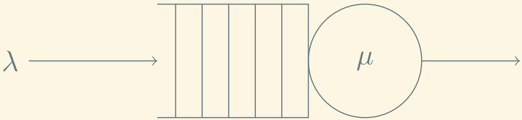
A 3-player game theoretic model of a choice between two queueing systems with strategic managerial decision making

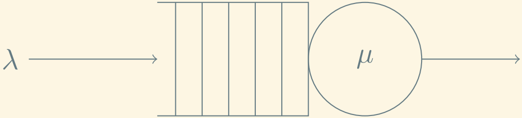
Michalis Panayides

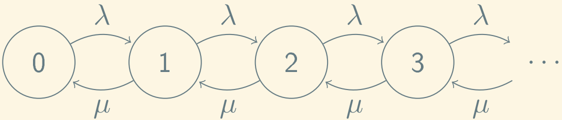
Supervisors:
Dr. Vince Knight,
Prof. Paul Harper

Queues - M/M/1

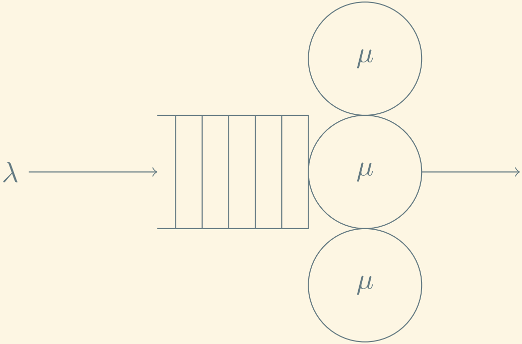


Queues - M/M/1

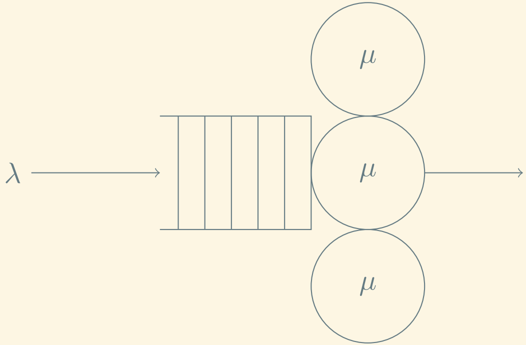




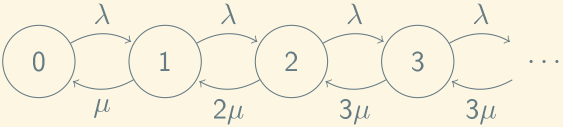
Queues - M/M/3



Queues - M/M/3



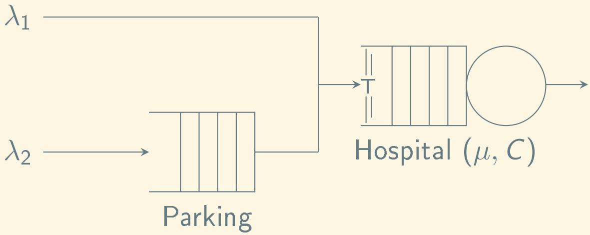
λ
μ
μ
μ



0
1
2
3
$\cdots$
λ
λ
λ
λ
μ
2μ
3μ
3μ

Queues - Custom network of queues

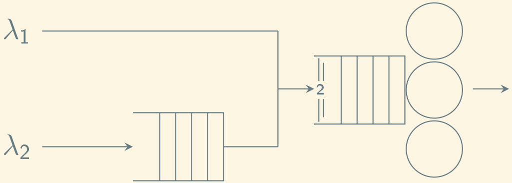
Queues - Custom network of queues

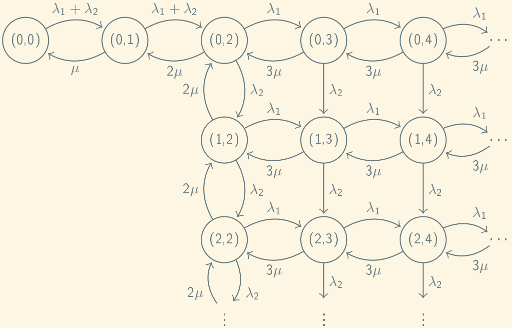


Parameters:

