

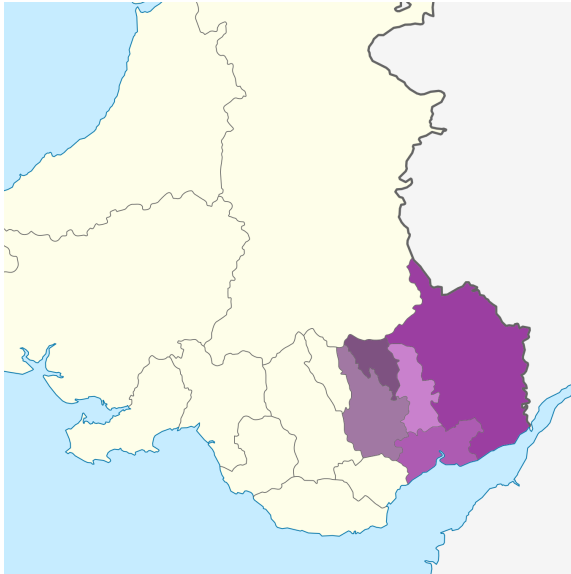
Queueing Networks for a Healthcare System

Deadlocking Properties

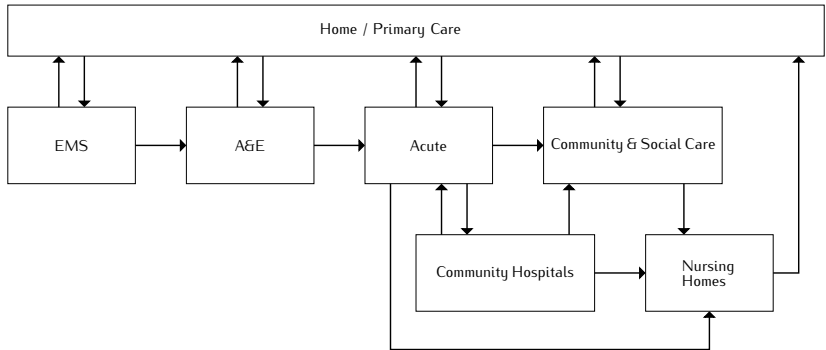
Geraint Palmer

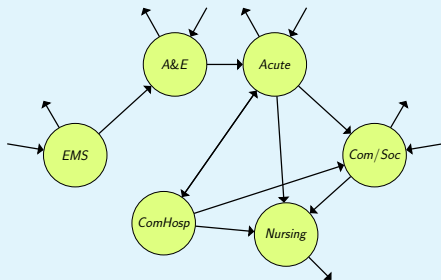
Wales Mathematics Colloquium 2015 - Gregynog

Aneurin Bevan University Health Board



Map of Healthcare System





Nodes:

EMS - [EDIT](#)

A&E - [EDIT](#)

Acute - [EDIT](#)

Community & Social Care - [EDIT](#)

Community Hospital - [EDIT](#)

Nursing Homes - [EDIT](#)

[Upload Parameters](#)

[Run Analysis](#)

[Run Simulation](#)

[View Results](#)

COMMUNITY & SOCIAL CARE

Winter Workforce Requirements:

Skill 1: ##### WTEs
Skill 2: ##### WTEs
Skill 3: ##### WTEs
Skill 4: ##### WTEs
Skill 5: ##### WTEs

Spring Workforce Requirements:

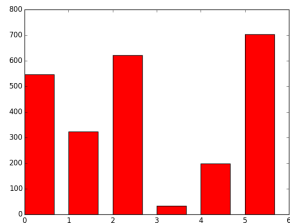
Skill 1: ##### WTEs
Skill 2: ##### WTEs
Skill 3: ##### WTEs
Skill 4: ##### WTEs
Skill 5: ##### WTEs

Summer Workforce Requirements:

Skill 1: ##### WTEs
Skill 2: ##### WTEs
Skill 3: ##### WTEs
Skill 4: ##### WTEs
Skill 5: ##### WTEs

Autumn Workforce Requirements:

Skill 1: ##### WTEs
Skill 2: ##### WTEs
Skill 3: ##### WTEs
Skill 4: ##### WTEs
Skill 5: ##### WTEs



Performance Measures:

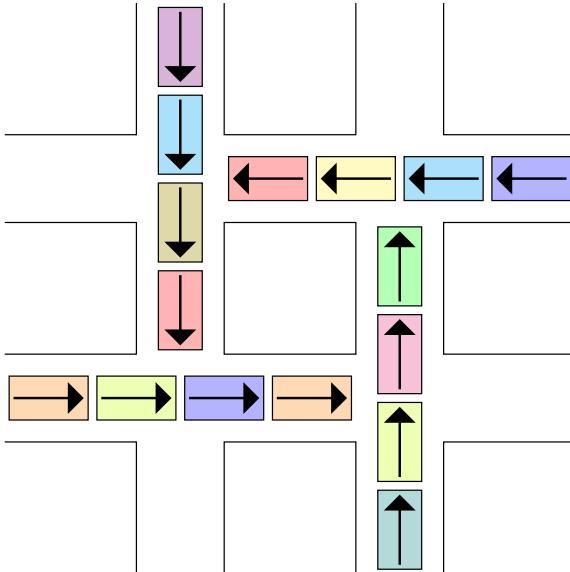
Winter Expected Occupancy: #####

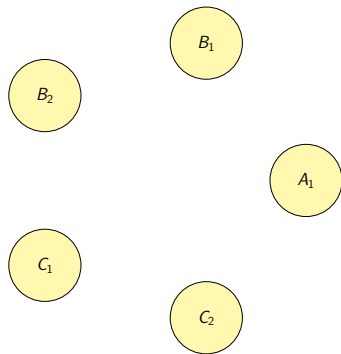
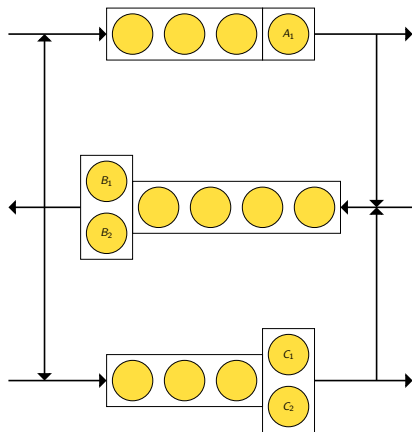
Spring Expected Occupancy: #####

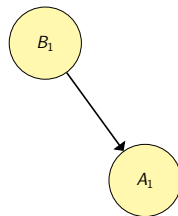
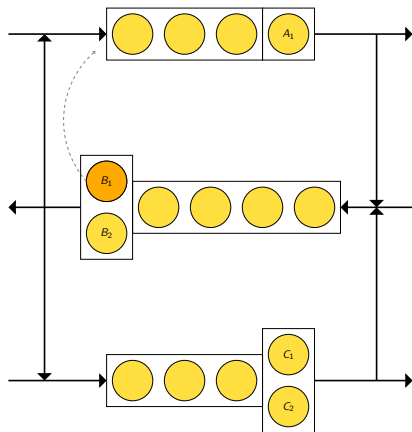
Summer Expected Occupancy: #####

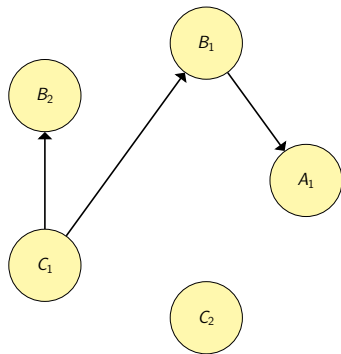
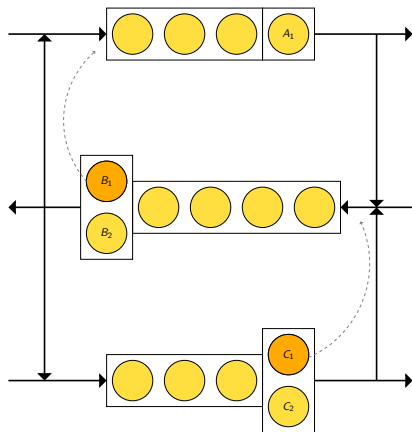
Autumn Expected Occupancy: #####

