Dynamic repairman assignment in a layered queueing network with correlated queues

Jan-Pieter Dorsman, Sandjai Bhulai, Maria Vlasiou

Technische Universiteit Eindhoven University of Technology

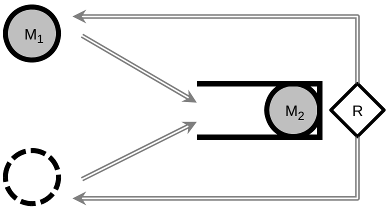
vrije Universiteit amsterdam

CWI Centrum Wiskunde & Informatica

YEQT-VI 2012, Nov 1

The model

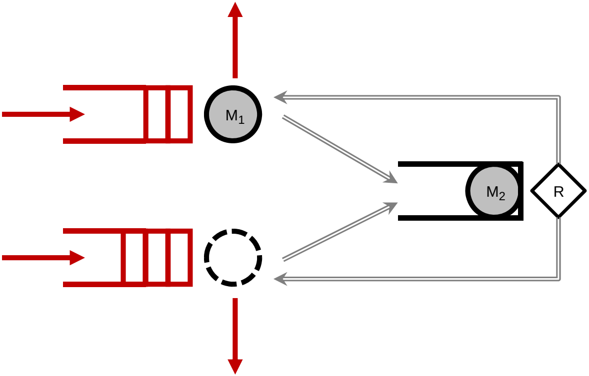
An extension of the machine repair model.



We are particularly interested in the behaviour of the ‘red’ queues.

The model

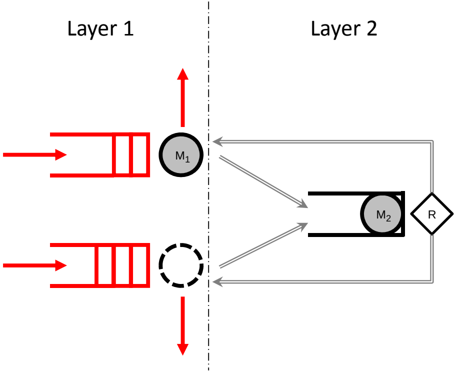
An extension of the machine repair model.



We are particularly interested in the behaviour of the 'red' queues.

The model

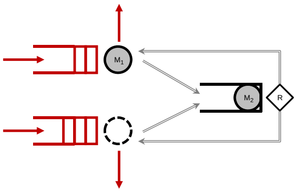
An extension of the machine repair model.



We are particularly interested in the behaviour of the ‘red’ queues.

FCFS repairs

When machines are repaired
in the order of breakdown, exact analysis
of the queue lengths is already hard.

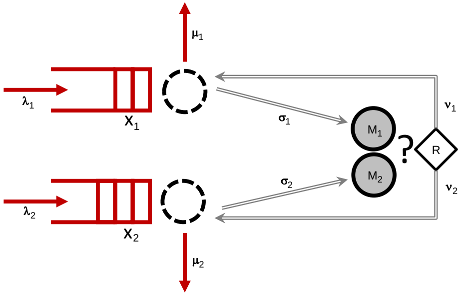


Approximations can be derived by, e.g.,

- assuming a certain dependence structure in consecutive downtimes;
- interpolating between light-traffic and heavy-traffic results.

Today's main question

FCFS repairs might not lead to optimal queue lengths. What is the optimal dynamic repair policy?



Or, how to dynamically allocate the repairman's capacity in order to minimise

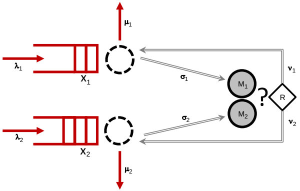
$$g = c_1\mathbb{E}[X_1] + c_2\mathbb{E}[X_2],$$

when given information on the current queue lengths and state of the machines?

Today's agenda

How to dynamically assign the fractions q_1 and q_2 of repair capacity to the machines?

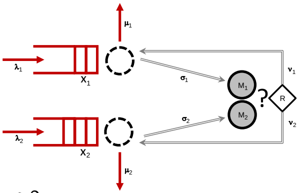
We study this question by using theory on Markov decision processes (MDPs).



- Formulation as a Markov decision process.
- Derivation of structural properties of the optimal policy.
- Derivation of a near-optimal policy.

Formulation as an MDP

Let
x_i be the number of type-i products in the system and $w_i = \mathbb{1}_{\{\text{Machine } i \text{ is operational}\}}$.



Then, the MDP is characterised by

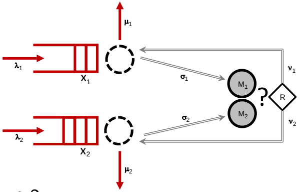
- State $s = (x_1, x_2, w_1, w_2) \in \mathcal{S} = \mathbb{N}^2 \times \{0,1\}^2$,
- Action $a = (q_1, q_2) \in \mathcal{A}_s = \{(q_1, q_2) : q_1 \in [0, 1 - w_1], q_2 \in [0, 1 - w_2], q_1 + q_2 \leq 1\}$,
- Transition probabilities, $i = 1, 2$:

$$
\begin{aligned}
P_a(s, s + e_i) &= \lambda_i, && \text{(product arrivals)} \\
P_a(s, s - e_i) &= \mu_i w_i \mathbb{1}_{\{x_i > 0\}}, && \text{(product services)} \\
P_a(s, s - e_{i+2}) &= \sigma_i w_i, && \text{(machine breakdowns)} \\
P_a(s, s + e_{i+2}) &= q_i \nu_i, && \text{(machine repairs)} \\
P_a(s, s) &= 1 - \lambda_i - w_i(\mu_i \mathbb{1}_{\{x_i > 0\}} + \sigma_i) && - q_i \nu_i,
\end{aligned}
$$

- Cost function $c(s) = c_1 x_1 + c_2 x_2$.

Formulation as an MDP

Let
x_i be the number of type-i products in the system and $w_i = \mathbb{1}_{\{\text{Machine } i \text{ is operational}\}}$.



Then, the MDP is characterised by

- State $s = (x_1, x_2, w_1, w_2) \in \mathcal{S} = \mathbb{N}^2 \times \{0,1\}^2$,
- Action $a = (q_1, q_2) \in \mathcal{A}_s = \{(q_1, q_2) : q_1 \in [0, 1 - w_1], q_2 \in [0, 1 - w_2], q_1 + q_2 \leq 1\}$,
- Transition probabilities, $i = 1, 2$:

$$
\begin{aligned}
P_a(s, s+e_i) &= \lambda_i, && \text{(product arrivals)}\\
P_a(s, s-e_i) &= \mu_i w_i \mathbb{1}_{\{x_i>0\}}, && \text{(product services)}\\
P_a(s, s-e_{i+2}) &= \sigma_i w_i, && \text{(machine breakdowns)}\\
P_a(s, s+e_{i+2}) &= q_i \nu_i, && \text{(machine repairs)}\\
P_a(s, s) &= 1 - \lambda_i - w_i(\mu_i \mathbb{1}_{\{x_i>0\}} + \sigma_i) && -q_i\nu_i,
\end{aligned}
$$

- Cost function $c(s) = c_1 x_1 + c_2 x_2$.

Formulation as an MDP

Ultimate goal

Analytic expression for optimal policy $\pi^{opt} : \mathcal{S} \to \mathcal{A}$ that minimises the long-run expected costs per time unit.

The relative value function $V^*(s)$ is the long-term difference in expected total costs accrued when starting in state s instead of some reference state under policy π^*.

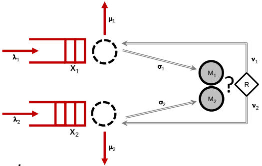
The value g^* represents the long-run expected costs per time unit under policy π^*.

V^{opt} and g^{opt} satisfy Bellman's optimality equations:

$$V^{opt}(s) + g^{opt} = \min_{a \in \mathcal{A}_s} \left\{ c(s) + \sum_{s' \in \mathcal{S}} P_a(s, s') V^{opt}(s') \right\}$$

Structural properties

We can prove that the optimal policy is work-conserving, i.e., $q_1 + q_2 = 1 - w_1 w_2$.



Minimising action in state s is given by

$$\pi^{opt}(s) = \underset{(q_1,q_2)\in\mathcal{A}_s}{\arg\min} \{q_1\nu_1(V^{opt}(x_1,x_2,1,w_2) - V^{opt}(x_1,x_2,0,w_2)) + q_2\nu_2(V^{opt}(x_1,x_2,w_1,1) - V^{opt}(x_1,x_2,w_1,0))\}.$$

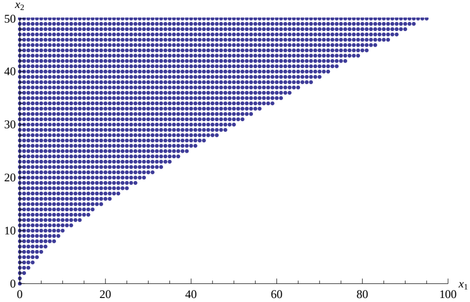
- Value iteration: iterate

$$V^{n+1}(s) = \min_{a\in\mathcal{A}_s} \left\{ c(s) + \sum_{s'\in\mathcal{S}} P_a(s,s')V^n(s') \right\}$$

  starting with arbitrary V^0.
- If a structural property 'survives' an iteration, it applies to V^{opt} by induction.
- Prove that $V_1(x_1,x_2,1,w_2) - V_1(x_1,x_2,0,w_2) \geq 0$, if $V^0(x_1,x_2,1,w_2) - V^0(x_1,x_2,0,w_2) \geq 0$.