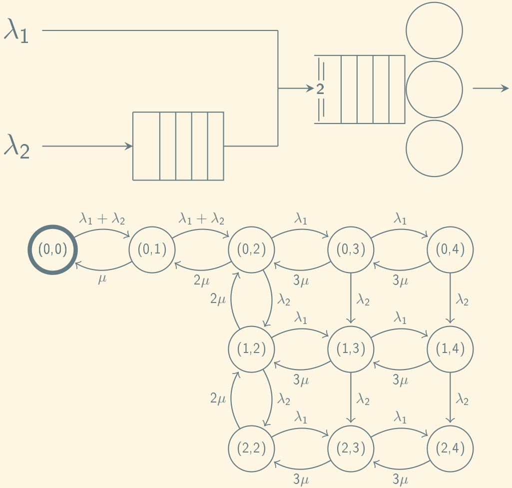
- λ_1: Arrival rate of type 1 individuals
- λ_2: Arrival rate of type 2 individuals
- μ: Service rate
- C: Number of servers
- T: Threshold

Markov Chain - Custom network

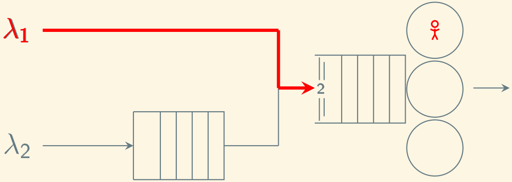


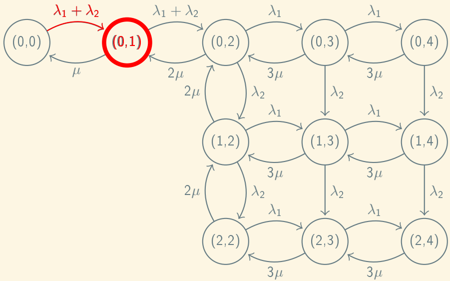


Markov Chain - Custom network - $N = 4, M = 2$

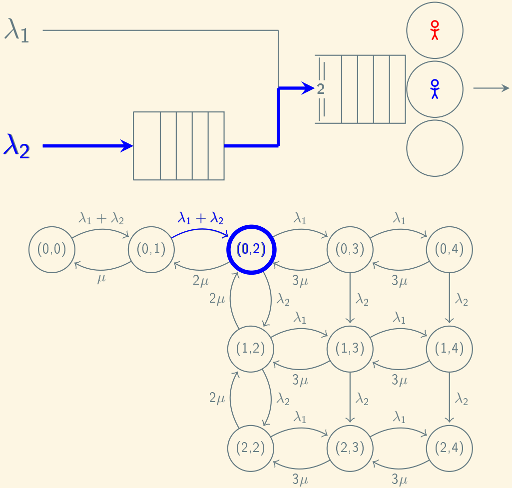


Markov Chain - Custom network - $N = 4, M = 2$

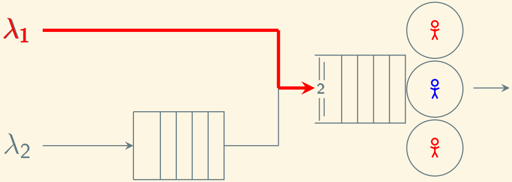


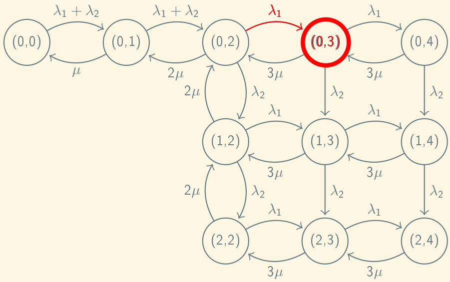


Markov Chain - Custom network - $N = 4, M = 2$

