

Queueing Networks for a Healthcare System

Deadlocking, Reinforcement Learning & Workforce Planning

Geraint Palmer

Paul Harper, Vincent Knight

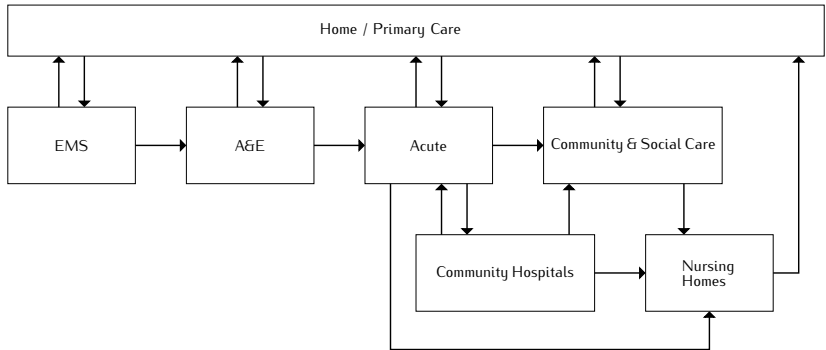
EURO 2015 - Glasgow

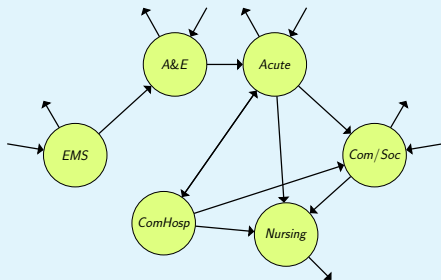


Aneurin Bevan University Health Board



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:

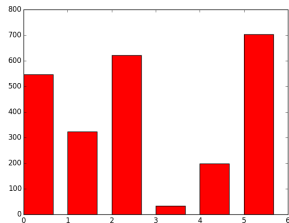
Skill 1: ##### WTEs
Skill 2: ##### WTEs
Skill 3: ##### WTEs
Skill 4: ##### WTEs
Skill 5: ##### WTEs

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:

