Learning Optimal Admission Control in Partially Observable Queueing Networks

Jonatha Anselmi, Bruno Gaujal, Louis-Sébastien Rebuffi
Inria and Univ. Grenoble Alpes

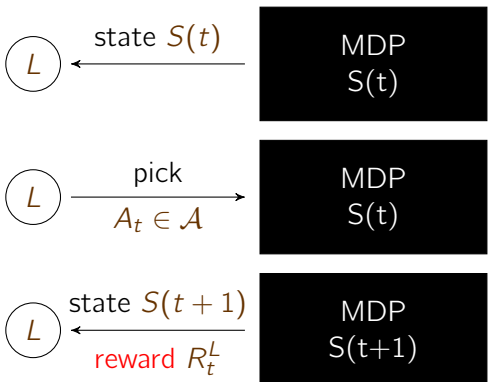
Toulouse, Nov. 2023

"Un coup de dés jamais n'abolira le hasard"

Reinforcement learning in MDP: framework

MDP $M = (\mathcal{S}, \mathcal{A}, r, P)$. Let L be a learning algorithm.

1. Observe
2. Pick action
3. Get feedback



Reinforcement learning in PO-MDPs

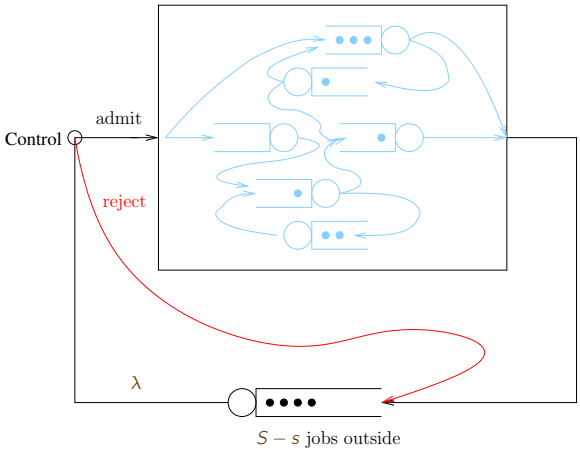
To go one step further: the state $S(t)$ is not perfectly observed by the learner: PO-MDP.

In general solving a POMDP is P-space complete and learning is also hard (lower bounds on the sample complexity are exponential).

One extensively studied case: games with hidden states (poker).

Another case (not studied): in/out systems seen as black boxes.

Admission control in a queueing network



Motivation: serverless plateforms with complex structure (Knative).

Problem setting

The problem is to learn the optimal admission control policy in the following setting.

Cost function:

- the learner may choose to reject/admit jobs arriving in the network, with cost γ_{reject} for each rejected job;
- for every time unit in the system, each job induces a holding cost γ_{hold} (see Borgs-2014);
- the leaner takes decisions only relying on its observations up to time t.

What does the learner know?

- The learner can observe the external events: arrivals and departures of jobs. This implies at t, the total number of jobs in the network, s_t is known (partially observed *state*).
- The learning algorithm knows T, learning horizon. This is not a strong requirement as one can make the algorithm oblivious to T by using a classical doubling trick on T.

Original MDP

System is modeled as a MDP $M^o = (\mathcal{X}^o, \mathcal{A}^o, P^o, r^o)$, with uniformization constant to see the process in discrete time: $U \geq \lambda + \sum_{i=1}^{N} \mu_i^o$.

- The state space $\mathcal{X}^o$ is the set of all tuples $(x_1, \ldots, x_N)$ given by the number of jobs x_i in each queue i.
- The action space is $\mathcal{A}^o := \{0, 1\}$ where 0 stands for rejection and 1 for admission.
- The transition matrix P^o is constructed by using the routing matrix L, the arrival rate λ and the service rates (μ_i^o).
- Reward: $r^o(\mathbf{x}, a) := r_{\max} - \dfrac{\lambda \gamma_{\text{reject}}(1-a) + \gamma_{\text{hold}} s}{U}$ where $s := \sum_{i=1}^{N} x_i$.

