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

Michalis Panayides

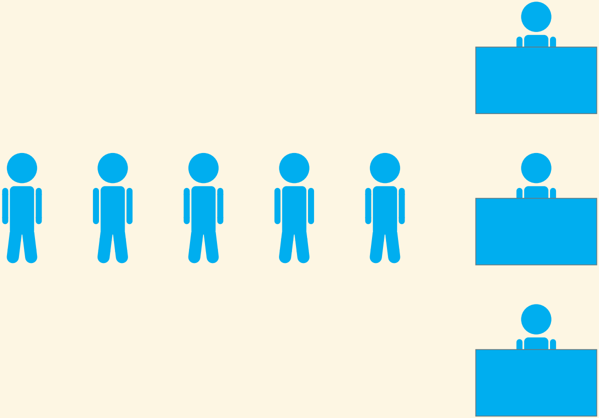
EURO 2021 Athens

About me

About me

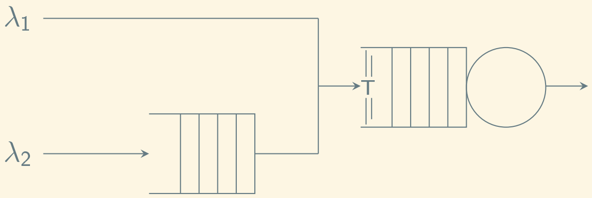
Queues - Examples

Queues - Examples

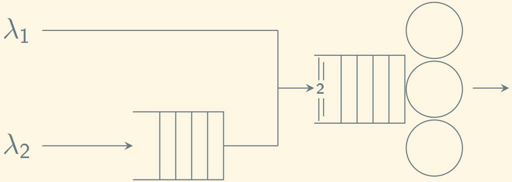


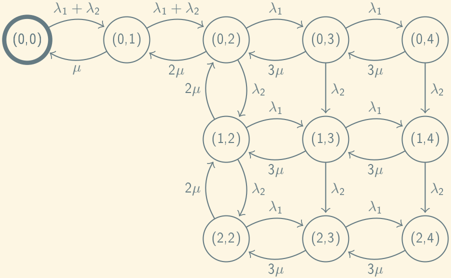
Queues - Examples

Queueing network structure

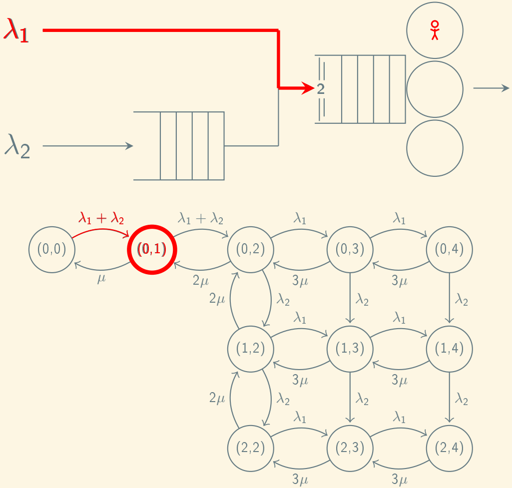


Markov Chain - Custom network

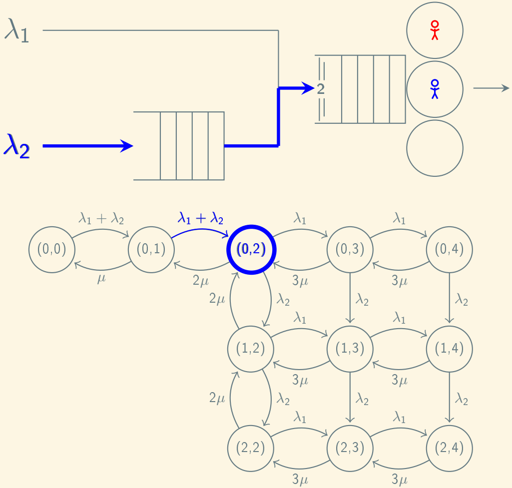




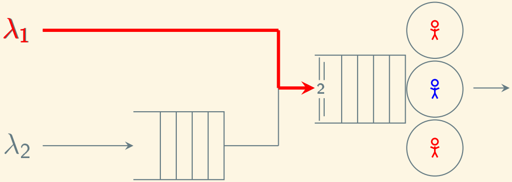
Markov Chain - Custom network

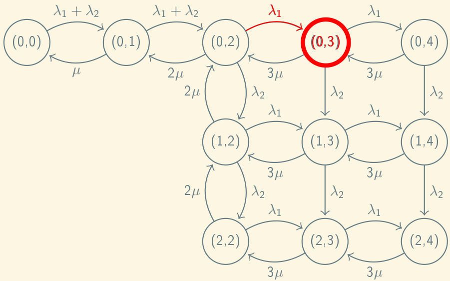


Markov Chain - Custom network

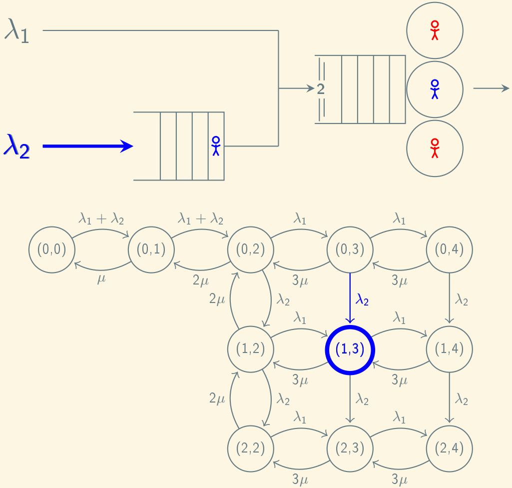


Markov Chain - Custom network

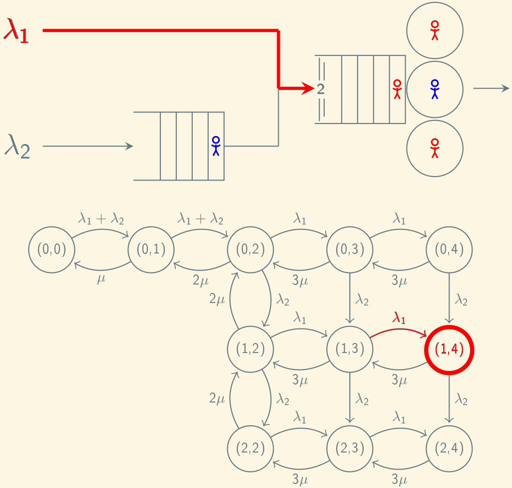




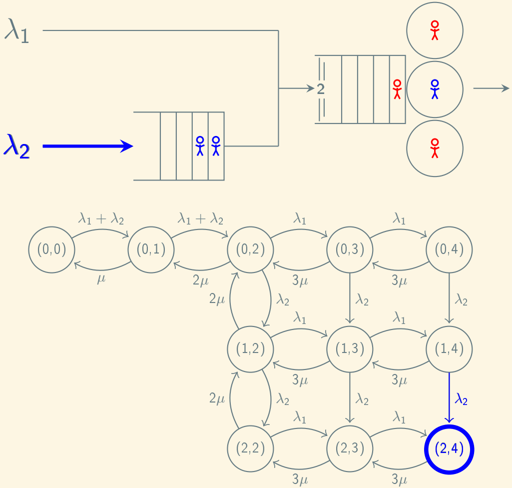
Markov Chain - Custom network



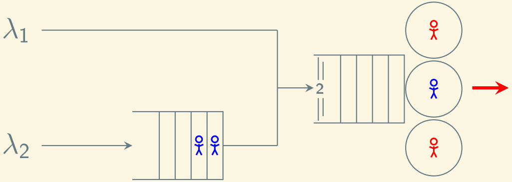
Markov Chain - Custom network

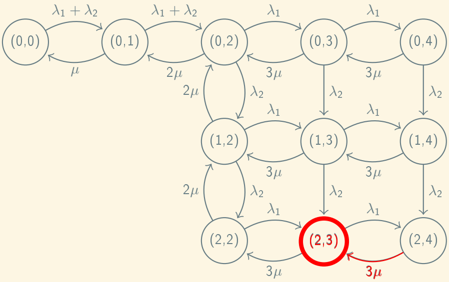


Markov Chain - Custom network

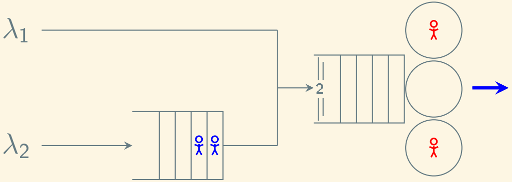


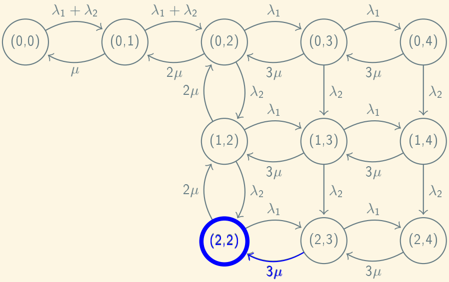
Markov Chain - Custom network



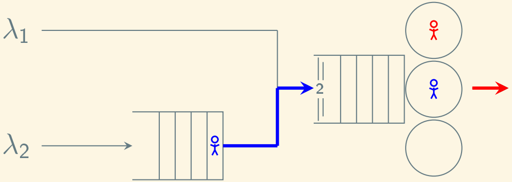


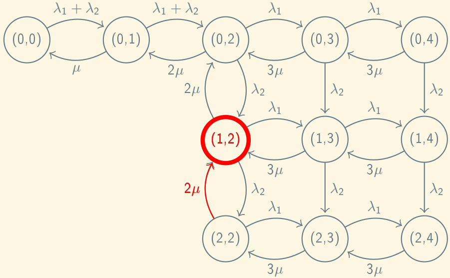
Markov Chain - Custom network





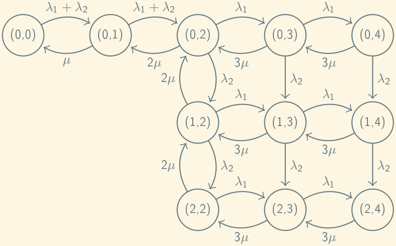
Markov Chain - Custom network





Steady state probabilities - Custom network

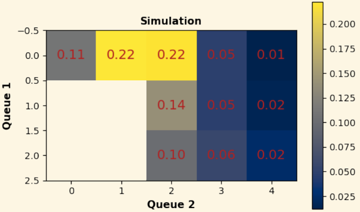
$$
\begin{array}{ccc}
