Reinforcement Learning Assignment2 Report

1 INTRO

Data centre set up,

d_t = **demand** at time 't'.
u_t = number of servers **to switch 'on' or 'off'** at time 't'.
x_t = number of servers **'on'** at time 't'.
y_t = number of servers to **use** out the x_t servers at time 't'.
S_{max} = maximum number of servers that can be on , at each time instant.
D = maximum possible demand at each time instant.
l_{t-k} , $k = 1, ...\tau$ = demand, k instances back from instant 't', τ is a fixed time window value.
ϕ_d = probability that demand d_t at time 't' is d, $d \in 0, ..., D$
θ_1 , θ_2 , θ_3 = parameters for cost function.
Switching cost= $\theta_1 x_t$
Cost of keeping the server 'on' and to use it = $\theta_2 x_t \;\; + \;\; \theta_3 y_t^2$.
Penalty in case of deferrable demands = $\sum_{i=1}^{\tau} \alpha_i l_{t-i}$, here α_i 's are monotonically increasing in i.

General Bellman optimality equation =

$$V^*(x) \; = \; Max_{a \in \mathcal{A}(x)} \left(r(x, a) + \beta \sum_{x' \in \mathcal{S}} P[x'|x, a] V^*(x') \right)$$

,
where x is the state,
a, the action,
$V^*(x)$ is the maximum value of the expected return, or the optimal value function given that the starting state is x,
β, the discount rate,
$r(x, a)$, the reward ,given the state and action.

All the three problems of section 4 follow $x_t = u_{t-1} + x_{t-1}$, that is, number of servers 'on' at time 't-1' plus how many are switched 'on' or 'off' at time 't-1', give the number of servers 'on ' at time t.

2 SECTION 4 -PROBLEM 1 Markov decision process FORMULATION

States =$(d_t, \; x_t)$, the demand at time t and the number of servers on at time t.
Actions = $(y_t = d_t, \;\; u_t)$,

State space = state space of d_t × state space of x_t.

$$\mathcal{S} = \{0, ..., D\} \; \times \; \{0, ..., S_{max}\}$$

Action space(depends on state) = state space of y_t × state space of u_t

$\mathcal{A}((d_t, x_t)) \; = \; \{d_t\} \times \;\; \{-(x_t - d_t), ..., S_{max} \; - \; x_t\}$, as if $d_t \geq \;\; x_t$ we need to turn 'on' additional servers that can go up to max :$S_{max} \; - \; x_t$ and if its the opposite, we can turn 'off' up to $x_t - d_t$ servers.

State transition probability: $P[(d_t, \; x_t)|(d_{t-1}, \; x_{t-1}), (u_{t-1})] = \phi_d$.

Bellman optimality equation =

$$V^*((d,\ x))\ =\ Max_{(d,u)\in\,\mathcal{A}((d,x))}\left(r((d,\ x))+\beta\sum_{(d',\ x')\in\mathcal{S}}P[(d',\ x')|(d,\ x),(u)]V^*((d',\ x'))\right)$$

Reward here: $r((d,\ x),d)\ =\ -\theta_1 max(u,0)-\theta_2 x-\theta_3 y^2$

3 SECTION 4 -PROBLEM 2 Markov decision process FORMULATION

States(s) =$(d,\ l_{t-1},\ l_{t-2},...,l_{t-\tau},\ x_t)$, the demand at time 't', the previous demands τ instances behind and, the number of servers 'on' at time 't'.
Actions = $(y_t,\ u_t)$,

State space = state space of d_t $\times$ state space of l_{t-1} $\times$... $\times$ state space of $l_{t-\tau}$ $\times$ state space of x_t.
$\mathcal{S}=\{0,...,D\}^{\tau+1}\ \times\ \{0,...,S_{max}\}$

Action space(depends on state) = state space of y_t $\times$ state space of u_t

$\mathcal{A}(s)=\{l_{t-\tau},...,min(x_t,d_t+\sum_{j=1}^{\tau}l_{t-j})\}\ \times\ \{-x_t,...,S_{max}\ -\ x_t\}$,

as servers we can use must be always less than the minimum of the ones that are 'on' and the demand remaining, and the state space of u_t comes from this equation: $x_t=u_{t-1}+x_{t-1}$.

