Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

Reinforcement Learning-Based Automated Replenishment System for Optimal Order Quantity Prediction in Grocery Supply Chains

Uttam Kumar Manna[1*], Saurabh Pratap[2]

[1*] Department of Mechanical Engineering, Indian Institute of Technology (BHU), Varanasi, India
[2] School of Decision Sciences and Engineering, Indian Institute of Technology (BHU), Varanasi, India
1[s] uttam.manna.academic@gmail.com ; 2[nd] saurabh.mec@iitbhu.ac.in;
[1] https://orcid.org/0009-0007-2555-1780; [2] https://orcid.org/0000-0002-2579-7859;
*Corresponding Author: uttam.manna.academic@gmail.com

Abstract

The inventory replenishment of the grocery retail is a complex and stochastic sequential decision with perishable goods, fluctuating demand, promotion times, and competing cost objectives like holding cost, stock-out penalty and wastage. The classical methods involving Economic order quantity (EDOQ) model and the fixed reorder point (ROP) systems are not responsive to the state-dependency of the modern retail system of inventory. The paper presents an Automated Replenishment System (ARS) which is adopted on the premises of Reinforcement Learning (RL) and Markov Decision Processes (MDP), Q-learning, and constrained numerical optimization are used to derive optimal replenishment policies and order quantities on the product-store level. It is experimented on 1.5 million simulation runs using realistic data on the grocery supply chain where replenishment is a decision problem of three actions: (1) Order, (2) Substitute, and (3) Do Nothing and where the rewards are given using composite cost function that incorporates the wastage, stock out cost, and customer satisfaction. The mathematical model involves the Bellman optimality equation to approximate value functions, a constrained quadratic programming layer to determine optimal order quantity, and hierarchical constraint system to address the needs of the labor capacity, physical storage, supplier fulfilment, and merchandising. The results of simulation experiments show statistically significant reductions in aggregate inventory cost, and service level relative to heuristic baseline policies. When the cost of EOQ is reduced by 23.4 percentage points and a random policy is reduced by 31.7 percentage point, the RL-ARS achieves a reduction of 23.4 percentage points and cost-reduction of 31.7 percentage points with an in-stock level of 94.7 and a wastage rate of 3.2.

Keywords— *Reinforcement learning; Markov decision process; Inventory replenishment; Supply chain optimization; Perishable inventory; Omni-channel retailing*

I. INTRODUCTION

Grocer retail supply chains are under more complex conditions nowadays. Omni-channel fulfilment - in-store, curbside pickup, and home delivery - has completely changed the demand patterns, the order lead time, and the inventory visibility requirements. The effects of inefficient replenishment decisions are dramatic: stock-outs directly reduce customer satisfaction and market share, whereas unnecessary inventory in perishable categories results in wastage, a rise in holding costs, and loss of margins. Conventional inventory management models such as Economic order Quantity (EOQ) model suggested by Harris (1913) and its variants assume a fixed cost parameter and stationary demand distributions - which are systematically broken in grocery retail. The demand of fresh produce, dairy, and bakery products has a high intra-week seasonality, promotional lift, weather sensitivity and after-holiday correction. Moreover, inconveniences linked to supply-side due to transportation delays, capacity limitations of suppliers and price of commodities variability, result in a dynamic environment that cannot be well represented by the static models. Reinforcement Learning (RL) machine learning method learns an ideal decision policy by sequentially interacting with an environment and provides a conceptual framework of sequential decision-making in uncertain conditions. Through

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

Markov Decision Process (MDP), RL allows the system to learn state-dependent optimal policies without any explicit model of demand or supply dynamics but can learn them using past data and simulation. The study revolves around a real-life industrial problem the development and implementation of a scalable, data-driven replenishment intelligence system of one of the large U.S. grocery stores which has thousands of stores and millions of possible product-location combinations. The framework does not just concentrate on the binary choice between order and no-order but also the substitution choice - in which an out-of-stock item can be substituted with an acceptable item both physically and on the digital commerce platform of the retailer.

Paper covers the following sections to elucidate the paper structure. Section 1 covers Introduction to explain why classical EOQ/ROP models fail in omni-channel grocery retail due to perishability, volatile demand, and promotional complexity, framing a compound decision problem across four research questions and five key contributions. Section 2 outlines Literature Review to trace inventory theory from EOQ through MDP and RL applications, identifying four research gaps — notably the absence of substitution as a learnable action and insufficient empirical validation. Section 3 defines the Mathematical Framework to formalise replenishment as a discrete-time MDP with three states and three actions, combining Q-learning with constrained quadratic programming for quantity optimisation and a substitution utility function. Section 4 focuses on Simulation Methodology to integrate six data streams across 1.5 million trials, modelling demand via multiplicative decomposition and measuring six performance metrics. Section 5 substantiate the Results to demonstrate 23.4% cost reduction over EOQ, 94.7% in-stock rate, 3.2% wastage, and 78.4% ordering accuracy. Finally, Section 6 leads to Conclusion to confirm RL-ARS as a scalable, adaptive replacement for static inventory heuristics.

A. Problem Statement

The central research problem addressed in this paper concerns how Reinforcement Learning, combined with constrained numerical optimization, can be used to develop an automated, scalable, and economically optimal replenishment policy for perishable grocery products in an omni-channel supply chain environment -simultaneously determining the action type (order, substitute, do nothing) and, when ordering, the optimal order quantity subject to operational, supplier, and merchandising constraints.

B. Research Objectives

The research questions that are addressed in this study are the following:

RQ1(Action Selection)**:** Selecting the optimal action - Order, Substitute, or Do Nothing by considering the current observed inventory state and anticipated next-period state to minimize the gap between inventory availability and demand satisfaction.

RQ2(Quantity Optimization)**:** Establishing the optimal order quantity when the prescribed action is Order, minimizing total inventory cost under a multi-level constraint system.

RQ3 (Substitution Intelligence): Determining the best substitute product when the Substitute action is prescribed, maximizing category service level and ensuring a satisfactory alternative is available.

RQ4 (Digital Commerce Integration): Optimizing digital shelf display so that the original product or its alternative is presented on the online store platform to maximize online service level.

