Optimal Hiring of Cloud Servers

A. Stephen McGough, Isi Mitrani

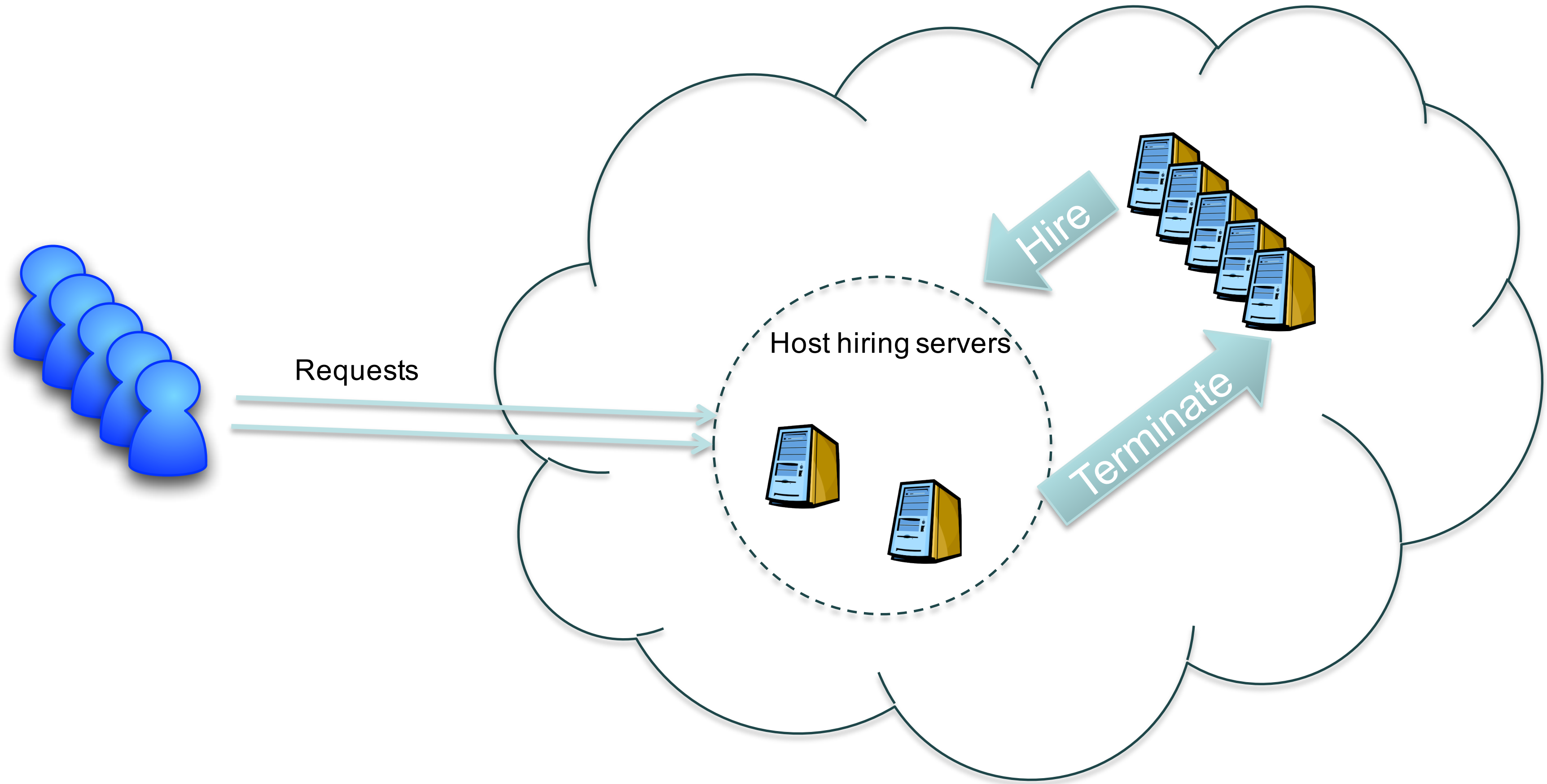
EPEW 2014, Florence

Durham University

Newcastle University

Scenario

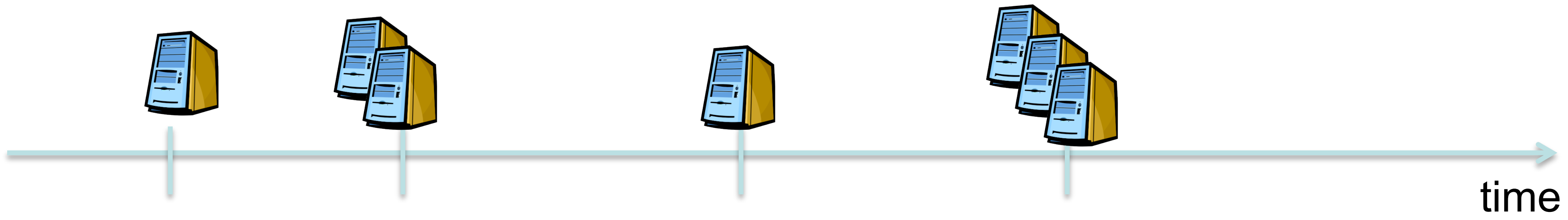
How many cloud instances should be hired?