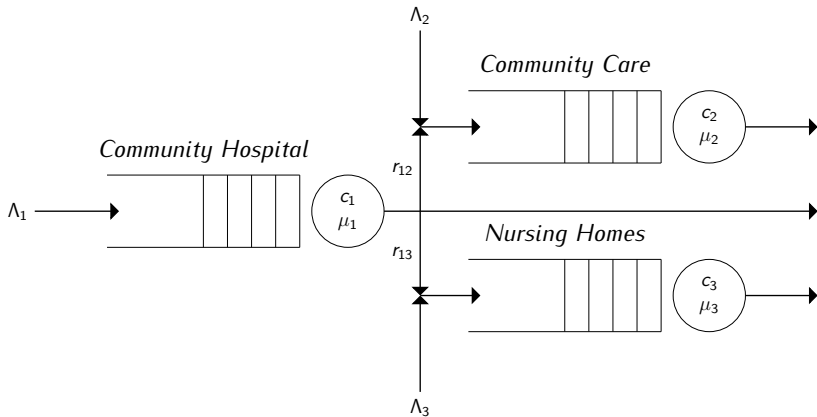
Winter Expected Occupancy: #####

Spring Expected Occupancy: #####

Summer Expected Occupancy: #####

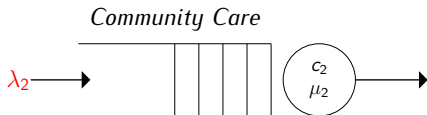
Autumn Expected Occupancy: #####

Jackson Networks



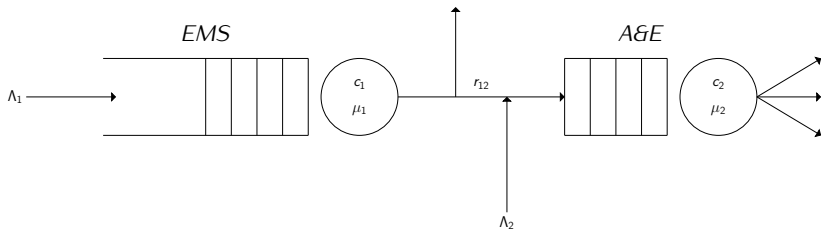
Jackson Networks

$$\lambda_i = \Lambda_i + \sum_j r_{ji} \lambda_j$$



$$P(k_1, k_2, \dots, k_M) = \prod_{i=1}^M P_i(k_i)$$

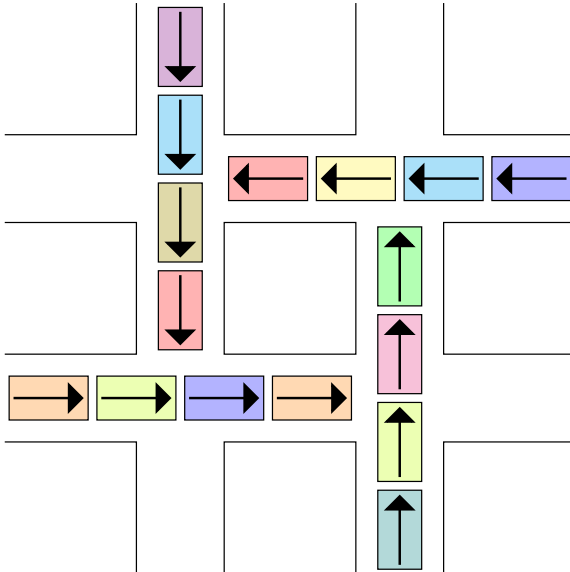
Restricted Networks

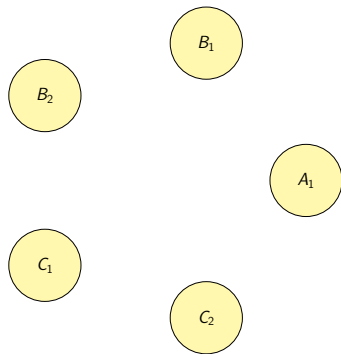
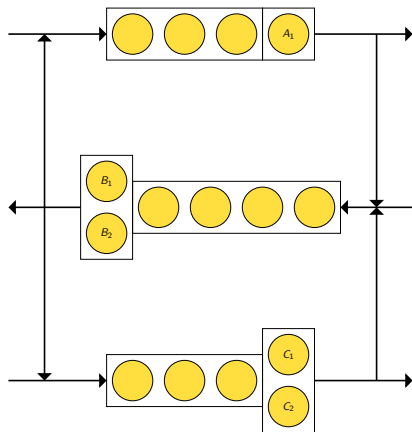


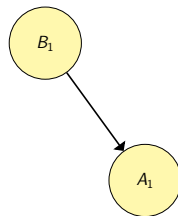
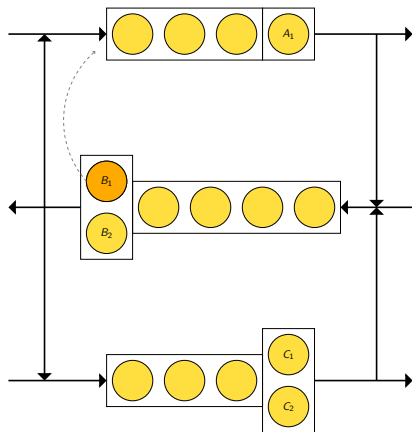
Restricted Networks

