Index based routing policies

monotonicity and perfect simulation

Jean-Marc Vincent

Laboratory ID-IMAG
MESCAL Project
Universities of Grenoble
Jean-Marc.Vincent@imag.fr
joint work with Bruno Gaujal, Vandy Berten (ULB) and Jérôme Vienne

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

INP Grenoble

INRIA INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

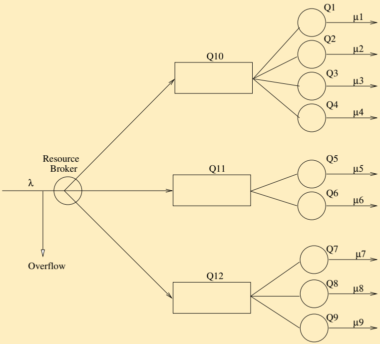
GRENOBLE 1 UNIVERSITÉ JOSEPH FOURIER SCIENCES,TECHNOLOGIE,MÉDECINE

Outline

1. Scheduling queues
2. Events in queueing networks
3. Application
4. Conclusion and future works

Resource broker

Grid model



Input rates
Allocation strategy
State dependent allocation
Index based routing :
destination minimize a
criteria

Problem

Optimization of throughput, response time,...
Comparison of policies, analysis of heuristics
...

Routing Customers in Parallel Queues

The problem:

- Find a routing policy maximizing the expected (discounted) throughput of the system.
- Several variations on this problem depend on the information available to the controller: current size of all queues (and size of the arriving batch).

The applications:

- improve batch schedulers for cluster and grid infrastructures.
- Assert the value of information in such cases.

Markov Decision Process

State space $\mathcal{X} = \{0, \cdots, C_1\} \times \cdots \times, \{0, , \cdots, C_K\}$,
Action $U = \{1, \cdots, K\}$,
Random space set of events $E = \{arrival, service_1, \ldots, service_K\}$.
Routing policy : $\pi = (u_1, u_2, u_3, \cdots)$,

$$u_k : \mathcal{X} \to U$$

action at step k.

Given $\pi \Rightarrow$ Continuous Time Markov Chain.
Uniformization

$$x_{k+1} = f(x_k, u_k, w_k),$$

Markov Decision Process (II)

x_0 fixed, find a policy $\pi = (u_1, u_2, \cdots)$ that minimizes the **discounted cost over an infinite horizon**

$$J_\pi(x_0) = \lim_{N\to\infty} \mathbb{E}_{w_1,\ldots w_N}\left(\sum_{k=0}^{N} \alpha^k g(x_k, u_k, w_k)\right).$$

- $g(x_k, u_k, w_k)$: immediate cost
  $= b.x_k$: b is the holding cost per customer per unit of time.
- α : discount factor ($0 < \alpha < 1$).

Bellman's Equation

Theorem [Bellman 1957]

Assume that the immediate costs are bounded :

$$|g(x,u,w)| < M \quad \forall (x,u,w) \in S \times U \times E.$$

Then, the optimal cost J^* satisfies Bellman's fixed point equation

$$J^*(x) = \min_{u \in U(x)} \mathbb{E}_w \left(g(x,u,w) + \alpha J^*(f(x,u,w)).\right)$$

The optimal policy π^* (the argmin in Bellman's equation) is stationary: $\pi^* = (u^*, u^*, \ldots)$.

Computational Issues

The Bellman operator is monotone and contracting.
$\Rightarrow$ fixed point can be computed in finite :

- **value iteration** : iteration of the Bellman operator starting with an arbitrary policy,
- **policy iteration** (iteration over the argmin), also called Howard's algorithm [Howard 1960]
- or linear programming.

BUT!
The state space is exponential in the size of the system.

All these algorithms have an exponential complexity in the worse case.

This is called the "curse of dimensionality".

Optimal Index policies

Index function : $\mathcal{X}_i \rightarrow \mathbb{R}$
Index policy :

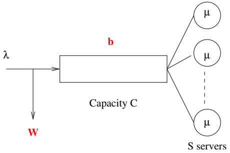
$$u(x_1, \cdots x_K) = F(argmin(I_1(x_1), \cdots, I_K(x_K)))$$

In some particular cases, Index policies are dominant.