The number of active servers is controlled by the host.

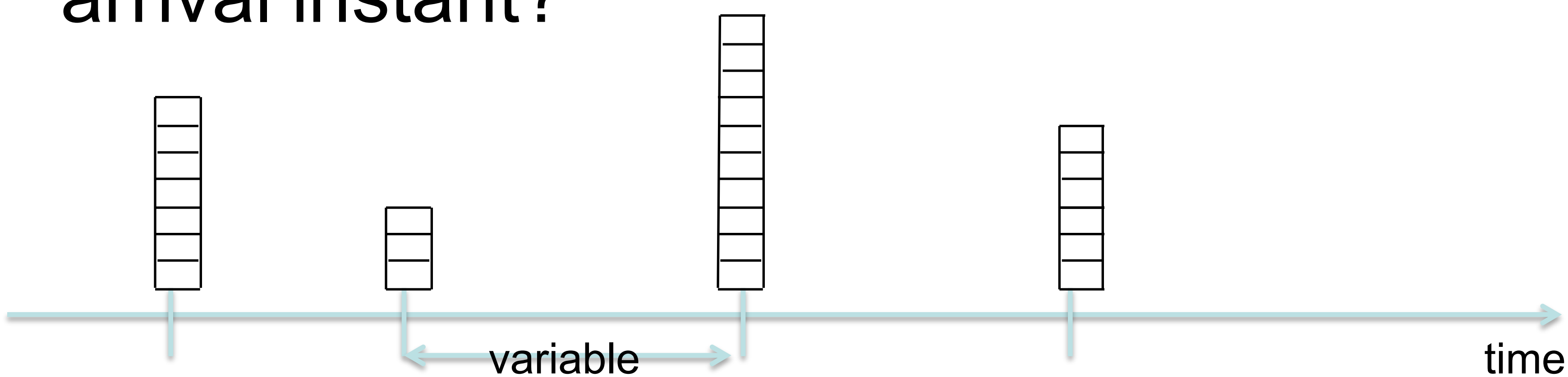
Dynamic optimization problems:

In a system whose state is a random process, decide at various moments in time how many servers to employ in order to minimize long-term performance and operating costs.



