Dynamic resource allocation problems in communication networks:

Weakly Coupled Markov decision processes

Alexandre Reiffers-Masson

Equipe Maths&Net, IMT Atlantique, CS department
LabSTICC (UMR CNRS 6285)

July 12, 2023

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Lab-STICC

Example: Load balancing and service rate planning in parallel queue networks

- **Scenario:** N queues are processing jobs. The rate of each queue can be controlled. Moreover, the scheduler can also decide to which queue a job can be sent.

Example: Load balancing and service rate planning in parallel queue networks

- **Scenario:** N queues are processing jobs. The rate of each queue can be controlled. Moreover, the scheduler can also decide to which queue a job can be sent.
- **Challenge:** In order to minimize the total load on the system:
  - which queue should a job be *allocated* at each instant t?
  - which queue should see its *service rate* increased or decreased at each instant t ?

Arrival Process

We consider that at every time instant αN new jobs arrives in the system with probability $p \in (0, 1)$. Let $T_n \in \mathbb{N}_+$ be the arrival time of the n-th batch new jobs.

Note that:

$$\mathbb{P}(T_n - T_{n-1} = \tau) = (1 - p)^{\tau - 1} p, \ \forall \tau \geq 1, \ \forall n \in \mathbb{N}_+.$$

Dynamic of the queue

The length of the k-th queue, denoted by $S_k(T_{n+1})$ at instant T_{n+1} is given by:

$$S_k(T_{n+1}) = S_k(T_n) - D_k(T_{n+1} - T_n) + I\{S_k(T_n) < K\}A_k(T_n)$$

where:

- K is the finite buffer size of a queue;
- $D_k(T_n)$ the number of process jobs between T_n and T_{n+1}. We assume that the probability that a job is processed during one-time unit is equal to $B_k(t) \in \{\underline{b}, \bar{b}\}$. We assume that between two arrivals $B_k(t)$ is constant for all t and k;
- $A_k(T_n) \in \{0, 1\}$ is equal to one if one job from n-th batch is sent to the queue k.

Transition Probability

From that fact that the arrival are i.d.d. and the departure only depends on the inter-arrival time, we can rewrite the dynamic of the queue:

$$S_k(t+1) = S_k(t) - D_k(\tau) + I\{S_k(t) < K\}A_k(t).$$

We have the following lemma:

Lemma

For $s + a < k$, we have that the probability $\mathbb{P}(S_k(t+1) = s' | S_k(t+1) = s,\ A_k(t) = a,\ B_k(t) = b)$ is equal to

$$\sum_{\tau=1}^{+\infty} (1-p)^{\tau-1} p I_{s' < \min\{\tau, s+a\}} \binom{\tau}{s'} b^{\tau - s'} (1-b)^{s'}.$$

Cost functions and constraints

Costs: We will assume that there are two instantaneous costs:

- *Energy cost:* $\sum_k C_s(S_k(t)) + \sum_k C_q(B_k(t))$, where $C_s(\cdot)$ and $C_q(\cdot)$ are convex increasing.
- *Job rejection cost:* $-\gamma \sum_k A_k(t)$, with $\gamma > 0$. This cost implies that we prefer to send jobs.

Constraints: We will also assume that there are two instantaneous constraints:

$$\sum_k A_k(t) \leq \alpha N, \quad (1)$$

$$\sum_k B_k(t) \leq \beta N. \quad (2)$$

Motivation

Weakly Coupled Markov decision processes

Construction of LP-Admissible Policy

Multistage Convex Stochastic Optimization Problems

Weakly Coupled Markov decision processes

A **Weakly Coupled Markov decision processes** is composed of N statistically equivalent MDPs where:

Weakly Coupled Markov decision processes

A **Weakly Coupled Markov decision processes** is composed of N statistically equivalent MDPs where:

- $S_k(t) \in \mathcal{S}$ is the state of the arm k at the discrete decision time $t \in \{0, \cdots, T\}$,

Weakly Coupled Markov decision processes

A **Weakly Coupled Markov decision processes** is composed of N statistically equivalent MDPs where:

- $S_k(t) \in \mathcal{S}$ is the state of the arm k at the discrete decision time $t \in \{0, \cdots, T\}$,
- $A_k(t) \in \mathcal{A}$ is the action taken by the decision maker at the discrete decision time $t \in \{0, \cdots, T\}$.