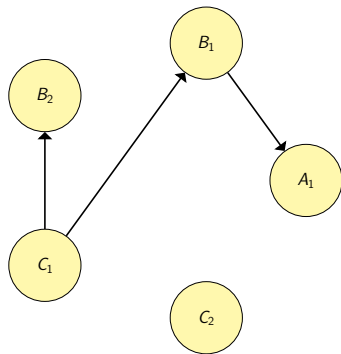
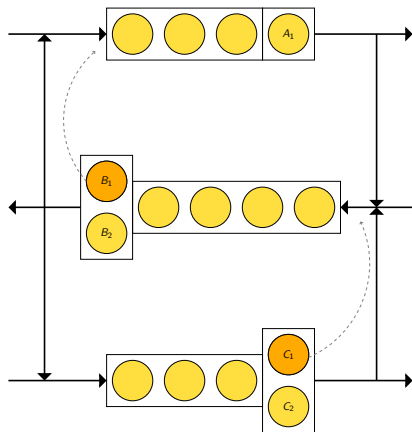
- Markov Chain Models
- Approximation Methods
- Simulation

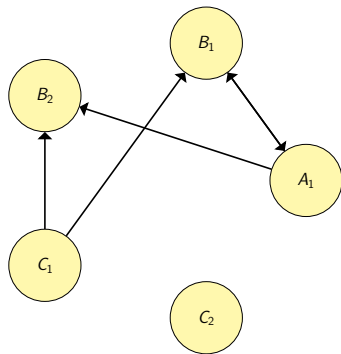
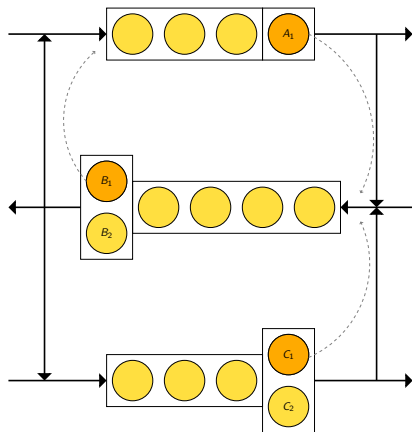
Deadlock

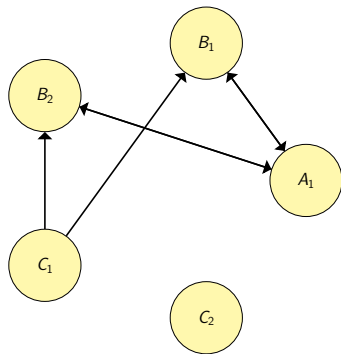
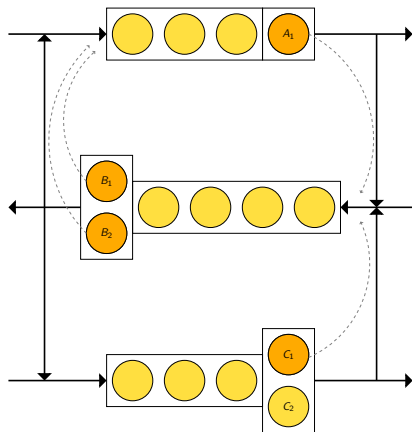


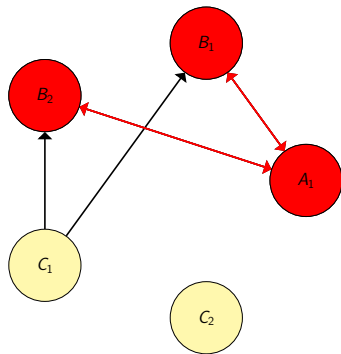
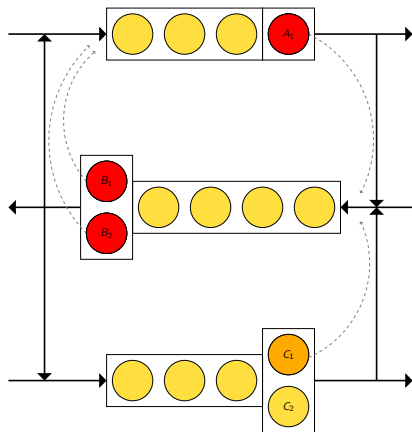




