Using Queueing Network Modelling to Assess the Impact of the OPICP

Geraint Palmer

Paul Harper, Vincent Knight

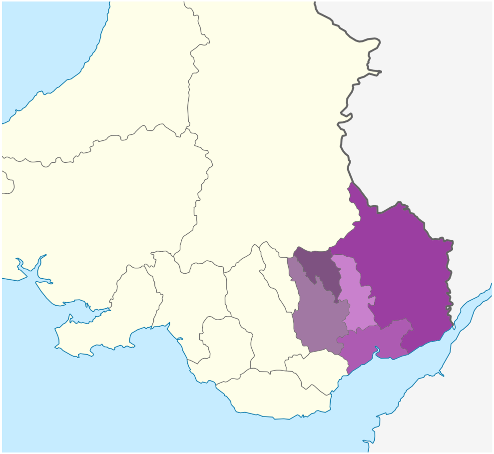
8th IMA International Conference on Quantitative Modelling in the Management of Health and Social Care

CARDIFF UNIVERSITY

PRIFYSGOL CAERDYDD

Aneurin Bevan University Health Board

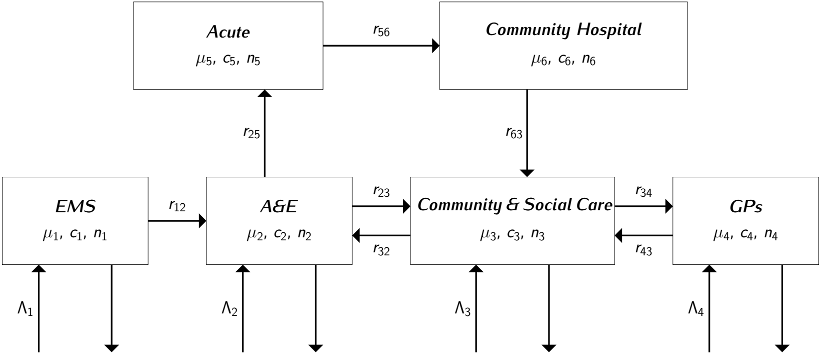
GIG CYMRU NHS WALES
Bwrdd Iechyd Prifysgol Aneurin Bevan University Health Board



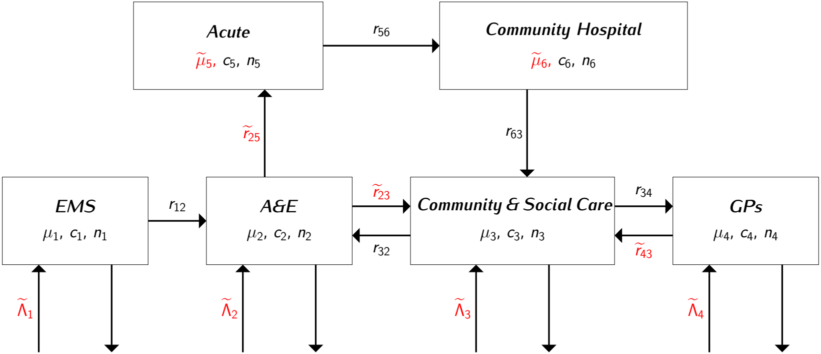
Older People's Integrated Care Pathway

- Pathway focused around pro-active patient centred coordinated care
- Individuals identified through ***risk stratification*** as being at risk of admission to institutionalised care or becoming frequent users of high cost care
- Develop holistic personal ***Stay Well Plans*** for these individuals, utilising ***low and no cost services***
- Aim to keep individuals and carers as well and as independent as possible

