0 & 2\mu & -2\mu - \lambda_1 - \lambda_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -\lambda_1 - 3\mu & \lambda_1 \\ 0 & 0 & 0 & \dots & 3\mu & -3\mu \end{pmatrix} \end{array} \begin{matrix} (0,0) \\ (0,1) \\ (0,2) \\ \\ \\ (2,3) \\ (2,4) \end{matrix}
$$

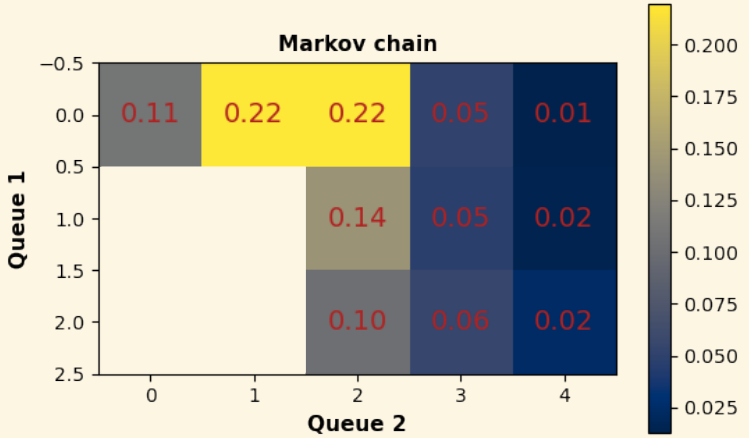


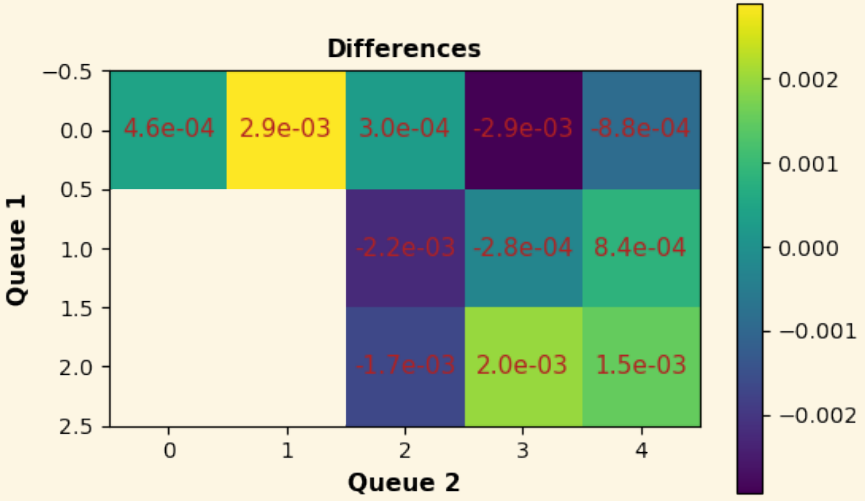
$$
\frac{d\pi}{dt} = \pi Q = 0, \qquad \sum \pi_{(u,v)} = 1
$$

$$
\pi = \begin{bmatrix} \pi_{(0,0)} \\ \pi_{(0,1)} \\ \pi_{(0,2)} \\ \vdots \\ \pi_{(2,3)} \\ \pi_{(2,4)} \end{bmatrix}
$$

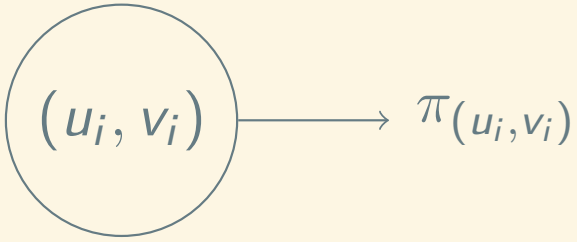
Steady state probabilities - Comparison



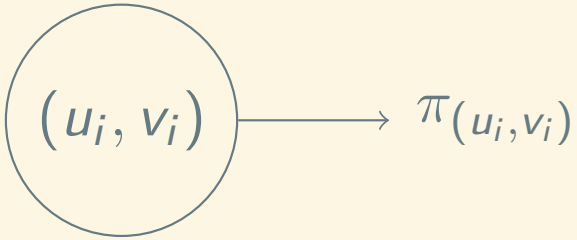




Performance Measures - Number of individuals



Performance Measures - Number of individuals



$$L = \sum_{i=1}^{|\pi|} \pi_i (u_i + v_i)$$

$$L_1 = \sum_{i=1}^{|\pi|} \pi_i u_i$$

$$L_2 = \sum_{i=1}^{|\pi|} \pi_i v_i$$

Performance Measures - Waiting time

$$W = \frac{\lambda_1 P_{L_1'}}{\lambda_2 P_{L_2'} + \lambda_1 P_{L_1'}} W^{(1)} + \frac{\lambda_2 P_{L_2'}}{\lambda_2 P_{L_2'} + \lambda_1 P_{L_1'}} W^{(2)}$$

$$W^{(1)} = \frac{\sum_{\substack{(u,v) \in S_A^{(1)} \\ v \geq C}} \frac{1}{C\mu} \times (v - C + 1) \times \pi(u,v)}{\sum_{(u,v) \in S_A^{(1)}} \pi(u,v)}$$

$$W^{(2)} = \frac{\sum_{\substack{(u,v) \in S_A^{(2)} \\ min(v,T) \geq C}} \frac{1}{C\mu} \times (\min(v+1,T) - C) \times \pi(u,v)}{\sum_{(u,v) \in S_A^{(2)}} \pi(u,v)}$$