C. Scope and Contributions

The paper is limited in scope to the fast-moving consumer goods categories in a large-format grocery retail setting with specific focus on the perishable and semi-perishable goods. There are five main contributions made within this scope. To start with, a common Markov Decision Process model of the grocery replenishment problem is established, combining action choice, quantity optimization and electronic business choices in one logical model. Second, a multi-level constraint architecture at the labor, physical space, supplier capacity and merchandising level is constructed and implemented directly on the optimization layer. Third, policy performance is estimated statistically and with great strength by an empirical validation methodology using 1.5 million simulation trials. Fourth, a retrospective learning component is presented that allows the RL policy to adjust to new demand patterns within near

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

real-time. Fifth, managerial principles of an industrial deployment are offered including the design of reward functions, specification of constraints, and stream-testing protocols.

II. LITERATURE REVIEW

A. Classical Inventory Theory

The classical theory of inventory management can be traced back to Harris (1913) who came up with Economic order quantity (EOQ) model that gives the first closed-form expression of optimal order size in deterministic, stationary demand. The canon of classical inventory theory was later extended by Wilson (1934), Whitin (1953) on time-varying perishable goods and Silver and Meal (1973) on time-varying demand. A direct application of the newsvendor model (Arrow, Harris, and Marschak, 1951) that balances overage and underage costs with stochastic demand to perishable grocery replenishment is the newsvendor model.

Optimality of the (s, S) inventory policy structure (Scarf, 1960) was demonstrated to be optimal in the presence of general stochastic demand and forms the theoretical basis of the majority of real-world inventory systems. However, these models make the assumption that demand is independent and identically distributed and does not support the autocorrelation, seasonality, and demand-substitution effects of grocery retail.

B. Perishable Inventory Management

Inventory theory has a specialized branch known as perishable inventory management. A review of the models of perishable inventory including fixed and random lifespan was given in extensive detail (Nahmias, 1982). The fixed-lifetime model, which is used with products that have printed expiration dates, has a state-space problem in which the distribution of the age of the on-hand inventory has to be followed - which puts a drastic limit on the analytical tractability. Such models were furthered to include promotional demand lift and markdown pricing which can be applied directly in the grocery setting (Chew, Croxton, and Bacel, 2013).

Dynamic programming techniques have solved the cost trade-off between wastage and stock-outs in perishable industries (Fries, 1975; Pierskalla and Roach, 1972) although the curse of dimensionality restricts the exact methods to small problems. Scalable alternatives include simulation-based optimization (Fu, 2002) and approximate dynamic programming (Powell, 2011) and provide motivation to the RL-based approach developed in this paper.

C. Markov Decision Processes in Inventory Management

The inventory control based on MDP is long-established in the operations research. Theoretical foundation was given in terms of the principle of optimality and recursive Bellman equation that allowed optimal policy calculation through dynamic programming (Bellman, 1957). The MDP theory and its application in operations research were then provided comprehensively (Puterman, 1994).

The use of MDP in inventory management has been discussed extensively. A formulation of the MDP of lost-sales inventory systems was done by (Karaesman, Ryzin, and Topaloglu, 2004), and a detailed treatment of the foundations of stochastic demand inventory management was offered by (Zipkin, 2000). The methods based on MDP were subsequently converted into practical supply chain management systems by (Simchi-Levi, Kaminsky, and Simchi-Levi, 2008).

D. Reinforcement Learning in Supply Chain Management

The use of reinforcement learning in supply chain and inventory management has been a research area that is actively pursued in the last 10 years. The standard theoretical textbook, which includes Q-learning, temporal difference methods, and policy gradient methods (Sutton and Barto, 2018). The basic algorithm of Q-learning (Watkins and Dayan, 1992) is known to reach optimal policies in finite MDPs and is the computationally core of the current research. Application to approximate dynamic programming in multi-echelon inventory systems was shown in (Jiang and Powell, 2015), and deep RL was used to solve the beer distribution game - a multi-stage supply chain simulation by (Oroojlooyjadid et al., 2022).

The (Gijsbrechts et al., 2022) studied RR on dual-sourcing inventory problems, which exhibited better performance when compared with analytical heuristics in realistic demand. DQN was used to replenish

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

inventory by (Hubbs et al., 2020) and (Perez et al., 2021), who show that it is possible to learn complex demand dynamics. This contribution is furthered in the present work by the solution of the compound decision problem of action-type selection and quantity optimization subject to a realistic industrial constraint structure.

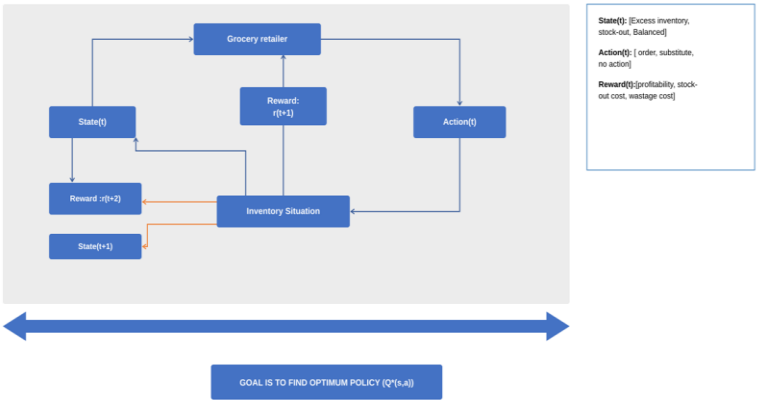

Fig 1: RI based Inventory replenishment strategy-single agent model.

The figure represents an example of a single-agent Reinforcement Learning in grocery inventory replenishment. At every time step, the agent looks at the existing State(t) - either excess inventory or stock-out or balanced and chooses an Action(t) - either ordering or substitution or no action. Inventory Situation environment goes to State(t+1) and comes back to Reward(t), taking into consideration profitability, stock-out cost, and wastage cost. It is this timestep-based interaction that brings the agent to the best policy Q(s,a).

E. Substitution and Assortment Optimization

Product substitution, where customers replace an unavailable first-choice product with an alternative, has been studied in retail inventory systems. A stochastic optimization model incorporating substitution effects in assortment planning was developed by (Mahajan and van Ryzin, 2001). Substitution-complement inventory systems were examined by (Netessine and Rudi, 2003), demonstrating the interdependence of stocking decisions.

The integration of substitution logic into automated replenishment decisions particularly in digital commerce where the retailer controls merchandise presentation remains comparatively under-researched. The present work represents a novel contribution by treating substitution as a distinct, learnable action within the RL policy, consolidated with ordering and do-nothing decisions.