- **JSQ** is optimal for symmetric queues (μ_i are all equal, C_i are all equal) [Winston 1977].
- An index policy is optimal for the Multi-armed Bandit Problem and for the Restless Bandit Problem. [Gittins 1979, Whittle 1988]

Index policies for routing

Optimal routing policy problem is still open for n different $M/M/1$
Heuristic : index policy inspired from the Multi-Armed Bandit
$\Rightarrow$ free parameter and compute an equilibrium point.
[Mitrani 2005] for routing and repair problems.



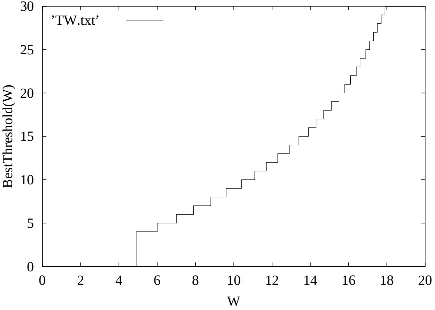
W is the rejection cost (free parameter).

Theorem

There exists an optimal policy of threshold type:
there exists θ such that :Reject if $x \geqslant \theta$ and accept otherwise.
- θ does not depend on C as long as $C > \theta$ (including if C is infinite).
- θ is a non-decreasing function of W.

Index policies for Routing(II)

Computation of $\theta(W)$ linear system of corresponding to Bellman's equation, after uniformization.

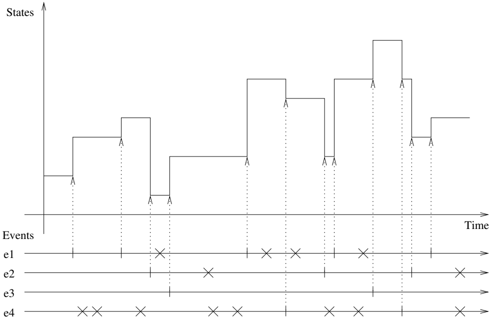


Index function $I(x) = \inf\{W \mid \theta(W) = x\}$.
Indifference case : when queue size is x, rejecting or accepting the next batch are both optimal choices if the rejection cost is $I(x)$.

Event modelling

Multidimensional state space : $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_K$ with $\mathcal{X}_i = \{0, \cdots, C_i\}$.

Event e :
$\rightsquigarrow$ transition function $\Phi(., e)$; (skip rule)
$\rightsquigarrow$ Poisson process λ_e



Uniformization

$$\Lambda = \sum_e \lambda_e \text{ and } \mathbb{P}(event\ e) = \frac{\lambda_e}{\Lambda}; \text{ Trajectory : } \{e_n\}_{n \in \mathbb{Z}} \text{ i.i.d. sequence.}$$

$\Rightarrow$ Homogeneous Discrete Time Markov Chain [Bremaud 99] $X_{n+1} = \Phi(X_n, e_{n+1})$.

Index routing in queueing networks

Index functions for event e

For queue i $\quad I_i^e : \{0, \cdots, C_i\} \longrightarrow \mathcal{O}$ (totally ordered set).

Property : $\forall x_i, x_j \;\; I_i^e(x_i) \neq I_j^e(x_j)$.

ex: inverse of a priority,...

Routing algorithm:

if $x_i > 0$ **then**
  { a client is available in the origin queue i}
  $x_i = x_i - 1$; { the client is removed from the origin queue}

  $j = arg \min_i I_i^e(x_i)$; { computation of the destination}

  if j $\neq$ -1 **then**
    $x_j = x_j$+1; { arrival of the client in queue j }
    { in the other case, the client goes out of the network}
  end if
end if

Monotonicity of index routing policies

State space

$\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_K$ with $\mathcal{X}_i = \{0, \cdots, C_i\}$, C_i : capacity of Q_i
$\mathcal{X}$ is partially ordered $\leqslant_{\mathcal{X}}$ (extension of natural ordering on $\mathbb{N}$)

$$x = (x_1, \cdots, x_K) \;\; y = (y_1, \cdots, y_K)$$

$$x \prec y \iff \forall i \quad x_i \leqslant y_i$$

An operator (transition function) Φ is monotone iff

$$x \prec y \Longrightarrow \Phi(x) \prec \Phi(y).$$

Proposition

If all index functions I_i^e are monotone then event e is monotone.