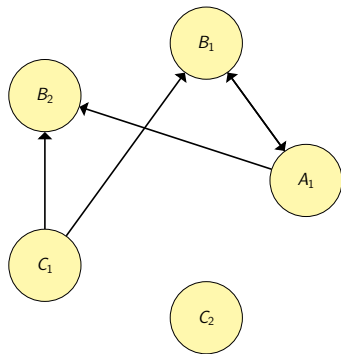
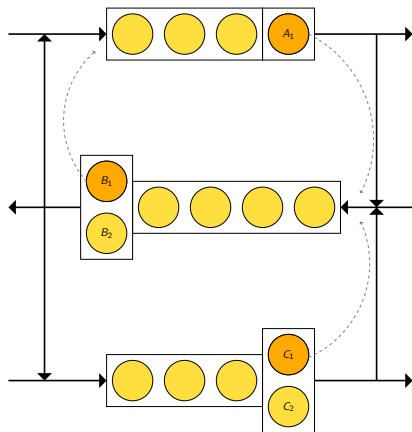
Deadlock

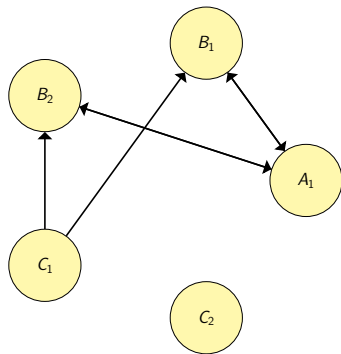
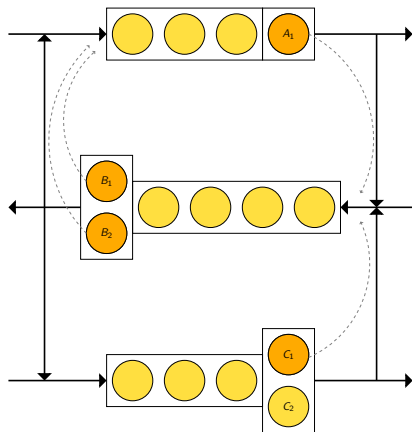


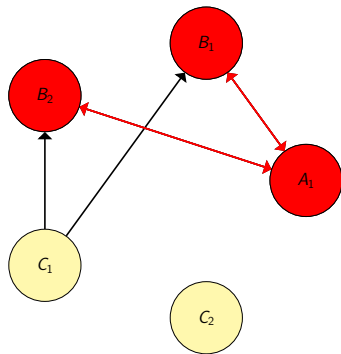
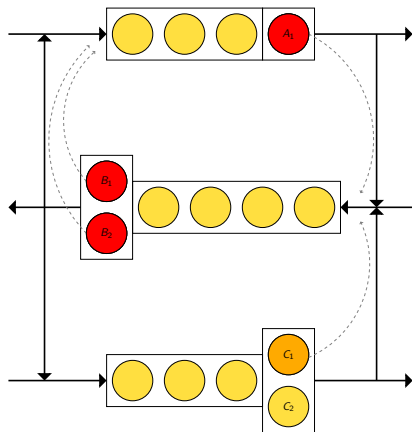




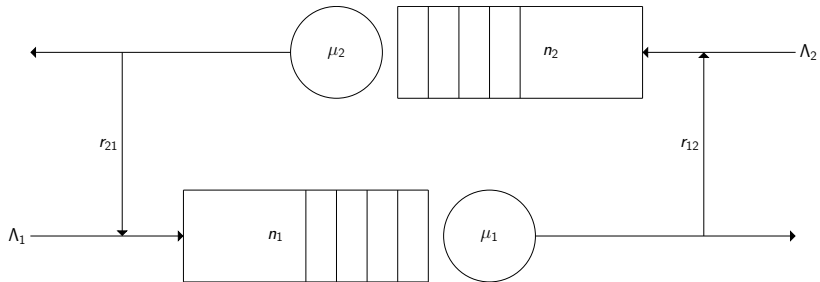








Markovian Model of Deadlock



(i, j)