 & \begin{array}{cccccc} (0,0) & (0,1) & (0,2) & & (2,3) & (2,4) \end{array} & \\
Q = & \begin{pmatrix}
-\lambda_1 - \lambda_2 & \lambda_1 + \lambda_2 & 0 & \dots & 0 & 0 \\
\mu & -\mu - \lambda_1 - \lambda_2 & \lambda_1 + \lambda_2 & \dots & 0 & 0 \\
0 & 2\mu & -2\mu - \lambda_1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

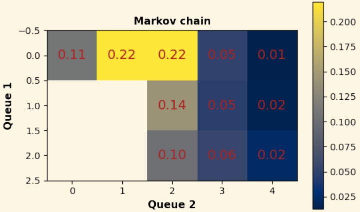


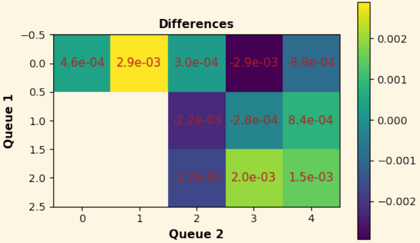
$$\frac{d\pi}{dt} = \pi Q = 0, \qquad \sum \pi_{(u,v)} = 1$$

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

Steady state probabilities - Comparison







Performance Measures - Number of individuals

$$(u_i, v_i) \longrightarrow \pi_{(u_i, v_i)}$$

Performance Measures - Number of individuals

$$(u_i, v_i) \longrightarrow \pi_{(u_i, v_i)}$$

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

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

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

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

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

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

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

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

Performance Measures - Waiting time

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

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

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

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

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

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

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

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

Performance Measures - Blocking time

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

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

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

Performance Measures - Blocking time

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

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

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

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

Performance Measures - Blocking time

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

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

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

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

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

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

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

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

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

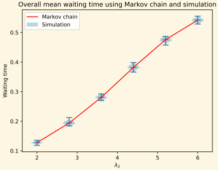
Performance Measures - Proportion within time