Weakly Coupled Markov decision processes

A **Weakly Coupled Markov decision processes** is composed of N statistically equivalent MDPs where:

- $S_k(t) \in \mathcal{S}$ is the state of the arm k at the discrete decision time $t \in \{0, \cdots, T\}$,
- $A_k(t) \in \mathcal{A}$ is the action taken by the decision maker at the discrete decision time $t \in \{0, \cdots, T\}$.
- We assume that the decision-maker has to respect the following resource allocation constraints:

$$\sum_k D_l(S_k(t), A_k(t)) \leq N\alpha_l, \ \forall l = 1, \dots, L$$

Weakly Coupled Markov decision processes

For each time-step $t = 0, \ldots, T-1$*:*

Weakly Coupled Markov decision processes

For each time-step $t = 0, \dots, T-1$*:*

1. The decision-maker gets full knowledge of the current system state $S(t) := [S_1(t), \dots, S_N(t)] \in \mathcal{S}^N$;

Weakly Coupled Markov decision processes

For each time-step $t = 0, \ldots, T-1$:

1. The decision-maker gets full knowledge of the current system state $S(t) := [S_1(t), \ldots, S_N(t)] \in \mathcal{S}^N$;
2. Once $S(t)$ has been observed, the decision-maker chooses a control $A(t) := [A_1(t), \ldots, A_N(t)] \in \mathcal{A}^N$, such that:

$$\sum_k D_l(S_k(t), A_k(t)) \leq N\alpha_l, \ \forall l = 1, \ldots, L$$

Weakly Coupled Markov decision processes

For each time-step $t = 0, \dots, T-1$:

1. The decision-maker gets full knowledge of the current system state $S(t) := [S_1(t), \dots, S_N(t)] \in \mathcal{S}^N$;
2. Once $S(t)$ has been observed, the decision-maker chooses a control $A(t) := [A_1(t), \dots, A_N(t)] \in \mathcal{A}^N$, such that:

$$\sum_k D_l(S_k(t), A_k(t)) \leq N\alpha_l, \; \forall l = 1, \dots, L$$

3. The decision-maker collects the reward $\sum_k r_{S_k(t)}^{A_k(t)}$;

Weakly Coupled Markov decision processes

For each time-step $t = 0, \dots, T-1$:

1. The decision-maker gets full knowledge of the current system state $S(t) := [S_1(t), \dots, S_N(t)] \in \mathcal{S}^N$;
2. Once $S(t)$ has been observed, the decision-maker chooses a control $A(t) := [A_1(t), \dots, A_N(t)] \in \mathcal{A}^N$, such that:

$$\sum_k D_l(S_k(t), A_k(t)) \leq N\alpha_l, \ \forall l = 1, \dots, L$$

3. The decision-maker collects the reward $\sum_k r_{S_k(t)}^{A_k(t)}$;
4. For every k, the arm k evolves to $S_k(t+1) = s'$ with probability $p_{S_k(t),s'}^{A_k(t)}$.

Weakly Coupled Markov decision processes

For each time-step $t = 0, \ldots, T-1$:

1. The decision-maker gets full knowledge of the current system state $S(t) := [S_1(t), \ldots, S_N(t)] \in \mathcal{S}^N$;
2. Once $S(t)$ has been observed, the decision-maker chooses a control $A(t) := [A_1(t), \ldots, A_N(t)] \in \mathcal{A}^N$, such that:

$$\sum_k D_l(S_k(t), A_k(t)) \leq N\alpha_l, \; \forall l = 1, \ldots, L$$

3. The decision-maker collects the reward $\sum_k r_{S_k(t)}^{A_k(t)}$;
4. For every k, the arm k evolves to $S_k(t+1) = s'$ with probability $p_{S_k(t),s'}^{A_k(t)}$.

Objective: Maximize the expected total sum of rewards over the T time-steps.

Weakly Coupled Markov decision processes

For each time-step $t = 0, \ldots, T-1$*:*

1. The decision-maker gets full knowledge of the current system state $S(t) := [S_1(t), \ldots, S_N(t)] \in \mathcal{S}^N$;
2. Once $S(t)$ has been observed, the decision-maker chooses a control $A(t) := [A_1(t), \ldots, A_N(t)] \in \mathcal{A}^N$, such that:

$$\sum_k D_l(S_k(t), A_k(t)) \leq N\alpha_l, \ \forall l = 1, \ldots, L$$

3. The decision-maker collects the reward $\sum_k r_{S_k(t)}^{A_k(t)}$;
4. For every k, the arm k evolves to $S_k(t+1) = s'$ with probability $p_{S_k(t),s'}^{A_k(t)}$.