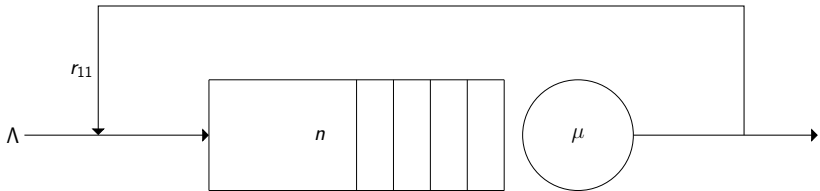








Markovian Model of Deadlock



(i)

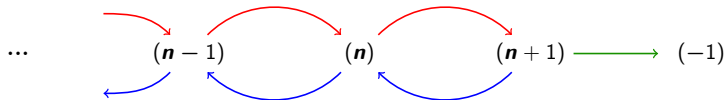
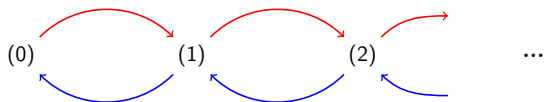
$$S = \{i \in \mathbb{N} | 0 \leq i \leq n+1\} \cup \{-1\}$$

$$\text{Define } \delta = i_2 - i_1$$

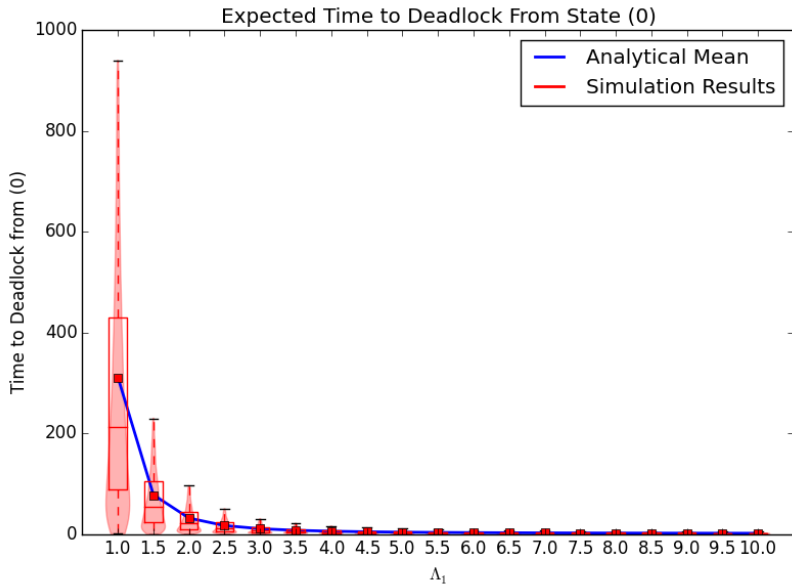
$$q_{i_1, i_2} = \begin{cases} \left\{ \begin{array}{ll} \Lambda & \text{if } i < n+1 \\ 0 & \text{otherwise} \end{array} \right\} & \text{if } \delta = 1 \\ \left\{ \begin{array}{ll} (1 - r_{11})\mu & \text{if } \delta = -1 \\ 0 & \text{otherwise} \end{array} \right\} & \end{cases}$$

$$q_{i, -1} = \begin{cases} r_{11}\mu & \text{if } i = n+1 \\ 0 & \text{otherwise} \end{cases}$$

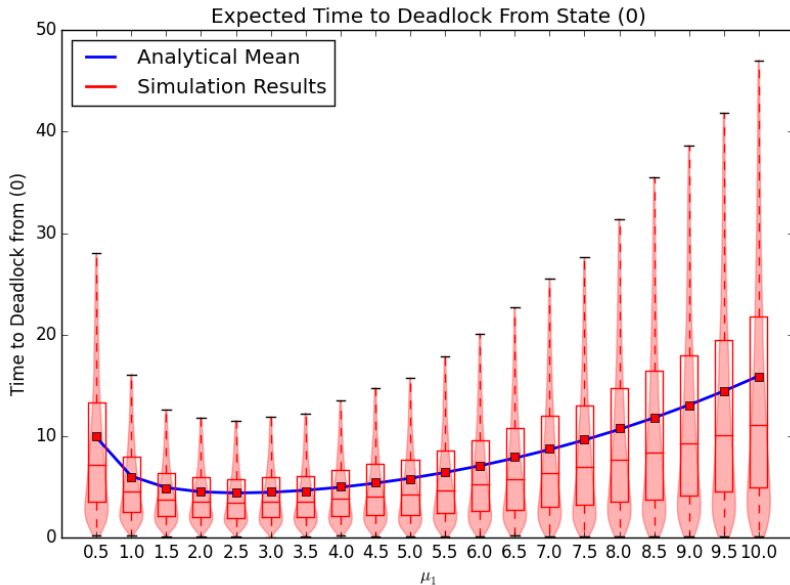
$$q_{-1, s} = 0$$



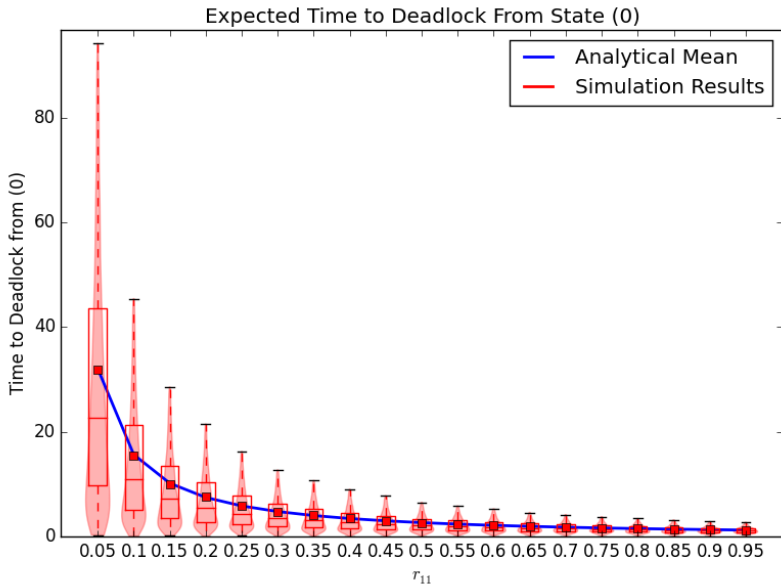
Times to Deadlock - Λ



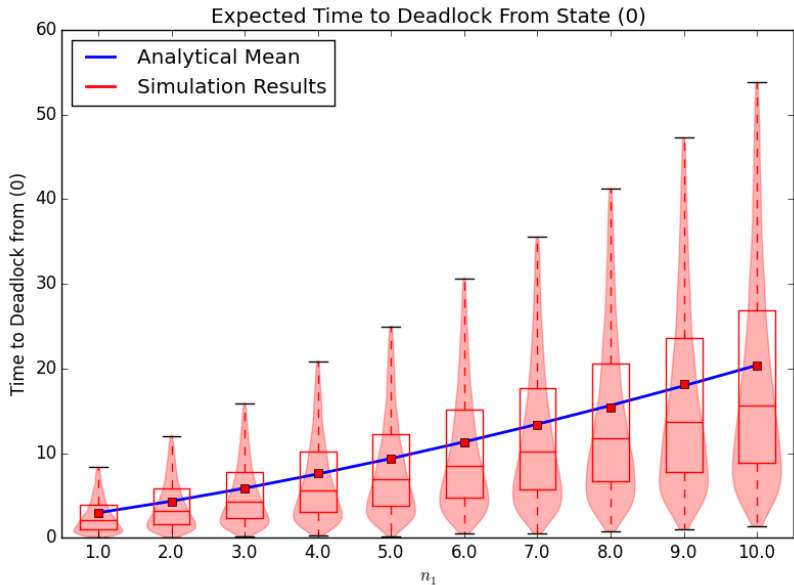
Times to Deadlock - μ



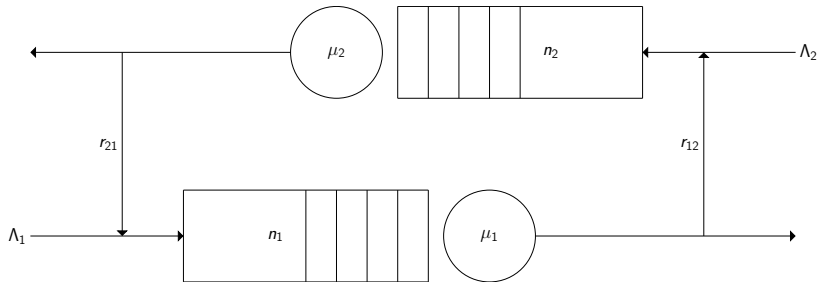
