SlateQ vs. Tabular Q-Learning

- In tabular Q-Learning:

$$Q^t(s, A) = Q^{t-1}(s, A) + \alpha \left(R(s, A, s') + \max_{A'} \left(\gamma Q^{t-1}(s', A') \right) - Q^{t-1}(s,$$

- Dimension of $Q(s, A)$: $K \times \binom{K}{N}$. Time-consuming for large K and N.
- SlateQ introduces $\bar{Q}(s, i)$ to avoid exhaustive exploration.
- Updated using:

$$\begin{aligned} \bar{Q}^t(s, i) &= \bar{Q}^{t-1}(s, i) \\ &\quad + \alpha \left(R(s, A, s') + \max_{A'} \left(\gamma Q^{t-1}(s', A') \right) - \bar{Q}^{t-1}(s, i) \right) \end{aligned}$$

Maximization Problem in SlateQ

- In our environment:

$$Q(s, A) = \sum_{i \in A} P(i|s, A)\bar{Q}(s, i)$$

- Revised update equation:

$$\bar{Q}^t(s, i) = \bar{Q}^{t-1}(s, i)$$
$$+ \alpha \left(R(s, i) + \max_{A'} \left(\gamma \sum_{j \in A'} P^{t-1}(j|i, A')\bar{Q}^{t-1}(i, j) \right) - \bar{Q}^{t-1}(\right.$$

- Maximization problem:

$$\max_A \sum_{i \in A} P(i|s, A)\bar{Q}(s, i) = \max_A \sum_{i \in A} \frac{1}{N}\bar{Q}(s, i)$$

Linear optimization:

$$\max_{\mathbf{x}} \sum_i x_i \frac{1}{N}\bar{Q}(s, i) \quad \text{s.t.} \quad x_i \in [0, 1],$$

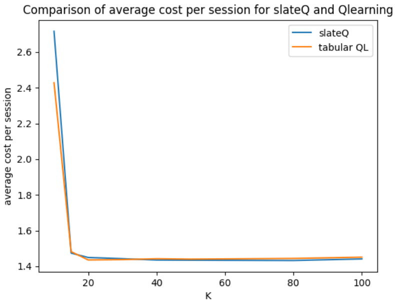
Evaluation Description and Optimal Average Cost

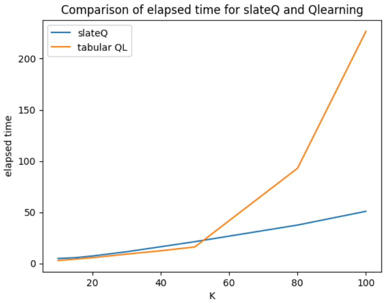
- Evaluate algorithm's effectiveness using a simulation function.
- Calculate average cost per session using the derived policy.
- For fixed number of cached items $C = 0.2K$, the optimal average cost, $E(S)$, is:

$$E[S] = 0.8 + 0.8(\frac{1}{q} - 1)(1 - \alpha)$$

- With $\alpha = 0.8$ and $q = 0.2$, we get $E[S] = 1.44$.
- Optimal policy should yield this average cost.

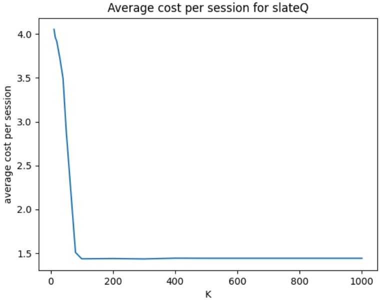
Simulation Results and Analysis

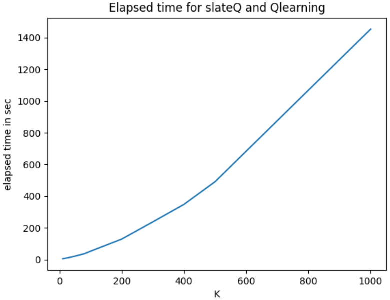




- Average cost for SlateQ aligns with Q-Learning and converges to 1.44.
- SlateQ time increases linearly with K, Q-Learning escalates exponentially.

Simulation Results and Analysis





- With a large library, the algorithm identifies optimal policy, cost remains 1.44.
- Elapsed time exhibits a near-linear increase. Algorithm operates as expected.