Adaptive Inventory Replenishment for Dynamic Supply Chains with Uncertain Market Demand

Viswanath Kumar Ganesan, David Sundararaj, and Ananda Padmanaba Srinivas

Abstract The flow of goods in supply chains starts from production plants to regional warehouses to local distribution centers and from these local distribution centers to point of sale at retail outlets. Uncertainties in global market such as trade wars and extreme weather conditions disrupt the flow of goods in the global supply chains. This paper presents a reinforcement learning approach for an autonomous inventory replenishment planning model that attempts to capture few aspects of the goods such as market demand, costs associated with the inventory, product life cycle, and/or seasonality along with a set of inventory policies. The proposed model has been evaluated using two different time horizons viz., weekly and monthly, and it is observed from our simulation runs that monthly planning provides around 30% cost reductions compared with weekly planning, and the algorithm is found to select the right policy in about 85–95% of the times across the experiments.

Keywords Inventory · Demand uncertainty · Replenishment · Reinforcement · Learning · Supply chains

1 Introduction

The supply and distribution chains that enable shipment of goods from manufacturing source to end customer locations have seen larger transformations over the last few decades. Supply chains have evolved to enable organizations to manufacture products with larger product variety to meet individual customer preferences as well

V. K. Ganesan (✉) · A. P. Srinivas
Innovation Labs, Tata Consultancy Services Ltd, Chennai, India
e-mail: viswanath.ganesan@tcs.com

A. P. Srinivas
e-mail: ap.srinivas@tcs.com

D. Sundararaj
Information Science & Technology, Anna University, Chennai, India
e-mail: davidsundararaj@outlook.com

A. Chakrabarti and M. Arora (eds.), *Industry 4.0 and Advanced Manufacturing*,
Lecture Notes in Mechanical Engineering,
https://doi.org/10.1007/978-981-15-5689-0_28

as act as a catalyst to provide technological innovation as perishability of the goods is more acceptable irrespective of the nature of the product or the commodity. The concept of perishability inherently refers to product life cycle dimension rather than taking it literally for the product perishability definition. Inventory control effectively contributes to significant amount of the total supply chain costs, and over the last decade, the combination of the factors such as market demand, inventory costs, changing customer preferences, and many other factors has ensured that many products do have minimal life cycle length of few years rather than decades. Chen et al. [1] present a case-based multi-criteria ABC analysis including factors such as lead time and criticality of SKUs that provides more flexibility in classification of SKUs. Jiang and Sheng [2] present a case-based reinforcement learning algorithm for dynamic inventory control in a multi-agent-based supply chain system with the premise that traditional time-triggered and event-triggered ordering policies become inaccurate causing excessive inventory (cost) or shortage. Kim et al. [3] employed the reinforcement learning algorithm for attaining a satisfactory service level in n-echelon supply chain. Hsueh [4] explores the effects of product life cycle on inventory control in manufacturing/re-manufacturing system and determines optimal production lot size, reorder point, and safety stock during each phase of product life cycle. Sun and Zhao [5] in their paper model Q-learning algorithm as a reinforcement learning approach for specifying ordering policies of the supply chain with five stages. In another work, Mortazavi et al. [6] use an agent-based simulation technique integrated with a reinforcement learning algorithm for a four-echelon supply chain that faces non-stationary customer demands. Kara and Dogan [7] address the inventory management problem of perishable products and present a reinforcement learning which is used in their research work, and the policies are optimized using Q-learning and Sarsa algorithms.

The markets are connected globally with enhanced visibility and ensuring that multiple nodes in the supply chain are sufficiently proactive. Push- and pull-based strategies in supply chain flows help organizations to carve out strategic and tactical plans based on market needs. A push-based supply chain, where upstream nodes drive the inventory flows, is used when the products or goods have low demand uncertainty. Pull-based strategy, where the downstream nodes pull inventories from upstream node, is used when products have high demand uncertainty. Baryannis et al. [8] provide a comprehensive review of supply chain literature that addresses problems relevant to SCRM using approaches that fall within the AI spectrum. Lolli et al. [9] use supervised classifiers based on the machine learning to perform a limited exhaustive search to define the best reorder policies per item by presenting the lowest cost classification of in-sample items. Priore et al. [10] employ an inductive learning algorithm for defining the most appropriate replenishment policy over time by reacting to the environmental changes in a three-echelon supply chain where the scenario is defined by seven variables viz., cost structure, demand variability, three lead times, and two partners' inventory policy.

This motivation of the research is to study and explore insights into dynamism and uncertainties in inventory management at various echelons in supply chain considering various factors such as market demand, costs associated with the inventory

flow, product life cycle, and/or seasonality along with a set of inventory policies. The research problem requires use of data and decision support models to foresee and predict uncertainties, capture the nature of the uncertainties, and provide model-driven decisions using intelligent learning algorithms to get adaptable answers to respond to practical needs. Little amount of work has been published in scientific journals/conferences considering combinations of factors related to products such as seasonality, life cycle, demand behavior, and planning horizon in relation to autonomous and dynamic selection of inventory policies over time. Dynamic selection essentially means that the system by itself selects the right inventory policy based on demand projections without manual intervention using reinforcement learning approach. The paper proposes replenishment system where manual intervention is almost minimal or non-existent which is applicable from a warehouse or distribution center in a supply chain. This same is perfectly applicable for an autonomous retail store. The paper is organized as follows. Section 2 introduces the inventory replenishment model detailing replenishment policies, states and actions considered for our study. The reinforcement learning algorithm based on Q-learning approach is presented in this section. Section 3 details out the simulation runs and experiments of our study followed by conclusion and scope for extended research.

2 The Replenishment Model

2.1 Problem Definition

The pull-based supply chain model is defined for our study that necessitates individual nodes in the supply chain to adopt right set of inventory replenishment policies for managing inventory flows from upstream supplies to satisfy the demand requirements downstream in the supply chain. The nodes could be either a central warehouse or local/regional distribution centers or point-of-sale outlets. Each of these nodes does handle good amount of products with each of these products having its own demand behavior, replenishment cycles, minimum and maximum order quantities, product life cycle, and seasonal or cyclical trends. In this work, we pick up one of the nodes in the supply chain say a distribution center and study the learning-based replenishment model considering the following assumptions.

1. Historical customer demand is generated using a compound Poisson process, where the size of the demand is defined to have the discrete uniform distribution.
2. Future demand data is generated using a time-series forecasting process, and any impact of market campaigns/promotions is incorporated into future demand profile.
3. Historical data is used for training the proposed model, and the future demand data is taken as one of the inputs to the autonomous replenishment model.
4. Order is placed for procurement with an order quantity Q that is defined based on the nature of the replenishment policy in operation.

5. Ordering cost or setup costs are incurred every time an order is placed.
6. Order can be replenished fully with an expected lead time *L,* and the standard deviation of lead time is defined as *l*.
7. Shortages in inventories are allowed, and the surplus inventories are carried forward to the next planning period.
8. Backorders are not considered, and shortages are assumed to be lost demand.
9. Joint replenishment between multiple products is not considered.
10. The life cycle of a product could be divided into the following phases: introduction (phase 1), growth (phase 2), maturity (phase 3), and decline (phases 4 and 5).

2.2 Inventory Policies

The following periodic and continuous review policies which are stock based were adapted and implemented in our simulation model. Stock-based inventory policies replenish inventory based on inventory levels at the replenishment centers based on market demand.

i. The periodic (*R*, *S*) policy, where every *R* time units (the review period) an order is placed to raise the inventory position to level *S*.
ii. The periodic (*R*, *s*, *S*) policy, where every *R* time units, an order is placed to raise the inventory position to level *S* provided the inventory position has reached or fallen below the reorder level s.
iii. The periodic (*R*, *s*, *Q*) policy, where every *R* time units an order of *Q* units is placed provided the inventory position is less than or equal to *s*.
iv. The continuous (*s*, *S*) policy, where an order is placed to raise the inventory position to level *S* when the inventory position falls to or below level *s* and
v. The continuous (*s*, *Q*) policy, where *Q* units are ordered when the inventory position falls to or below level s.

The values of maximum inventory position *S* are defined for our experiments as the maximum or average of weekly demands observed over the past historical demand periods, and *Q* can be computed using economic order quantity (EOQ) formulae or can be set to be equal to the expected forecast demand in the future time periods. EOQ works well when the demand is more or less flat with no trend or seasonality.

2.3 States and Actions

The following set of propositions is introduced to enable the proposed model to learn and train itself over the historical data in a supervised manner to enable us to set the algorithmic configuration parameters. These propositions are characterizations obtained from the above-mentioned five policies to estimate reorder quantity, number of reorder points, under and overstocking implications, and demand behavior.

Proposition 1 Increase in inventory holding cost necessitates re-computation of reorder quantity, i.e., minimization of reorder quantity.

Proposition 2 Increase in set up costs necessitates minimization of number of reorders which eventually turns out increasing the reorder quantity for each order.

Proposition 3 Overstocking in inventories results in a need to move from periodic review policy to continuous review policy.

Proposition 4 Continual shortages in inventories require a shift from periodic review policy to continuous review policy.

Proposition 5 Change in demand behavior from flat pattern to an increasing or decreasing trend requires switch to a continuous review policy.

Proposition 6 Change in demand behavior to the flat pattern from increasing or decreasing trend requires switch from continuous to periodic review policy.

Proposition 7 Demand in maturity phase of the product with flat pattern can be fulfilled with periodic review policy.

The last two propositions essentially imply that it uses combination of either surplus or shortages in inventory with demand trend.

2.4 Reinforcement Learning Algorithm

Reinforcement learning (RL) provides an alternative approach to solve difficult control problems which are impossible to solve using supervised learning and traditional dynamic programming methods [11]. RL algorithms usually estimate the value functions of Markov decision process by observing state transition data. Tabular algorithms such as tabular Q-learning and tabular Sarsa-learning have be widely studied. In tabular RL algorithms, value functions are represented and estimated in tabular forms for each state-action pair [11]. In our work, we use the combination of shortages (*avg shortage*), frequency of shortages (*freq shortage*), and surplus (*avg surplus*) inventories over the past ***n*** periods to trigger the need for evaluation of policy change. An adaptation of Q-learning algorithm is used in our reinforcement learning process to select and use the mentioned replenishment policies in our model. The following notations are defined before presenting the algorithm.

Action	An array that contains different inventory policies that can be applied.
s	The policy index.
A	The action index.
Reward	The associated costs
S_t	The *state* tuple which represents an array containing average shortage, frequency of shortages, and average surplus during a time period.
Policy	A dictionary that maps an action to a state.
Returns	The reward which is the total cost incurred.
state_action_reward	An array that keeps track of the states, actions, and their corresponding rewards.
state_action_return	An array that contains the mean episodic reward for the specified time period.
seen_action_state	An array containing the state-action pairs that have been visited already. It is used as a caching mechanism.
DISCOUNT_FACTOR	A factor that is set to 0.9 to ensure that the RL algorithm places higher weightage on long-term rewards. (Values closer to 1.0 are used to 'discount' the immediate reward)
EPSILON	A factor used to balance exploration and exploitation by assigning a 20% chance to choose a random inventory policy.
Demand	An array containing the forecasted demand values.
G	The reward for an episode.
Q	The Q-table which contains the reward for each action taken in each state
V	Contains the maximum return corresponding to the best action for a given state in the Q-table

We use the following symbols:

- { } to refer to dictionaries or key-value pairs.
- () to refer to sets where only unique elements are permitted.
- [] to refer to arrays.

Reinforcement Learning Algorithm

1. Initialize variables
   policy = {},
   Q = {},
   returns = {},
   actions = set of all inventory policies under consideration,
   state_action_reward = [],
   DISCOUNT_FACTOR = 0.9,
   EPSILON = 0.2
2. for time period *t* in *Demand*:
   a. S_t = [*avg shortage*$_t$, *freq shortage*$_t$, *avg surplus*$_t$]
   b. If random(0, 1) < (1 – *EPSILON*)
      i. *action* = *policy*[*S*]
   c. else: *action* = random(*actions*)
   d. *reward* = -1 x *cost*$_t$
   e. if end of *Demand* has been reached:
      i. *state_action_reward.append*([S_t, null, *reward*])
      ii. break
   f. else:
      i. *state_action_reward.append*([S_t, *action*, *reward*])
   g. *G* = 0, *state_action_return* = []
   h. for *state, action, reward* in reversed(*state_action_reward*)[1:]:
      i. *state_action_return.append(*[*state, action, G*])
      ii. *G* = *reward* + *DISCOUNT_FACTOR* x *G*
3. *seen_state_action* = ()
4. Loop until *Q*[*s*] converges:
   a. for *state, action, G* in *state_action_return*:
      i. if [*state, action*] not in *seen_state_action*:
         1. returns[[*state, action*]].append(G)
         2. Q[s][a] = mean(returns[[*state, action*]])
         3. *seen_state_action.append*([*state, action*])
   b. for *s* in *policy.keys*():
      i. policy[s] = argmax(Q[s])
5. *V* = {}
6. for *s* in *policy.keys*():
   a. V[s] = max(Q[s])
   b. return V, policy

In our implementation, the values of average shortages, frequency of shortages, and average surplus are pre-estimated based on training data set with a planning horizon of *n* periods considering the cost vs benefit trade-off for the mentioned five policies. The forecast data together with the historical data is eventually used on the rolling window basis to decide the need for trigger for the policy change. Any

significant change in the cost data (i.e., ordering costs and holding costs) necessitates the recompilation of the state-action transition matrix using the historical as well as forecast data to redefine the selection of policies.

3 Simulation Runs and Experiments

The objective of our experiments is to ensure that the demand is satisfied at stated service levels, and shortages are avoided as much as possible with simultaneous minimization of total costs. Total cost comprises total annual ordering costs and annual inventory holding costs. The following three demand patterns are simulated to experiment and validate the proposed model:

1. Demand being flat and increasing.
2. Demand decreasing and turns flat.
3. Considering the complete product life cycle with four phases.

Demand forecast is assumed to be given on daily basis for the next one year planning period using historical data for the last three to five years. Inventory holding cost is assumed to be defined at 5% of the product cost. Product cost is assumed to be $10 for our experimental runs using simulated data for training out models. Ordering costs are defined as 10% of the cost of the reorder quantity. Experiments for policy definitions are performed at weekly and monthly planning horizon granularity levels.

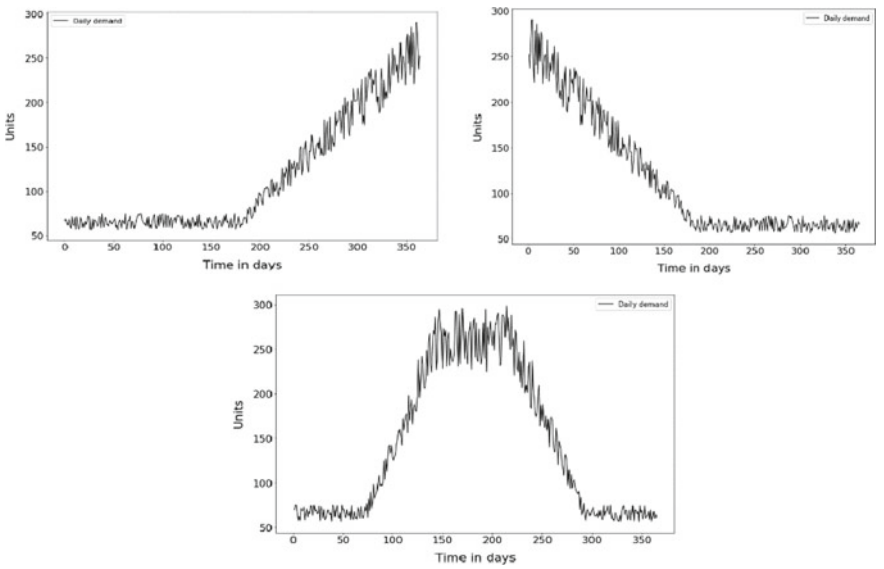

Fig. 1 Three types of demand profiles

Table 1 Evaluation of costs at weekly run

Demand profile	Total costs	Average surplus	Average deficit
Flat and increasing	2702554.18	187.98	43.23
Decreasing and flat	3942057.90	229.19	42.25
Full product life cycle	1234665.63	170.20	83.53

Table 2 Evaluation of costs at monthly run

Demand profile	Total costs	Average surplus	Average deficit
Flat and increasing	2006172.33	1069.73	34.76
Decreasing and flat	2489461.39	1187.54	34.90
Full product life cycle	1849730.37	1063.63	77.81

The three presented scenarios (Fig. 1) have been extracted from the historical profiles of the products from consumer product company supply chain at the local distribution center that receives supplies on either weekly or monthly basis, while the deliveries to different retail outlets happen on daily basis based on the orders received from the retail outlets across a major metropolitan city in India. The reinforcement learning algorithm together with inventory models and policies is implemented in Anacondo Phython 3.7 programming language and PostgreSQL. Tables 1 and 2 presents the total annual costs (includes ordering cost and inventory carrying cost), average daily surplus, and average daily deficit for weekly and monthly planning models with the average of the results presented over 10 simulation runs for three selected product profiles over an yearly planning cycle. The cumulative demand for each demand profile is kept constant for with respect to each one of the demand profiles and their corresponding simulation runs.

Figures 2 and 3 present the computation of surplus and shortages for the three demand profiles over weekly and monthly planning horizon for one simulated demand scenario. The results are presented for service levels 95% service levels. The inventory carrying costs are computed on daily basis, while ordering cost is defined for each order placed for procurement. Average surplus and shortages presented in the tables are computed using the end of the week/month surpluses and shortages in inventory. We found that the proposed algorithm is observed to select 85–95% of the times the right policy in our experiments and in those cases where the selection was a mismatch got corrected within a period length of 2 weeks in the worst case.

The proposed approach looks into the future demand profile and revises the order quantities placed with the upstream supplier based on surplus and shortages in inventory on-hand. It was observed from our experiments that when the demand is flat