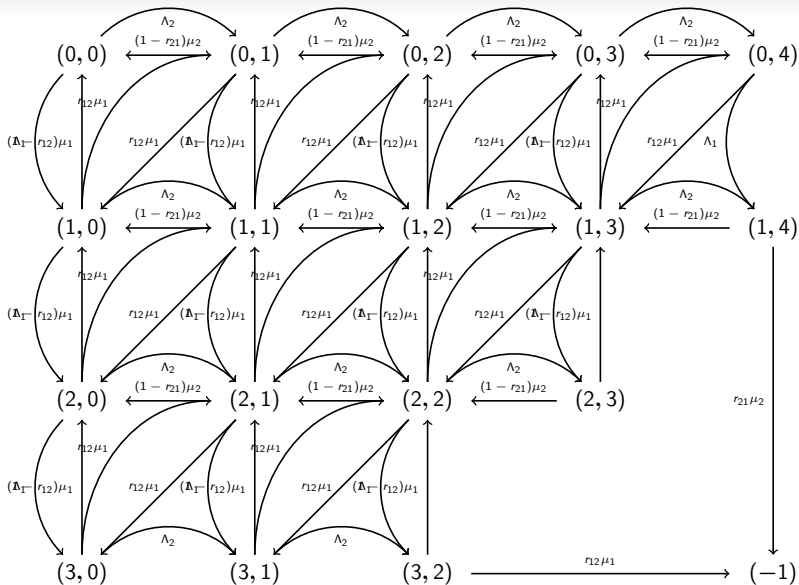
$$S = \{(i, j) \in \mathbb{N}^{(n_1+2 \times n_2+2)} | 0 \leq i + j \leq n_1 + n_2 + 2\} \cup \{(-1)\}$$

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

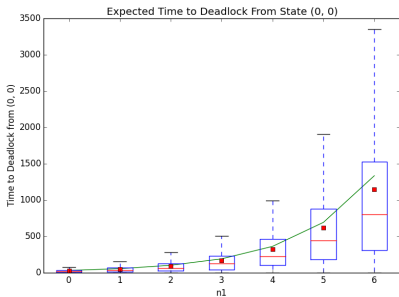
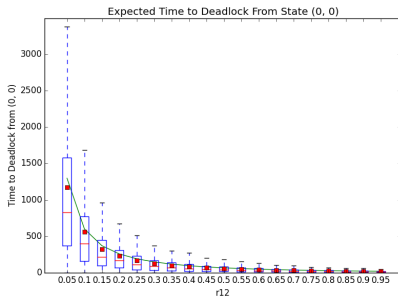
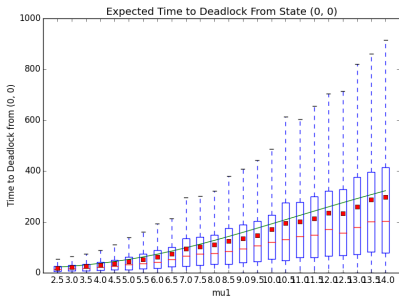
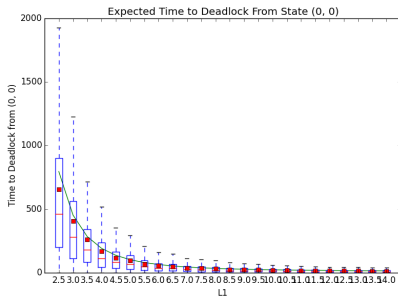
$$q_{(i_1, j_1), (i_2, j_2)} = \begin{cases} \left. \begin{array}{ll} \Lambda_1 & \text{if } i_1 \leq n_1 \\ 0 & \text{otherwise} \end{array} \right\} & \text{if } \delta = (1, 0) \\ \left. \begin{array}{ll} \Lambda_2 & \text{if } j_1 \leq n_2 \\ 0 & \text{otherwise} \end{array} \right\} & \text{if }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

$$q_{(i_1, j_1), (-1)} = \begin{cases} r_{21}\mu_2 & \text{if } (i, j) = (n_1, n_2 + 2) \\ r_{12}\mu_1 & \text{if } (i, j) = (n_1 + 2, n_2) \\ 0 & \text{otherwise} \end{cases}$$