Times to Deadlock - r_{11}



Times to Deadlock - n



Markovian Model of Deadlock



(i, j)

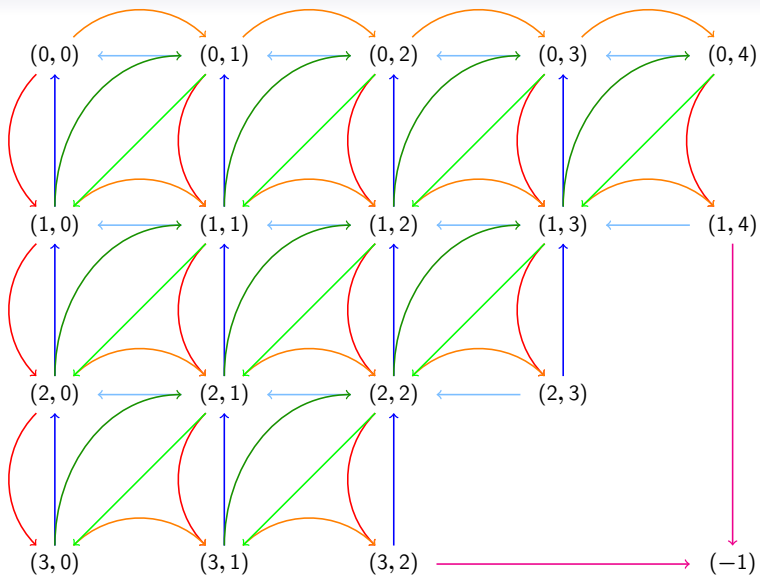
$$S = \{(i, j) \in \mathbb{N}^{(n_1+2 \times n_2+2)} | 0 \leq i + j \leq n_1 + n_2 + 2\} \cup \{(-1)\}$$