Objective: Maximize the expected total sum of rewards over the T time-steps.

Discussion with respect to the constraints

We assume that all terms in $D_l(s, a)$ and α_l are non-negative numbers, and that $D(s, 0) = 0$.

This is a natural assumption under the resource allocation context in which $a = 0$ corresponds to a passive action that consumes no resources.

Implication: The later also implies that our resource constraint problem has at least a feasible solution by always choosing the passive action.

Mathematical Formulation

$$\min_{\pi} \quad \mathbb{E} \sum_{t=0}^{T-1} \sum_{s,a} r_s^a Y_{a,s}^{(N)}(t) := V_{opt}^{(N)}(m(0), T) \tag{3a}$$

s.t. Arms follow the Markovian evolution generated by $\Pi_n p^{a_n}_{s_n, s'_n}$, (3b)

$$\sum_a Y_{a,s}^{(N)}(t) = M_s^{(N)}(t),\ \forall t \in [[0, T-1]],\ \forall s \in \mathcal{S}, \tag{3c}$$

$$\sum_s D_l(s,a) Y_{s,a}^{(N)}(t) \leq \alpha_l\ \forall t \in [[0, T-1]],, \tag{3d}$$

$$M_s^{(N)}(0) = m_s(0),\ \forall s \in \mathcal{S}, \tag{3e}$$

where $m_s(0) = \frac{1}{N} \sum_{k=1}^{N} I\{S_k(0) = s\}$, for all $s \in \mathcal{S}$.

Difficulty

The key difficulty of Weakly Coupled Markov decision processes is coming from:

$$\sum_{s} D_l(s,a) Y_{s,a}^{(N)}(t) \leq \alpha_l \; \forall t \in [[0, T-1]],$$

which couples all the arms together.

Challenge of the day:
How to design an efficient heuristic to solve such problem?
A different one that the projection policy.

Outline of the approach

1. **Relaxation:** Classical approach is to relax this constraint and consider a problem where this constraint has to be satisfied only in expectation:

Outline of the approach

1. **Relaxation:** Classical approach is to relax this constraint and consider a problem where this constraint has to be satisfied only in expectation:

$$\sum_s D_l(s,a)\mathbb{E}[Y_{s,a}^{(N)}(t)] \leq \alpha_l \;\forall t \in [[0, T-1]],$$

Outline of the approach

1. **Relaxation:** Classical approach is to relax this constraint and consider a problem where this constraint has to be satisfied only in expectation:

$$\sum_s D_l(s,a)\mathbb{E}[Y_{s,a}^{(N)}(t)] \leq \alpha_l \ \forall t \in [[0, T-1]],$$

2. **Interpolation:** Construct a sequence of decision rules $\pi_t : \Delta^d \to \Delta^{2d}$ which is optimal for the relaxed problem.

Relaxed problem

$$\min_{\pi} \quad \mathbb{E}[\sum_{t=0}^{T-1} \sum_{s,a} r_s^a Y_{a,s}^{(N)}(t)] =: V_{rel}^{(N)}(m(0), T) \tag{4a}$$

$$\text{s.t.} \quad \text{Arms follow the Markovian evolution generated by } \Pi_n p_{s_n, s'_n}^{a_n}, \tag{4b}$$

$$\sum_a Y_{a,s}^{(N)}(t) = M_s^{(N)}(t), \ \forall t \in [[0, T-1]], \ \forall s \in \mathcal{S}, \tag{4c}$$

$$\sum_s D_l(s,a)\mathbb{E}[Y_{s,a}^{(N)}(t)] \leq \alpha_l \ \forall t \in [[0, T-1]], \ \forall l, \tag{4d}$$

$$M_s^{(N)}(0) = m_s(0), \ \forall s \in \mathcal{S}, \tag{4e}$$

LP formulation

Let us define the following LP problem:

$$\begin{aligned}
\min_{y \geq 0} \quad & \sum_{t=0}^{T-1} \sum_{s,a} r_s^a y_{s,a}(t) =: V_{LP}(m(0), T) \\
\text{s.t.} \quad & \sum_{a} y_{s,a}(t) = m_s(t),\ \forall t \in [[0, T-1]],\ \forall s \in \mathcal{S}, \\
& m_s(t) = \sum_{s'} \sum_{a} y_{s',a}(t-1) p_{s',s}^a \ \forall t \in [[1, T-1]],\ \forall s \in \mathcal{S}, \\
& \sum_{s} D_l(s,a) y_{s,a}(t) \leq \alpha_l \ \forall t \in [[0, T-1]],\ \forall l, \\
& m_s(0) = m^0,\ \forall s \in \mathcal{S}
\end{aligned} \tag{5}$$

LP formulation

Let us define the following LP problem:

$$\begin{aligned}
\min_{y \geq 0} \quad & \sum_{t=0}^{T-1} \sum_{s,a} r_s^a y_{s,a}(t) =: V_{LP}(m(0), T) \\
\text{s.t.} \quad & \sum_{a} y_{s,a}(t) = m_s(t),\ \forall t \in [[0, T-1]],\ \forall s \in \mathcal{S}, \\
& m_s(t) = \sum_{s'} \sum_{a} y_{s',a}(t-1) p_{s',s}^a\ \forall t \in [[1, T-1]],\ \forall s \in \mathcal{S}, \\
& \sum_{s} D_l(s,a) y_{s,a}(t) \leq \alpha_l\ \forall t \in [[0, T-1]],\ \forall l, \\
& m_s(0) = m^0,\ \forall s \in \mathcal{S}
\end{aligned} \tag{5}$$

We denote by $y^* := [[[y^*_{s,a}(t)]]]_{s,a,t}$ the optimal solution of (6) and we also define $m^* := [[m_s(t) := \sum_a y^*_{s,a}(t)]]_{s,t}$.

Feasible control

We define the set of feasible control at time t by:

$$\mathcal{Y}(M^{(N)}(t)) := \Big\{ y \in \mathbb{R}_+^{2S} | \sum_a y_{s,a} = M_s^{(N)}(t) \ \forall s \in \mathcal{S};$$
$$\sum_s \sum_a D_l(s,a) y_{s,a} \leq \alpha_l \Big\}$$

Feasible control

We define the set of feasible control at time t by:

$$\mathcal{Y}(M^{(N)}(t)) := \Big\{ y \in \mathbb{R}_+^{2S} | \sum_a y_{s,a} = M_s^{(N)}(t) \; \forall s \in \mathcal{S};$$

$$\sum_s \sum_a D_l(s,a) y_{s,a} \leq \alpha_l \Big\}$$

Some observations:

Feasible control

We define the set of feasible control at time t by:

$$\mathcal{Y}(M^{(N)}(t)) := \Bigg\{ y \in \mathbb{R}_+^{2S} | \sum_a y_{s,a} = M_s^{(N)}(t) \ \forall s \in \mathcal{S};$$
$$\sum_s \sum_a D_l(s,a) y_{s,a} \leq \alpha_l \Bigg\}$$

Some observations:

1. In general, note that $y^*(t)$ are not an integers;
2. In general $y^*(t) \notin \mathcal{Y}(M^{(N)}(t))$;

Feasible control

We define the set of feasible control at time t by:

$$\mathcal{Y}(M^{(N)}(t)) := \Big\{ y \in \mathbb{R}_+^{2S} | \sum_a y_{s,a} = M_s^{(N)}(t) \ \forall s \in \mathcal{S};$$
$$\sum_s \sum_a D_l(s,a) y_{s,a} \leq \alpha_l \Big\}$$

Some observations:

1. In general, note that $y^*(t)$ are not an integers;
2. In general $y^*(t) \notin \mathcal{Y}(M^{(N)}(t))$;
3. In general $y^*(t) \in \mathcal{Y}(m^*(t))$.

Feasible control

We define the set of feasible control at time t by:

$$\mathcal{Y}(M^{(N)}(t)) := \Big\{ y \in \mathbb{R}_+^{2S} | \sum_a y_{s,a} = M_s^{(N)}(t) \; \forall s \in \mathcal{S};$$
$$\sum_s \sum_a D_l(s,a) y_{s,a} \leq \alpha_l \Big\}$$

Some observations:

1. In general, note that $y^*(t)$ are not an integers;
2. In general $y^*(t) \notin \mathcal{Y}(M^{(N)}(t))$;
3. In general $y^*(t) \in \mathcal{Y}(m^*(t))$.