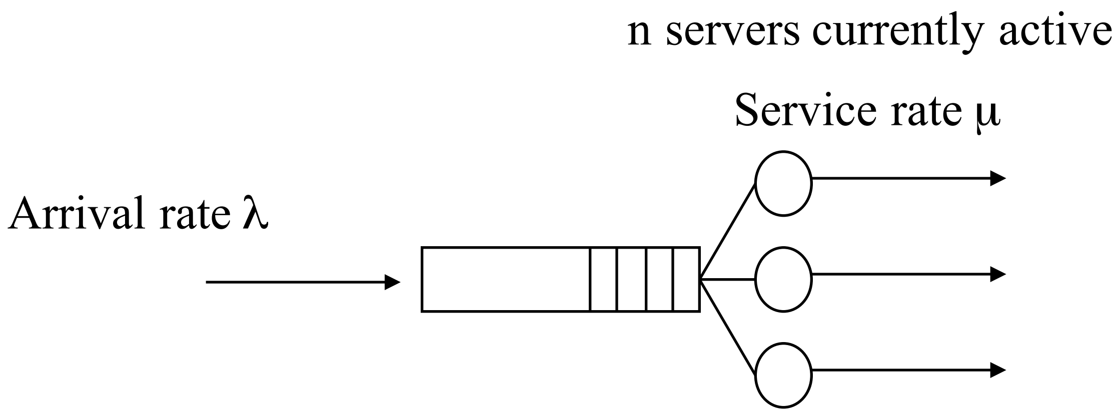
Case 1: Batch Arrivals

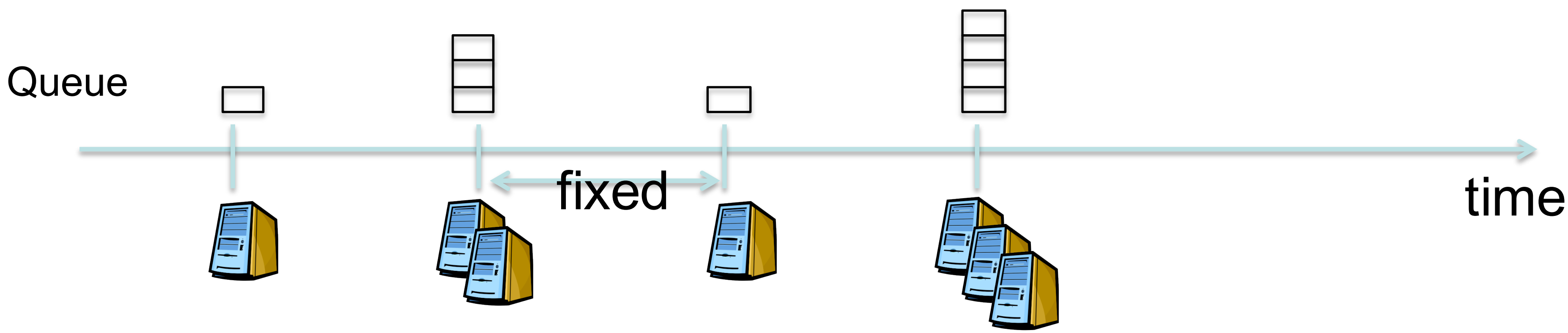
- Decision instants are when jobs arrive
  - In batches
  - Arrival rate λ
  - Service rate μ
  - Batch size distribution b_i
- How many servers should be hired at each arrival instant?