Elderly People's Flows Through Health System



Elderly People's Flows Through Health System



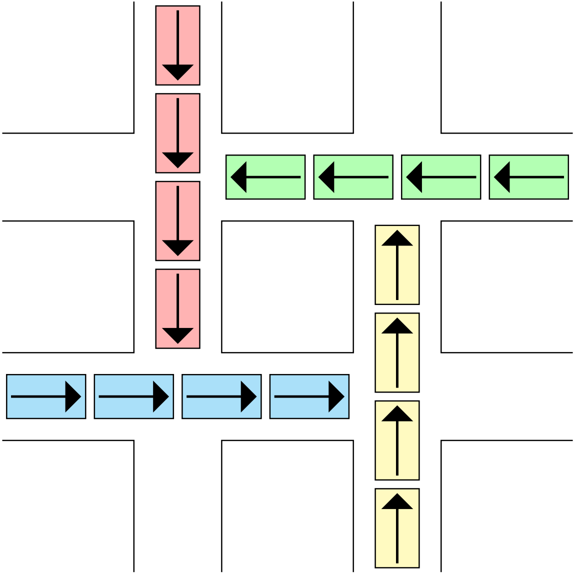
Simulation with Ciw

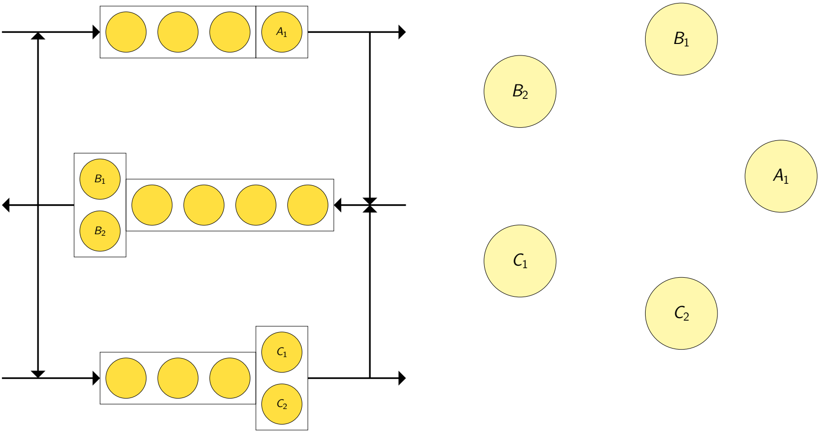
Open Source Python Library
Three-Phase Simulation

https://github.com/geraintpalmer/Ciw
https://pypi.python.org/pypi/Ciw
http://ciw.readthedocs.org

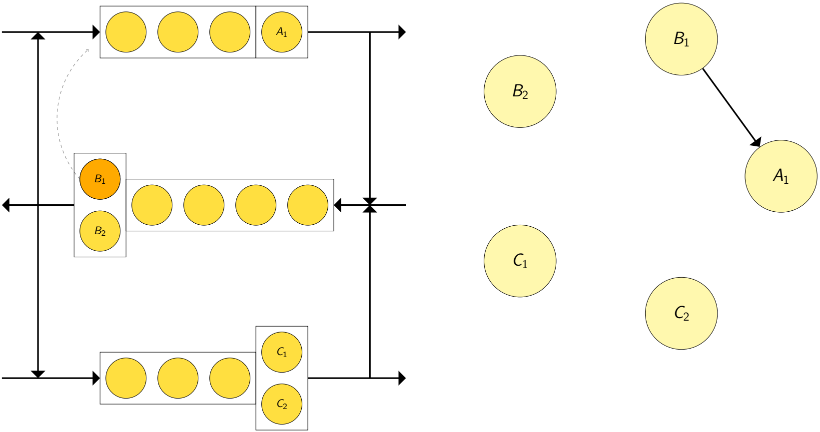
```
>>> import ciw
>>> params = {
...   'Number_of_nodes': 2,
...   'Arrival_distributions':
...       [['Exponential', 6.0],
...        ['Uniform', 0.2, 0.4]],
...   'Service_distributions':
...       [['Lognormal', 0.4, 0.3],
...        ['Deterministic', 0.25]],
...   'Transition_matrices':
...       [[0.5, 0.1],
...        [0.0, 0.4]],
...   'Number_of_servers': [3, 1],
...   'Simulation_time': 2000}
>>> Q = ciw.Simulation(params)
>>> Q.simulate_until_max_time()
>>> recs = Q.get_all_records()
```