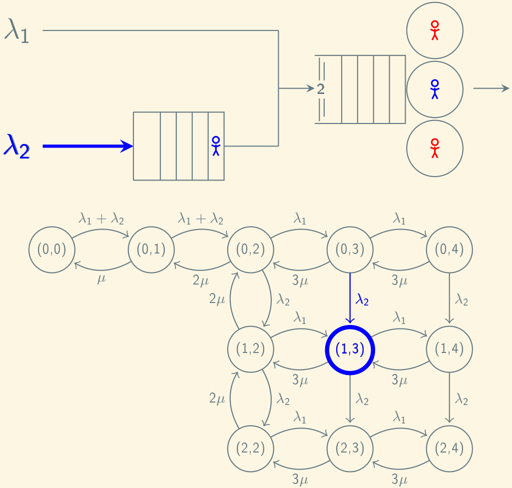


Markov Chain - Custom network - $N = 4, M = 2$

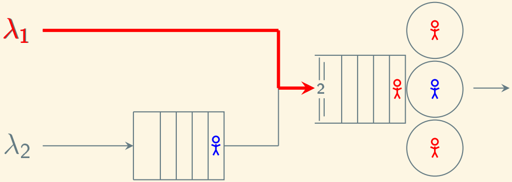


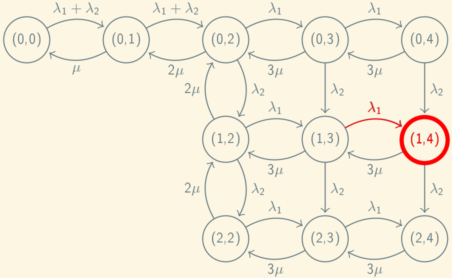


Markov Chain - Custom network - $N = 4, M = 2$

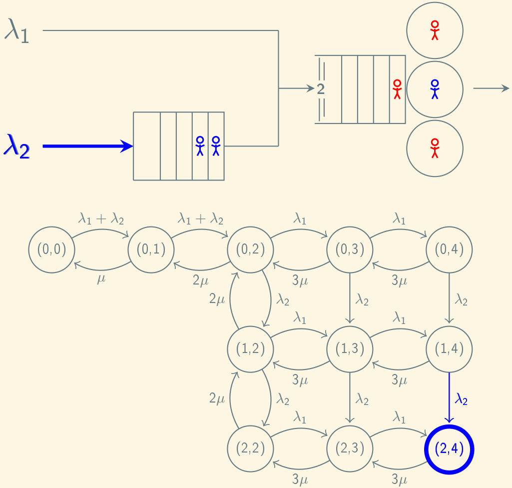


Markov Chain - Custom network - $N = 4, M = 2$

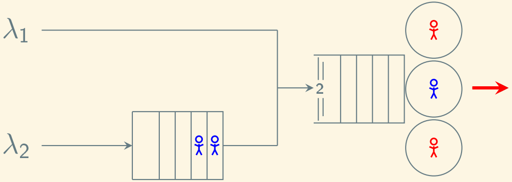


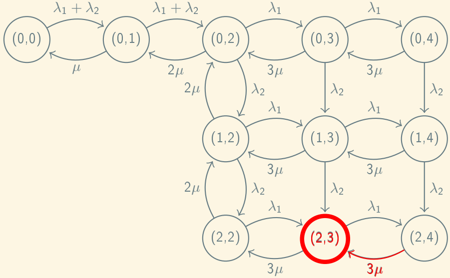


Markov Chain - Custom network - $N = 4, M = 2$

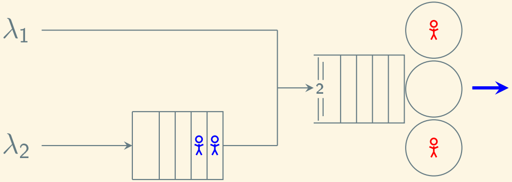


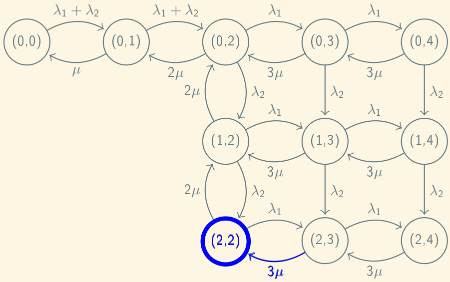
Markov Chain - Custom network - $N = 4, M = 2$