Resolving policy

We redefine the following LP:

$$\begin{aligned}
\min_{y\geq 0} \quad & \sum_{t=0}^{T-t-1} \sum_{s,a} r_s^a y_{s,a}(t) =: V_{LP}(m(0), \textbf{T-t}) \\
\text{s.t.} \quad & \sum_a y_{s,a}(t) = m_s(t),\ \forall t \in [[0, \textbf{T-t-1}]],\ \forall s \in \mathcal{S}, \\
& m_s(t) = \sum_{s'} \sum_a y_{s',a}(t-1) p^a_{s',s}\ \forall t \in [[1, \textbf{T-t-1}]],\ \forall s \in \mathcal{S}, \\
& \sum_s y_{s,1}(t) \leq \alpha,\ \forall t \in [[0, \textbf{T-t-1}]],, \\
& m_s(0) = m^0,\ \forall s \in \mathcal{S}
\end{aligned} \tag{6}$$

Resolving policy

We redefine the following LP:

$$
\begin{aligned}
\min_{y \geq 0} \quad & \sum_{t=0}^{T-t-1} \sum_{s,a} r_s^a y_{s,a}(t) =: V_{LP}(m(0), \textbf{T-t}) \\
\text{s.t.} \quad & \sum_a y_{s,a}(t) = m_s(t),\ \forall t \in [[0, \textbf{T-t-1}]],\ \forall s \in \mathcal{S}, \\
& m_s(t) = \sum_{s'} \sum_a y_{s',a}(t-1) p_{s',s}^a\ \forall t \in [[1, \textbf{T-t-1}]],\ \forall s \in \mathcal{S}, \\
& \sum_s y_{s,1}(t) \leq \alpha,\ \forall t \in [[0, \textbf{T-t-1}]],, \\
& m_s(0) = m^0,\ \forall s \in \mathcal{S}
\end{aligned} \tag{6}
$$

The solution of this LP is denoted by

$$y^{Res}(m(0), T-t) = [y_{t'}^{Res}(m(0), T-t)]_{0 \leq t' \leq T-t-1}.$$

Algorithm to solve the LP

What could be a possible algorithm to solve this LP?

Solution 1: Simplex or Convex optimisation?

Solution 2: Dynamic programming. Observe that:

$$V_{LP}(m, T-t) = \min_{y \in \mathcal{Y}(m)} \sum_{s,a} r_s^a y_{s,a} + V_{LP}(\phi(m,y), T-t-1),$$

where $\phi_s(m,y) = \sum_{s'} \sum_a y_{s',a} p_{s',s}^a$ for all s.

Resolving operator

We define the following operator:

$$\pi_t^{Res}(M^{(N)}) := y_0^{Res}(M^{(N)}, T - t). \qquad (7)$$

Resolving operator

We define the following operator:

$$\pi_t^{Res}(M^{(N)}) := y_0^{Res}(M^{(N)}, T-t). \tag{7}$$

Note that:

Resolving operator

We define the following operator:

$$\pi_t^{Res}(M^{(N)}) := y_0^{Res}(M^{(N)}, T-t). \tag{7}$$

Note that:

1. In general, note that $\pi_t^{Res}(M^{(N)})$ are not an integers;

Resolving operator

We define the following operator:

$$\pi_t^{Res}(M^{(N)}) := y_0^{Res}(M^{(N)}, T - t). \tag{7}$$

Note that:

1. In general, note that $\pi_t^{Res}(M^{(N)})$ are not an integers;
2. $\pi_t^{Res}(M^{(N)}) \in \mathcal{Y}(M^{(N)}(t))$;

Resolving operator

We define the following operator:

$$\pi_t^{Res}(M^{(N)}) := y_0^{Res}(M^{(N)}, T-t). \tag{7}$$

Note that:

1. In general, note that $\pi_t^{Res}(M^{(N)})$ are not an integers;
2. $\pi_t^{Res}(M^{(N)}) \in \mathcal{Y}(M^{(N)}(t))$;
3. $y^*(t) = pi_t^{Res}(m^*(t))$. (P-Admissible Policy)

Algorithm

Resolving Policy

- **Input:** Initial system configuration vector $m(0)$ and time horizon T.
- **Set** $\hat{M}$:= m(0);
- **For** $t = 0, 2, \ldots, T-1$ **do:**
  1. **Compute** $y^{Res}(\hat{M}, T-t)$; Set $\hat{y}(t) = y_0^{Res}(\hat{M}, T-t)$
  2. *Rounding step:* For all $s \in \mathcal{S}$, set:

$$\hat{Y}_{s,a}^{(N)}(t) = \begin{cases} N^{-1}\lfloor N\hat{y}_{s,1}(t) \rfloor & \text{if } a = 1, \\ \hat{M}_s - N^{-1}\lfloor N\hat{y}_{s,1}(t) \rfloor & \text{otherwise.} \end{cases}$$

  3. Use control $\hat{Y}^{(N)}$ to advance to the next time-step ;
  4. Set $\hat{M}$:= current empirical distribution;

Certainty equivalent control

Our policy is inspired from the *certainty equivalent control* (CEC).