Monotonicity of index routing policies (proof)

Proof : Let $x \prec y$ two states and let be an index routing event. Let i be the origin queue for the event.

$$j_x = argmin_j I_j^e(x_j) \text{ and } j_y = argmin_j I_j^e(y_j)$$

Case 1 $x_i = y_i = 0$ nothing happens and
$\Phi(x, e) = x \prec y = \Phi(y, e)$

Case 2 $x_i = 0,\ y_i > 0$ then $\Phi(x, e) = x \prec y - e_i + e_{j_y} = \Phi(y, e)$

Case 3 $x_i > 0,\ y_i > 0$ then

$$I_{j_x}^e(x_{j_x}) < I_{j_y}^e(x_{j_y}) \leqslant I_{j_y}^e(y_{j_y}) < I_{j_x}^e(y_{j_x});$$

then $x_{j_x} < y_{j_x}$ and

$$\Phi(x, e) = x - e_i + e_{j_x} \leqslant y - e_i \leqslant y - e_i + e_{j_y} = \Phi(y, e)$$

Monotonicity of routing

[Glasserman and Yao] provides a more general framework for monotone events

Examples

All of these events could be expressed as index based routing policies :
- external arrival with overflow and rejection
- routing with overflow and rejection or blocking
- routing to the shortest available queue
- routing to the shortest mean available response time
- general index policies [Palmer-Mitrani]
- rerouting inside queues
...

Monotonicity of routing : examples

Stateless routing

Overflow routing with rejection

$$I_j^e(x_j) = \begin{cases} prio(j) & \text{if } x_j < C_j; \\ +\infty & \text{elsewhere} \end{cases}$$

$$I_{-1}^e = \max_j C_j.$$

Overflow routing with rejection

$$I_j^e(x_j) = \begin{cases} prio(j) & \text{if } x_j < C_j; \\ +\infty & \text{elsewhere} \end{cases}$$

$$I_i^e = \max_j C_j.$$

State dependent routing

Join the shortest queue

$$I_j^e(x_j) = \begin{cases} x_j & \text{if } x_j < C_j; \\ +\infty & \text{elsewhere;} \end{cases}$$

$$I_{-1}^e = \max_j C_j.$$

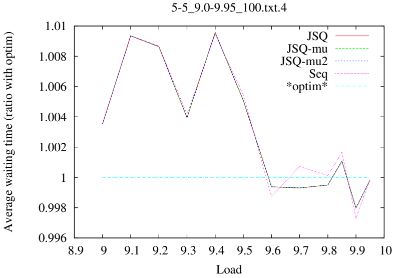
Join the shortest response time

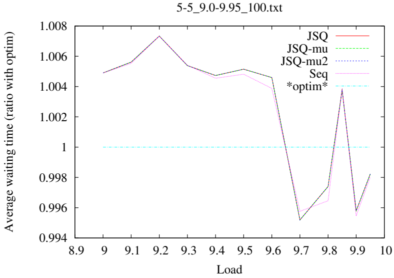
$$I_j^e(x_j) = \begin{cases} \frac{x_j+1}{\mu_j} & \text{if } x_j < C_j; \\ +\infty & \text{elsewhere;} \end{cases}$$

$$I_{-1}^e = \max_i C_i.$$

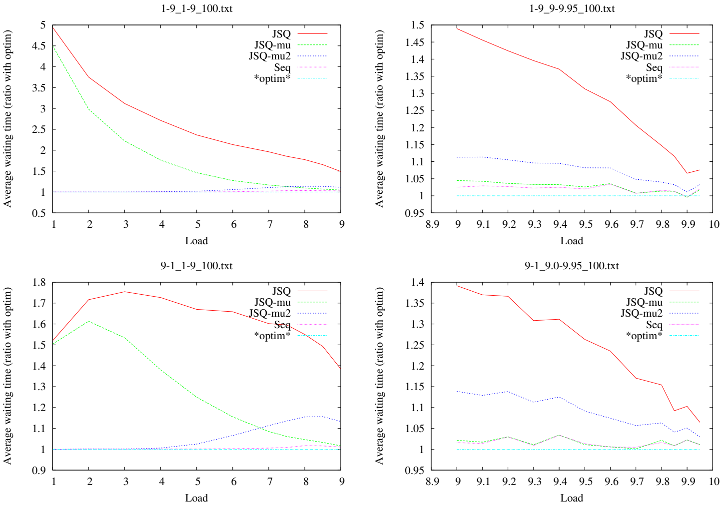
Some numerical experiments

Two queues with the same service rate ($\mu = 5$)



