Requirements:

- Convex cost function f
- Power down costs are w.l.o.g. equal to 0.
- $\lambda_0 = \lambda_T = 0$
- $\forall t \in [T-1] : \lambda_t \in [0, 2]$
- All servers are powered down at $t = 0$ and $t = T$.

Input:

- β: Power up costs.
- $\lambda_1 \dots \lambda_{T-1}$: Arrival rates

We construct a directed acyclic graph as follows:
For each timestep $t \in [T-1]$ we vertices $(t,0), (t,1)$ and $(t,2)$ 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 $\forall t \in [T-1]$ there are at least $\lceil \lambda_t \rceil$ active servers, we define an auxiliary function which calculates the costs for handling an arrival rate λ with x active servers:

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

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

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

Finally, $\forall i \in \{0, 1, 2\}$ 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 0.

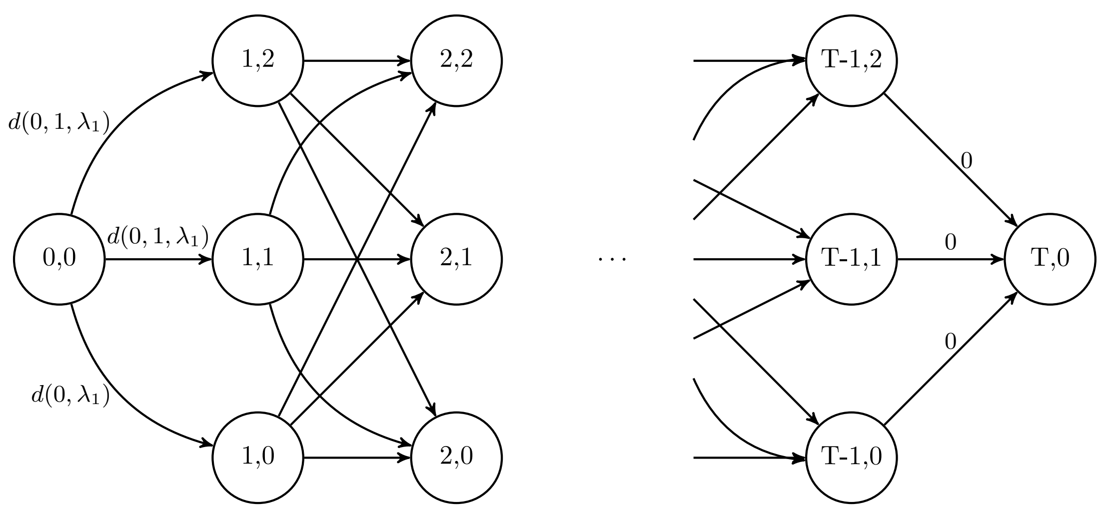

Figure 1: All edges from (t, i) to $(t+1, j)$ have weight $d(i, j, \lambda_{t+1})$

Algorithm 1 Optimal schedule for $m = 2$ homogeneous servers

```
Require: Convex cost function f, λ_0 = λ_T = 0, ∀t ∈ [T − 1] : λ_t ∈ [0, 2]
function SCHEDULE(T, β, λ_1, ..., λ_{T−1})
    if T < 2 then
        return
    let p[2 ... T − 1, 2] and m[1 ... T − 1, 2] be new arrays
    for j ← 0 to 2 do
        m[1, j] ← d(0, j, λ_1)
    for t ← 1 to T − 2 do
        for j ← 0 to 2 do
            opt ← ∞
            for i ← 0 to 2 do
                m[t + 1, j] ← m[t, i] + d(i, j, λ_{t+1})
                if m[t + 1, j] < opt then
                    opt ← m[t + 1, j]
                    p[t + 1, j] ← i
    return p and m
```

Algorithm 2 Extract schedule for m=2 homogeneous servers

```
function EXTRACT(p, m, T)
    let x[0 ... T] be a new array
    x[0] ← x[T] ← 0
    x[T − 1] ← arg min_{0≤i≤2} {m[T − 1, i]}
    for t ← T − 2 to 1 do
        x[t] ← p[t + 1, x[t + 1]]
    return x
```