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

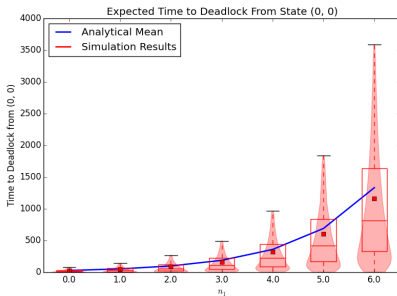
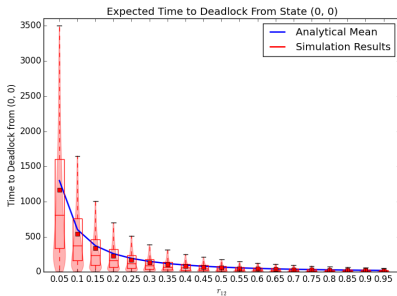
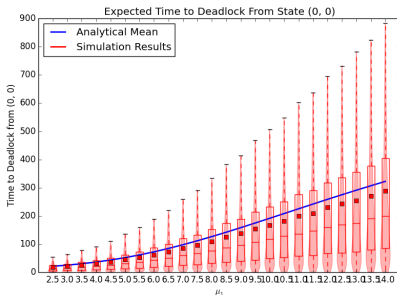
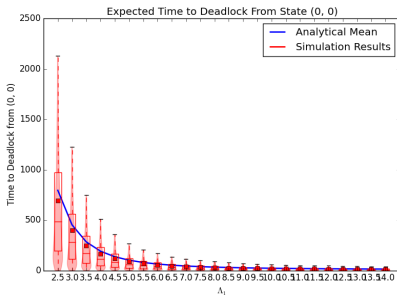
$$q_{(i_1, j_1), (i_2, j_2)} = \begin{cases} \left. \begin{array}{ll} \Lambda_1 & \text{if } i_1 \leq n_1 \\ 0 & \text{otherwise} \end{array} \right\} & \text{if } \delta = (1, 0) \\ \left. \begin{array}{ll} \Lambda_2 & \text{if } j_1 \leq n_2 \\ 0 & \text{otherwise} \end{array} \right\} & \text{if }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

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

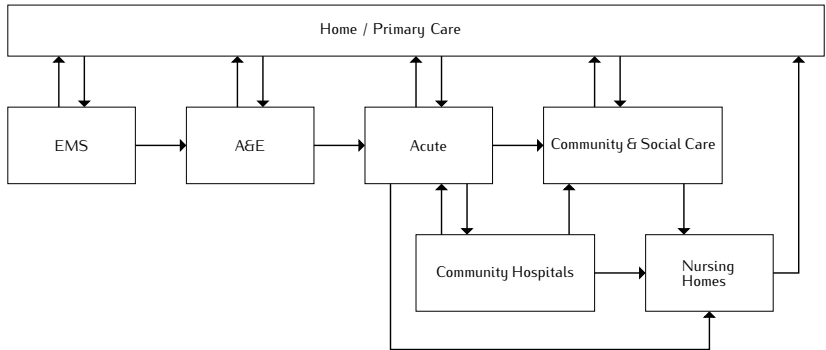
$$q_{-1, s} = 0$$



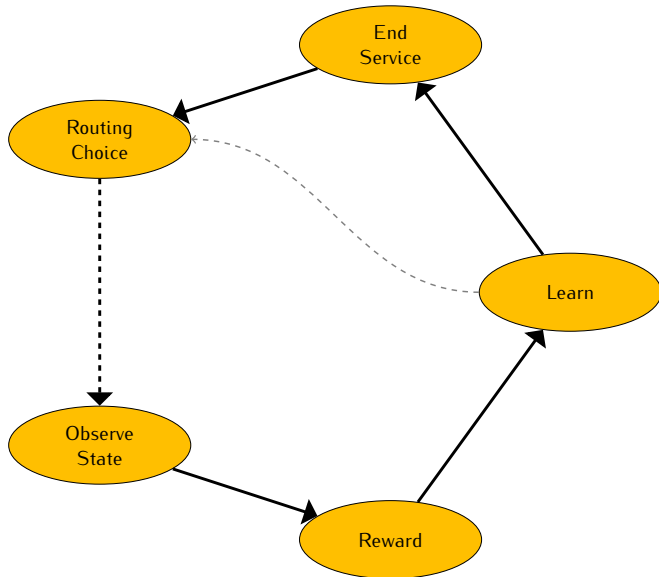
Times to Deadlock



Reinforcement Learning



Reinforcement Learning



Reinforcement Learning

Q-Learning

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

Diolch - Thank You

<https://github.com/geraintpalmer/Presentations>
palmergi1@cardiff.ac.uk