Norton equivalent queue

The input/output behavior of the network in its stationary regime is equivalent to a single queue

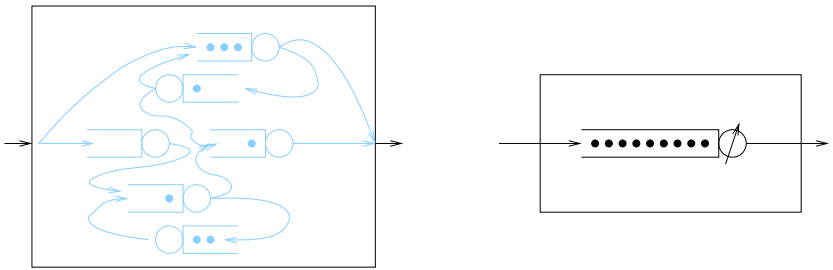


Figure: Illustration of Norton Equivalence theorem.

Arrival rate: λ. Service rate: $\mu(s)$ (Throughput of the network with s jobs in total).

Aggregated MDP

$M = (\mathcal{S}, \mathcal{A}, r, P)$.

- The state space $\mathcal{S} = \{0, \ldots, S\}$
- The actions are the same as for the original MDP: $\mathcal{A} = \mathcal{A}^o = \{0, 1\}$ (reject or accept).
- The reward are also the same: $r(s, a) = r_{\max} - \dfrac{\lambda\gamma_{\text{reject}}(1-a) + \gamma_{\text{hold}}s}{U}$
- Let π be a policy (a function from $\mathcal{S} \to \mathcal{A}$) on M.

$$P(s' \mid s, \pi(s)) = \sum_{\mathbf{x}, |\mathbf{x}|=s} \sum_{\mathbf{y}, |\mathbf{y}|=s'} \frac{\nu^{o,\pi}(\mathbf{x})}{\nu^{\pi}(s)} P^o(\mathbf{y} \mid \mathbf{x}, \pi(\mathbf{x})), \tag{1}$$

where $\nu^{\pi}(s) = \sum_{\mathbf{x}, |\mathbf{x}|=s} \nu^{o,\pi}(\mathbf{x})$ is the equivalent stationary measure. These probabilities are also those for the Norton equivalent queue.

Comparison between both MDPs

The original MDP M^o has a greater set of policies than the aggregated MDP M.
Therefore, $g^o(M^o, \pi^*) \geq g^*(M)$.
However, if we only consider the set Π^o_{sum} of policies for M^o that take the same action in all the states with the same total number of jobs, then optimal gains coincide.

$$\max_{\pi \in \Pi^o_{sum}} g^o(M^o, \pi) = g^*(M).$$

As for learning when the full state is not observable, the best one can hope for is to learn $\max_{\pi \in \Pi^o_{sum}} g^o(M^o, \pi)$, so we will consider the regret with respect to this gain, in the following.

Regret

Definition (Regret)

The *regret* at time T of the learning algorithm $\mathcal{L}$ is

$$\mathrm{Reg}(M, \mathcal{L}, T) := Tg^*(M) - \sum_{t=1}^{T} r_t. \tag{2}$$

Here, $g^*(M)$ is the optimal gain in the aggregated MDP. The reward r_t is the reward of the state visited at time t under the current policy used by the learning algorithm.