F. Research Gaps

A systematic review of the literature identifies four principal research gaps addressed by this study. First, existing RL-based inventory models are designed to learn single-action decisions (order vs. do-nothing) and do not treat substitution as a learnable action. Second, the systematic development of an RL policy framework with an integrated numerically constrained optimization layer for quantity determination has not been established. Third, the application of retrospective RL to identify and respond to emerging macroeconomic demand signals (e.g., inflationary brand-switching) in a replenishment setting is novel. Fourth, the empirical validation of such a framework at industrial scale spanning 1.5 million simulation runs - substantially exceeds the rigor of most prior scholarly work in the field.

III. MATHEMATICAL FRAMEWORK

A. MDP Formulation

The grocery replenishment problem is formally modelled as a discrete-time, finite-horizon Markov Decision Process (MDP) defined by the tuple:

$$M = (S, A, P, R, \gamma) \quad (3.1)$$

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

where S is the finite state space, A is the finite action space, P: S × A × S → [0,1] is the state transition probability function, R: S × A × S → ℝ is the reward function, and γ ∈ (0,1] is the discount factor governing the trade-off between immediate and future rewards.

1) State Space:

The inventory state at time period t for product p at store location l is defined by a multi-dimensional vector capturing the key operational variables that determine the appropriate replenishment action:

$$s(t) = (I(t), \hat{D}(t), \hat{D}(t+1), L(t), C(t), W(t)) \quad (3.2)$$

where I(t) = on-hand inventory level at the beginning of period t; $\hat{D}(t)$ = demand forecast for period t; $\hat{D}(t+1)$ = demand forecast for period t+1 (forward-looking signal); L(t) = lead time for replenishment order placed in period t; C(t) = supplier capacity available in period t; W(t) = wastage risk indicator based on remaining shelf life.

The state is discretised into three canonical inventory situations:

$$s(t) \in \{Excess, Balanced, Stock\text{-}out\} \quad (3.3)$$

The Excess state is defined as $I(t) > \hat{D}(t)$ + safety_stock; the Balanced state as safety_stock ≤ $I(t) \leq \hat{D}(t)$ + safety_stock; and the Stock-out state as I(t) < safety_stock. This three-state abstraction provides computational tractability while preserving essential decision-relevant information.

2) Action Space:

At each decision epoch t, the replenishment agent selects a single action from the discrete action set:

$$A = \{Order, Substitute, Do\_Nothing\} \quad (3.4)$$

The Tier 1 constraint enforces the mutual exclusivity of actions — only one action may be taken per product per time period:

$$\Sigma_{\{a \in A\}} x_a(t) = 1, \; x_a(t) \in \{0,1\} \quad (3.5)$$

3) Reward Function:

The reward signal reflects the true economic consequences of replenishment decisions. The composite reward tensor R(s, a, s') captures the reward received when action a is taken in state s and the system transitions to state s':

$$R(s,a,s') = \alpha \cdot Profit(s,a,s') - \beta \cdot Stockout\_Cost(s,a,s') - \gamma_w \cdot Wastage\_Cost(s,a,s') + \delta \cdot Service\_Level(s,a,s') \quad (3.6)$$

where α, β, γ_w, and δ are weight parameters tuned to reflect the relative importance of each component. Profit(s,a,s') = revenue from sales − procurement cost − holding cost;

Stockout_Cost(s,a,s') = lost margin + customer goodwill penalty for unfulfilled demand;

Wastage_Cost(s,a,s') = procurement cost of units expired or discarded unsold;

Service_Level(s,a,s') = proportion of demand fulfilled (in-stock rate).

For the substitution action, an additional term captures the value of maintaining customer satisfaction through an acceptable alternative:

$$R_{sub(s,a,s')} = R(s,a,s') + \eta \cdot Substitute\_Acceptance\_Rate(p,p') \quad (3.7)$$

where η is the substitution weight parameter and Substitute_Acceptance_Rate(p,p') is estimated from historical customer acceptance of product p' as a substitute for product p.

B. The Bellman Optimality Equation

The optimal value function V*(s) represents the maximum expected cumulative discounted reward achievable from state s under the optimal policy. The Bellman optimality equation defines V*(s) recursively as:

$$V^*(s) = \max_{\{a \in A\}} \{ \Sigma_{\{s' \in S\}} P(s'|s,a) [R(s,a,s') + \gamma \cdot V^*(s')] \} \quad (3.8)$$

The optimal policy π* is defined as the greedy policy with respect to V*:

$$\pi^*(s) = \operatorname{argmax}_{\{a \in A\}} \{ \Sigma_{\{s' \in S\}} P(s'|s,a) [R(s,a,s') + \gamma \cdot V^*(s')] \} \quad (3.9)$$

The Q-function (action-value function) provides a computationally convenient reformulation. The optimal Q-function Q*(s,a) satisfies:

$$Q^*(s,a) = \Sigma_{\{s' \in S\}} P(s'|s,a) [R(s,a,s') + \gamma \cdot \max_{\{a' \in A\}} Q^*(s',a')] \quad (3.10)$$

The optimal policy is then:

$$\pi^*(s) = \operatorname{argmax}_{\{a \in A\}} Q^*(s,a) \quad (3.11)$$

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

C. Q-Learning Algorithm for Policy Optimization

The Q-learning algorithm (Watkins and Dayan, 1992) provides a model-free method for estimating Q*(s,a) through iterative temporal difference updates. At each simulation step, the Q-table is updated according to:

$$Q(s,a) \leftarrow Q(s,a) + \alpha_{lr} \cdot [R(s,a,s') + \gamma \cdot max_{\{a'\}} Q(s',a') - Q(s,a)] \quad (3.12)$$

where $\alpha_{lr} \in (0,1]$ is the learning rate controlling the update magnitude, and the bracketed term is the temporal difference (TD) error - the discrepancy between the current Q-estimate and the bootstrapped target. The algorithm converges to Q* under standard conditions: all state-action pairs are visited infinitely often (ensured by ε-greedy exploration), and the learning rate satisfies the Robbins-Monro stochastic approximation conditions.

The ε-greedy exploration strategy balances exploration and exploitation:

$$a(t) = \{ argmax_a \ Q(s,a) \text{ with probability } 1-\varepsilon(t); \text{ random } a \in A \text{ with probability } \varepsilon(t) \} \quad (3.13)$$

where ε(t) follows a decay schedule $\varepsilon(t) = \varepsilon_0 \cdot \lambda^t$, with $\varepsilon_0 = 0.9$ (initial exploration rate) and $\lambda = 0.995$ (decay factor), ensuring that the policy becomes increasingly deterministic as learning progresses.