Deadlock

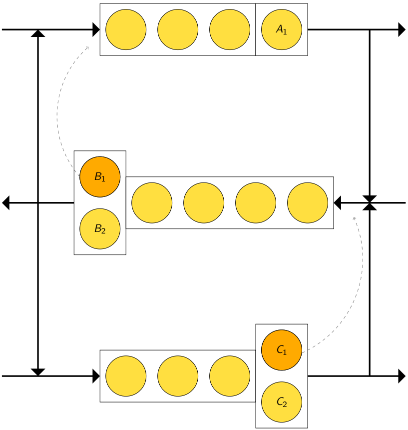




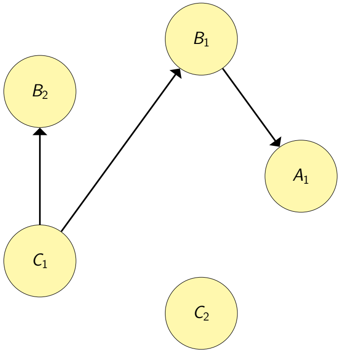
A_1
B_1
B_2
C_1
C_2
B_1
B_2
A_1
C_1
C_2



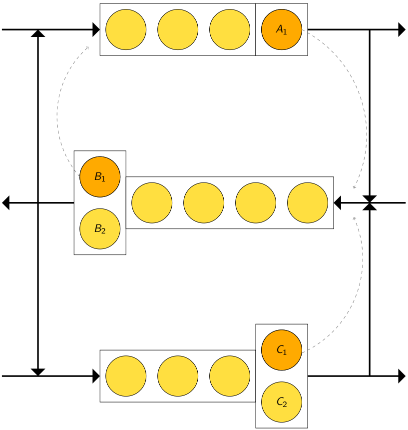
A_1
B_1
B_2
C_1
C_2
B_1
B_2
A_1
C_1
C_2

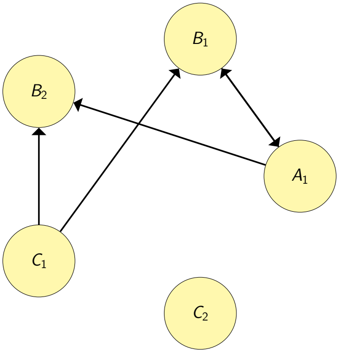


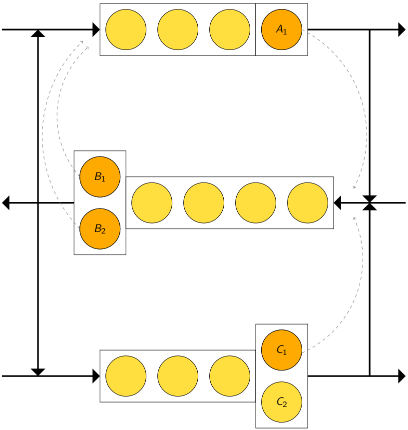
A_1
B_1
B_2
C_1
C_2



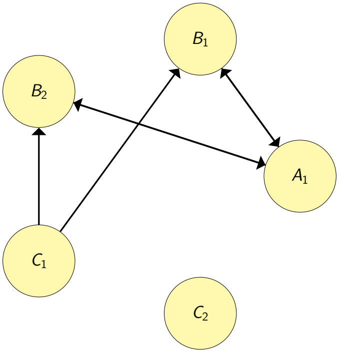
B_1
B_2
A_1
C_1
C_2


A_1
B_1
B_2
C_1
C_2

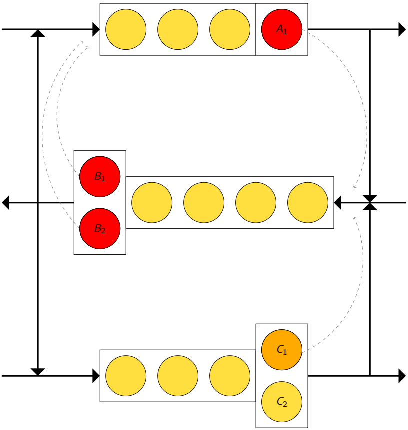

B_1
B_2
A_1
C_1
C_2



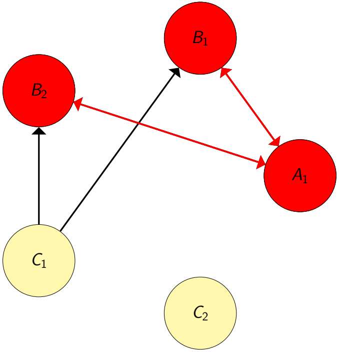
A_1
B_1
B_2
C_1
C_2



B_1
B_2
A_1
C_1
C_2



A_1
B_1
B_2
C_1
C_2

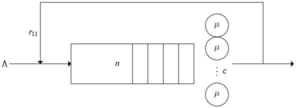


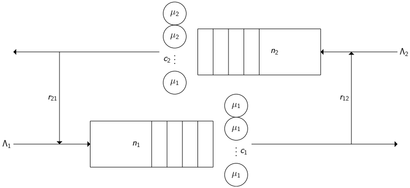
B_1
B_2
A_1
C_1
C_2

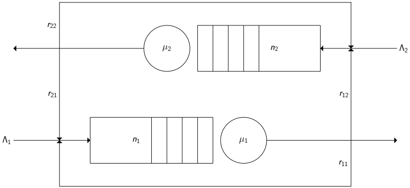
Deadlock Detection in Ciw

```
>>> import ciw
>>> params = {'Number_of_nodes': 1,
...           'Arrival_distributions': [['Exponential', 6.0]],
...           'Service_distributions': [['Exponential', 5.0]],
...           'Transition_matrices': [[0.5]],
...           'Number_of_servers': [1],
...           'Queue_capacities': [3],
...           'Detect_deadlock': True}
>>> Q = ciw.Simulation(params)
>>> times_to_deadlock = Q.simulate_until_deadlock()
>>> times_to_deadlock[((0, 0),)]
1.1707879982560288
```

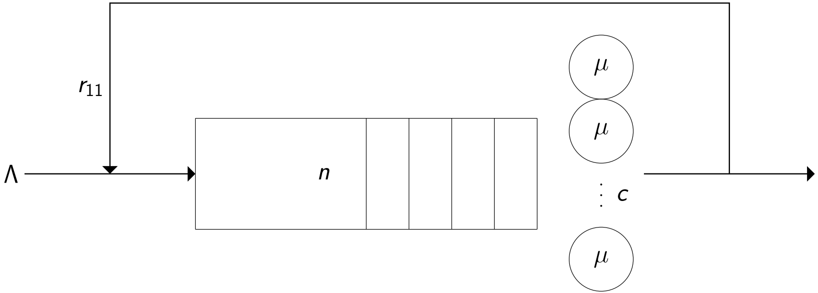
Three Deadlocking Queueing Networks







Markovian Model of Deadlock



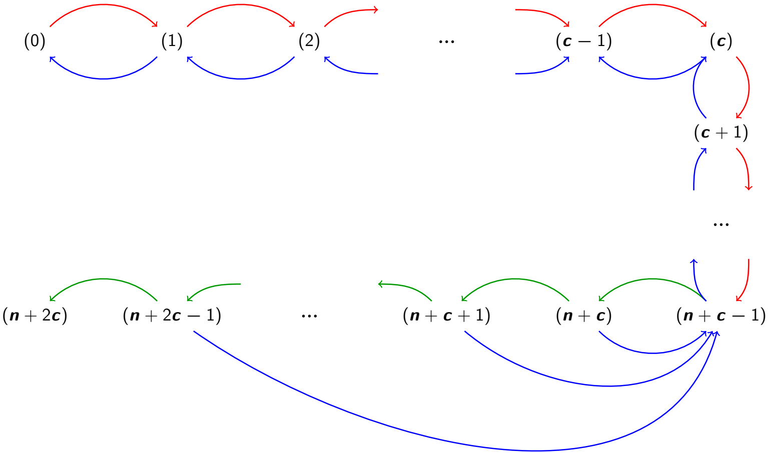
(i)

$$S = \{i \in \mathbb{N} \mid 0 \leq i \leq n + 2c\}$$

Define $\delta = i_2 - i_1$

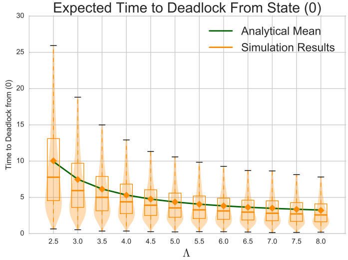
$$q_{i_1,i_2} = \begin{Bmatrix} \Lambda & \text{if } \delta = 1 \\ (1 - r_{11})\mu \min(i, c) & \text{if } \delta = -1 \\ 0 & \text{otherwise} \end{Bmatrix} \quad \text{if } i_1 < n + c$$