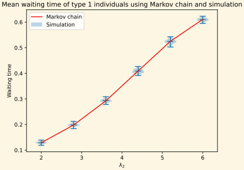
$$P(X^{(i)} < t) = \frac{\sum_{(u,v)\in S_A^{(i)}} P(X_{u,v}^{(i)} < t)\pi_{u,v}}{\sum_{(u,v)\in S_A^{(i)}} \pi_{u,v}}, \quad \textbf{for } i = \{1, 2\}$$

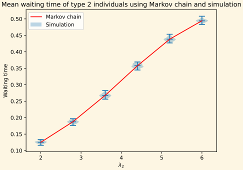
$$X_{(u,v)}^{(1)} \sim \begin{cases} \textbf{Erlang}(v, \mu), & \textbf{if } C = 1 \textbf{ and } v > 1 \\ \textbf{Hypo}\left([v - C, 1], [C\mu, \mu]\right), & \textbf{if } C > 1 \textbf{ and } v > C \\ \textbf{Erlang}(1, \mu), & \textbf{if } v \leq C \end{cases}$$

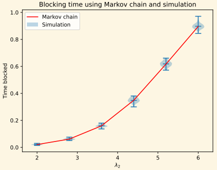
$$X_{(u,v)}^{(2)} \sim \begin{cases} \textbf{Erlang}(\min(v, T), \mu), & \textbf{if } C = 1 \textbf{ and } v, T > 1 \\ \textbf{Hypo}\left([\min(v, T) - C, 1], [C\mu, \mu]\right), & \textbf{if } C > 1 \textbf{ and } v, T > C \\ \textbf{Erlang}(1, \mu), & \textbf{if } v \leq C \textbf{ or } T \leq C \end{cases}$$