Learning algorithm: Main ideas

The total number of jobs $(s_t)_{t \leq T}$ is not Markovian. Instead, consider $\{s_1, \ldots, s_{\tau_{\text{norton}}}\}$.
If τ_{norton} is larger than the mixing time of the MDP, then each subsequence $s_i, s_{i+\tau_{\text{norton}}}, s_{i+2\tau_{\text{norton}}}, \ldots$ is "almost" independent and can be used for statistical estimations.
We set a collection of τ_{norton} learning algorithms $\mathcal{L}_1, \ldots, \mathcal{L}_{\tau_{\text{norton}}}$, using resp. the subsequence $(s_{i+k\tau_{\text{norton}}})_{k \in \mathbb{N}}$ of observations, called *modules*.
Each module $\mathcal{L}_i$ behaves similarly as the classical optimistic algorithm. There are no interactions between modules except for the number of visits that contributes to the construction of the global confidence region.
However,

1. The modules $\mathcal{L}_1, \ldots, \mathcal{L}_{\tau_{\text{norton}}}$ are not independent of each other
2. For each learning module $\mathcal{L}_i$, its sequence of observations $s_{i+\tau_{\text{norton}}}, s_{i+2\tau_{\text{norton}}}, s_{i+3\tau_{\text{norton}}}, \ldots$ is not really stationary and independent, but only weakly correlated.

UCRL-M

Algorithm 1: The UCRL-M algorithm.

while $t \leq T$ **do**
  Initialize episode k with $t_k := t$
  Compute the confidence region $\mathcal{M}_k$;
  Find a policy $\tilde{\pi}_k$ and an optimistic MDP $\tilde{M}_k \in \mathcal{M}_k$ s.t. $g(\tilde{M}_k, \tilde{\pi}_k) \geq \max_{M_k \in \mathcal{M}_k} \max_{\pi} g(M_k, \pi) - \frac{\delta_{\max}}{\sqrt{t_k}}$;
  Ramping phase (Φ): Iterate the MDP with policy $\tilde{\pi}_k$ for τ_{norton} time-steps, discard the observations and set $t := t + \tau_{\text{norton}}$.
  Exploration: while $V_k^{(m_t)}(s_t, \tilde{\pi}_k(s_t)) < \max\{1, N_{t_k}^{(m_t)}(s_t, \tilde{\pi}_k(s_t))\}$ **do**
    Use action $a_t = \tilde{\pi}_k(s_t)$; collect $r(s_t, a_t)$; move to $t + 1$;
  end
end

Confidence region $\mathcal{M}_k$

$$\forall (s,a), \quad \left|\tilde{r}(s,a) - \hat{r}_k^{(m_k(s,a))}(s,a)\right| \leq \delta_{\max} \sqrt{\frac{2\log(2At_k)}{\max\left\{1, N_{t_k}^{(m_k(s,a))}(s,a)\right\}}}; \tag{3}$$

$$\forall (s,a), \quad \left\|\tilde{p}(\cdot \mid s,a) - \hat{p}_k^{(m_k(s,a))}(\cdot \mid s,a)\right\|_1 \leq \sqrt{\frac{8\log(2At_k)}{\max\{1, N_{t_k}^{(m_k(s,a))}(s,a)\}}}, \tag{4}$$

where $m_k(s,a)$ is the most frequent module at time t_k under (s,a). This is the only interaction between modules.

Regret of UCRL-M

Theorem

Define $\kappa := 228(\gamma_{\text{reject}} + \frac{\gamma_{\text{hold}}}{U})\frac{U}{\mu(1)} C_1 \left(1 - \sqrt{\frac{\lambda}{\mu(s_0)}}\right)^{-3}$.

For the choice $\tau_{\text{norton}} = 5\frac{\log T}{\log 1/\rho}$, *we have:*

$$\mathbb{E}\left[Reg(M, UCRL\text{-}M, T)\right] \leq \kappa \log(2T) \sqrt{T \log^{-1}(1/\rho)} + R_{\text{LO}}, \tag{5}$$

where $R_{\text{LO}} := O(T^{1/4})$ *is a lower order term of the regret.*

Time complexity of UCRL-M

Theorem

The time complexity of UCRL-M is $O(KS\tau_{\text{norton}} + Kt_{\text{evi}} + T)$*, where* K *is the number of episodes and* t_{evi} *the time complexity of extended value iteration. Furthermore,* $\mathbb{E}(K) = O(\log T)$.

Proof. The expected number of episodes, $\mathbb{E}K = O(\log T)$ because of the doubling test used to end the episodes. The time complexity of line 3 is $O(KS\tau_{\text{norton}})$. The complexity of line 4 is $O(Kt_{\text{evi}})$. The complexity of line 5 is $O(K\tau_{\text{norton}})$. The complexity of lines 6-7 is $O(T - K\tau_{\text{norton}})$, the number of useful observations. As for the expected number of episodes, $\mathbb{E}K = O(\log T)$ because of the doubling trick used to end the episodes.
The number of useful samples (excluding the ramping phases) is $T - K\tau_{\text{norton}}$, and each module uses $\dfrac{T - K\tau_{\text{norton}}}{\tau_{\text{norton}}}$ samples.
The complexity barely depends on τ_{norton} or t_{evi} (one per episode) since K is small w.r.t. T.

Controlling the convergence rate ρ

The efficiency of UCRL-M is critically based on controlling τ_{norton} and ρ. In particular, the regret of UCRL-M depends on $W := \log^{-1/2}(1/\rho)$.
Also τ_{norton} is defined as $\tau_{\text{norton}} := 5 \log T / \log(1/\rho)$. where ρ is such that

$$\max_{\pi} \sup_{x_0 \in \mathcal{S}} \left\| \mathbb{P}_{x_0}^{o,\pi}(x_t = \cdot) - \nu^o(\pi) \right\|_{TV} \leq C\rho^t \quad \forall t > 0. \tag{6}$$

Relations with mixing and coupling times:
Let $d(t) := \sup_{x_0 \in \mathcal{S}} \|\mathbb{P}_{x_0}(x_t = \cdot) - \nu\|_{TV}$. The *mixing time* is $t_{mix} := \min\{t : d(t) \leq 1/4\}$.
A classical bound is $\rho \leq \dfrac{1}{2^{t_{mix}^{-1}}}$. This implies $W \leq \sqrt{t_{mix} \log(2)}$.
The *coupling time* $\tau_{x,y} := \min\{t : X_t = Y_t\}$, with X_t and Y_t coupled and start at $X_0 = x$ and $Y_0 = y$ resp. Then, $d(t) \leq \max_{x,y} \mathbb{P}(\tau_{x,y} > t)$. Using Markov inequality,

$$t_{mix} \leq 4 \max_{x,y} \mathbb{E}[\tau_{x,y}]. \tag{7}$$

Acyclic networks

The policy where the coupling time is the largest is when all jobs are admitted. The coupling time is upper bounded by the coupling in an open network where all the N queues have buffers bounded by S. In this case, the coupling time has been studied in Dopper-2006, where the following result is proved

$$\max_{x,y} \mathbb{E}[\tau_{x,y}] \leq \sum_{i=1}^{N} \frac{U^2}{(\lambda_i^o + \mu_i^o)(\mu_i^o - \lambda_i^o)} S, \tag{8}$$

where U is the uniformization constant and $(\lambda_i)_{i \leq N}$ is the solution of the traffic equations. Therfore,

$$W \leq \kappa_0 \sqrt{NS},$$

where κ_0 is a constant: $\kappa_0 = \max_i \sum_{i=1}^{N} \frac{U}{\lambda_i^o + \mu_i^o}$.

Hyperstable networks

A network is called *hyperstable* if for each queue i, $\sum_j L_{ji}\mu_j^o + L_{0i}\lambda < \mu_i^o$.

The policy under which the coupling time is the largest is when all jobs are admitted. Under this policy, the coupling time is upper bounded by the coupling in an open network where all the N queues have buffers bounded by S.