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

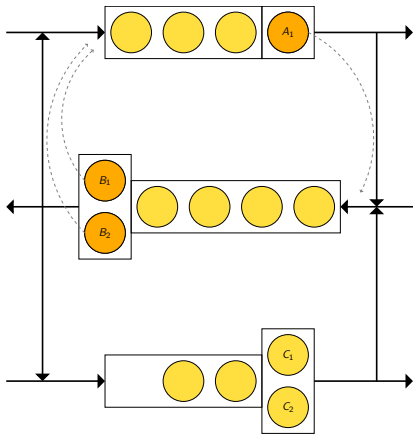


Times to Deadlock

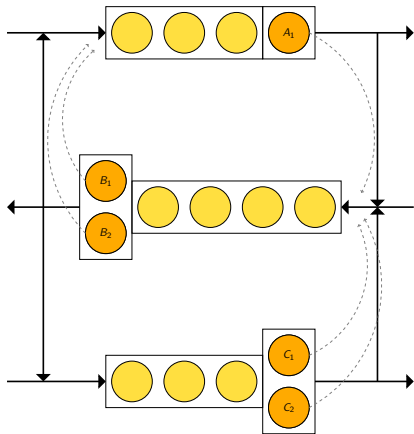


Types of Deadlock

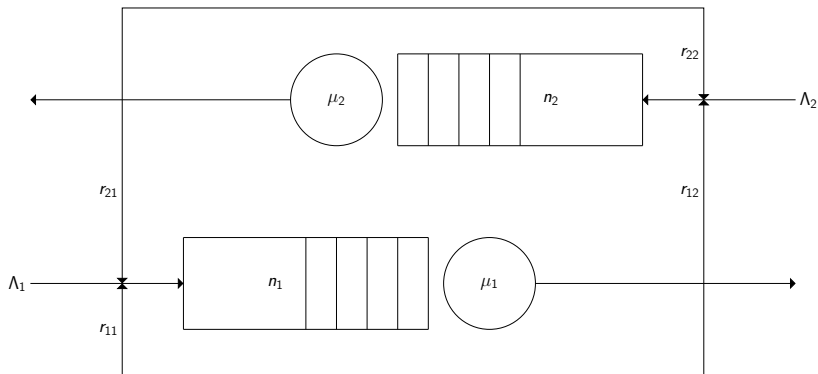
Transient Deadlock



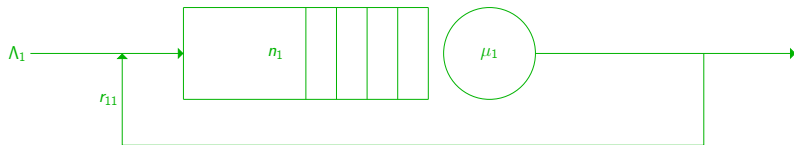
Absorbing Deadlock



Future Work...

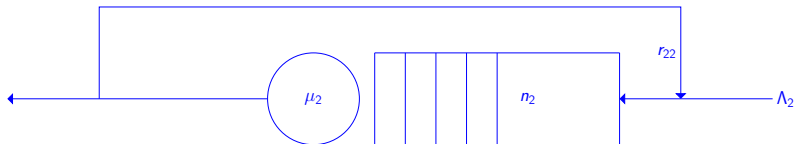


Future Work...



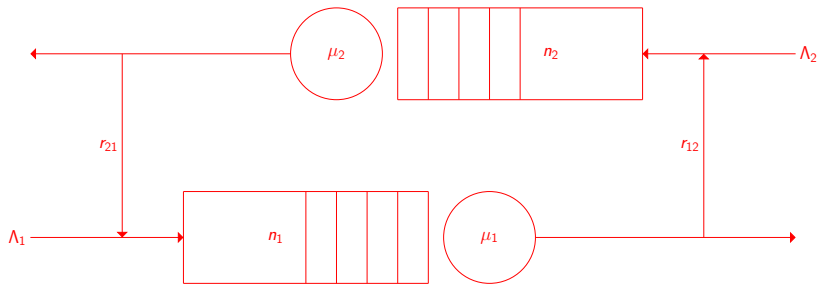
Q_1

Future Work...