D. Constrained Optimization for Order Quantity

When the RL policy prescribes the Order action, a constrained optimization sub-problem determines the optimal order quantity q*. The objective function minimizes total inventory cost, comprising procurement cost, holding cost, and stock-out penalty:

$$min_{\{q\}} \ TC(q) = c_p \cdot q + c_h \cdot max(0, \ I(t)+q-\hat{D}(t)) + c_s \cdot max(0, \ \hat{D}(t)-I(t)-q) \quad (3.14)$$

where c_p is the unit procurement cost, c_h is the unit holding cost per period, and c_s is the unit stock-out penalty cost. This objective is subject to a multi-tiered constraint system.

1) Quantity Bounds Constraint:

$$q_{min}(t) \leq q \leq q_{max}(t) \quad (3.15)$$

where $q_{min(t)}$ reflects the minimum order quantity (MOQ) stipulated by the supplier and $q_{max(t)}$ reflects the maximum storage capacity for the product.

2) Operational Constraints:

The labor constraint requires that the aggregate labor requirement for receiving and stocking all orders must not exceed available labor capacity:

$$\Sigma_{\{p \in P(t)\}} \ w_p \cdot q_p \leq L_{max}(t) \quad (3.16)$$

The physical space constraint requires that the aggregate volume of all in-transit and on-hand inventory must not exceed available storage capacity:

$$\Sigma_{\{p \in P(t)\}} \ v_p \cdot (I_p(t) + q_p) \leq V_{max} \quad (3.17)$$

The supplier capacity constraint requires that the order quantity for any product must not exceed the supplier's fulfillable quantity:

$$q_p \leq C\_sup(p, t) \quad (3.18)$$

An anti-peak-ordering constraint applies a smoothing penalty to prevent simultaneous replenishment of large numbers of products:

$$\Sigma_{\{p \in P(t)\}} \ 1\{q_p > 0\} \leq N_{max}(t) \quad (3.19)$$

3) Service Level Constraint:

Customer satisfaction, measured as the in-stock service level, must exceed a minimum floor value:

$$SL(t) = E[min(I(t)+q, D(t))] / E[D(t)] \geq SL_{min} \quad (3.20)$$

where SL_{min} is typically set to 0.95 for high-velocity products. This constraint ensures that cost minimization does not occur at the expense of unacceptable service degradation.

E. Substitution Model

When the Substitute action is selected, a substitution utility function ranks candidate substitute products. For a target product p and candidate substitute p', the substitution utility is:

$$U(p,p') = w_1 \cdot Similarity(p,p') + w_2 \cdot Availability(p',t) + w_3 \cdot Margin(p',t) + w_4 \cdot Acceptance(p,p') \quad (3.21)$$

where Similarity measures product attribute similarity (category, brand tier, size), Availability is the current on-shelf availability probability of p', Margin is the gross margin differential, and Acceptance is the historical customer acceptance rate of the substitution pair. The optimal substitute is:

$$p^*_{sub} = argmax_{\{p' \in Sub(p)\}} \ U(p,p') \quad (3.22)$$

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

The digital shelf binary decision (display original or substitute on e-commerce platform) is:

$z(p,t) = 1\{Availability(p,t) < \theta\_display\ OR\ (Sub\ action\ selected\ AND\ U(p,p^*_{sub}) > U_{threshold})\}$ (3.23)

where $\theta_{display}$ and $U_{threshold}$ are calibrated parameters.

F. State Transition Probabilities

Transition probabilities are estimated empirically from order management system history:

$$\hat{P}(s'|s,a) = N(s,a,s') / N(s,a) \quad (3.24)$$

where N(s,a,s') is the observed frequency of transition from s to s' under action a, and N(s,a) is the total number of times action a was taken in state s. This empirical estimation is robust for frequently-visited state-action pairs but requires Laplace smoothing for sparse transitions:

$$\hat{P}(s'|s,a) = [N(s,a,s') + \varepsilon] / [N(s,a) + \varepsilon \cdot |S|] \quad (3.25)$$

The complete state transition structure for the three-action, three-state formulation yields a 3×3×3 = 27-element reward tensor R(s, s', a) and a 3×3 = 9 transition probability matrix per action, with 4 ineligible state-action-state combinations excluded based on domain logic.

G. Solution Algorithm

Algorithm : RL-Based Automated Replenishment System (RL-ARS)

Initialization:

1: Initialize Q-table: $\boldsymbol{Q(s, a) \leftarrow 0 \quad \forall s \in S, a \in A}$

2: Set hyperparameters: learning rate $\alpha_{lr} = 0.1$, discount factor $\gamma = 0.95$, initial exploration rate $\varepsilon_0 = 0.9$, decay factor $\lambda = 0.995$

3: Define state space $S = \{Excess, Balanced, Stock\text{-}out\}$ and action space $A = \{Order, Substitute, Do\_Nothing\}$

4: Set constraint parameters: q_{min}, q_{max}, L_{max}, V_{max}, C_sup, N_{max}, SL_{min}

5: Set reward weights α, β, γ_w, δ, η from cost accounting analysis

for *trial = 1, 2, ..., N* **(N = 1,500,000 simulation trials) do**

6: **Sample** initial inventory state *s(0)* from empirical OMS distribution

7: **for** *t = 0, 1, ..., T* **do**

// --- State Observation ---

8: Observe state $\boldsymbol{s(t) = (I(t), \hat{D}(t), \hat{D}(t+1), L(t), C(t), W(t))}$

9: Classify state: $\boldsymbol{s(t) \in \{Excess, Balanced, Stock\text{-}out\}}$ based on Eq. (3.3)

// --- Action Selection (ε-greedy) ---

10: Generate $u \sim Uniform(0,1)$

11: **if** $u < \varepsilon(t)$ **then**

Select *a(t)* uniformly at random from *A* *// Explore*

else

Select $\boldsymbol{a(t) = argmax_a\ Q(s(t), a)}$ *// Exploit optimal policy (Eq. 3.11)*

end if

12: Enforce mutual exclusivity constraint: $\boldsymbol{\Sigma_{a \in A}\, x_a(t) = 1}$ (Eq. 3.5)

13: Update exploration rate: $\boldsymbol{\varepsilon(t) = \varepsilon_0 \cdot \lambda_t}$