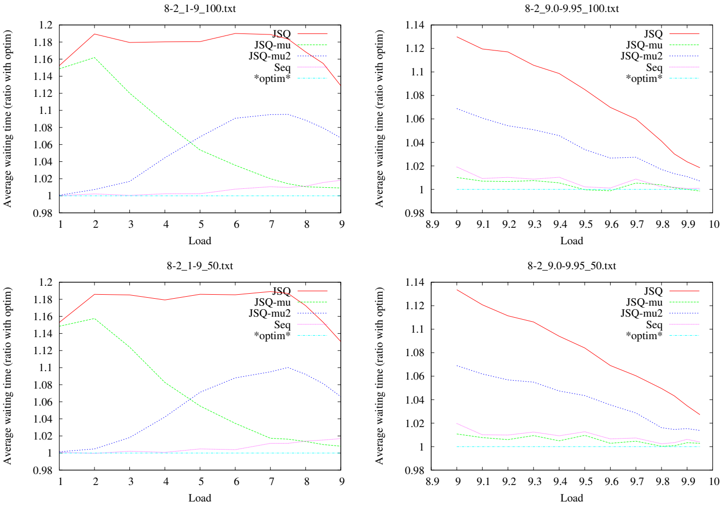


Some numerical experiments(II)



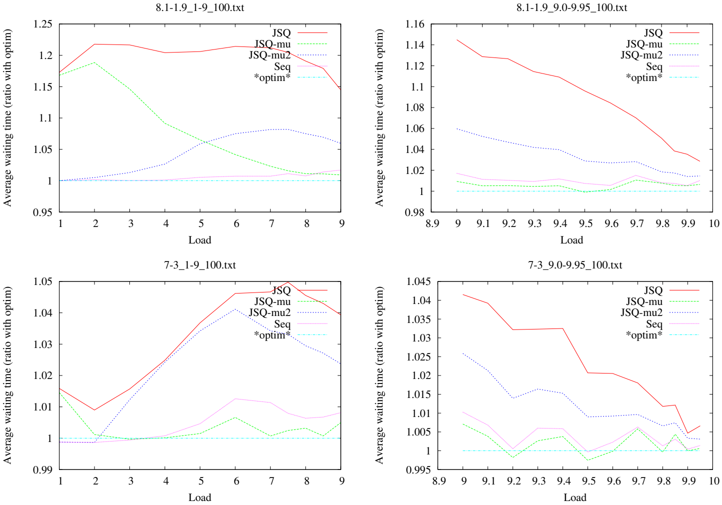
Several cases with two queues with respective parameters ($\mu = 9, 1$), $C = 100$

Some numerical experiments(III)



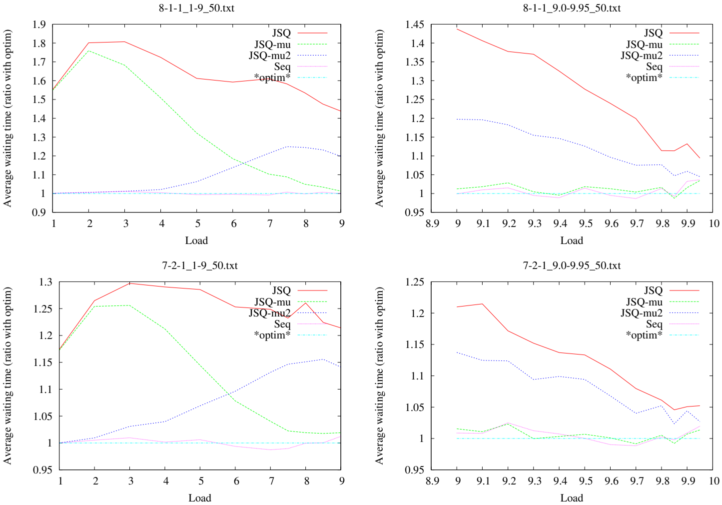
Several cases with two queues with respective parameters ($\mu = 8, 2$), $C = 100$.