Performance Measures - Waiting time

$$W^{(1)} = \frac{\sum_{(u,v)\in S_A^{(1)}} w^{(1)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(1)}} \pi_{(u,v)}}, \quad W^{(2)} = \frac{\sum_{(u,v)\in S_A^{(2)}} w^{(2)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}}$$

Performance Measures - Waiting time

$$W^{(1)} = \frac{\sum_{(u,v)\in S_A^{(1)}} w^{(1)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(1)}} \pi_{(u,v)}}, \quad W^{(2)} = \frac{\sum_{(u,v)\in S_A^{(2)}} w^{(2)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}}$$

$$S_A^{(1)} = \{(u,v) \in S \mid v < N\}, \quad S_A^{(2)} = \begin{cases} \{(u,v) \in S \mid u < M\} & \textbf{if } T \leq N \\ \{(u,v) \in S \mid v < N\} & \textbf{otherwise} \end{cases}$$

Performance Measures - Waiting time

$$W^{(1)} = \frac{\sum_{(u,v)\in S_A^{(1)}} w^{(1)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(1)}} \pi_{(u,v)}}, \quad W^{(2)} = \frac{\sum_{(u,v)\in S_A^{(2)}} w^{(2)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}}$$

$$S_A^{(1)} = \{(u,v) \in S \mid v < N\}, \quad S_A^{(2)} = \begin{cases} \{(u,v) \in S \mid u < M\} & \textbf{if } T \leq N \\ \{(u,v) \in S \mid v < N\} & \textbf{otherwise} \end{cases}$$

$$S_W = \{(u,v) \in S \mid v > C\}$$

Performance Measures - Waiting time

$$W^{(1)} = \frac{\sum_{(u,v)\in S_A^{(1)}} w^{(1)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(1)}} \pi_{(u,v)}}, \quad W^{(2)} = \frac{\sum_{(u,v)\in S_A^{(2)}} w^{(2)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}}$$

$$S_A^{(1)} = \{(u,v) \in S \mid v < N\}, \quad S_A^{(2)} = \begin{cases} \{(u,v) \in S \mid u < M\} & \text{if } T \leq N \\ \{(u,v) \in S \mid v < N\} & \text{otherwise} \end{cases}$$

$$S_W = \{(u,v) \in S \mid v > C\}$$

$$c^{(1)}(u,v) = \begin{cases} 0, & \text{if } u > 0 \text{ and } v = T \\ \frac{1}{\min(v,C)\mu}, & \text{otherwise} \end{cases}, \quad c^{(2)}(u,v) = \begin{cases} 0, & \text{if } u > 0 \\ \frac{1}{\min(v,C)\mu}, & \text{otherwise} \end{cases}$$

Performance Measures - Waiting time

$$W^{(1)} = \frac{\sum_{(u,v)\in S_A^{(1)}} w^{(1)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(1)}} \pi_{(u,v)}}, \quad W^{(2)} = \frac{\sum_{(u,v)\in S_A^{(2)}} w^{(2)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}}$$

$$S_A^{(1)} = \{(u,v) \in S \mid v < N\}, \quad S_A^{(2)} = \begin{cases} \{(u,v) \in S \mid u < M\} & \text{if } T \leq N \\ \{(u,v) \in S \mid v < N\} & \text{otherwise} \end{cases}$$

$$S_W = \{(u,v) \in S \mid v > C\}$$

$$c^{(1)}(u,v) = \begin{cases} 0, & \text{if } u > 0 \text{ and } v = T \\ \frac{1}{\min(v,C)\mu}, & \text{otherwise} \end{cases}, \quad c^{(2)}(u,v) = \begin{cases} 0, & \text{if } u > 0 \\ \frac{1}{\min(v,C)\mu}, & \text{otherwise} \end{cases}$$

$$w^{(i)}(u,v) = \begin{cases} 0, & \text{if } (u,v) \notin S_w \\ c^{(i)}(u,v) + w^{(i)}(u-1,v), & \text{if } u > 0 \text{ and } v = T \\ c^{(i)}(u,v) + w^{(i)}(u,v-1), & \text{otherwise} \end{cases}$$

Performance Measures - Blocking time

$$B = \frac{\sum_{(u,v) \in S_A^{(2)}} \pi_{(u,v)} \; b(u,v)}{\sum_{(u,v) \in S_A^{(2)}} \pi_{(u,v)}}$$

Performance Measures - Blocking time

$$B = \frac{\sum_{(u,v) \in S_A^{(2)}} \pi_{(u,v)} \; b(u,v)}{\sum_{(u,v) \in S_A^{(2)}} \pi_{(u,v)}}$$

Performance Measures - Blocking time

$$B = \frac{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}\ b(u,v)}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}}$$

$$b(u,v) = \begin{cases} 0, & \text{if } (u,v) \notin S_b \\ c(u,v) + b(u-1,v), & \text{if } v = N = T \\ c(u,v) + b(u,v-1), & \text{if } v = N \neq T \\ c(u,v) + p_s(u,v)b(u-1,v) + p_a(u,v)b(u,v+1), & \text{if } u > 0 \text{ and } v = T \\ c(u,v) + p_s(u,v)b(u,v-1) + p_a(u,v)b(u,v+1), & \text{otherwise} \end{cases}$$