Coupling times of hyperstable networks with finite buffer queues have been studied in Anselmi-Gaujal-2014:

$$\max_{x,y} \mathbb{E}[\tau_{x,y}] \leq \kappa_2 N^2 S \sum_{i=1}^{N} \frac{\lambda_i^o}{\mu_i^o - \lambda_i^o}, \tag{9}$$

where κ_2 is a constant. This induces a similar bound on the term W in the regret:

$$W \leq \kappa_3 N \sqrt{S},$$

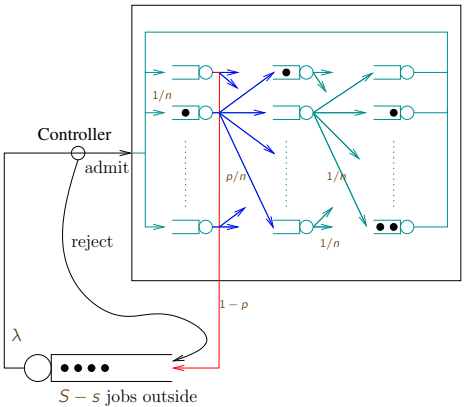
where κ_3 is yet another constant.

Making the algorithm oblivious to ρ

UCRL-M uses $\tau_{\text{norton}} = 5 \log T / \log(1/\rho)$ modules.
This implies an *a priori* knowledge of ρ, (or at least an upper bound) of the network being learned.

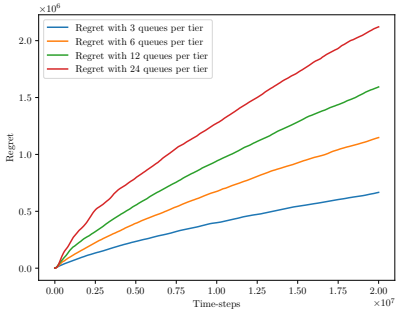
UCRL-M can be made oblivious to ρ by using $\tau_{\text{norton}} \geq 5 \log T / \log \rho^{-1}$ for any large enough T.
For example, one can chose $\tau_{\text{norton}} := \log^2(T)$. This patch adds a multiplicative $\log(T)$ term in the asymptotic bound of the regret.

Numerical experiments



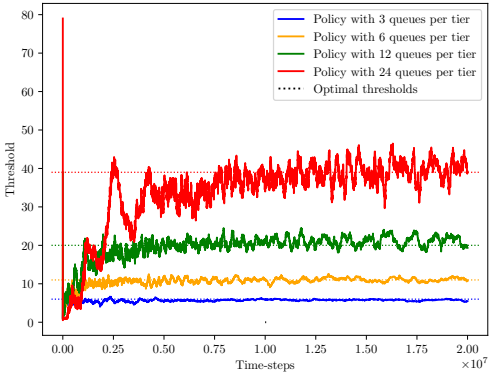
This multi-tier structure is common in empirical studies of computer systems and is the default architecture of web applications deployed on Amazon Elastic Compute Cloud (EC2).

Regret scaling with S



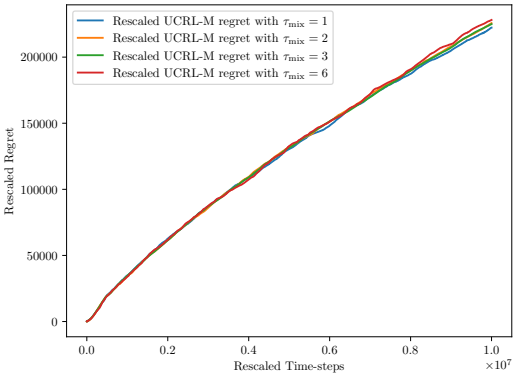
The number of queues n (and the number of jobs S) scales multiplicatively while the regret is increasing in $\log(S)$. Knowing that the dependency in S of the regret bound mainly comes from ρ, this is much slower than the square root bounds given for acyclic and hyperstable networks.

Optimal admission



The optimal threshold scales linearly with N.

Regret scaling with the number of modules, τ_{norton}.



The modules do not seem to bring any practical upside because the regret is almost perfectly linear in the number of modules.

That's all folks!