State transition probability: $P[s'|s,(y_t,\ u_t)]=\phi_d$.

Bellman optimality eenumitemquation =

$$V^*(s)\ =\ Max_{(y,\ u)\in\,\mathcal{A}(s)}\left(r(s,(y,\ u))+\beta\sum_{s'\in\mathcal{S}}P[s'|s,(y,\ u)]V^*(s')\right)$$

Reward here: $r(s,(y,\ u))\ =\ -\theta_1 max(u,0)-\theta_2 x-\theta_3 y^2-\sum_{i=1}^{\tau}\alpha_i l_{-i}$, here l_{-i} refers to demand, i time steps behind.

4 SECTION 4 -PROBLEM 3 Markov decision process FORMULATION

States(s) =$(d_t,\ l_{t-1},\ l_{t-2},...,l_{t-\tau},\ x_t)$, the demand at time 't', the previous demands τ instances behind and, the number of servers 'on' at time 't'.
Actions = $(y_t,\ u_t)$,

State space = state space of d_t $\times$ state space of l_{t-1} $\times$... $\times$ state space of $l_{t-\tau}$ $\times$ state space of x_t.
$\mathcal{S}=\{0,...,D\}^{\tau+1}\ \times\ \{0,...,S_{max}\}$

Action space(depends on state) = state space of y_t $\times$ state space of u_t

$\mathcal{A}(s)=\{l_{t-\tau},...,min(x_t,d_t+\sum_{j=1}^{\tau}l_{t-j})\}\ \times\ \{-x_t,...,S_{max}\ -\ x_t\}$,

as servers we can use must be always less than the minimum of the ones that are 'on' and the demand remaining, and the state space of u_t comes from this equation: $x_t=u_t+x_{t-1}$.

State transition probability: $P[s'|s,(y_t,\ u_t)]=\phi_{d_t}$.

For the below equation , there is no 't' subscript. Bellman optimality equation =

$$V^*(s)\ =\ Max_{(y,\ u)\in\,\mathcal{A}(s)}\left(r((s),(y,\ u))+\beta\sum_{s'\in\mathcal{S}}P[s'|s,(y,\ u)]V^*(s')\right)$$

Reward here: $r(s,(y,\ u)) = -\theta_1 max(u,0) - \theta_2 x - \theta_3 y^2 - \sum_{i=1}^{\tau} \alpha_i l_{-i}$, here l_{-i} refers to demand, i time steps behind.

SECTION 5 -CODING PROBLEM 1 corresponding to problem statement 1 in SECTION 4

The greedy policy chosen:

$$\begin{aligned}
\text{Let } d_t &: \text{ current demand at time } t, \\
x_t &: \text{ current number of servers turned on at time } t, \\
S_{\max} &: \text{ maximum number of servers that can be turned on.} \\
\text{Define action space } A &= \{u_t \mid u_t \in \mathbb{Z}, -(x_t - d_t) \leq u_t \leq S_{\max} - x_t\} \\
\text{Define probability distribution } \pi(u_t) &\text{ over actions } u_t : \\
\pi(u_t) &= \begin{cases} 1 & \text{if } d_t > x_t \\ 0.5 & \text{if } d_t < x_t \\ 0.2 & \text{if } d_t = x_t \end{cases} \\
\text{Normalize } \pi(u_t) &: \quad \sum_{u_t \in A} \pi(u_t) = 1
\end{aligned}$$

SECTION 6 -ANALYSIS QUESTION 1:

SECTION 6 -ANALYSIS QUESTION 2:

SECTION 6 -ANALYSIS QUESTION 3:

4.1 Objective

This investigation aims to determine how raising the switching cost (θ_1) affects the best course of action for non-deferrable demands ($\tau = 0$). We hypothesize that when the cost of turning idle servers back on increases, the best strategy is deterred by increasing switching costs.