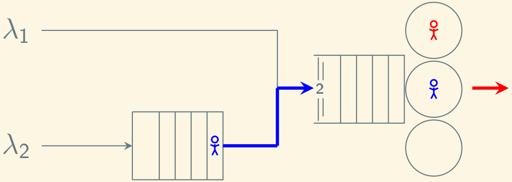


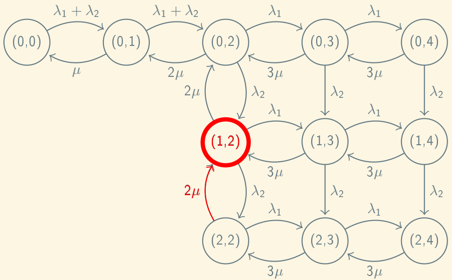
Markov Chain - Custom network - $N = 4, M = 2$





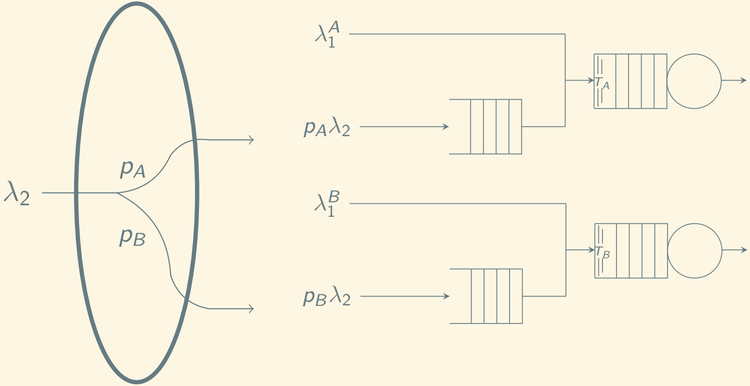
Markov Chain - Custom network - $N = 4, M = 2$





Game - Definition

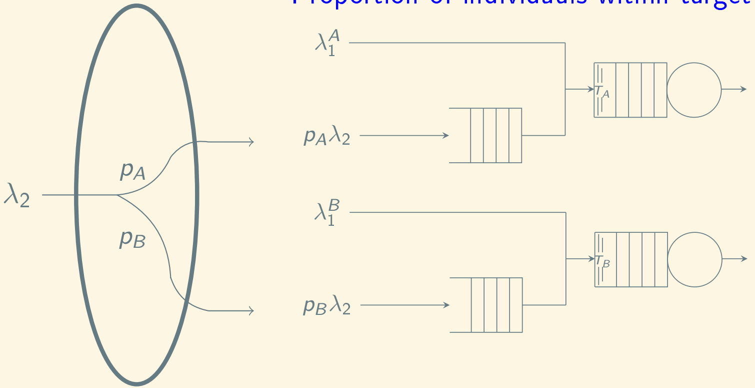
Game - Players and objectives



Game - Players and objectives

Blocking time

Proportion of individuals within target



Performance Measures - Blocking time

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

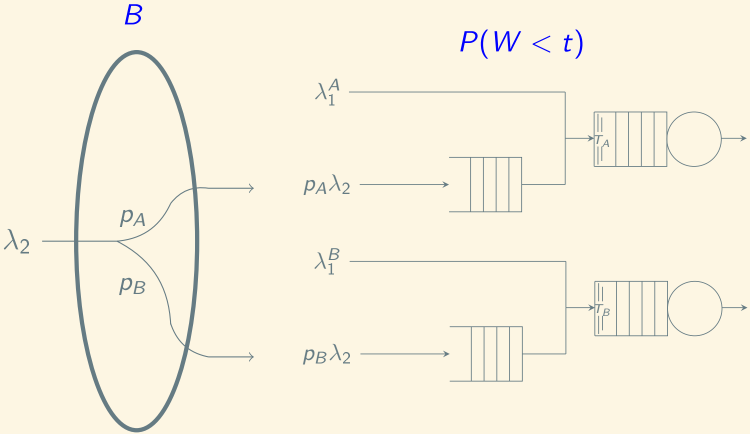
Performance Measures - Proportion within target