Structural properties

The optimal policy is a threshold policy.



No dot: full capacity to machine 1, blue dot: full capacity to machine 2.

Derivation of a near-optimal policy

Near-optimal policies can be derived by using a one-step policy improvement.

- Policy iteration: for any π^*,

$$\pi'(s) = \underset{(q_1,q_2)\in\mathcal{A}_s}{\arg\min} \{q_1\nu_1(V^*(x_1,x_2,1,w_2) - V^*(x_1,x_2,0,w_2)) + q_2\nu_2(V^*(x_1,x_2,w_1,1) - V^*(x_1,x_2,w_1,0))\}$$

  If $\pi' = \pi^*$, then this is the optimal policy. Otherwise, set $\pi^* = \pi'$ and repeat.
- Problem: $V'(x_1,x_2,w_1,w_2)$ usually does not allow for analytic, and sometimes not even for numerical solutions.
- Norman (1972):
  1. Choose an initial policy that allows decomposition of the large Markov process into multiple small Markov processes;
  2. Perform one step of policy iteration.

Derivation of a near-optimal policy

Near-optimal policies can be derived by using a one-step policy improvement.

- Policy iteration: for any π^*,

$$\pi'(s) = \arg\min_{(q_1,q_2)\in\mathcal{A}_s} \{q_1\nu_1(V^*(x_1,x_2,1,w_2) - V^*(x_1,x_2,0,w_2)) + q_2\nu_2(V^*(x_1,x_2,w_1,1) - V^*(x_1,x_2,w_1,0))\}$$

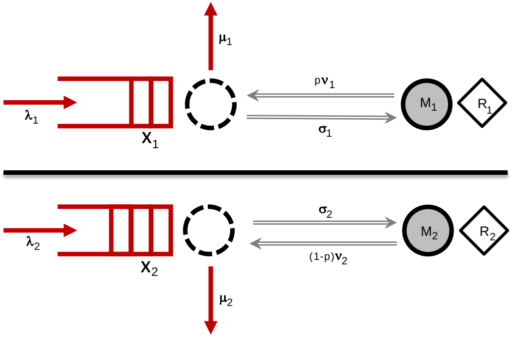
  If $\pi' = \pi^*$, then this is the optimal policy. Otherwise, set $\pi^* = \pi'$ and repeat.

- Problem: $V'(x_1, x_2, w_1, w_2)$ usually does not allow for analytic, and sometimes not even for numerical solutions.
- Norman (1972):
  1. Choose an initial policy that allows decomposition of the large Markov process into multiple small Markov processes;
  2. Perform one step of policy iteration.

Static policy as an initial policy

Suppose the repairman always reserves a fraction p of his capacity for machine 1, and $(1-p)$ for machine 2.

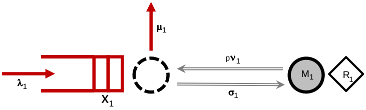
The system can then be decomposed into simpler subsystems:



Analysis of M/M/1 queue with exponential server vacations!

Static policy as an initial policy

Solving the subsystem:



Poisson equations

$$\begin{aligned} V(x,w)+g =& x+\lambda V(x+1,w)+\mu w V((x-1)^+,0) \\ &+\sigma w V(x,0)+\nu(1-w)V(x,1) \\ &+(1-\lambda-(\mu+\sigma)w-\nu(1-w))V(x,w) \end{aligned}$$

Solution of $V(x,w)$ is a second-order polynomial in x with coefficients dependent on w. Back to the complete model:

$$V^{sta}(x_1,x_2,w_1,w_2)=c_1V_1(x_1,w_1)+c_2V_2(x_2,w_2)$$

Static policy as an initial policy

One-step policy improvement:

- Determine optimal static policy, i.e., optimal value p.
- Obtain improved policy:

$$\begin{aligned} \arg\min_{(q_1,q_2)\in\mathcal{A}_{(x_1,x_2,w_1,w_2)}} & \{q_1\nu_1(V^{sta}(x_1,x_2,1,w_2) - V^{sta}(x_1,x_2,0,w_2)) \\ & + q_2\nu_2(V^{sta}(x_1,x_2,w_1,1) - V^{sta}(x_1,x_2,w_1,0))\}. \end{aligned}$$

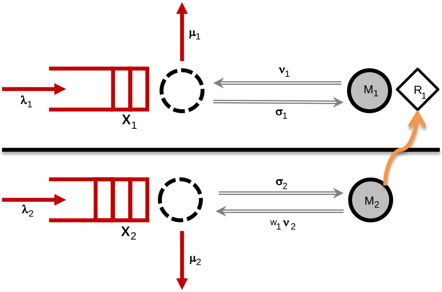
The improved policy is expressed analytically and meets the structural properties of the optimal policy:

$$\pi^{oss}(x_1,x_2,w_1,w_2) = \begin{cases} (0,0) & \text{if } w_1 = w_2 = 1, \\ (1,0) & \text{if } w_1 = 1 - w_2 = 0, \text{ or if } w_1 w_2 = 0 \text{ and} \\ & \frac{\frac{c_1\nu_1}{c_2\nu_2}((\alpha_{1,1}-\alpha_{2,1})x_1-\alpha_{3,1})+\alpha_{3,2}}{\alpha_{1,2}-\alpha_{2,2}} \leq x_2, \\ (0,1) & \text{otherwise} \end{cases}$$

Priority policy as an initial policy

There might not be a 'stable' static policy available. Alternative: take a priority policy as initial policy.

Under this policy, the repairman always gives priority to M_1 by taking the action $(q_1, q_2) = ((1 - w_1), w_1(1 - w_2))$.



$$V^{prio}(x_1, x_2, w_1, w_2) = c_1 V_1(x_1, w_1) + c_2 V_2(x_2, w_1, w_2)$$

No complete decomposition, and therefore hard to analyse!

Priority policy as an initial policy

$$V^{prio}(x_1, x_2, w_1, w_2) = V_1(x_1, w_1) + V_2(x_2, w_1, w_2)$$

No complete decomposition, and therefore hard to analyse!

However we conjecture that, as $x_2 \to \infty$, $V_2(x_2, w_1, w_2)$ behaves like a second-order polynomial in x_2:

- $V_2(x_2, w_1, w_2) - V_2(x_2 - 1, w_1, w_2)$ asymptotically equals expected time for the queue to empty when starting in (x_2, w_1, w_2).
- $(w_1(t), w_2(t))$ moves to equilibrium, so any service delaying effect imposed by $w_1 = w_1(0)$ is conjectured to build up to constant.
- Other than the effect of w_1, the system behaves like a $M/Ph/1$ queue with vacations.

First-order and second-order coefficients can be obtained by studying the Poisson equations.

Priority policy as an initial policy

$$V^{prio}(x_1, x_2, w_1, w_2) = V_1(x_1, w_1) + V_2(x_2, w_1, w_2)$$

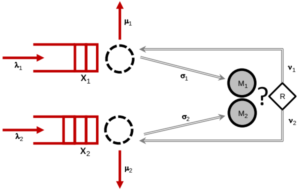
By using the asymptotic version of V_2 in the above, we obtain an improved policy:

- Determine optimal priority policy.
- Perform one-step policy improvement. The result is again a work-conserving threshold policy:

$$\pi^{osp}(x_1, x_2, w_1, w_2) = \begin{cases} (0,0) & \text{if } w_1 = w_2 = 1, \\ (1,0) & \text{if } w_1 = 1 - w_2 = 0, \text{ or if } w_1 w_2 = 0 \\ & \text{ and } \frac{\nu_1 c_1((\alpha_1 - \alpha_2)x_1 - \alpha_3)}{c_2(\nu_1 \Delta_{1,0} - \nu_2 \Delta_{0,1})} \leq x_2, \\ (0,1) & \text{otherwise} \end{cases}$$

Conclusion

Suggested policy: use one-step improved static policy if it exists, otherwise use one-step improved priority policy.



The obtained policy

- can be expressed analytically;
- allows for more machines due to decomposition properties;
- is almost always feasible;
- performs generally well. Cumulative breakdown of the relative performance of this policy w.r.t. the optimal policy in 1296 systems:

	$< 0.1\%$	$< 1\%$	$< 5\%$	$< 10\%$
Cumulative % of perf.	32.02%	56.17%	85.49%	95.14%