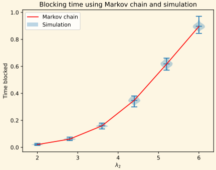
Comparisons

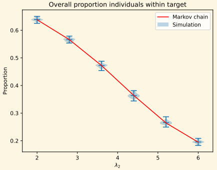


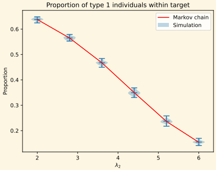


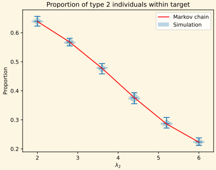






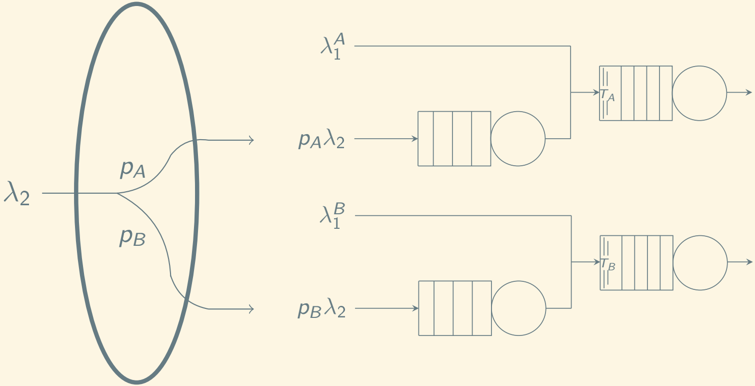




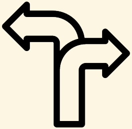


Game - Definition

Game - Players

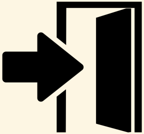


Game - Strategies

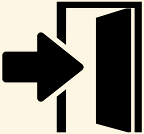


$p_A, p_B \in [0, 1]$

$p_A + p_B = 1$

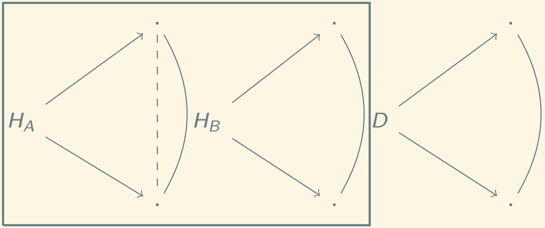


$T_A \in [1, N_A]$



$T_B \in [1, N_B]$

Game - Formulation



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

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

Solution concepts in games

$$A \in \mathbb{R}^{m \times n}, \qquad B \in \mathbb{R}^{m \times n}$$

Solution concepts in games

$$A \in \mathbb{R}^{m\times n}, \qquad B \in \mathbb{R}^{m\times n}$$

$$\frac{dx}{dt}_i = x_i((f_x)_i - \phi_x), \quad \text{for all } i$$

$$\frac{dy}{dt}_i = y_i((f_y)_i - \phi_y), \quad \text{for all } i$$

Solution concepts in games

$$A \in \mathbb{R}^{m \times n}, \qquad B \in \mathbb{R}^{m \times n}$$

$$\frac{dx}{dt}_i = x_i((f_x)_i - \phi_x), \quad \text{for all } i$$

$$\frac{dy}{dt}_i = y_i((f_y)_i - \phi_y), \quad \text{for all } i$$

- Fudenberg, Drew, et al. The theory of learning in games. Vol. 2. MIT press, 1998.
- Elvio, Accinelli and Carrera, Edgar. 2011. Evolutionarily Stable Strategies and Replicator Dynamics in Asymmetric Two-Population Games. 10.1007/978-3-642-11456-4_3.

Inefficiency measure

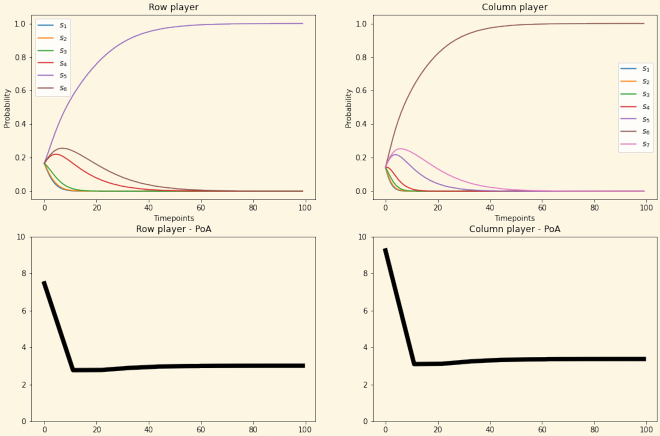
$$PoA = \frac{\max_{s \in E} Cost(s)}{\min_{s \in S} Cost(S)}$$

Inefficiency measure

$$PoA = \frac{\max_{s \in E} Cost(s)}{\min_{s \in S} Cost(S)}$$

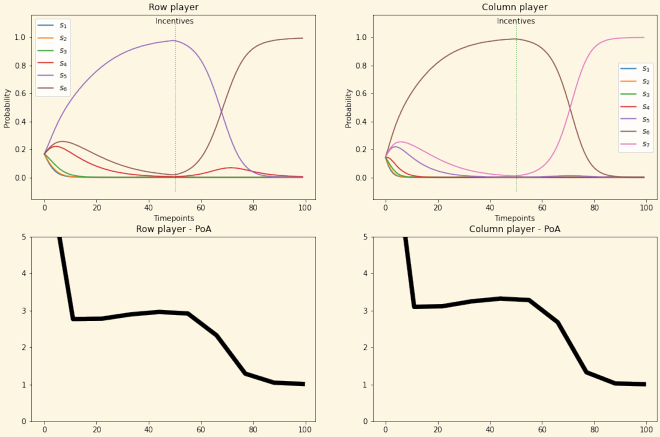
$$PoA_A(s_r) = \frac{Cost(s_r)}{\min_{s \in S} Cost(S)}, \qquad PoA_B(s_c) = \frac{Cost(s_c)}{\min_{s \in S} Cost(S)}$$

Learning algorithms - Asymmetric replicator dynamics



"Inefficiencies can be learned and emerged naturally in an interactive system"

Learning algorithms - Asymmetric replicator dynamics



"Targeted incentivisation of behaviours can help escape learned inefficiencies"

Code / Test

```python
def get_probability_of_accepting( ···
):
    """
    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,      Michalis Panayides, 3 weeks ago via PR
        ),
        [0.8, 0.85],
    )
```

GitHub repository