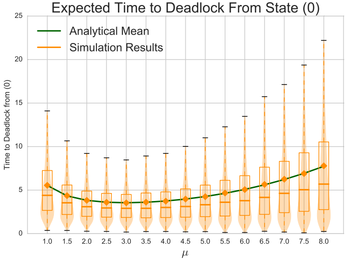
$$q_{i_1,i_2} = \begin{Bmatrix} (c - b)r_{11}\mu & \text{if } \delta = 1 \\ (1 - r_{11})(c - b)\mu & \text{if } \delta = -b - 1 \\ 0 & \text{otherwise} \end{Bmatrix} \quad \text{if } i_1 = n + c + b \quad \forall \quad 0 \leq b \leq c$$

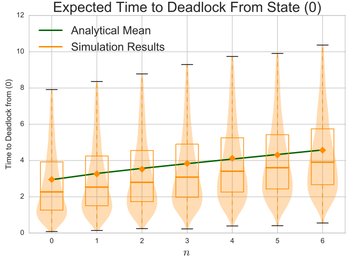


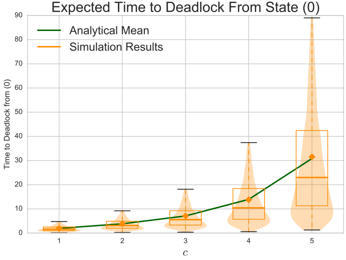
(0)
(1)
(2)
...
$(c-1)$
(c)
$(c+1)$
...
$(n+2c)$
$(n+2c-1)$
...
$(n+c+1)$
$(n+c)$
$(n+c-1)$

Times to Deadlock

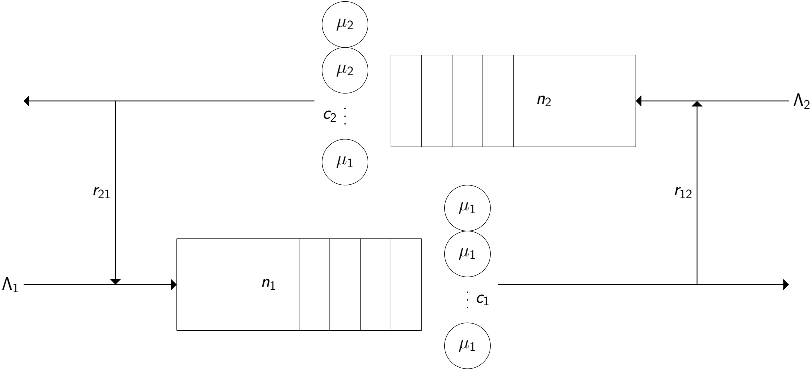








Markovian Model of Deadlock

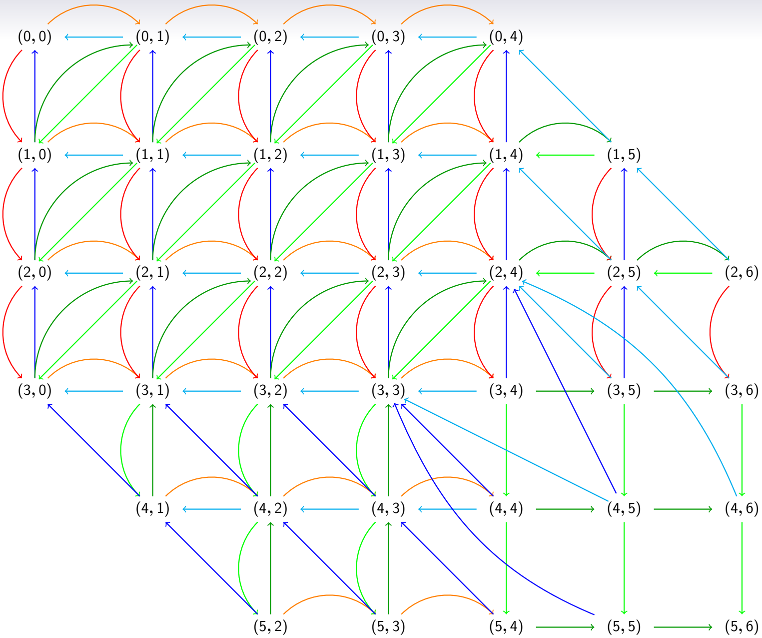


$$(i, j)$$

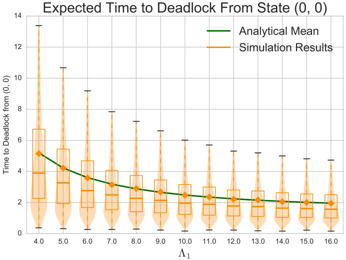
$$S = \{(i,j) \in \mathbb{N}^{(n_1+c_1+c_2)\times(n_2+c_2+c_1)} \mid i \le n_1 + c_1 + j,\ j \le n_2 + c_2 + i\}$$

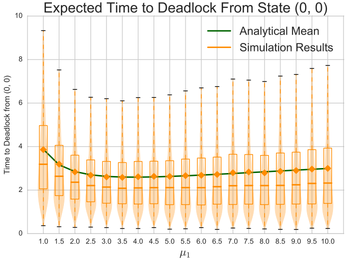
$$\begin{aligned}
\delta &= (i_2, j_2) - (i_1, j_1) \\
b_1 &= \max(0, i_1 - n_1 - c_1) \\
b_2 &= \max(0, i_2 - n_2 - c_2) \\
s_1 &= \min(i_1, c_1) - b_2 \\
s_2 &= \min(i_2, c_2) - b_1
\end{aligned}$$

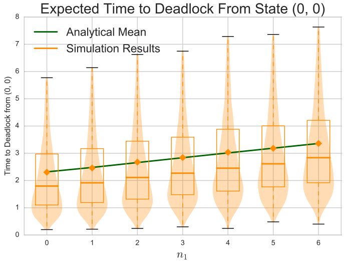
	$j_1 < n_2 + c_2$	$j_1 = n_2 + c_2$	$j_1 > n_2 + c_2$
$i_1 < n_1 + c_1$	Λ_1 if $\delta = (1,0)$ Λ_2 if $\delta = (0,1)$ $r_{12}s_1\mu_1$ if $\delta = (-1,1)$ $r_{21}s_2\mu_2$ if $\delta = (1,-1)$ $(1-r_{12})s_1\mu_1$ if $\delta = (-1,0)$ $(1-r_{21})s_2\mu_2$ if $\delta = (0,-1)$	Λ_1 if $\delta = (1,0)$ $r_{12}s_1\mu_1$ if $\delta = (0,1)$ $r_{21}s_2\mu_2$ if $\delta = (1,-1)$ $(1-r_{12})s_1\mu_1$ if $\delta = (-1,0)$ $(1-r_{21})s_2\mu_2$ if $\delta = (0,-1)$	Λ_1 if $\delta = (1,0)$ $r_{12}s_1\mu_1$ if $\delta = (0,1)$ $r_{21}s_2\mu_2$ if $\delta = (0,-1)$ $(1-r_{12})s_1\mu_1$ if $\delta = (-1,0)$ $(1-r_{21})s_2\mu_2$ if $\delta = (-1,-1)$
$i_1 = n_1 + c_1$	Λ_2 if $\delta = (0,1)$ $r_{12}s_1\mu_1$ if $\delta = (-1,1)$ $r_{21}s_2\mu_2$ if $\delta = (1,0)$ $(1-r_{12})s_1\mu_1$ if $\delta = (-1,0)$ $(1-r_{21})s_2\mu_2$ if $\delta = (0,-1)$	$r_{12}s_1\mu_1$ if $\delta = (0,1)$ $r_{21}s_2\mu_2$ if $\delta = (1,0)$ $(1-r_{12})s_1\mu_1$ if $\delta = (-1,0)$ $(1-r_{21})s_2\mu_2$ if $\delta = (0,-1)$	$r_{12}s_1\mu_1$ if $\delta = (0,1)$ $r_{21}s_2\mu_2$ if $\delta = (1,0)$ $(1-r_{12})s_1\mu_1$ if $\delta = (-1,0)$ $(1-r_{21})s_2\mu_2$ if $\delta = (-1,-1)$
$i_1 > n_1 + c_1$	Λ_2 if $\delta = (0,1)$ $r_{12}s_1\mu_1$ if $\delta = (-1,0)$ $r_{21}s_2\mu_2$ if $\delta = (1,0)$ $(1-r_{12})s_1\mu_1$ if $\delta = (-1,-1)$ $(1-r_{21})s_2\mu_2$ if $\delta = (0,-1)$	$r_{12}s_1\mu_1$ if $\delta = (0,1)$ $r_{21}s_2\mu_2$ if $\delta = (1,0)$ $(1-r_{12})s_1\mu_1$ if $\delta = (-1,-1)$ $(1-r_{21})s_2\mu_2$ if $\delta = (0,-1)$	$r_{12}s_1\mu_1$ if $\delta = (0,1)$ $r_{21}s_2\mu_2$ if $\delta = (1,0)$ $(1-r_{12})s_1\mu_1$ if $\delta = (-\min(b_1+1, b_2+1), -\min(b_1, b_2+1))$ $(1-r_{21})s_2\mu_2$ if $\delta = (-\min(b_1+1, b_2), -\min(b_1+1, b_2+1))$