Performance Measures - Blocking time

$$B = \frac{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}\ b(u,v)}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}}$$

$$b(u,v) = \begin{cases} 0, & \text{if } (u,v) \notin S_b \\ c(u,v) + b(u-1,v), & \text{if } v = N = T \\ c(u,v) + b(u,v-1), & \text{if } v = N \neq T \\ c(u,v) + p_s(u,v)b(u-1,v) + p_a(u,v)b(u,v+1), & \text{if } u > 0 \text{ and } v = T \\ c(u,v) + p_s(u,v)b(u,v-1) + p_a(u,v)b(u,v+1), & \text{otherwise} \end{cases}$$

$$S_b = \{(u,v) \in S \mid u > 0\}$$

Performance Measures - Blocking time

$$B = \frac{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}\ b(u,v)}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}}$$

$$b(u,v) = \begin{cases} 0, & \text{if } (u,v) \notin S_b \\ c(u,v) + b(u-1,v), & \text{if } v = N = T \\ c(u,v) + b(u,v-1), & \text{if } v = N \neq T \\ c(u,v) + p_s(u,v)b(u-1,v) + p_a(u,v)b(u,v+1), & \text{if } u > 0 \text{ and } v = T \\ c(u,v) + p_s(u,v)b(u,v-1) + p_a(u,v)b(u,v+1), & \text{otherwise} \end{cases}$$

$$S_b = \{(u,v) \in S \mid u > 0\}$$

$$c(u,v) = \begin{cases} \frac{1}{\min(v,C)\mu}, & \text{if } v = N \\ \frac{1}{\lambda_1 + \min(v,C)\mu}, & \text{otherwise} \end{cases}$$

Performance Measures - Blocking time

$$B = \frac{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}\ b(u,v)}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}}$$

$$b(u,v) = \begin{cases} 0, & \text{if } (u,v) \notin S_b \\ c(u,v) + b(u-1,v), & \text{if } v = N = T \\ c(u,v) + b(u,v-1), & \text{if } v = N \neq T \\ c(u,v) + p_s(u,v)b(u-1,v) + p_a(u,v)b(u,v+1), & \text{if } u > 0 \text{ and } v = T \\ c(u,v) + p_s(u,v)b(u,v-1) + p_a(u,v)b(u,v+1), & \text{otherwise} \end{cases}$$

$$S_b = \{(u,v) \in S \mid u > 0\}$$

$$c(u,v) = \begin{cases} \frac{1}{\min(v,C)\mu}, & \text{if } v = N \\ \frac{1}{\lambda_1 + \min(v,C)\mu}, & \text{otherwise} \end{cases}$$

$$p_s(u,v) = \frac{\min(v,C)\mu}{\lambda_1 + \min(v,C)\mu}, \qquad p_a(u,v) = \frac{\lambda_1}{\lambda_1 + \min(v,C)\mu}$$

Performance Measures - Proportion within time

$$P(X < t) = \frac{\lambda_1 P_{L_1'}}{\lambda_2 P_{L_2'} + \lambda_1 P_{L_1'}} P(X^{(1)} < t) + \frac{\lambda_2 P_{L_2'}}{\lambda_2 P_{L_2'} + \lambda_1 P_{L_1'}} P(X^{(2)} < t)$$

$$P(X^{(1)} < t) = \frac{\sum_{(u,v) \in S_A^{(1)}} P(X_{(u,v)}^{(1)} < t) \pi_{u,v}}{\sum_{(u,v) \in S_A^{(1)}} \pi_{u,v}}$$

$$P(X^{(2)} < t) = \frac{\sum_{(u,v) \in S_A^{(2)}} P(X_{(u,v)}^{(2)} < t) \pi_{u,v}}{\sum_{(u,v) \in S_A^{(2)}} \pi_{u,v}}$$

Performance Measures - Proportion within time

$$P(X^{(i)} < t) = \frac{\sum_{(u,v)\in S_A^{(i)}} P(X_{u,v}^{(i)} < t)\pi_{u,v}}{\sum_{(u,v)\in S_A^{(i)}} \pi_{u,v}}, \quad \textbf{for } i = \{1, 2\}$$

Performance Measures - Proportion within time

$$P(X^{(i)} < t) = \frac{\sum_{(u,v)\in S_A^{(i)}} P(X_{u,v}^{(i)} < t)\pi_{u,v}}{\sum_{(u,v)\in S_A^{(i)}} \pi_{u,v}}, \quad \textbf{for } i = \{1, 2\}$$

$$P(X_{(u,v)}^{(\mathbf{1})} < t) = \begin{cases} 1 - \sum_{i=\mathbf{0}}^{v-\mathbf{1}} \frac{\mathbf{1}}{i!} e^{-\mu t}(\mu t)^i, & \textbf{if } C = 1 \textbf{ and } v > 1 \\ 1 - (\mu C)^{v-C} \mu \sum_{k=\mathbf{1}}^{|\vec{r}|} \sum_{l=\mathbf{1}}^{r_k} \frac{\Psi_{k,l}(-\lambda_k)t^{r_k-l}e^{-\lambda_k t}}{(r_k-l)!(l-\mathbf{1})!}, & \textbf{if } C > 1 \textbf{ and } v > C \\ \quad \textbf{where } \vec{r} = (v - C, 1) \textbf{ and } \vec{\lambda} = (C\mu, \mu) & \\ 1 - e^{-\mu t}, & \textbf{if } v \leq C \end{cases}$$

$$P(X_{(u,v)}^{(\mathbf{2})} < t) = \begin{cases} 1 - \sum_{i=\mathbf{0}}^{\min(v,T)-\mathbf{1}} \frac{\mathbf{1}}{i!} e^{-\mu t}(\mu t)^i, & \textbf{if } C = 1 \textbf{ and } v, T > 1 \\ 1 - \mu(\mu C)^{\min(v,T)-C} \times \sum_{k=\mathbf{1}}^{|\vec{r}|} \sum_{l=\mathbf{1}}^{r_k} \frac{\Psi_{k,l}(-\lambda_k)t^{r_k-l}e^{-\lambda_k t}}{(r_k-l)!(l-\mathbf{1})!}, & \textbf{if } C > 1 \textbf{ and } v, T > C \\ \quad \textbf{where } \vec{r} = (\min(v, T) - C, 1) \textbf{ and } \vec{\lambda} = (C\mu, \mu) & \\ 1 - e^{-\mu t}, & \textbf{if } v \leq C \textbf{ or } T \leq C \end{cases}$$