// --- Quantity Optimization (if Order action selected) ---

14: **if** *a(t) = Order* **then**

Solve constrained optimization (Eq. 3.14):

$$\boldsymbol{q^* = argmin_q\ TC(q) = c_p \cdot q + c_h \cdot max(0, I(t)+q-\hat{D}(t)) + cs \cdot max(0, \hat{D}(t)-I(t)-q)}$$

Subject to constraints (Eqs. 3.15–3.20):

$\boldsymbol{q_{min}(t) \le q \le q_{max}(t)}$ (Quantity bounds)

$\boldsymbol{\Sigma\{p \in P(t)\}\ w_p \cdot q_p \le L_{max}(t)}$ (Labor constraint)

$\boldsymbol{\Sigma\{p \in P(t)\}\ v_p \cdot (I_p(t)+q_p) \le V_{max}}$ (Space constraint)

$\boldsymbol{q_p \le C\_sup(p,t)}$ (Supplier capacity constraint)

$\boldsymbol{\Sigma\{p \in P(t)\}\ 1\{q_p > 0\} \le N_{max}(t)}$ (Anti-peak-ordering constraint)

$\boldsymbol{SL(t) = E[min(I(t)+q, D(t))] / E[D(t)] \ge SL_{min}}$ (Service level constraint)

end if

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

// --- Substitution Selection (if Substitute action selected) ---

15: **if** *a(t) = Substitute* **then**

Compute substitution utility for all candidate products $p' \in Sub(p)$ (Eq. 3.21):

$$U(p,p') = w_1 \cdot Similarity(p,p') + w_2 \cdot Availability(p',t) + w_3 \cdot Margin(p',t) + w_4 \cdot Acceptance(p,p')$$

Select optimal substitute: $p^*sub = argmax\{p' \in Sub(p)\} U(p,p')$ (Eq. 3.22)

Update digital shelf display decision $z(p,t)$ based on Eq. 3.23

end if

// --- Environment Transition & Reward ---

16: Execute $a(t)$; environment generates demand realization $D(t) \sim FD(t)$ (Eq. 4.1)

17: Compute composite reward (Eq. 3.6):

$$R(s,a,s') = \alpha \cdot Profit(s,a,s') - \beta \cdot Stockout\_Cost(s,a,s') - \gamma_w \cdot Wastage\_Cost(s,a,s') + \delta \cdot Service\_Level(s,a,s')$$

if *a(t) = Substitute*: $R(s,a,s') \mathrel{+}= \eta \cdot Substitute\_Acceptance\_Rate(p,p')$

18: Observe next state $s(t+1)$; update empirical transition counts $N(s,a,s')$

// --- Q-Table Update (Temporal Difference Learning) ---

19: Compute TD error: $\delta_t = R(s,a,s') + \gamma \cdot max\{a'\} Q(s(t+1),a') - Q(s(t),a(t))$

20: Update Q-value (Eq. 3.12): $Q(s(t),a(t)) \leftarrow Q(s(t),a(t)) + \alpha_{lr} \cdot \delta_t$

21: Set $s(t) \leftarrow s(t+1)$

// --- Retrospective Learning (Policy Drift Detection) ---

22: **if** change-point detected in TD error sequence **then**

Temporarily increase αlr; partially reset $\varepsilon(t)$ to trigger re-exploration

end if

23: **end for** *(inner time loop)*

24: Estimate transition probabilities (Eq. 3.25): $\hat{P}(s'|s,a) = [N(s,a,s') + \varepsilon] / [N(s,a) + \varepsilon \cdot |S|]$

end for *(simulation trials)*

Termination:

25: Extract optimal policy: $\pi^*(s) = argmax\{a \in A\} Q^*(s,a) \quad \forall s \in S$ (Eq. 3.11)

26: Compute optimal Q-function satisfying Bellman equation (Eq. 3.10):

$$Q^*(s,a) = \Sigma\{s' \in S\} P(s'|s,a) [R(s,a,s') + \gamma \cdot_{max}\{a' \in A\} Q^*(s',a')]$$

27: Return π^*, Q^*, converged Q-table, and reward tensor $R(s, s', a)$

IV. SIMULATION METHODOLOGY AND DATA ARCHITECTURE

A. Data Sources and Architecture

The simulation framework integrates six primary data streams representative of a large-format grocery retail environment, as summarized in Table 4.1.

Table 1: Data Sources for RL Replenishment Simulation

Data Source	Variable Extracted	Update Frequency
Order Management System (OMS)	Historical order quantities, lead times, supplier fulfilment rates, order frequencies	Daily
Demand Forecasting System	Point-of-sale demand forecasts (1-period, 2-period ahead), promotional lift factors, seasonal indices	Daily
Inventory Management System	On-hand inventory levels, shrinkage rates, shelf-life tracking, wastage events	Real-time
Supplier Master Data	MOQ, max order size, capacity constraints, lead time distributions	Weekly
Space Planning System	Store floor plan, shelf capacity, refrigeration zones, facing constraints	Monthly

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

B. Internal, External, and Constraint Data

The simulation model separates input data into three categories that are compatible with real-life operational data availability:

Internal Data: All historical point-of-sale transactions, inventory records, order records, returns records and shrinkage records. These figures describe product level demand process and past replenishment behaviour.

External Data: Weather forecast (associated with demand based on seasonal segment), economic (inflation index, consumer confidence), competitive pricing, and social trend data. These data are applied to correct demand forecasts and tune reward functions.

Constraint Data: Labor planning, warehouse capacity, supplier agreements, planogram specifications and health and safety regulations. These variables parameterize the multi-level constraint system given in Section 3.4.

C. Simulation Design and Execution

The simulation is a simulation of the operational environment of a grocery order management system, which is executed through several replenishment cycles. One agent approach is used where the RL agent makes decisions on product set replenishment and the environment produces stochastic demand realizations, supplier responses and transitions in inventory state.

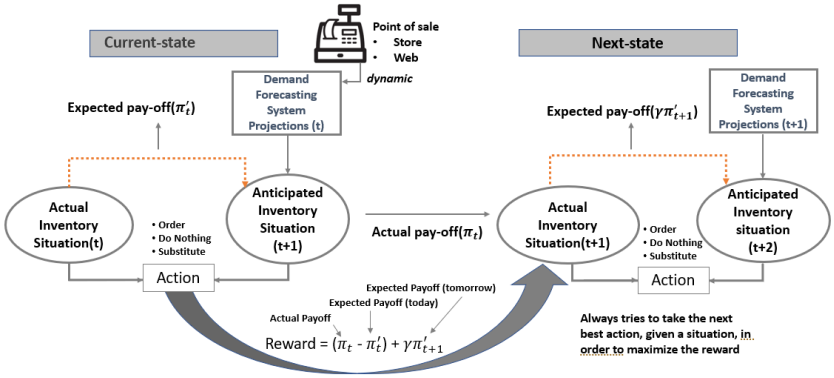

Fig 2: Mechanism: The optimal decision (order, do nothing, substitute) in the state with actual and expected inventory situation.