Certainty equivalent control

Our policy is inspired from the *certainty equivalent control* (CEC).

Principle of the CEC

Sub-optimal control that applies at each stage the control that would be optimal if some or all of the uncertain quantities were fixed at their expected values.

Challenges

What are the remaining challenges:

1. How to handle asymmetric arms?

Challenges

What are the remaining challenges:

1. How to handle asymmetric arms?
2. How to handle average cost ($\lim_{T\to+\infty} \mathbb{E}[\frac{1}{T}\sum_{t=0}^{T} Y_{s,a}]$ instead of $\mathbb{E}[\frac{1}{T}\sum_{t=0}^{T} Y_{s,a}]$.)

Challenges

What are the remaining challenges:

1. How to handle asymmetric arms?
2. How to handle average cost ($\lim_{T\to+\infty} \mathbb{E}[\frac{1}{T}\sum_{t=0}^{T} Y_{s,a}]$ instead of $\mathbb{E}[\frac{1}{T}\sum_{t=0}^{T} Y_{s,a}]$.)
3. How to handle continuous state?

Challenges

What are the remaining challenges:

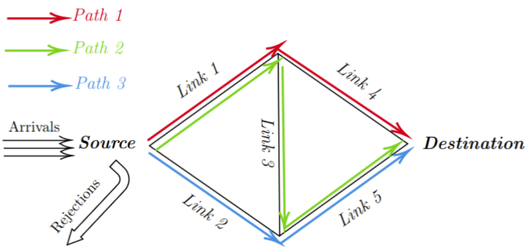
1. How to handle asymmetric arms?
2. How to handle average cost ($\lim_{T\to+\infty} \mathbb{E}[\frac{1}{T}\sum_{t=0}^{T} Y_{s,a}]$ instead of $\mathbb{E}[\frac{1}{T}\sum_{t=0}^{T} Y_{s,a}]$.)
3. How to handle continuous state?
4. Efficient algorithm to solve the LP when the parameters are unknown.

Example: Access control and Utility Maximization

- **Scenario:** How bandwidth should I allocate on each path?

Example: Access control and Utility Maximization

- **Scenario:** How bandwidth should I allocate on each path?
- **Challenge:** Maximize the total amount of Bandwidth sent into the network?



Demand arrival and allocation

- A path is a sequence of consecutive directed links that connect the source to the destination, enumerated by $1 \leq p \leq 3$.

Demand arrival and allocation

- A path is a sequence of consecutive directed links that connect the source to the destination, enumerated by $1 \leq p \leq 3$.
- We assume that $(W_1(t), W_2(t), W_3(t))$ are the arrivals of new demands of bandwidth on each path at time-step t.

Demand arrival and allocation

- A path is a sequence of consecutive directed links that connect the source to the destination, enumerated by $1 \leq p \leq 3$.
- We assume that $(W_1(t), W_2(t), W_3(t))$ are the arrivals of new demands of bandwidth on each path at time-step t.
- The control is actual allocation $(U_1(t), U_2(t), U_3(t))$ of bandwidth among each path.

Demand arrival and allocation

- A path is a sequence of consecutive directed links that connect the source to the destination, enumerated by $1 \le p \le 3$.
- We assume that $(W_1(t), W_2(t), W_3(t))$ are the arrivals of new demands of bandwidth on each path at time-step t.
- The control is actual allocation $(U_1(t), U_2(t), U_3(t))$ of bandwidth among each path.
- We suppose that the demand is elastic, so that the non-satisfied demand incurs no cost.

Demand arrival and allocation

- A path is a sequence of consecutive directed links that connect the source to the destination, enumerated by $1 \leq p \leq 3$.
- We assume that $(W_1(t), W_2(t), W_3(t))$ are the arrivals of new demands of bandwidth on each path at time-step t.
- The control is actual allocation $(U_1(t), U_2(t), U_3(t))$ of bandwidth among each path.
- We suppose that the demand is elastic, so that the non-satisfied demand incurs no cost.
- A first set of constraints on the model can then be expressed as for all $t \in \{1, \ldots, T\}$.