Case 2: Dynamically Controlled M/M/n/J Queue

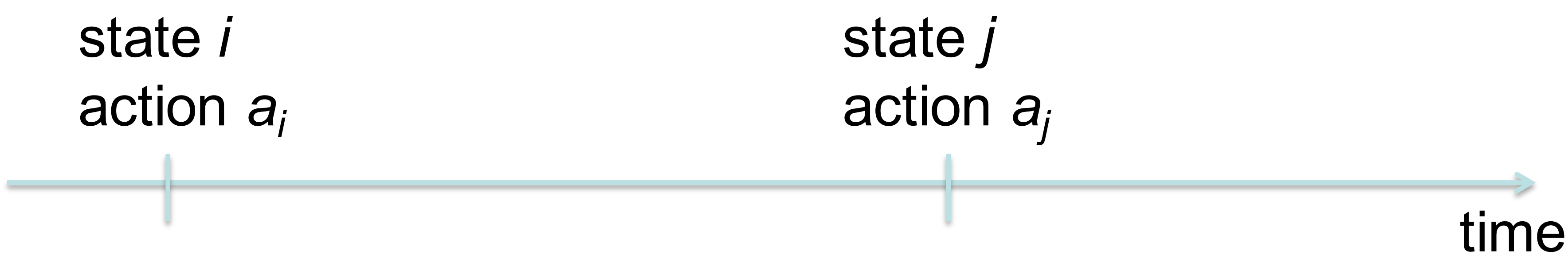
J – maximum jobs





How many servers should be hired at each hiring instant?

General framework
(Semi-Markov Decision Process)



Identify best action a_i to take for state *i*

Characteristics:

- Average interval to next decision instant: $\tau_i(a)$
- One step cost, i.e. average cost incurred until next decision instant c_i(a)
- Transition probability to next state $p_{i,j}(a)$
- Policy set A = a_i, i=1, …, … (an action for each state)

Policy Set

- Stationary policy A
  - Actions depend only on state not on prior history
- Average cost incurred during interval (0,t):
  - $Z_A(t)$
- Long-term average cost per unit time:

$$g(A) = \lim_{t \to \infty} \frac{1}{t} E[Z_A(t)]$$

  - g(A) does not depend on initial state

Determining Cost

- For a given policy set A
  - The average cost g(A) can be computed by introducing auxiliary variables v_j
    - One for each state
  - And solving the set of simultaneous liner equations:

$$v_j = c_j(A) - \tau_j(A)g(A) + \sum_{k=1}^{J} p_{j,k}(A)v_k \ ; \ j = 1, 2, \ldots, J$$

  - Make unique solution by setting $v_k = 0$ for some state k

Determining A*

- Find an optimal policy using a 'policy improvement' algorithm:
  1. Choose an initial stationary policy A
  2. Compute v_i and g(A) by solving the set of simultaneous liner equations
  3. For each i find the action a* which minimizes the right hand side of:

$$v_j = c_j(A) - \tau_j(A)g(A) + \sum_{k=1}^{J} p_{j,k}(A)v_k \;\; ; \;\; j = 1, 2, \ldots, J$$

  1. If A* = A we're finished
     - Else let A = A* and repeat from 2

The algorithm is guaranteed to terminate in a finite number of iterations

Heuristics and Policies

- Greedy Heuristic:
  - For every state j choose the action which minimizes the cost in the current interval
    - The one-step-cost
  - $c_j(n)$
- Fixed policy – fixed number of servers
  - To cope with most extreme events aim for average server occupancy of 70%

Case 1: $n^* = \left\lceil \frac{\lambda b}{0.7\mu} \right\rceil$ Case 2: $n^* = \left\lceil \frac{\lambda}{0.7\mu} \right\rceil$

Results

Case 1:Batch Arrivals

- Decision instants: batch arrivals
- System state: number of jobs present – j
- Action taken: n servers hired
- Average length of decision interval: $1/\lambda$
- Transition probabilities: $p_{j,k}(t)$
  - Closed form expressions
- One-step cost of decision n:

$$c_j(n) = c_1 T_j(n) + c_2 n \frac{1}{\lambda}$$

  - Recurrence relation for T_j – holding time

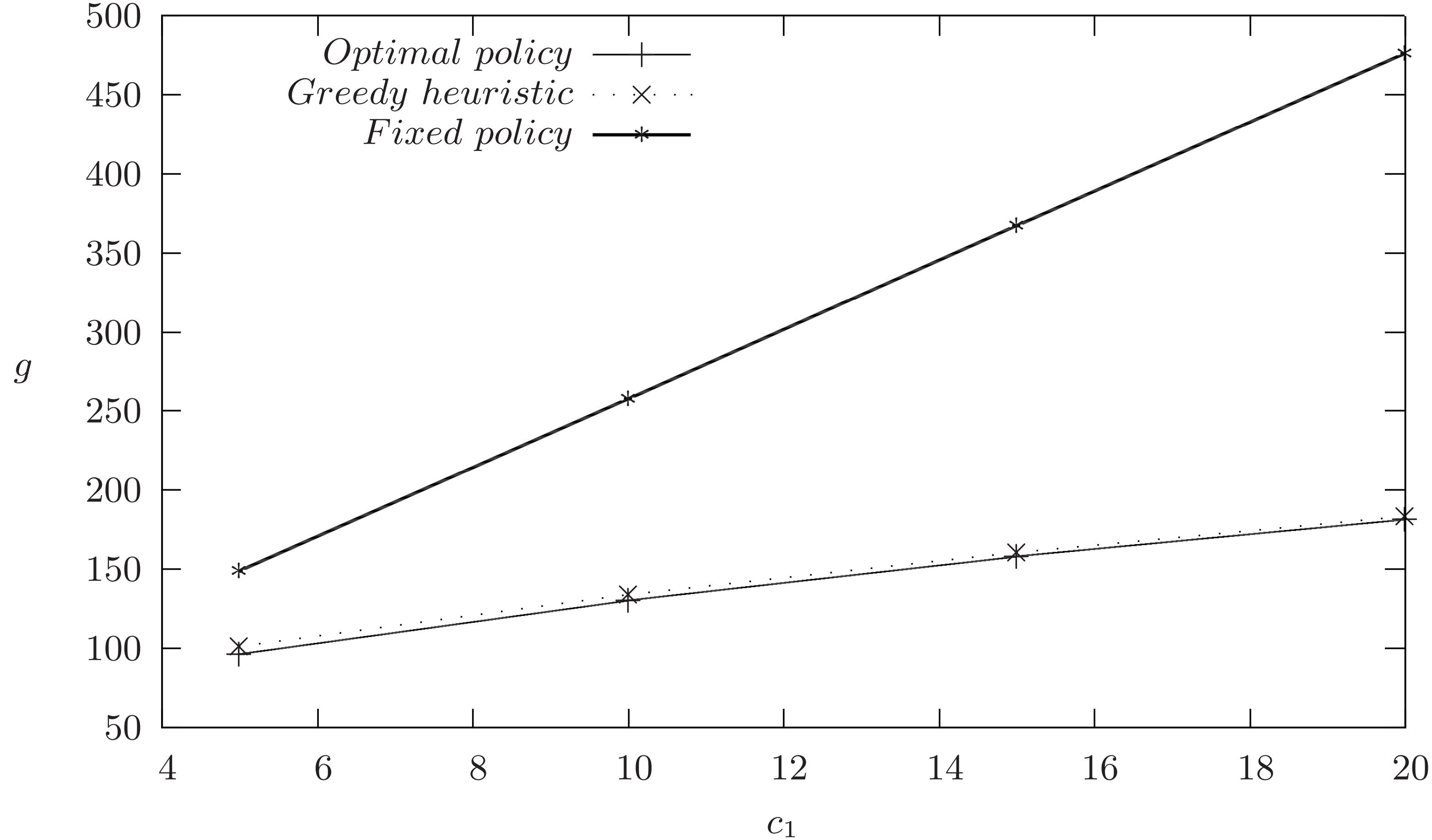
Case 2:Dynamically Controlled M/M/n/J Queue

- Decision instants: discrete
- System state: number of jobs present – j
- Action taken: n servers hired
- Average length of decision interval: τ
- Transition probabilities: $p_{j,k}(t)$
  - Numerical solution for transient transition probabilities
- One-step cost of decision n:

$$c_j(n) = \left[c_1 \frac{j + L_j}{2} + c_2 n \right] \tau$$

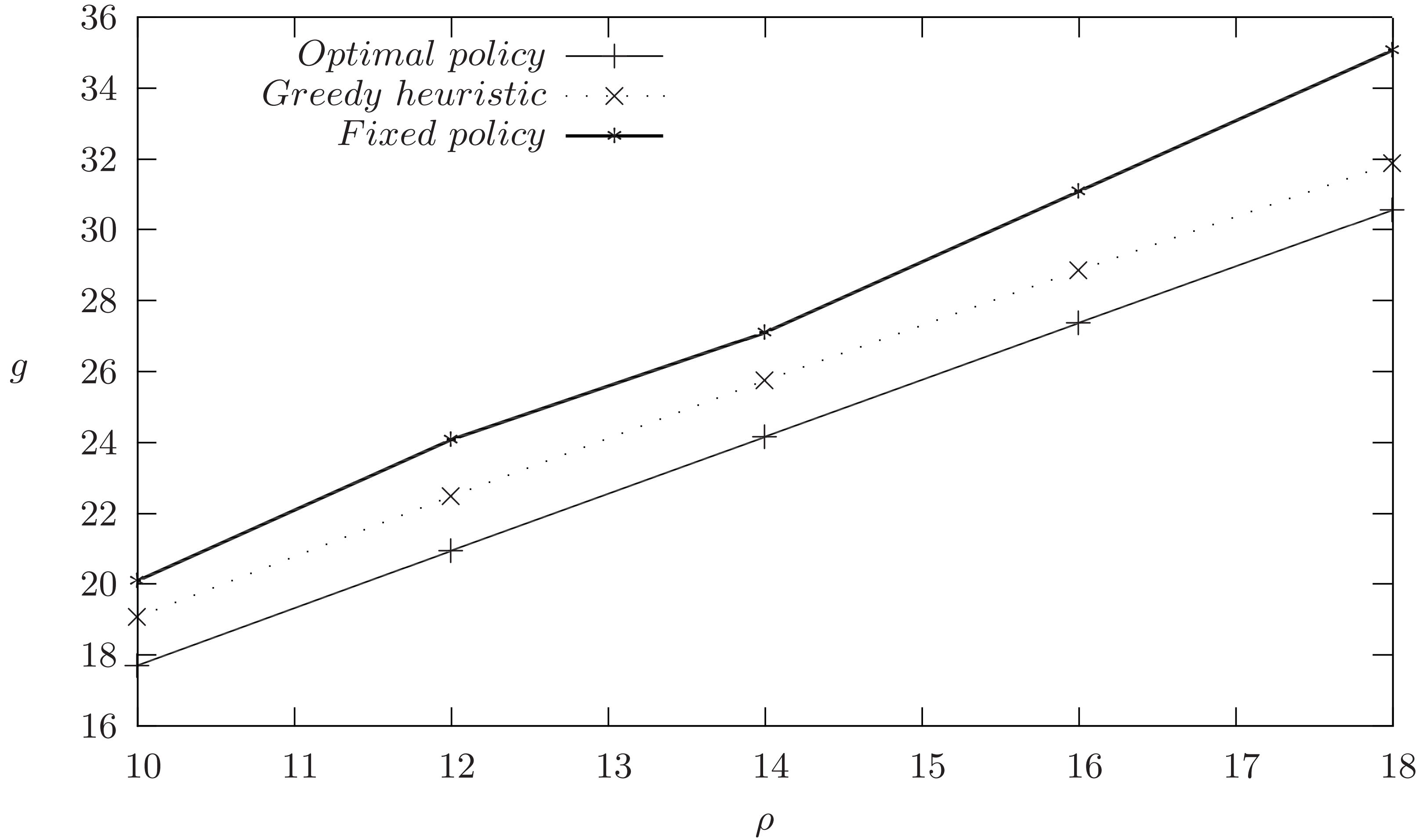
  - L_j – average number of jobs in the system during interval τ

Case 1:Batch Arrivals



Batch arrivals: varying unit holding cost

Case 2:Dynamically Controlled M/M/n/J Queue



Fixed hiring periods: varying offered load

Questions

stephen.mcgough@durham.ac.uk

Isi.mitrani@newcastle.ac.uk

Durham University

Newcastle University