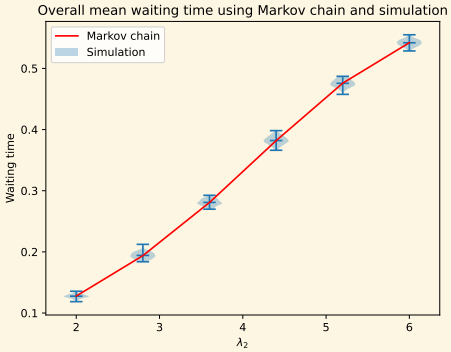
Performance Measures - Proportion within time

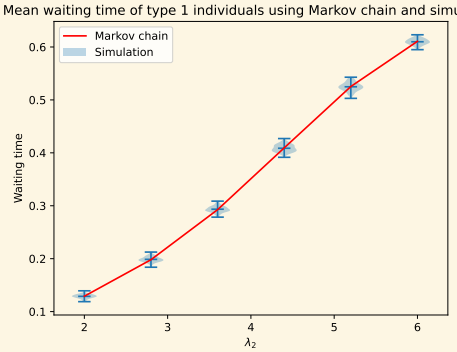
$$P(X^{(i)} < t) = \frac{\sum_{(u,v)\in S_A^{(i)}} P(X_{u,v}^{(i)} < t)\pi_{u,v}}{\sum_{(u,v)\in S_A^{(i)}} \pi_{u,v}}, \quad \textbf{for } i = \{1, 2\}$$

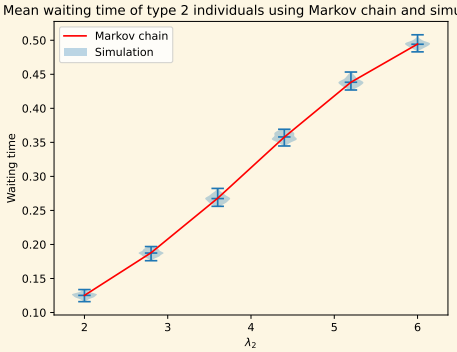
$$X_{(u,v)}^{(1)} \sim \begin{cases} \textbf{Erlang}(v, \mu), & \textbf{if } C = 1 \textbf{ and } v > 1 \\ \textbf{Hypo}\left([v - C, 1], [C\mu, \mu]\right), & \textbf{if } C > 1 \textbf{ and } v > C \\ \textbf{Erlang}(1, \mu), & \textbf{if } v \leq C \end{cases}$$

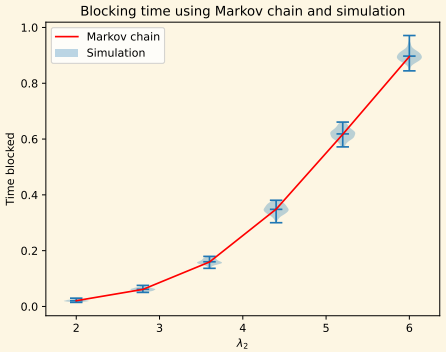
$$X_{(u,v)}^{(2)} \sim \begin{cases} \textbf{Erlang}(\min(v, T), \mu), & \textbf{if } C = 1 \textbf{ and } v, T > 1 \\ \textbf{Hypo}\left([\min(v, T) - C, 1], [C\mu, \mu]\right), & \textbf{if } C > 1 \textbf{ and } v, T > C \\ \textbf{Erlang}(1, \mu), & \textbf{if } v \leq C \textbf{ or } T \leq C \end{cases}$$