Demand arrival and allocation

- A path is a sequence of consecutive directed links that connect the source to the destination, enumerated by $1 \leq p \leq 3$.
- We assume that $(W_1(t), W_2(t), W_3(t))$ are the arrivals of new demands of bandwidth on each path at time-step t.
- The control is actual allocation $(U_1(t), U_2(t), U_3(t))$ of bandwidth among each path.
- We suppose that the demand is elastic, so that the non-satisfied demand incurs no cost.
- A first set of constraints on the model can then be expressed as for all $t \in \{1, \ldots, T\}$.

$$0 \leq U_1(t) \leq W_1(t) \quad 0 \leq U_2(t) \leq W_2(t) \quad 0 \leq U_3(t) \leq W_3(t)$$

Bandwidth occupation

- $(X_1(t), X_2(t), X_3(t))$ the bandwidth occupation of the three routing paths just arriving at time-step t.

Bandwidth occupation

- $(X_1(t), X_2(t), X_3(t))$ the bandwidth occupation of the three routing paths just arriving at time-step t.
- We assume that the evolution of $X_p(t)$ is given by:

Bandwidth occupation

- $(X_1(t), X_2(t), X_3(t))$ the bandwidth occupation of the three routing paths just arriving at time-step t.
- We assume that the evolution of $X_p(t)$ is given by:

$$X_p(t+1) = (X_p(t) + U_p(t)) \cdot q_p + \epsilon_p(t+1), \quad \text{for } 1 \le p \le 3 \text{ and } 1 \le t \le T,$$

where $\epsilon_p(t+1)$ is a r.v. with mean zero and support,

$$[-(X_p(t) + U_p(t)) \cdot q_p, (X_p(t) + U_p(t)) \cdot (1 - q_p)].$$

Capacity constraints

- Each directed edge of the graph is called a *link*, enumerated by $1 \leq l \leq 5$. Each link has a maximum bandwidth capacity, denoted as $c_l > 0$.

Capacity constraints

- Each directed edge of the graph is called a *link*, enumerated by $1 \leq l \leq 5$. Each link has a maximum bandwidth capacity, denoted as $c_l > 0$.
- The constraints that each link should satisfy are given by for all $1 \leq t \leq T$:

$$\begin{aligned}
Y_1(t) &:= U_1(t) + X_1(t) + U_2(t) + X_2(t) \leq c_1 \\
Y_2(t) &:= U_3(t) + X_3(t) \leq c_2 \\
Y_3(t) &:= U_2(t) + X_2(t) \leq c_3 \\
Y_4(t) &:= U_1(t) + X_1(t) \leq c_4 \\
Y_5(t) &:= U_2(t) + X_2(t) + U_3(t) + X_3(t) \leq c_5.
\end{aligned}$$

Utility funcion

The decision-maker aims to maximize the following α-fairness utility (with $\alpha > 0$)

$$\mathbb{E}\sum_{t=1}^{T}\sum_{p=1}^{3}\frac{(X_p(t)+U_p(t))^{1-\alpha}}{1-\alpha}$$

gained by allocating and rejecting the bandwidth demands over a finite horizon T, while respecting the dynamics and constraints described in the previous slides.

Constrained Finite Horizon Stochastic Optimization Problem

For each time-step $t = 1, \ldots, T$:

1. The decision-maker gets the current system state $X(t)$;

Constrained Finite Horizon Stochastic Optimization Problem

For each time-step $t = 1, \dots, T$:

1. The decision-maker gets the current system state $X(t)$;
2. (The environment) independently draws $W(t) \sim f(w)$;

Constrained Finite Horizon Stochastic Optimization Problem

For each time-step $t = 1, \ldots, T$:

1. The decision-maker gets the current system state $X(t)$;
2. (The environment) independently draws $W(t) \sim f(w)$;
3. The decision-maker chooses a control $U(t)$ that satisfies constraints

$$g_{t,i}(X(t), W(t), U(t)) \leq 0,\ h_{t,j}(X(t), W(t), U(t)) = 0,$$

Constrained Finite Horizon Stochastic Optimization Problem

For each time-step $t = 1, \ldots, T$:

1. The decision-maker gets the current system state $X(t)$;
2. (The environment) independently draws $W(t) \sim f(w)$;
3. The decision-maker chooses a control $U(t)$ that satisfies constraints

$$g_{t,i}(X(t), W(t), U(t)) \leq 0,\ h_{t,j}(X(t), W(t), U(t)) = 0,$$

4. The decision-maker collects a reward $R_t(X(t), W(t), U(t))$
5. The system evolves to the next state $(t+1)$ such that $X(t+1) \sim \phi\,(X(t), W(t), U(t)) + \epsilon(X(t), W(t), U(t))$.