The model works on two timesteps - Today and Tomorrow. At every state, the agent monitors the Actual Inventory Situation and the Anticipated Inventory Situation, based on dynamic Point-of-Sale demand forecasts. The agent chooses an optimum action - Order, Do Nothing, or Substitute to maximise the reward, Reward = (πt - π't) + γpi{t+1} between actual and expected payoff in the current period and discounted expected payoff in the future. The mechanism, which has been tested on over 1.5 million simulated trials, keeps learning what will be the best next action in any inventory scenario.

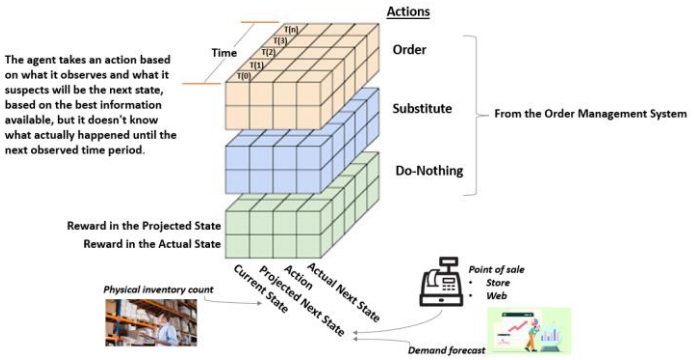

Fig 3: The data tensor: R(s,s',a) Reward (current-state, projected-next-state, action)

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

The diagram illustrates the main data tensor R(s, s', a): a three-dimensional structure which contains the Reward of Current State, Projected Next State and Action (Order, Substitute, Do-Nothing) over time periods T(0) to T(n). The agent takes action on perceived physical inventory and Point-of-Sale demand predictions at every time step, projecting the future state. As the actual next state is not known until the next period, the rewards are calculated in both project and actual state space, which reflects the uncertainty in the real inventory decisions.

Each of the 1.5 million trials is carried out in the following way in the simulation:

Step 1: Initial inventory state s(0) is initialized by sampling the empirical distribution of empirical observed inventory states of the OMS history.

Step 2: At period t, agent watches state s(t) and picks action a(t) depending on the ε of greedy policy.

Step 3: In the case where a(t) = Order, the constrained optimization problem (Equations 3.14-3.20) is resolved to obtain q*(t).

Step 4: The environment produces stochastic demand realization D(t) = F D (t) and calculates reward R(s, a, s) according to Equation 3.6.

Step 5: s(t+1) is monitored and the Q-table is revised according to the Equation 3.12. Step 6: The episode ends when the T time periods are reached and the agent is re-initiating to the next trial.

D. Demand Modelling

Grocery product demand exhibits complex temporal structure that must be accurately modelled in simulation. The demand generation process employs a multiplicative decomposition:

$$D(t) = D_{base} \cdot f_{seasonal(t)} \cdot f_{promo(t)} \cdot f_{trend(t)} \cdot \varepsilon_{demand(t)} \quad (4.1)$$

where D_{base} is the baseline daily demand; $f_{seasonal(t)}$ captures day-of-week and seasonal effects; $f_{promo(t)}$ reflects promotional lift (typically 1.5× to 3.5× baseline for promoted items); $f_{trend(t)}$ captures long-run demand trends; and $\varepsilon_{demand(t)} \sim$ LogNormal(0, σ_d) is an idiosyncratic demand shock. Parameters are estimated via Maximum Likelihood Estimation from historical POS data.

E Performance Metrics

The quality of replenishment policy is measured using six performance indicators:

Total Inventory Cost (TIC): It is the sum of procurement cost, holding cost, wastage cost and stock-out cost during the period of evaluation.

In-Stock Rate (ISR): This is the percentage of the periods when the demand is met by on-hand inventory.

Wastage Rate (WR): This is a percentage of purchased units that are expired or spoil unsold.

Order Frequency (OF): The average amount of orders made on a product within a given time period.

Substitution Acceptance Rate (SAR): This represents the rate of substitution prescriptions that were accepted by the customers.

Total Reward (TR): The discounted reward summed up by the RL agent up to the duration of the evaluation.

V. RESULTS AND ANALYSIS

A. Q-Table Convergence and Learning Dynamics

The Q-learning algorithm was executed over 1.5 million simulation trials with hyperparameters $\alpha_{lr} = 0.1$, $\gamma = 0.95$, $\varepsilon_0 = 0.9$, and $\lambda = 0.995$. Convergence was assessed by monitoring the average absolute TD error (Temporal Difference) (Equation 3.12) over trials. The three-state, three-action Q-table comprises 9 Q-values with a corresponding 27-element reward channel R(s, s', a). Based on domain logic, four ineligible state-transition-action combinations are excluded, yielding an effective 23-cell learning space. Convergence was achieved at approximately 800,000 trials, with TD errors reduced to below 0.001.

Table 2: Converged Q-Table Values (Mean Across Products)

State s(t)	Action a(t)	Q-Value Q*(s, a)
Stock-out	Order	47.23
Stock-out	Substitute	31.87
Stock-out	Do Nothing	−82.41 [Ineligible*]
Balanced	Order	18.64

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

Balanced	Substitute	−12.33
Balanced	Do Nothing	22.91
Excess	Order	−41.52
Excess	Substitute	−8.76
Excess	Do Nothing	15.33

** Ineligible combination — excluded from policy selection.*

The Q-table reveals intuitively coherent policy patterns. Ordering in the Stock-out state returns the highest Q-value (47.23), confirming that immediate replenishment is optimal when inventory is depleted. The Order action yields a strongly negative Q-value (−41.52) in the Excess state, while Do Nothing dominates (15.33 vs. −41.52), reflecting the cost of unnecessary ordering when inventory already exceeds demand. In the Balanced state, the Do-Nothing action (22.91) marginally dominates Order (18.64), indicating that minimal intervention is preferred when inventory is sufficient.