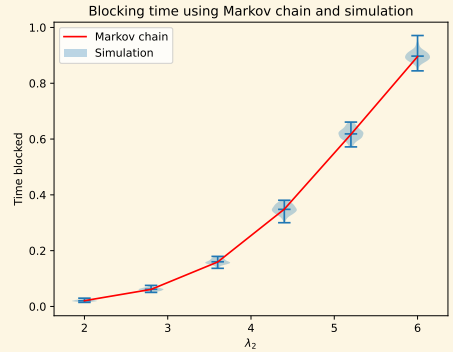
Comparisons

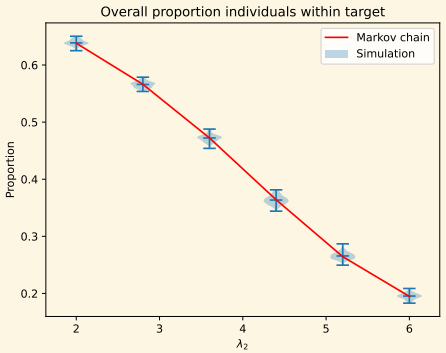


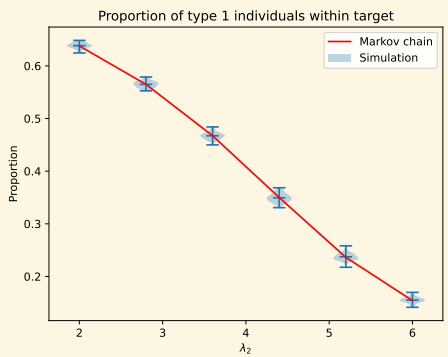


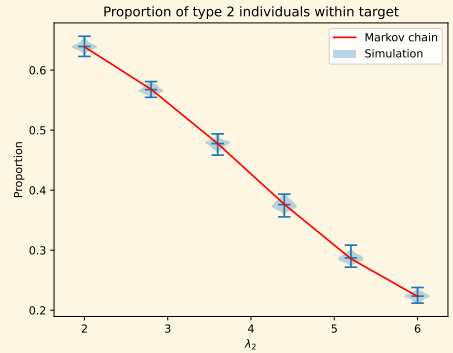






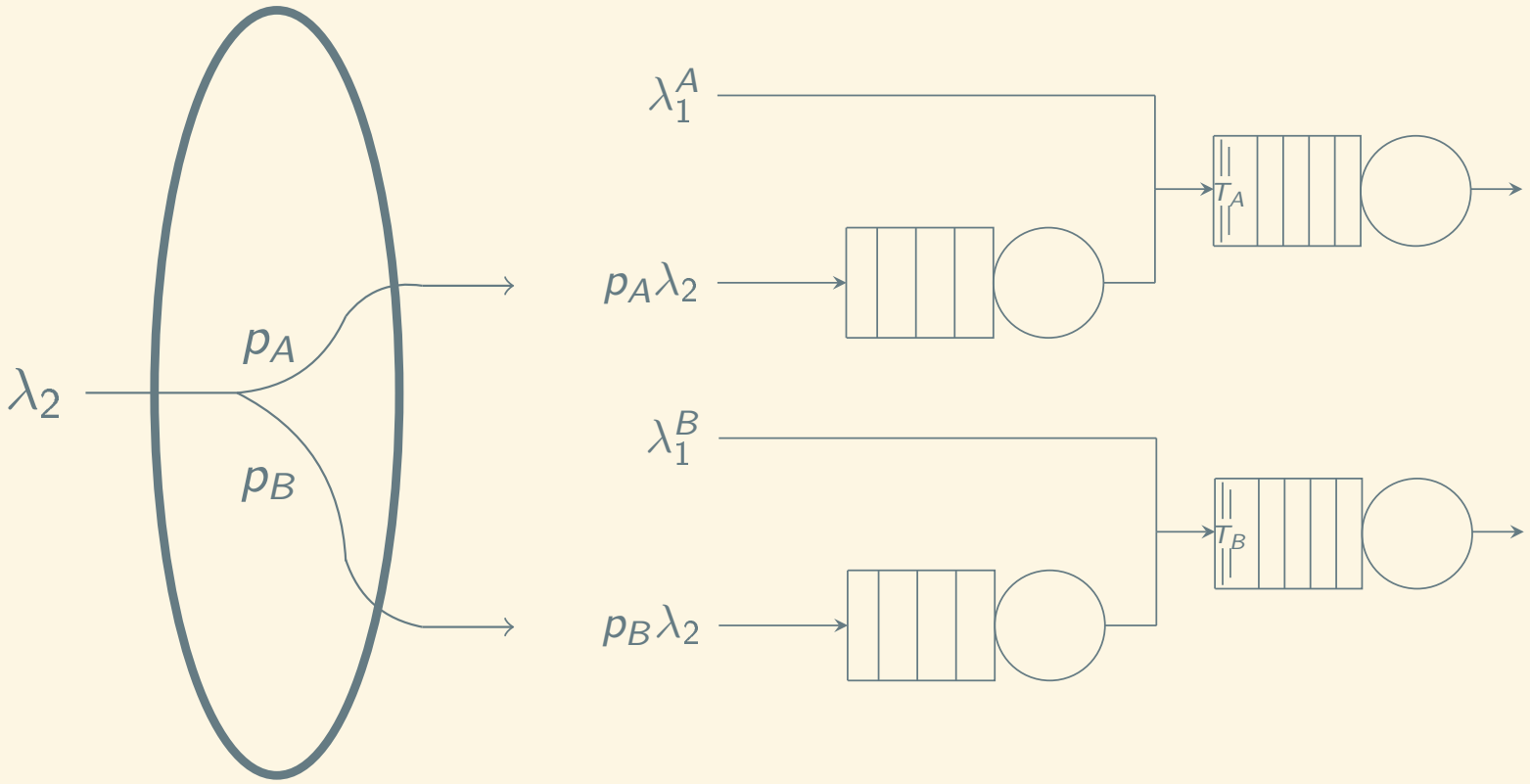




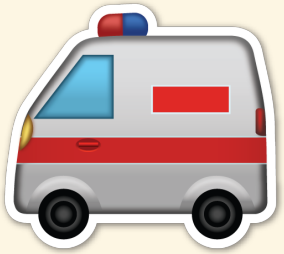


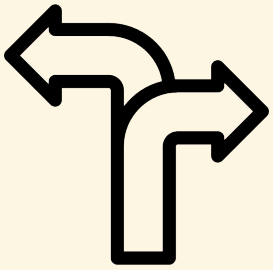
Game - Definition

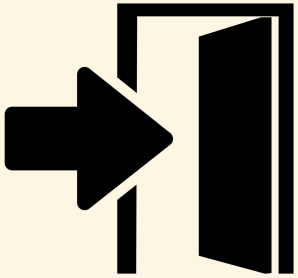
Game - Players

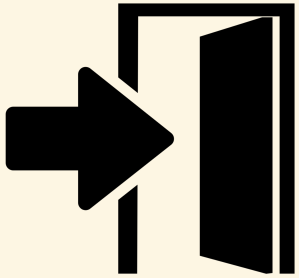


Game - Strategies









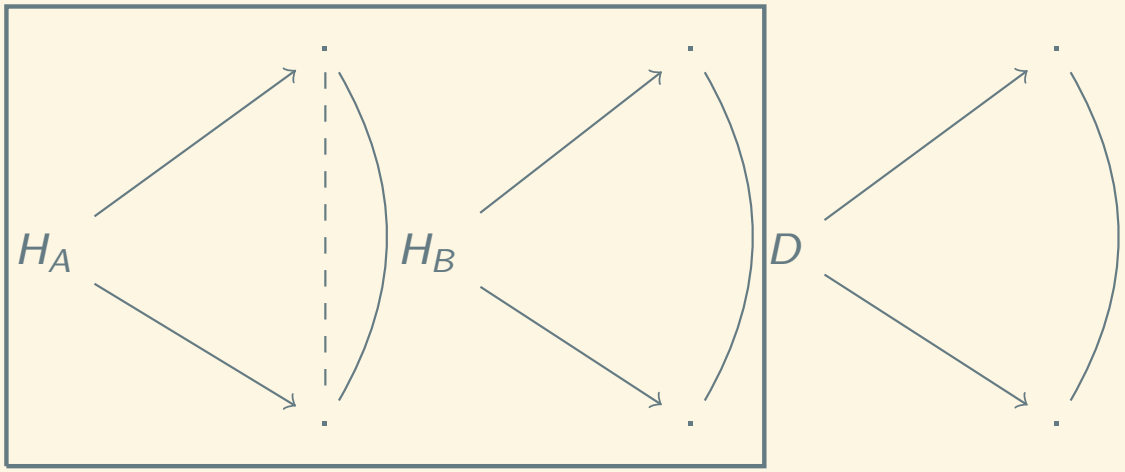
$p_A, p_B \in [0, 1]$

$T_A \in [1, N_A]$

$T_B \in [1, N_B]$

$p_A + p_B = 1$

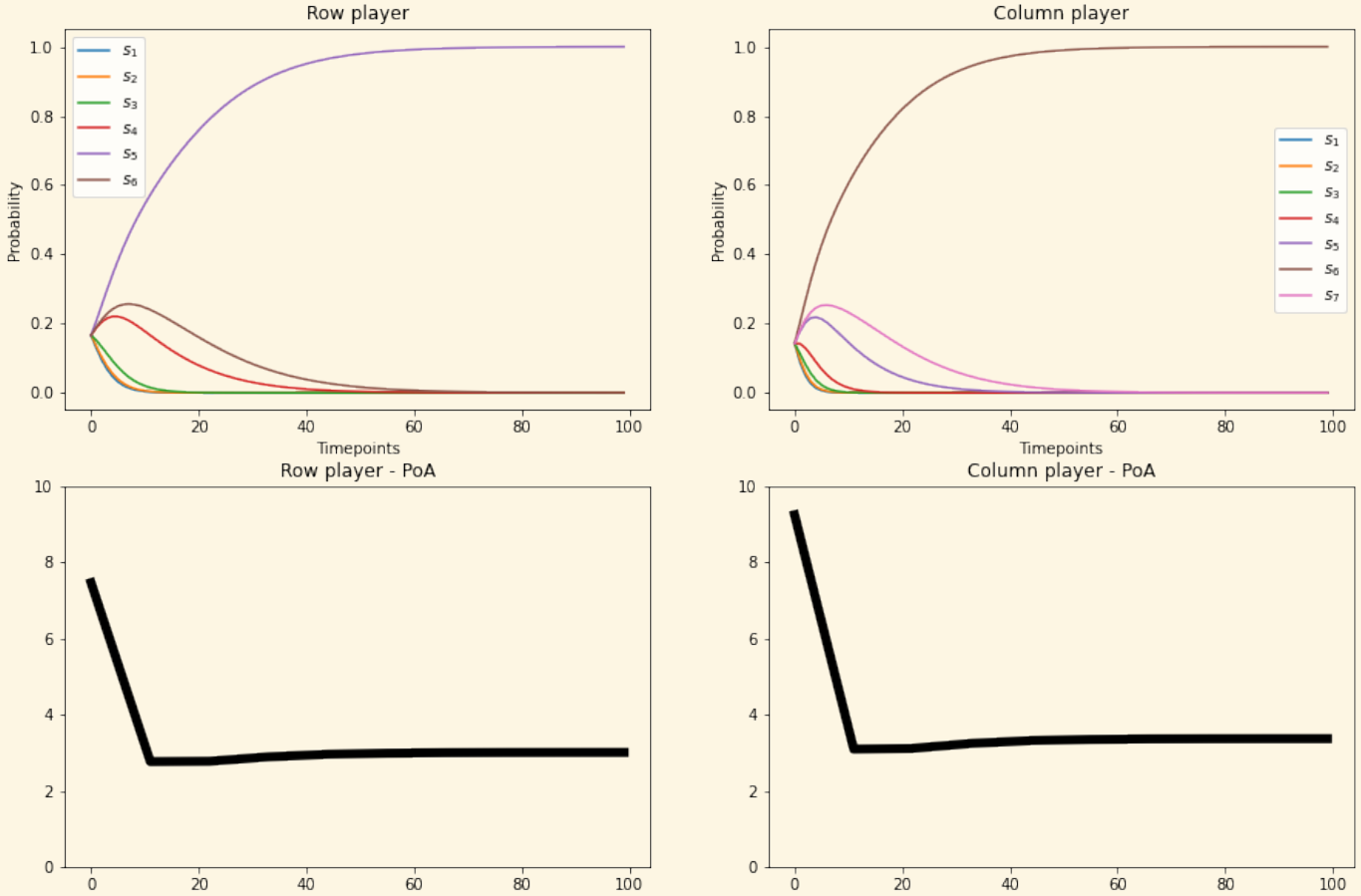
Game - Formulation



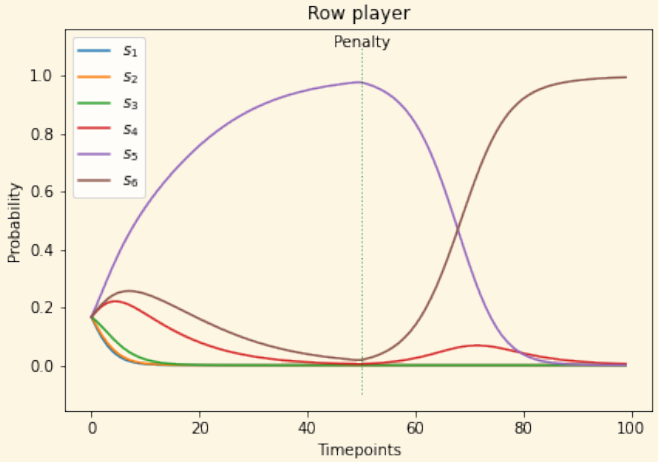
$$A = \begin{pmatrix} U^A_{1,1} & U^A_{1,2} & \cdots & U^A_{1,N_B} \\ U^A_{2,1} & U^A_{2,2} & \cdots & U^A_{2,N_B} \\ \vdots & \vdots & \ddots & \vdots \\ U^A_{N_A,1} & U^A_{N_A,2} & \cdots & U^A_{N_A,N_B} \end{pmatrix}, \quad B = \begin{pmatrix} U^B_{1,1} & U^B_{1,2} & \cdots & U^B_{1,N_B} \\ U^B_{2,1} & U^B_{2,2} & \cdots & U^B_{2,N_B} \\ \vdots & \vdots & \ddots & \vdots \\ U^B_{N_A,1} & U^B_{N_A,2} & \cdots & U^B_{N_A,N_B} \end{pmatrix}$$

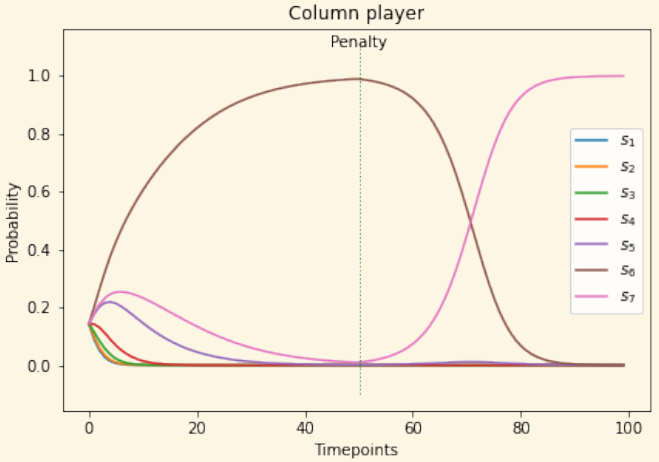
$$R = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,N_B} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,N_B} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N_A,1} & p_{N_A,2} & \cdots & p_{N_A,N_B} \end{pmatrix}$$

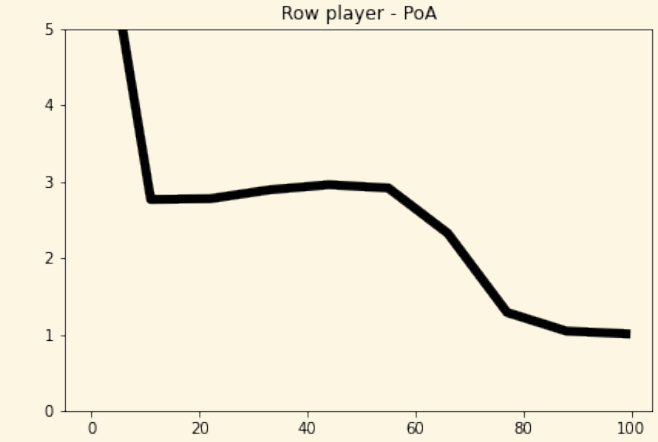
Learning algorithms - Asymmetric replicator dynamics