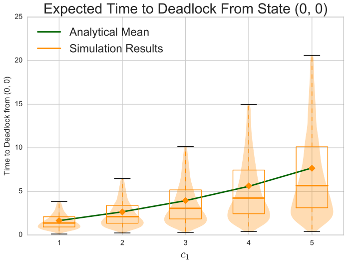

(0, 0) (0, 1) (0, 2) (0, 3) (0, 4)
(1, 0) (1, 1) (1, 2) (1, 3) (1, 4) (1, 5)
(2, 0) (2, 1) (2, 2) (2, 3) (2, 4) (2, 5) (2, 6)
(3, 0) (3, 1) (3, 2) (3, 3) (3, 4) (3, 5) (3, 6)
(4, 1) (4, 2) (4, 3) (4, 4) (4, 5) (4, 6)
(5, 2) (5, 3) (5, 4) (5, 5) (5, 6)

Times to Deadlock

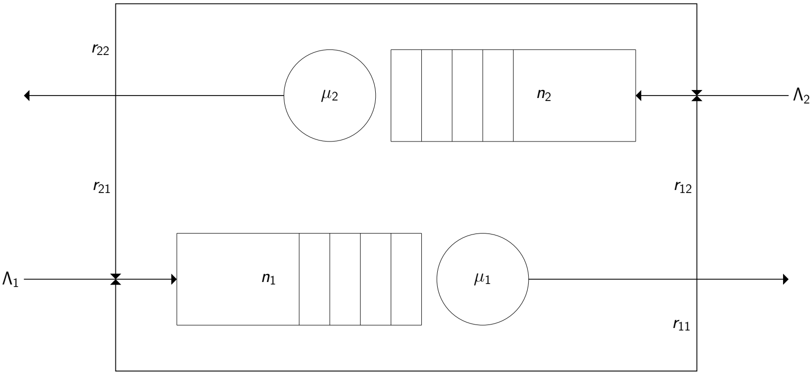








Markovian Model of Deadlock



$$(i, j)$$

$$S = \{(i,j) \in \mathbb{N}^{(n_1+2\times n_2+2)} \mid 0 \le i+j \le n_1+n_2+2\} \cup \{(-1)\}$$