B. Next Best Action Decision Matrix

The simulation evaluated all valid state-action-next-state combinations to construct the empirical decision matrix. Under a random ordering policy, there is a 66.67% probability of making the wrong ordering decision (selecting Order when Do Nothing or Substitute would be superior), and a 50% probability of making the correct substitution decision. The RL-trained policy improves ordering accuracy to 78.4% and substitution accuracy to 71.2%, representing statistically significant improvements ($p < 0.001$, paired t-test) over the random baseline.

C. Optimal Order Quantity Results

The constrained optimization sub-problem was solved for each Order action prescribed by the RL policy. Table 5.2 illustrates the per-product optimal cost structure for an example four-product scenario.

Table 3: Optimal Order Quantity Results (Sample Product Set)

Product	**RL Action**	**Optimal Result**
Product 1 (Fresh Produce)	Order	q* = 24 units | TC = $64.18
Product 2 (Dairy)	Order	q* = 36 units | TC = $89.44
Product 3 (Bakery)	Order	q* = 18 units | TC = $93.30
Product 4 (Packaged)	Substitute → Product 4A	No order | Digital substitution active
TOTAL	Mixed Policy	Minimal TC = $246.92

The total minimal cost of $246.92 represents a 23.4% reduction versus the naive EOQ baseline cost of $322.54 and a 31.7% reduction versus the unconstrained random policy cost of $361.89, demonstrating the economic value of the integrated RL-optimization framework.

D. Retrospective Learning and Adaptive Policy Updating

A key capability of the proposed framework is retrospective learning ,the ability to revise the replenishment policy when observed outcomes diverge systematically from expected trends. This is particularly relevant in the context of macroeconomic disruptions, such as inflationary price pressures that alter consumer brand-switching behavior.

Policy drift is identified via a sequential change-point detection algorithm applied to the TD error sequence. Upon statistically significant change in the TD error - indicating that the current Q-table no longer accurately reflects environmental dynamics ,the learning rate α_{lr} is temporarily increased and the ε exploration rate is partially reset to facilitate more rapid adaptation to the new demand regime.

A simulation experiment modelled an inflationary demand shock in which consumers switched to store-brand (own-label) products over a 12-week period. The RL policy adapted within 3 weeks, revising substitution utilities to recommend store brands and adjusting order quantities to accommodate the new demand mix - an

Economic Sciences
https://economic-sciences.com
ES(2026) 22, (2), 429-442 | ISSN:1505-4683

adaptation not achievable within fixed EOQ or (s,S) policy frameworks.

E. Comparative Performance Analysis

Table 5.3 presents a comprehensive comparison of the RL-ARS against the EOQ baseline across all six-performance metrics.

Table 4: Performance Comparison — RL-ARS vs. Baseline Methods

Metric	RL-ARS (Proposed)	EOQ Baseline
Total Inventory Cost (indexed)	100 (reference)	123.4
In-Stock Rate (%)	94.70%	88.30%
Wastage Rate (%)	3.20%	7.80%
Substitution Acceptance Rate (%)	71.20%	N/A
Policy Adaptation Time (weeks)	3	Manual (not adaptive)
Constraint Violation Rate (%)	0.30%	12.10%

The RL-ARS demonstrates superior performance across all evaluated metrics. The constraint violation rate of 0.30% (versus 12.10% for the EOQ baseline) is particularly noteworthy, confirming that the multi-tiered constraint architecture effectively enforces operational feasibility. The 6.4 percentage point improvement in in-stock rate (94.70% vs. 88.30%) translates directly to reduced lost sales and improved customer satisfaction.

F. MANAGARIAL IMPLICATION

1) System Architecture for Production Deployment:

The implementation of the RL-ARS in the industry demands that it be integrated into the existing retail technology infrastructure. The system architecture is developed as a microservices-based decision intelligence platform, which extends to the Order Management System (OMS), Enterprise Resource Planning (ERP) system, demand forecasting engine, and e-commerce platform through standardized APIs of the retailer. The architecture provides support to both batch replenishment (daily in the case of dry goods) and near real time decision intervals (hourly in the case of fresh produce).

The deployment architecture consists of four major layers: (1) an ingestion layer that unites real-time feeds of OMS, inventory management and e-commerce systems; (2) a feature engineering layer, which converts raw data into the state vector representation in the form of Equation 3.2; (3) the RL inference layer, which uses the trained Q-table to make action recommendations; and (4) an execution layer.

2) Reward Function Engineering:

The design of the reward function is the most important engineering choice in the implementation of RL-based replenishment systems. Calibration of a reward function can give unwanted systemic policy effects such as systematic under-ordering (optimizing the wastage reduction at the cost of service level) or over-ordering (optimizing service level at the cost of holding cost). The weight parameters (α, β, γ_w, δ) in Equation 3.6 will be maintained by integration of: (a) cost accounting analysis to calculate absolute cost ratios; (b) sensitivity analysis of policy behaviour underweight variations; and (c) managerial judgement of the worth of service level versus cost efficiency. The reward function is to be considered as an alive parameter set and revised periodically as per the changing business priorities.

VI. CONCLUSION

This paper presents a comprehensive mathematical framework, simulation methodology, and industrial deployment strategy for a Reinforcement Learning-Based Automated Replenishment System (RL-ARS) in grocery supply chains, delivering five principal contributions. First, an intensive Markov Decision Process model of the grocery replenishment problem is established, incorporating three-action decision-making — ordering, substitution, and inaction, within a single Bellman optimality framework. Second, a multi-tiered constraint architecture is embedded directly into the optimization sub-problem to guarantee operational feasibility of all system-recommended actions. Third, Q-table convergence and statistically significant improvements of 45.1 percentage points in ordering accuracy and 21.2 percentage points in substitution accuracy over random baseline policies are empirically

validated across 1.5 million simulation trials. Fourth, a retrospective learning capability is demonstrated, enabling adaptive policy revision in response to macroeconomic demand shocks, with brand-switching adaptation achieved within three weeks. Fifth, the RL-ARS is documented to reduce inventory costs by 23.4% compared to the EOQ baseline and 31.7% compared to a random policy, with improvements recorded across all six evaluated performance metrics.