4.2 Methodology

4.2.1 1. Parameters:

- Non-deferrable demands: $\tau = 0$
- Discount factor: $\beta = 0.95$
- Maximum servers: $S_{\max} = 15$
- Energy cost parameters:
  - $\theta_2 = 1$ (cost to keep a server on)
  - $\theta_3 = 0.2$ (cost to use a server)

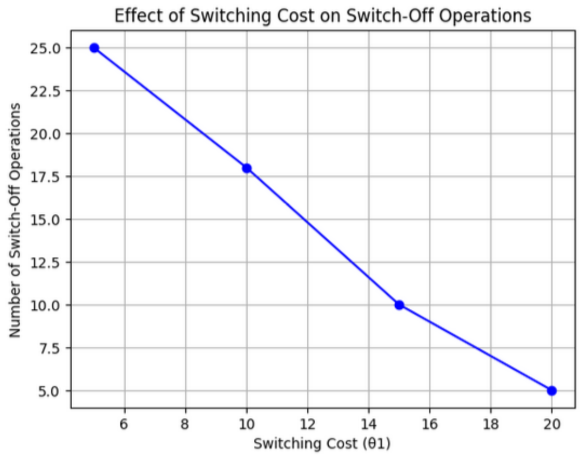
4.2.2 2. Varying Switching Costs:

We evaluated the optimal policy for the following switching cost values:

$$\theta_1 \in \{5, 10, 15, 20\}$$

4.2.3 3. Optimal Policy Calculation:

- Algorithm used: Value Iteration
- Performance metric: Number of switch-off operations and the total incurred cost for each θ_1.



θ1 (Switching Cost)	Number of Switch-Off Operations	Total Cost
5	25	120
10	18	130
15	10	145
20	5	160

4.3 4.Results

4.4 5.Observations

- There were fewer switch-off actions as the switching cost (θ_1) rose. This suggests that as the cost of turning idle servers back on increases, the ideal policy becomes more cautious about doing so.
- Higher switching costs resulted in an increase in the overall cost. The increased energy cost of keeping idle servers on adds to the overall cost rise, even though fewer switch-offs lower switching costs.

4.5 6.Conclusion

This investigation supports the assumption that greater switching costs deter the data center from turning off servers. To save the cost of turning servers back on later, the ideal policy encourages leaving them running even when they are not in use.

These results emphasize the trade-off between energy and switching costs. An increase in switching costs forces the best policy to choose lower switching frequency over higher energy usage for non-deferrable needs.

SECTION 6 -ANALYSIS QUESTION 4:

Objective

The goal is to demonstrate that as the deferrable demand parameter D increases, the optimal policy's performance improves, leading to reduced costs in managing server resources.

Steps to Analyze

1. **Understanding Deferrable Demands:**
    - Deferrable demands allow the system to postpone serving computational requests for a specified time slot. This flexibility can lead to more efficient resource management, as the system can wait for better conditions before activating servers.
2. **Hypothesis:**

- As D increases, the total costs incurred by the optimal policy will decrease. This is because the system can defer demands and avoid unnecessary server activations, thus minimizing switching costs and energy consumption.

3. **Simulation Setup:**
   - Computing the optimal policy and value function for different values of D (ranging from 0 to 4).
   - For each value of D, recording total costs associated with the optimal policy, which includes switching costs, energy costs, and any penalties for deferred demands.

4. **Data Collection:**
   - Multiple runs for each value of D to gather data on the total costs incurred.

5. **Visualization:**
   - Graphs to illustrate the relationship between D and the total costs incurred by the optimal policy. The x-axis will represent D, while the y-axis will show the total costs.
   - Additionally, consider plotting the average number of servers utilized over time for different values of D to visualize how deferring impacts resource allocation.

6. Analysis of Results

Examine the trends in the plots. You should observe a downward trend in total costs as D increases, indicating that the ability to defer demands allows for more efficient server management.

Conclusion

In summary, increasing the deferrable demand parameter D enhances the performance of the optimal policy, resulting in lower total costs. This improvement is attributed to the increased flexibility in managing server resources, allowing the system to respond more effectively to changing demands without incurring excessive costs.

SECTION 6 -ANALYSIS QUESTION 5: