Replenishment of Retail Stores
TCS Quantum Computing Challenge 2023
Idea Summary

Nikhil Londhe, Pratik Kumar De, Grishma Prasad, Jay Patel, and Sreekuttan L S

May 22, 2024

Contents

1 Introduction

In the current era, the competitiveness of the retail market is intensifying with technological advances. Prompt restocking of products is essential to ensure customer satisfaction and foster business expansion. For small-scale retailers who operate a limited number of outlets, achieving effective product replenishment poses a significant challenge. This challenge escalates even further for large retail chains that operate many stores and warehouses, making it increasingly intricate and challenging to manage efficiently. Current solutions use either rule-based approaches or very small-scale machine-learning-based approaches. This problem is a good candidate for exploring niche technologies like quantum computing.

2 Problem Statement

Store replenishments are currently based on calculations made in supply chain software solutions and algorithms. These calculations depend on several parameters and data sets such as Store stock, Store predicted sales, In-transit stock to store, store stock targets, and Distribution Centre (DC) available stock. Again, DC stocks depend on plenty of other parameters like Safety stock, based on demand and demand variability, lead time and lead time variability, and desired service levels. So, the complexity of calculations has multiple dimensions, including the rate of sale, range, delivery frequency, pack sizes, sales forecast accuracy, seasonality, DC capacity and throughput limitations, physical variations between stores, store access limitations, etc. Therefore, it is very difficult to optimize the balance between customer availability, working capital/inventory, and operating costs. This problem influences customer availability, Cost to Serve, and working capital, which are key business metrics that affect customer experience and the company's profitability.

In today's retail landscape, where a single retailer may operate over a thousand stores and manage tens of thousands of diverse SKUs, the delicate balance between granularity and computational speed is often compromised. As a result, both stores and SKUs are aggregated into broader categories. This aggregation, while expedient, leads to solutions that are less than ideal. However, there lies a tremendous opportunity to harness cutting-edge quantum-based solution strategies. These innovative approaches promise to revolutionize SKU-based service levels, optimizing them to achieve the lowest possible working capital and cost to serve, thus catapulting the retail operations into a new era of efficiency and precision.

3 Solution Approach

Our task involves utilizing quantum computing techniques to address the challenge of optimal retail store replenishment. This issue covers two optimization layers. Initially, at the store level, our objective is to determine the ideal quantity of goods that a store should order from the distribution center (DC) to satisfy demand and maximize profits, considering factors such as store inventory, cost of holding inventory, and anticipated demand. Subsequently, at the DC level, the DC must process replenishment orders from various stores and decide the appropriate amount to restock each store based on their inventory levels, delivery lead times, and projected profits. Our focus is on optimizing at the store level and we will outline the strategies we have developed in the subsequent discussion.

3.1 Quantum Reinforcement Learning

Markov Decision Processes (MDP) optimize problems using states, actions, transition probabilities, and rewards, ensuring future states depend solely on the current state and action to maximize expected rewards. Previous studies have applied MDP to replenishment issues[1][4]. We enhance this model by incorporating available capital into state definitions. The state $\mathcal{S}$ is defined as $\mathcal{S} = (FD, C, OH)$, where OH represents on-hand inventory, FD forecasted demand, and C available capital, formatted as a $N_p \times N_s$ matrix for N_p products across N_s stores. Actions, structured in a similar way, involve weekly replenishments X. The rewards, derived from sales, are calculated based on forecasted demand and a probabilistic sales model, constrained between zero and the forecasted demand. Sales predictions are drawn from this model, and actual sales are influenced by total store stock and replenishments.

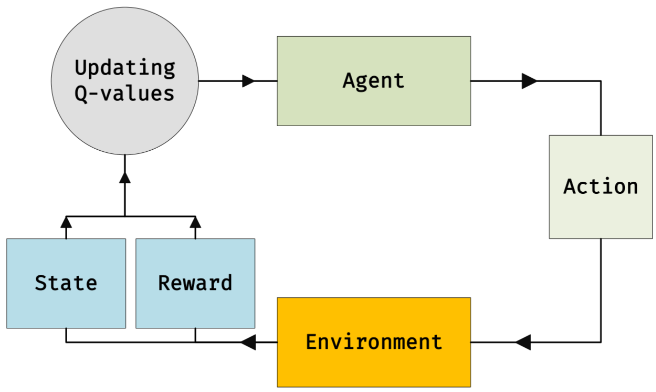

Figure 1: Flowchart for Q-Learning.

We apply Quantum Reinforcement Learning (QRL) to address the specified MDP. In Reinforcement Learning (RL), an agent learns to make decisions sequentially by interacting with its environment, receiving feedback in the form of rewards or punishments. This feedback helps guide the agent to optimal strategies. RL requires both exploration of the environment to find rewarding actions and the exploitation of known strategies. The essential elements of RL include states, actions, rewards, policies, and value functions[3]. QRL[2] mirrors the classical RL, using a quantum circuit as an agent to interact with its environment through actions, evaluated by rewards. Actions form a policy aimed at maximizing cumulative rewards. Notable algorithms such as Q-learning and Deep Q Networks (DQN) have proven effective in diverse areas, including gaming and robotics. Figure 1 illustrates a Q-learning based algorithm, a model-free method where the agent learns action values across states. Detailed descriptions of Parametric Quantum Circuit (PQC), Q-Learning, and Deep Q Networks are available in the appendix section of the final submission document.

$$P = min(S, OH + X) \times (P_c + P_r) - X \times (P_c + P_h) - OH \times P_h - max(X - D, 0) \times P_h \tag{1}$$

Where,

- P: Weekly profit.
- D: Demand quantity.
- S: Predicted Sales.
- OH: On-Hand quantity.
- X: Replenishment.
- P_r: Profit per unit for product.
- P_c: Procurement cost per unit for product.
- P_h: Holding cost per unit per week for product.

1 is the reward function we designed and implemented using an Open AI gym environment.

4 Results

The quantum model was evaluated on the Aer Simulator and IonQ Aria 1, contrasted with a classical neural network-based reinforcement learning model. Performance during training and testing for both quantum and classical are shown in Figures 3 , 4 ,5 and 6.Optimal rewards and execution times are detailed in tables 1 and 2, respectively. The Quantum Reinforcement Learning (QRL) with Parameterized Quantum Circuits (PQC) 2 performs comparably or better than the Classical Reinforcement Learning (CRL) model. However, QRL has longer runtimes due to its batch-based training algorithm, resulting in more model evaluations (circuit simulations) than CRL. The maximum circuit simulations are given by $(N_e + bN_t) \times 25$, where N_e is the number of episodes, N_t is the training episodes and b is the batch size. QRL decisions are documented in 7, executed on the IonQ QPU.

Model	Best Training Reward	Testing Reward
QRL (AerSimulator)	327	218 (mean — 25 episodes)
QRL (IonQ Aria 1)	-	206.8
CRL	391	219 (mean — 100 episodes)

Table 1: Reward comparison for all models.

Model	Time taken	Episodes
QRL (AerSimulator)	$\approx$ 35 min	1000
QRL (IonQ Aria 1)	$\approx$ 3 hrs	1
CRL	$\approx$ 4 min	1000

Table 2: Model Runtime.

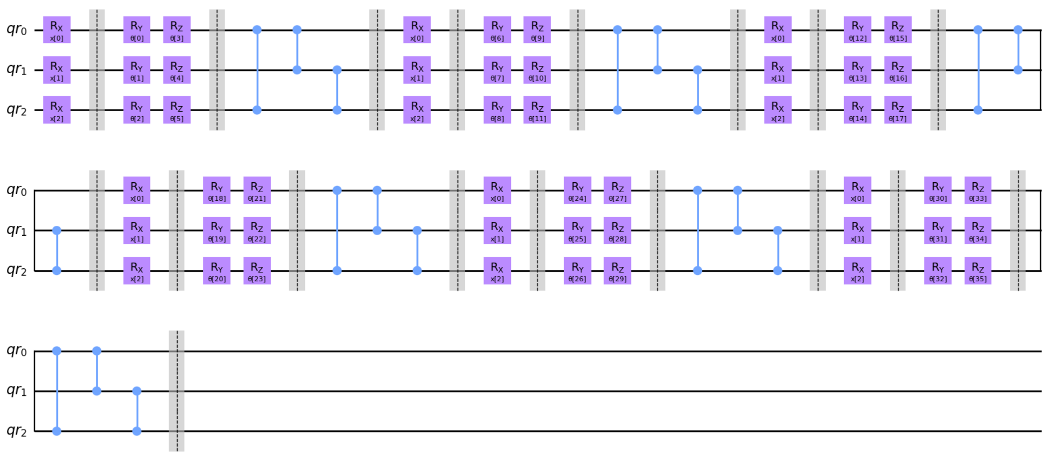

Figure 2: Quantum circuit we used for our quantum agent.

5 Conclusion

Our team has implemented the Quantum Reinforcement Learning (QRL) approach to optimize retail store replenishment. We modeled this as a Markov Decision Process (MDP), where agents learn replenishment strategies based on store conditions, including inventory, demand forecasts, and capital. Our Q-learning algorithm trains agents to maximize profits and minimize inventory and procurement costs. Our quantum RL has matched the performance of the classical RL using significantly fewer parameters, validated on both simulators and real quantum hardware, specifically IonQ Aria. **Our QRL model has yielded results comparable to those of the classical RL model, while utilizing a significantly lower number of parameters. The parameter ratio between QRL and Classical is an impressive 1:132. More comprehensive results have been provided in the defense presentation**. Later, if more parameters are added to the environment, our model will be very adaptive with fewer changes in computing resources. Scaling to multiple stores involves mapping state variables to qubits, but creating unique observables for actions like replenishment, constrained by capital, remains a challenge due to the limitations of

Figure 3: Training QRL on an ideal simulator (AerSimulator) for 1000 episodes.

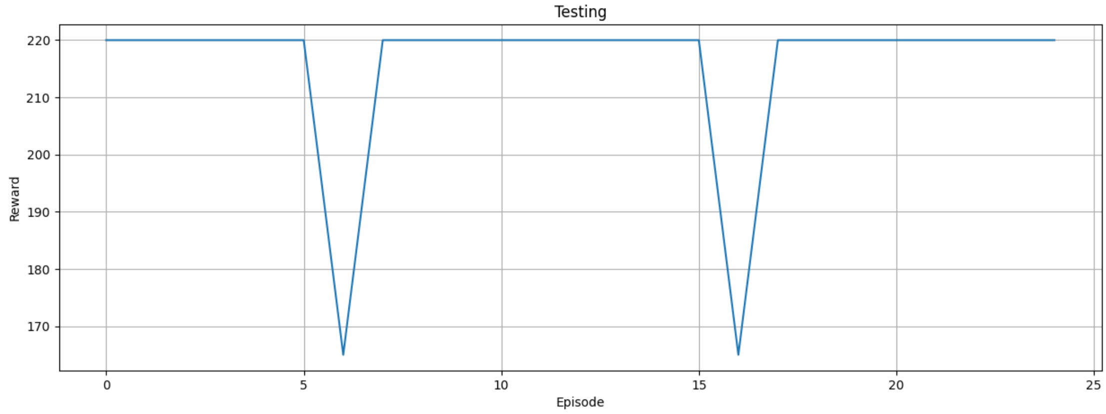

Figure 4: Testing the trained QRL on an ideal simulator (AerSimulator) for 25 episodes.

Figure 5: CRL model training for 1000 episodes.

the n-qubit Pauli group. This development suggests a promising future for quantum computing in achieving positive ROI with reduced computational needs. **As quantum hardware becomes more affordable, our approach could become a viable alternative to classical methods, potentially saving significant costs.**

6 Data Availability

The complete codebase can be found here, the final submission document can be found here and the defense presentation is available here.

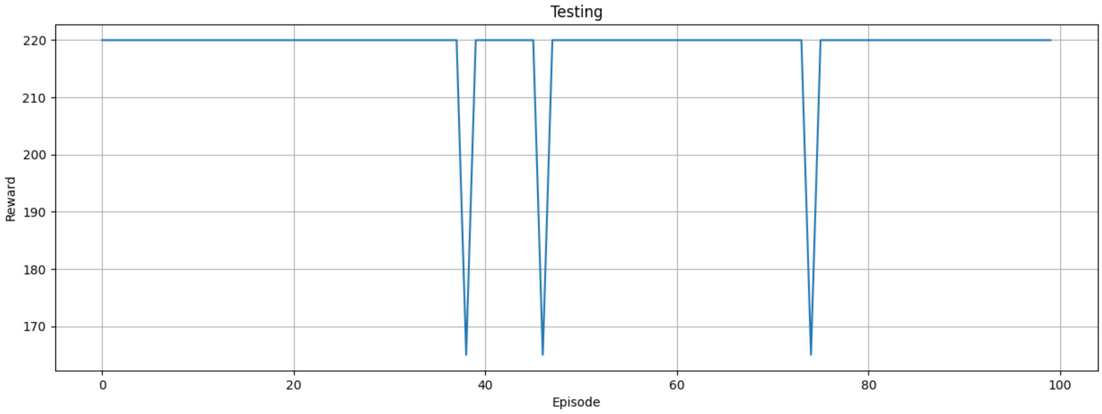

Figure 6: Testing the trained CRL model for 1000 episodes.

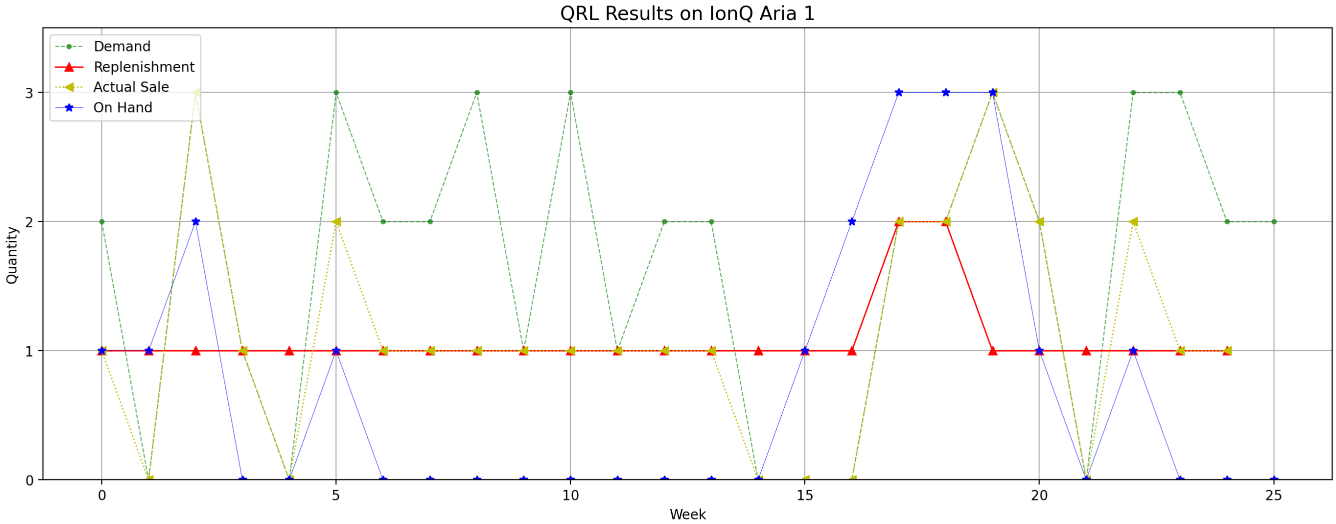

Figure 7: The Quantum Reinforcement Learning (QRL) model, executed on the Quantum Processing Unit (QPU) **IonQ Aria 1**, for the replenishment period of 25 weeks.

References

[1] J. Goedhart, R. Haijema, R. Akkerman, and S. de Leeuw. Replenishment and fulfilment decisions for stores in an omni-channel retail network. *European Journal of Operational Research*, 311(3):1009–1022, 2023.

[2] W. Hu and J. Hu. Distributional reinforcement learning with quantum neural networks. *Intelligent Control and Automation*, 2019.

[3] R. Karthikeyan. *Quantum inspired algorithms for learning and control of stochastic systems*. PhD thesis, 2015.

[4] A. G. Zheng Sui and L. Lin. A reinforcement learning approach for inventory replenishment in vendor-managed inventory systems with consignment inventory. *Engineering Management Journal*, 22(4):44–53, 2010.