Define $\delta = (i_2, j_2) - (i_1, j_1)$

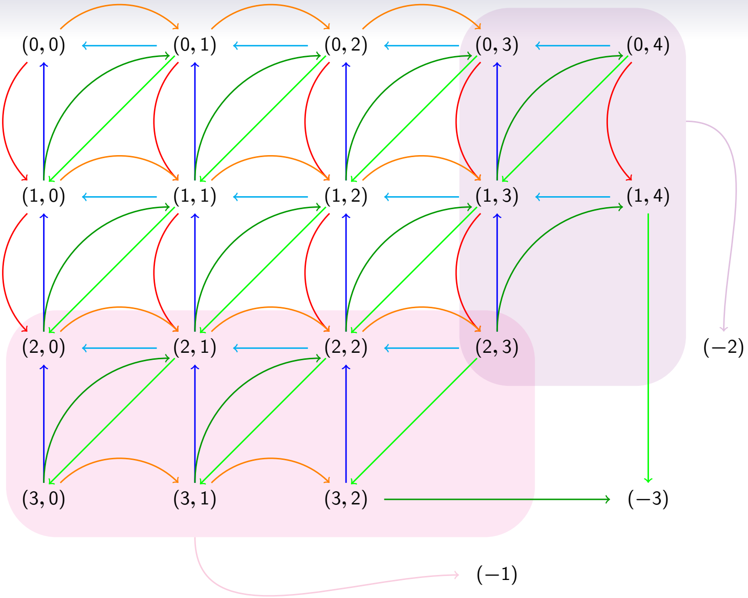
$$q_{(i_1,j_1),(i_2,j_2)} = \begin{cases} \left.\begin{array}{ll} \Lambda_1 & \text{if } i_1 \le n_1 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,0) \\ \left.\begin{array}{ll} \Lambda_2 & \text{if } j_1 \le n_2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,1) \\ \left.\begin{array}{ll} (1-r_{12})\mu_1 & \text{if } j_1 < n_2+2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,0) \\ \left.\begin{array}{ll} (1-r_{21})\mu_2 & \text{if } i_1 < n_1+2 \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (0,-1) \\ \left.\begin{array}{ll} r_{12}\mu_1 & \text{if } j_1 < n_2+2 \text{ and } (i_1,j_1) \ne (n_1+2, n_2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (-1,1) \\ \left.\begin{array}{ll} r_{21}\mu_2 & \text{if } i_1 < n_1+2 \text{ and } (i_1,j_1) \ne (n_1, n_2+2) \\ 0 & \text{otherwise} \end{array}\right\} & \text{if } \delta = (1,-1) \\ 0 & \text{otherwise} \end{cases}$$

$$q_{(i_1,j_1),(-1)} = \begin{cases} r_{11}\mu_1 & \text{if } i > n_1 \text{ and } j < n_2+2 \\ 0 & \text{otherwise} \end{cases}$$

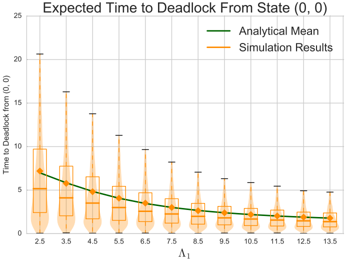
$$q_{(i_1,j_1),(-2)} = \begin{cases} r_{22}\mu_2 & \text{if } j > n_2 \text{ and } i < n_1+2 \\ 0 & \text{otherwise} \end{cases}$$

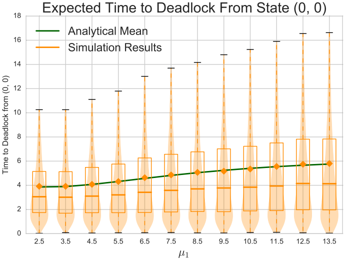
$$q_{(i_1,j_1),(-3)} = \begin{cases} r_{21}\mu_2 & \text{if } (i,j) = (n_1, n_2+2) \\ r_{12}\mu_1 & \text{if } (i,j) = (n_1+2, n_2) \\ 0 & \text{otherwise} \end{cases}$$

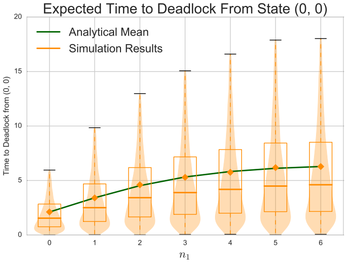
$$q_{-1,s} = q_{-2,s} = q_{-3,s} = 0$$


(0, 0)
(0, 1)
(0, 2)
(0, 3)
(0, 4)
(1, 0)
(1, 1)
(1, 2)
(1, 3)
(1, 4)
(2, 0)
(2, 1)
(2, 2)
(2, 3)
(−2)
(3, 0)
(3, 1)
(3, 2)
(−3)
(−1)

Times to Deadlock







Summary

Summary

- Developed a general use simulation library in Python
- Investigated potential for deadlock, automatic detection
- Modelled some deadlocking queueing networks

To Do...

- Build and parameterise patient flow networks from data
- Use queueing network analysis and simulation to investigate impact of the OPICP
- Determine the OPICP's effect on demand and workforce needs

Thank You

palmergi1@cardiff.ac.uk