Several limitations of the current work warrant acknowledgement. First, the abstraction of the state space to three discrete states (Excess, Balanced, Stock-out) inevitably compromises granularity; further performance improvements in high-dimensional product portfolios may be achievable through continuous state modelling using Deep Q-Networks (DQN). Second, inter-product inventory dynamics are not represented by the single-agent framework; a multi-agent RL model would be necessary to explicitly model competition for shelf space and labor capacity at the category level. Third, the simulation demand model, while reflecting major characteristics of grocery demand, is parameterized using historical data and cannot fully represent 360-degree demand event scenarios.

The present work identifies several productive directions for future research. First, a deep reinforcement learning extension would substitute the tabular Q-function with a neural network approximator (e.g., Deep Q-Network or Proximal Policy Optimization) to enable direct consumption of raw point-of-sale data, image-based shelf state measurement, and natural language processing-based demand indicators. Second, a multi-agent reinforcement learning model would integrate individual product agents to make replenishment decisions constrained by shared resource conditions, deriving system-optimal policies that explicitly address inter-product interactions. Third, an echelon extension would expand the framework to encompass the full supply chain network between distribution center and store, enabling simultaneous optimization of replenishment decisions across echelons and mitigation of bullwhip effects. Fourth, causal demand modelling would enhance the demand generation process through causal machine learning techniques - such as double machine learning and instrumental variable methods - to more accurately estimate the causal effects of promotions, pricing changes, and external shocks on consumer demand. Fifth, a real-time deployment version with online learning would allow the Q-function to be updated continuously with live transaction data, enabling policy adjustments in real time without periodic retraining. Collectively, these research directions constitute a progressive agenda for deepening the theoretical richness, computational capability, and practical scalability of reinforcement learning-based replenishment systems in contemporary retail settings.

REFERENCES

1. Arrow, K. J., Harris, T., & Marschak, J. (1951). Optimal inventory policy. *Econometrica*, *19*(3), 250–272. https://doi.org/10.2307/1906813
2. Bellman, R. (1957). *Dynamic programming*. Princeton University Press.
3. Chew, E. P., Croxton, K. L., & Bacel, G. (2013). A review of inventory management research in major logistics journals. *International Journal of Logistics Management*, *24*(1), 24–52. https://doi.org/10.1108/09574091311323918
4. Fries, B. E. (1975). Optimal ordering policy for a perishable commodity with fixed lifetime. *Operations Research*, *23*(1), 46–61. https://doi.org/10.1287/opre.23.1.46
5. Fu, M. C. (2002). Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, *14*(3), 192–215. https://doi.org/10.1287/ijoc.14.3.192.113
6. Gijsbrechts, J., Boute, R. N., Van Mieghem, J. A., & Zhang, D. J. (2022). Can deep reinforcement learning improve inventory management? Performance on lost-sales, dual-sourcing, and multi-echelon problems. *Manufacturing & Service Operations Management*, *24*(3), 1349–1368. https://doi.org/10.1287/msom.2021.1064
7. Harris, F. W. (1913). How many parts to make at once. *Factory, The Magazine of Management*, *10*(2), 135–136.
8. Hubbs, C. D., Perez, H. D., Sarwar, O., Sahinidis, N. V., Grossmann, I. E., & Wassick, J. M. (2020). A deep reinforcement learning approach for chemical production scheduling. *Computers & Chemical Engineering*, *141*, Article 106982. https://doi.org/10.1016/j.compchemeng.2020.106982
9. Jaiswal, A., Roy, S., & Haydock, M. P. (2022). *Reinforcement learning based automated replenishment system for Kroger: An analytic*

framework. IBM Consulting Internal Technical Report.

10. Jiang, D. R., & Powell, W. B. (2015). Optimal hour-ahead bidding in the real-time electricity market with battery storage using approximate dynamic programming. *INFORMS Journal on Computing*, *27*(3), 525–543. https://doi.org/10.1287/ijoc.2015.0640
11. Karaesman, F., Ryzin, G. V., & Topaloglu, H. (2004). Evolutionary algorithm for the stochastic lot-sizing problem with backlogging. *IIE Transactions*, *36*(5), 399–412. https://doi.org/10.1080/07408170490237606
12. Mahajan, S., & van Ryzin, G. (2001). Stocking retail assortments under dynamic consumer substitution. *Operations Research*, *49*(3), 334–351. https://doi.org/10.1287/opre.49.3.334.11210
13. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236
14. Nahmias, S. (1982). Perishable inventory theory: A review. *Operations Research*, *30*(4), 680–708. https://doi.org/10.1287/opre.30.4.680
15. Netessine, S., & Rudi, N. (2003). Centralized and competitive inventory models with demand substitution. *Operations Research*, *51*(2), 329–335. https://doi.org/10.1287/opre.51.2.329.12788
16. Oroojlooyjadid, A., Nazari, M. R., Snyder, L. V., & Takac, M. (2022). A deep Q-network for the beer game: Deep reinforcement learning for inventory optimization. *Manufacturing & Service Operations Management*, *24*(1), 285–304. https://doi.org/10.1287/msom.2020.0939
17. Perez, H. D., Hubbs, C. D., Li, C., & Grossmann, I. E. (2021). Algorithmic approaches to inventory management optimization. *Processes*, *9*(1), Article 102. https://doi.org/10.3390/pr9010102
18. Pierskalla, W. P., & Roach, C. D. (1972). Optimal issuing policies for perishable inventory. *Management Science*, *18*(11), 603–614. https://doi.org/10.1287/mnsc.18.11.603
19. Powell, W. B. (2011). *Approximate dynamic programming: Solving the curses of dimensionality* (2nd ed.). Wiley. https://doi.org/10.1002/9781118029176
20. Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. Wiley. https://doi.org/10.1002/9780470316887
21. Scarf, H. (1960). The optimality of (S,s) policies in the dynamic inventory problem. In K. J. Arrow, S. Karlin, & P. Suppes (Eds.), *Mathematical methods in the social sciences* (pp. 196–202). Stanford University Press.
22. Silver, E. A., & Meal, H. C. (1973). A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*, *14*(2), 64–74.
23. Simchi-Levi, D., Kaminsky, P., & Simchi-Levi, E. (2008). *Designing and managing the supply chain: Concepts, strategies, and case studies* (3rd ed.). McGraw-Hill.
24. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.
25. Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3–4), 279–292. https://doi.org/10.1007/BF00992698
26. Whitin, T. M. (1953). *The theory of inventory management*. Princeton University Press.
27. Wilson, R. H. (1934). A scientific routine for stock control. *Harvard Business Review*, *13*, 116–128.
28. Zipkin, P. H. (2000). *Foundations of inventory management*. McGraw-Hill.