Some numerical experiments(IV)



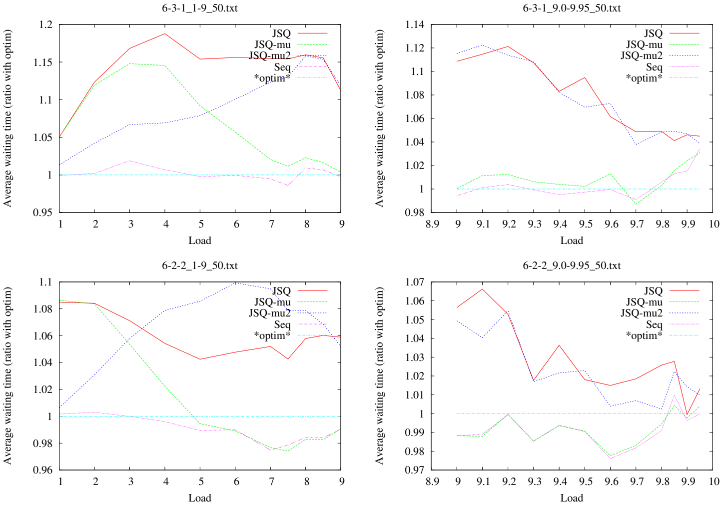
Some other cases

Some numerical experiments(V)



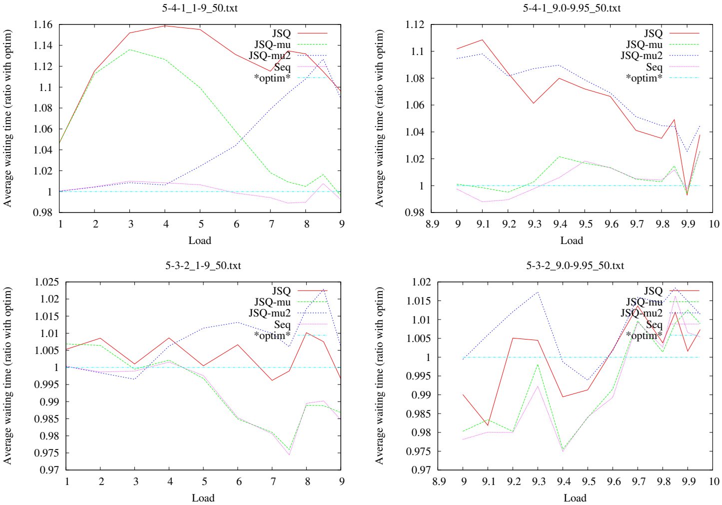
Three queues. Respectively, $\mu = 8, 1, 1$ and $\mu = 7, 2, 1$.

Numerical experiments(VI)



Now, $\mu = 6, 3, 1$ and $\mu = 6, 2, 2$.

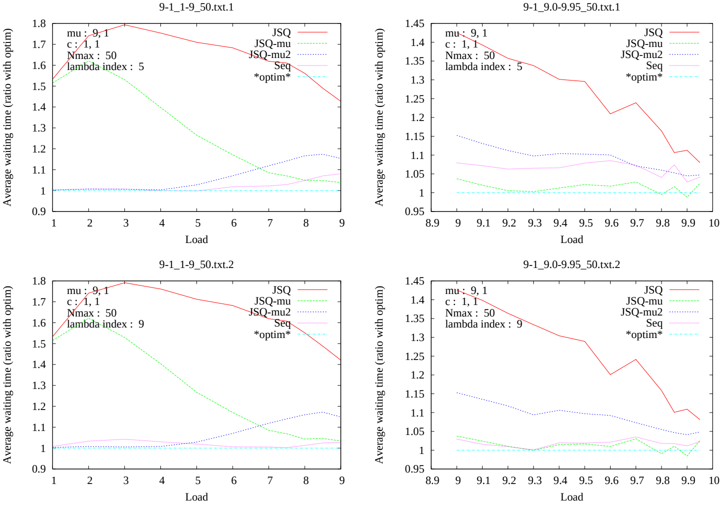
Numerical experiments(VII)



Now, $\mu = 5, 4, 1$ and $\mu = 5, 3, 2$.

Robustness of Index policies

The index policy was computed for $\lambda = 5$ or 9 and used over the whole range $\lambda = 1$ to 10.



Conclusion and future works

Summary

- monotone structure of scheduling inqueueing networks
- perfect simulation $\Rightarrow$ direct sample of steady-state
- general formalism of events : “look like independent queues”
- practically efficient $\Rightarrow$ quality of heuristic

Future works

- Generalized index function
- Multiple routing
- Batch arrivals
- Software environment PSI2