1 Optimal scheduling for n homogeneous servers

Input:

- n: Number of homogeneous servers
- T: Number of time steps
- β: Power up costs
- $\lambda_1, \ldots, \lambda_{T-1} \in [0, n]$: Arrival rates

Requirements:

- Convex cost function f
- Power down costs are w.l.o.g. equal to 0.
- $\lambda_0 = \lambda_T = 0$
- All servers are powered down at $t = 0$ and $t = T$.

We construct a directed acyclic graph as follows:
$\forall t \in [T-1], 0 \leq i \leq m$ we add vertices (t, i) modelling the number of active servers at time t. Furthermore, we add vertices $(0, 0)$ and $(T, 0)$ for our initial and final state respectively.
In order to warrant that there are at least $\lceil \lambda_t \rceil$ active servers $\forall t \in [T-1]$, we define an auxiliary function which calculates the costs for handling an arrival rate λ with x active servers:

$$c(x, \lambda) \coloneqq \begin{cases} x * f(\lambda/x), & \text{if } \lambda \leq x \\ \infty, & \text{otherwise} \end{cases} \tag{1}$$

Then, $\forall t \in [T-2], i, j \in \{0, \ldots, m\}$ we add edges from (t, i) to $(t+1, j)$ with weight

$$d(i, j, \lambda_{t+1}) \coloneqq \underbrace{\beta * \min\{0, j - i\}}_{\text{power up costs}} + c(j, \lambda_{t+1}) \tag{2}$$

Finally, for $0 \leq i \leq m$ we add edges from $(0, 0)$ to $(1, i)$ with weight $d(0, i, \lambda_1)$ and from $(T-1, i)$ to $(T, 0)$ with weight $d(i, 0, \lambda_T) = 0$.

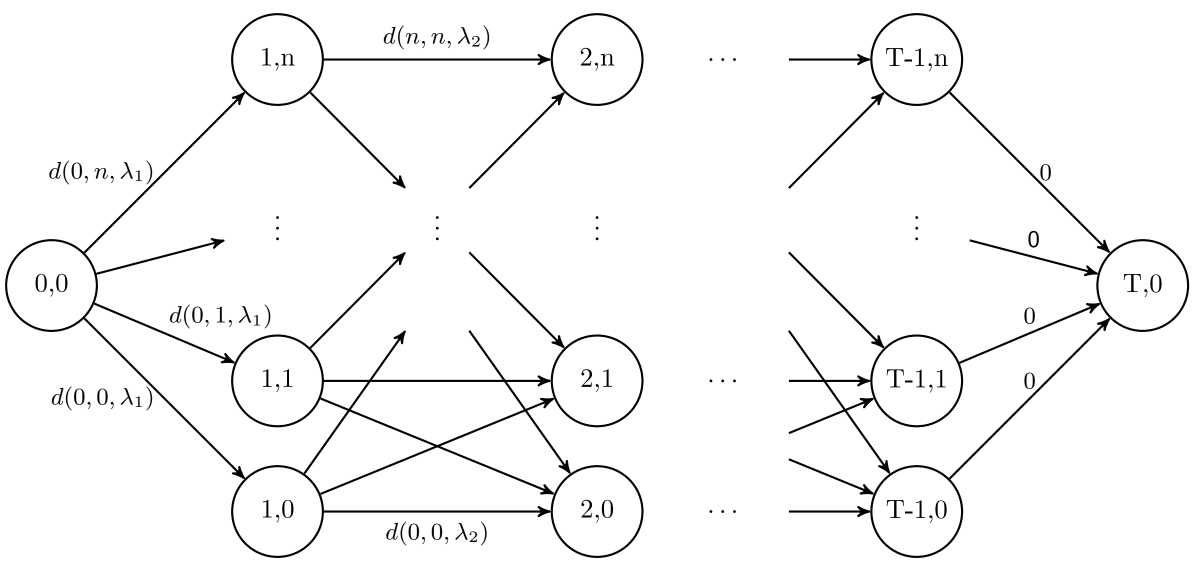

Figure 1: All edges from (t, i) to $(t+1, j)$ have weight $d(i, j, \lambda_{t+1})$

Algorithm 1 Calculate costs for n homogeneous servers

Require: Convex cost function f, $\lambda_0 = \lambda_T = 0$, $\forall t \in [T-1] : \lambda_t \in [0, n]$
1: **function** SCHEDULE($n, T, \beta, \lambda_1, \ldots, \lambda_{T-1}$)
2: **if** $T < 2$ **then**
3: **return** $(0, 0)$
4: **let** $p[2 \ldots T-1, 2]$ and $m[1 \ldots T-1, n]$ **be** new arrays
5: **for** $j \leftarrow 0$ to n **do**
6: $m[1, j] \leftarrow d(0, j, \lambda_1)$
7: **for** $t \leftarrow 1$ to $T-2$ **do**
8: **for** $j \leftarrow 0$ to n **do**
9: $opt \leftarrow \infty$
10: **for** $i \leftarrow 0$ to n **do**
11: $m[t+1, j] \leftarrow m[t, i] + d(i, j, \lambda_{t+1})$
12: **if** $m[t+1, j] < opt$ **then**
13: $opt \leftarrow m[t+1, j]$
14: $p[t+1, j] \leftarrow i$
15: **return** p and m

Algorithm 2 Extract schedule for n homogeneous servers

1: **function** EXTRACT(n, p, m, T)
2: **let** $x[0 \ldots T]$ **be** a new array
3: $x[0] \leftarrow x[T] \leftarrow 0$
4: $x[T-1] \leftarrow \underset{0 \leq i \leq n}{arg\ min}\{m[T-1, i]\}$
5: **for** $t \leftarrow T-2$ to 1 **do**
6: $x[t] \leftarrow p[t+1, x[t+1]]$
7: **return** x

1.1 Runtime analysis

Schedule: Loop 5,8 and 10 run $n+1$ times, loop 7 runs $T-2$ times
Extract: Loop 5 runs $T-2$ times, argmin 4 takes time $n+1$.
For $T, n \to \infty$ it holds:

$$\mathcal{O}(n+1+(T-2)*(n+1)^2+T-2+n+1) = \mathcal{O}(2*n+T+(T-2)*(n+1)^2) = \mathcal{O}(T*n^2) \quad (3)$$

1.2 Proof of correctness

First, we show that for x active servers and an arrival rate λ, the best method is to assign each server a load of λ/x:
$\forall x \in \mathbb{N}, \mu_i \in [0,1] : \sum_{i=1}^{x} \mu_i = 1 :$

$$\begin{aligned} f\Big(\frac{\lambda}{x}\Big) &= f\Big(\sum_{i=1}^{x} \frac{\mu_i * \lambda}{x}\Big) \overset{\text{Jensen's inequality}}{\leq} \sum_{i=1}^{x} \frac{1}{x} f(\mu_i * \lambda) \\ \Leftrightarrow \quad x * f\Big(\frac{\lambda}{x}\Big) &\leq \sum_{i=1}^{x} f(\mu_i * \lambda) \end{aligned} \quad (4)$$