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

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

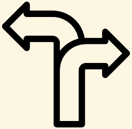
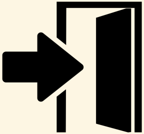
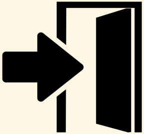
$$P(W^{(2)} < t) = \frac{\sum_{(u,v) \in S_A^{(2)}} P(W^{(2)}_{(u,v)} < t) \pi_{u,v}}{\sum_{(u,v) \in S_A^{(2)}} \pi_{u,v}}$$

Game - Players and objectives



Game - Strategies



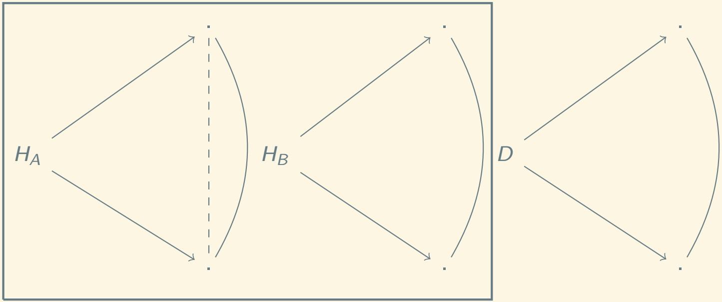
$p_A, p_B \in [0, 1]$

$T_A \in [1, N_A]$

$T_B \in [1, N_B]$

$p_A + p_B = 1$

Game - Formulation



Hospital's utility

$$U^{(i)}_{T_A,T_B} = 1 - \left[\left(P(X^{(i)} < t) - 0.95 \right)^2 \right]$$

Game - Payoff matrices

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

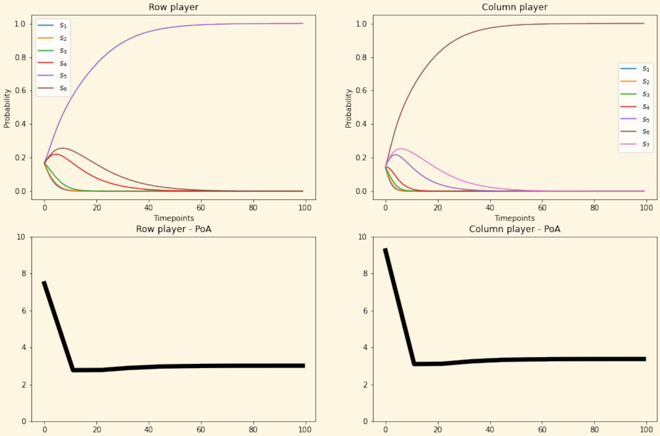
$$R = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,N_B} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,N_B} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N_A,1} & p_{N_A,2} & \cdots & p_{N_A,N_B} \end{pmatrix}$$

Asymmetric Replicator Dynamics

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

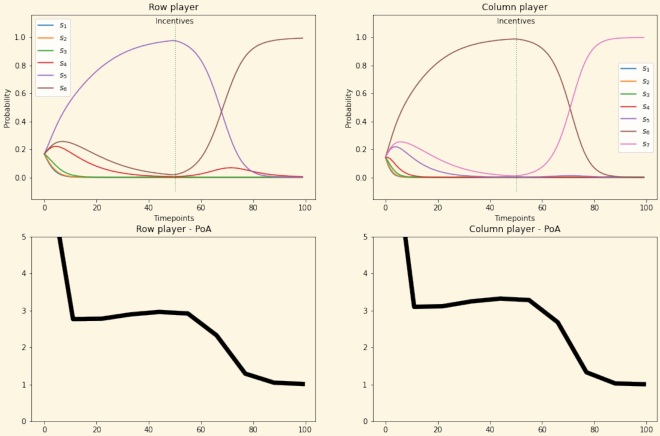
$$\frac{dy}{dt}_i = y_i((f_y)_i - \phi_y), \qquad \text{for all } i$$

Learning algorithms - Asymmetric replicator dynamics



Inefficiencies can be learned and emerge naturally

Learning algorithms - Asymmetric replicator dynamics



Targeted incentivisation of behaviours can help escape learned inefficiencies

Thank you!

$ pip install ambulance_game

https://github.com/11michalis11/AmbulanceDecisionGame

PanayidesM@cardiff.ac.uk

@Michalis_Pan

@11michalis11