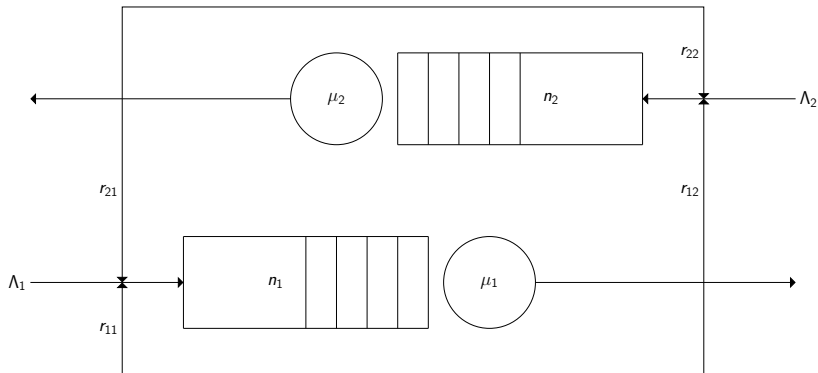
Q_1 Q_2

Future Work...



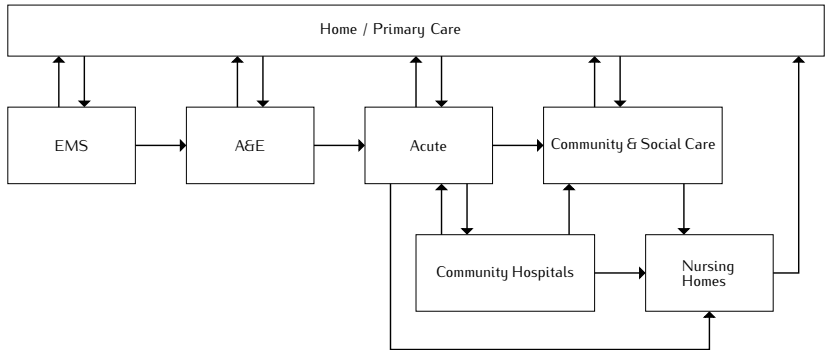
Q_1 Q_2 Q_3

Future Work...

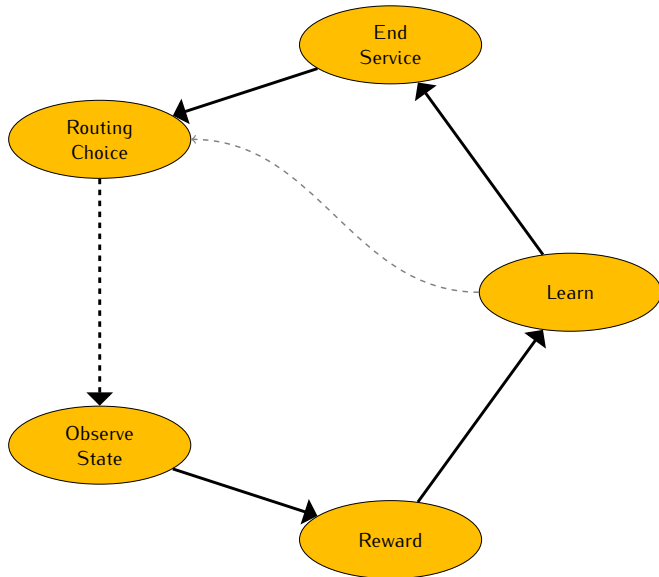


$$Q = f(\textcolor{green}{Q}_1, \textcolor{blue}{Q}_2, \textcolor{red}{Q}_3)$$

Reinforcement Learning



Reinforcement Learning



Reinforcement Learning

Q-Learning

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$