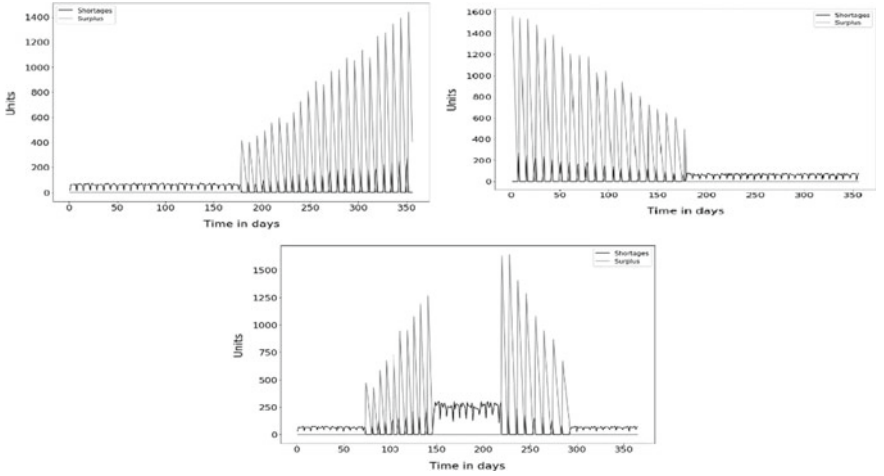

Fig. 2 Surplus and shortages for weekly planning mode for the demand profiles

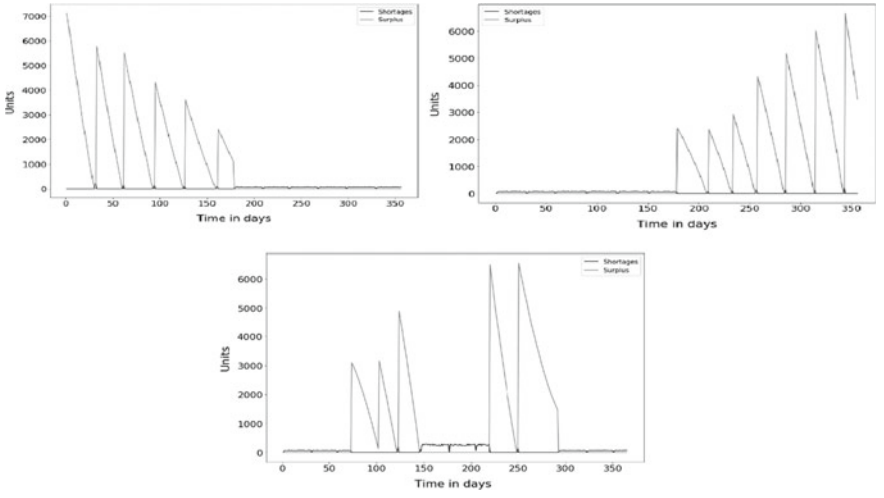

Fig. 3 Surplus and shortages for monthly planning mode for the demand profiles

over a period of time (R, S), policy was found to be working well. (R, s, S) or (R, s, Q) was found to be working good under the conditions of demand uncertainty, and (s, S) or (s, Q) is found to be working well under conditions of increasing or decreasing demand.

The higher service levels introduced in our experimentations resulted in higher surplus scenarios during the periods of increasing and decreasing demand profiles. It is found that weekly planning model is cost effective for all the demands profiles.

4 Conclusions

In this work, we attempt to introduce a replenishment model using a supervised learning-based approach that reflects the characteristics of the inventory operations in a typical supply chain and validate the results against three demand scenarios that are quite evident in real life. The proposed model considers multiple factors such as product life cycle, demand behavior and seasonality, planning horizon granularity along with set of replenishment policies to cost effectively plan and define replenishment cycles, and order quantities considering lead time variations. This essentially automates the replenishment process to a good extent and enables operations of autonomous warehouses in supply chains or motivates to run autonomous retail stores. The presented work limits the study for evaluation of replenishment model with no back orders, and shortages are allowed which need not be true in many practical scenarios. The research can be further extended considering perishable products, joint replenishment options, as well as policies considering back orders, supply variations, market campaigns, etc.

References

1. Chen, Y., Li, K. W., Kilgour, M. D., & Hipel, K. W. (2006). A case-based distance model for multiple criteria ABC analysis. *Computers & Operations Research, 35,* 776–796.
2. Jiang, C., & Sheng, Z. (2009). Case-based reinforcement learning for dynamic inventory control in a multi-agent supply-chain system. *Expert Systems with Applications, 36,* 6520–6526.
3. Kim, C. O., Kwon, I.-H., & Kwak, C. (2010). Multi-agent based distributed inventory control model. *Expert Systems with Applications, 37,* 5186–5191.
4. Hsueh, C. F. (2011). An inventory control model with consideration of remanufacturing and product life cycle. *International Journal of Production Economics, 133,* 645–652.
5. Sun, R. & Zhao, G. (2012). Analyses about efficiency of reinforcement learning to supply chain ordering management. In *IEEE international conference on industrial informatics (INDIN)* (pp. 124–127).
6. Mortazavi, A., Khamseh, A. A., & Azimi, P. (2015). Designing of an intelligent self-adaptive model for supply chain ordering management system. *Engineering Applications of Artificial Intelligence, 37,* 207–220.
7. Kara, A., & Dogan, I. (2018). Reinforcement learning approaches for specifying ordering policies of perishable inventory systems. *Expert Systems with Applications, 37,* 150–158.
8. Baryannis, G., Validi, S., Dani, S., & Antoniou, G. (2019). Supply chain risk management and artificial intelligence: state of the art and future research directions. *International Journal of Production Research, 57,* 2179–2202.
9. Lolli, F., Balugani, E., Ishizaka, A., Gamberini, R., Rimini, B., & Regattieri, A. (2018). Machine learning for multi-criteria inventory classification applied to intermittent demand. *Production Planning & Control, 30,* 76–89.
10. Priore, P., Ponte, B., Rosillo, R., & de la Fuente, D. (2018). Applying machine learning to the dynamic selection of replenishment policies in fast-changing supply chain environments. Production Planning & Control.
11. Xu, X., Zuo, L., & Huang, Z. (2014). Reinforcement learning algorithms with function approximation: Recent advances and applications. *Information Sciences, 261,* 1–31.