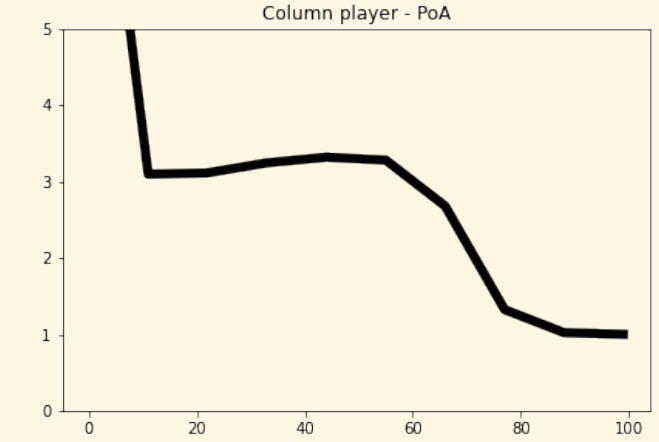


Learning algorithms - Asymmetric replicator dynamics









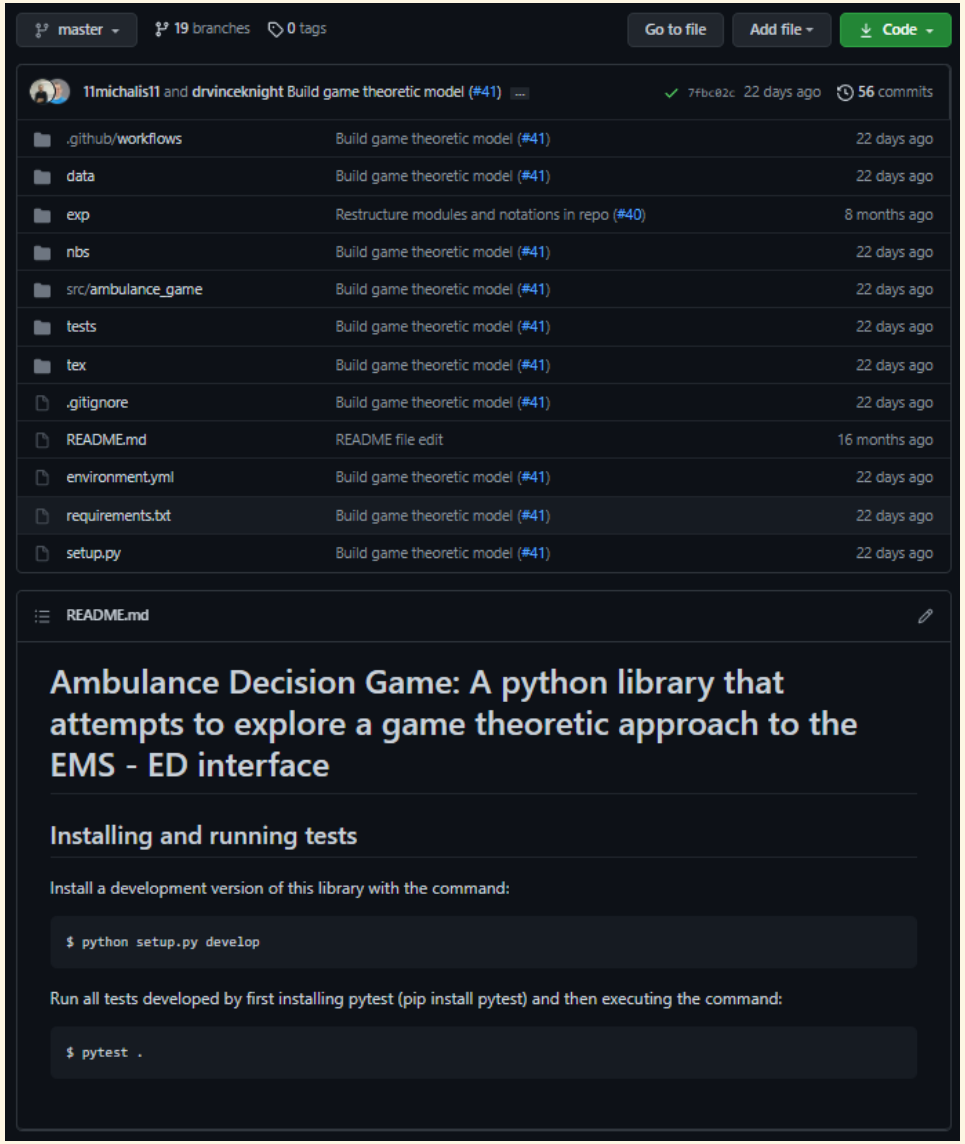
Code / Test

```python
def get_probability_of_accepting(
    all_states,
    pi,
    threshold,
    system_capacity,
    buffer_capacity,
):
    """
    Generates the probability of acceptance for both class types of a given
    Markov model.

    Parameters
    ----------
    all_states : list
    pi : numpy.array
    threshold : int
    system_capacity : int
    buffer_capacity : int

    Returns
    -------
    list
        The probability of accepting an individual upon its arrival for class 0
        and class 1
    """
    prob_accept = [
        np.sum(
            [
                pi[state]
                for state in all_states
                if is_accepting_state(
                    state=state,
                    class_type=class_type,
                    threshold=threshold,
                    system_capacity=system_capacity,
                    buffer_capacity=buffer_capacity,
                )
            ]
        )
        for class_type in range(2)
    ]
    return prob_accept
```

```python
def test_get_probability_of_accepting_example():
    """
    Test that the probability of accepting an individual is as expected
    """
    all_states = [
        (0, 0),
        (0, 1),
        (0, 2),
        (0, 3),
        (1, 3),
        (2, 3),
        (0, 4),
        (1, 4),
        (2, 4),
        (0, 5),
        (1, 5),
        (2, 5),
    ]
    pi = np.array(
        [
            [0.1, 0.1, 0.1, 0.1, 0.1, 0.1],
            [np.nan, np.nan, np.nan, 0.1, 0.1, 0.05],
            [np.nan, np.nan, np.nan, 0.05, 0.05, 0.05],
        ]
    )

    assert np.allclose(
        get_probability_of_accepting(
            all_states=all_states,
            pi=pi,
            threshold=2,
            system_capacity=5,
            buffer_capacity=2,
        ),
        [0.8, 0.85],
    )
```

Michalis Panayides, 3 weeks ago via PR

GitHub repository



master · 19 branches · 0 tags · Go to file · Add file · Code

11michalis11 and drvinceknight Build game theoretic model (#41) ... 7fbc02c 22 days ago 56 commits

.github/workflows	Build game theoretic model (#41)	22 days ago
data	Build game theoretic model (#41)	22 days ago
exp	Restructure modules and notations in repo (#40)	8 months ago
nbs	Build game theoretic model (#41)	22 days ago
src/ambulance_game	Build game theoretic model (#41)	22 days ago
tests	Build game theoretic model (#41)	22 days ago
tex	Build game theoretic model (#41)	22 days ago
.gitignore	Build game theoretic model (#41)	22 days ago
README.md	README file edit	16 months ago
environment.yml	Build game theoretic model (#41)	22 days ago
requirements.txt	Build game theoretic model (#41)	22 days ago
setup.py	Build game theoretic model (#41)	22 days ago

README.md

Ambulance Decision Game: A python library that attempts to explore a game theoretic approach to the EMS - ED interface

Installing and running tests

Install a development version of this library with the command:

```
$ python setup.py develop
```

Run all tests developed by first installing pytest (pip install pytest) and then executing the command:

```
$ pytest .
```