Constrained Finite Horizon Stochastic Optimization Problem

For each time-step $t = 1, \dots, T$:

1. The decision-maker gets the current system state $X(t)$;
2. (The environment) independently draws $W(t) \sim f(w)$;
3. The decision-maker chooses a control $U(t)$ that satisfies constraints

$$g_{t,i}(X(t), W(t), U(t)) \leq 0, \; h_{t,j}(X(t), W(t), U(t)) = 0,$$

4. The decision-maker collects a reward $R_t(X(t), W(t), U(t))$
5. The system evolves to the next state $(t+1)$ such that $X(t+1) \sim \phi(X(t), W(t), U(t)) + \epsilon(X(t), W(t), U(t))$.

Objective: Maximize the expected total sum of rewards over the T time-steps.

Mathematical model

$$\max_{[1,T]} \quad \mathbb{E}\left[\sum_{t=1}^{T} R_t\left(X(t), W(t), U(t)\right)\right] =: V_{\text{opt}}(x(1), T) \tag{8a}$$

$$\text{s.t.} \quad X(1) = x(1) \; a.s., \tag{8b}$$

$$g_{t,i}(X(t), W(t), U(t)) \leq 0, \; \forall t \text{ and } \forall j, \tag{8c}$$

$$h_{t,j}(X(t), W(t), U(t)) = 0, \; \forall t \text{ and } \forall j, \tag{8d}$$

$$X(t+1) = \phi\left(X(t), W(t), U(t)\right) + \epsilon(X(t), W(t), U(t)), \; \forall t \tag{8e}$$

The Certainty Equivalent Control (CEC)

Based on CEC, we apply the following relaxation to the original problem: define $\mathbb{E}X(t) := x(t)$ and $\mathbb{E}U(t) = u(t)$ where the expectation is taken with the whole trajectory. By Jensen's inequality, we have

$$\mathbb{E}[\sum_{t=1}^{T} R_t \left(X(t), W(t), U(t)\right)] \leq \sum_{t=1}^{T} R_t \left(x(t), \mathbb{E}[w], u(t)\right). \tag{9}$$

$$\mathbb{E}[g_{t,i}(X(t), W(t), U(t))] \geq g_{t,i}(x(t), \mathbb{E}[w], u(t)),\ \forall t, i \tag{10}$$

$$\mathbb{E}[h_{t,j}(X(t), W(t), U(t))] = h_{t,j}(x(t), \mathbb{E}[w], u(t))\ \ \forall t, j \tag{11}$$

Relaxed mathematical program

All this consideration leads to the following relaxed mathematical program with decision variables $u(t)$:

$$\begin{aligned}
\max_{u[1,T]} \quad & \sum_{t=1}^{T} R_t\left(x(t), \overline{w}, u(t)\right) && \text{(12a)}\\
\text{s.t.} \quad & x(1) = x, && \text{(12b)}\\
& g_{t,i}(x(t), \overline{w}, u(t)) \leq 0,\ \forall t, i, && \text{(12c)}\\
& h_{t,j}(x(t), \overline{w}, u(t)) = 0,\ \forall t, i, && \text{(12d)}\\
& x(t+1) = \phi\left(x(t), \overline{w}, u(t)\right),\ \forall t, i && \text{(12e)}
\end{aligned}$$

Algorithms

You can apply the resolving algorithm or the projection policy in this case. You can even have a combination of both.

But we don't have the symmetry property. So our error will be this time controlled by the **variances** of the different variables.

Bibliography

- The proof of the main theorem and more advance theorem can be found here: Gast, Nicolas, Bruno Gaujal, and Chen Yan. "The LP-update policy for weakly coupled Markov decision processes." arXiv preprint arXiv:2211.01961 (2022).
- If you want to have a quick introduction to dynamic programming, please have a look to the lecture note of Nahum Shimkin: `https://webee.technion.ac.il/shimkin/LCS11/LCS11index.html`