Computers & Operations Research 147 (2022) 105968

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

Intelligent inventory management approaches for perishable pharmaceutical products in a healthcare supply chain

Ehsan Ahmadi [a,*], Hadi Mosadegh [b], Reza Maihami [c], Iman Ghalehkhondabi [d], Minghe Sun [e], Gürsel A. Süer [f]

[a] Stetson-Hatcher School of Business, Mercer University, Atlanta, GA 30341, USA
[b] Department of Industrial Engineering & Management Systems, Amirkabir University of Technology, Tehran, Iran
[c] Department of Management and Marketing, East Tennessee State University, Johnson City, TN 37614, USA
[d] Department of Information Systems & Supply Chain Management, School of Business, Howard University, Washington, D.C. 20059, USA
[e] Department of Management Science and Statistics, University of Texas at San Antonio, San Antonio, TX 78249, USA
[f] Department of Industrial and Systems Engineering, Ohio University, Athens, OH 45701, USA

ARTICLE INFO

Keywords:
Healthcare systems
Inventory management
Reinforcement learning
Q-learning
Deep Q-network
Stochastic programming
Perishable pharmaceutical products

ABSTRACT

This study develops intelligent inventory management (IIM) approaches for managing perishable pharmaceutical products in a healthcare supply chain consisting of multiple regional hospitals and a central warehouse. The purchase price of the product is assumed to be age dependent, and the age distribution of the product is assumed to be known and is shared among all the supply chain members. Each agent representing a regional hospital or the central warehouse in the supply chain replenishes its stocks from its upstream agent. In IIM approaches, reinforcement learning methods, specifically Q-learning and Deep Q-network, are used to construct inventory policies. The IIM policies provide hospitals with near-optimal order quantities and remaining life distributions for products they order. Through many test instances, the performance of the IIM policies is compared to that of periodic review (R,s,S) policies obtained through a stochastic mixed integer programming model, solved using CPLEX and a genetic algorithm. The computational results demonstrate that the IIM policies are more cost-effective than the (R,s,S) policies, although the genetic algorithm has a speed advantage. Moreover, as compared with the (R,s,S) polices, the IIM polices have a lower possibility of product shortage, and thereby a higher service level for the patients, and a lower risk of product expiration. By implementing the IIM approaches, a healthcare system may also save storage space, in addition to having a lower total inventory cost. Sensitivity analyses are performed to derive managerial insights on the performance of the policies when the values of the key cost parameters change.

1. Introduction

The global healthcare expenditures contributed around 9.8 % to the global GDP in 2019 by reaching $8.5 trillion (WHO, 2021). Medicines constitute about 10–30 % of the global healthcare expenditure (McKone-Sweet et al., 2005), while a significant portion of these medicines is lost due to poor inventory and logistics management (Boerma et al., 2009). Managing inventory and logistics of pharmaceutical products is among the most challenging tasks of healthcare supply chains due to the perishability and limited lifespan of these products. These products become obsolete if they are not consumed before their expiration dates.

Pharmaceutical manufacturers and healthcare supply chains often need to deal with the risk of their products becoming obsolete due to uncertainty in the market demand. To reduce this risk, manufacturers usually sell excess inventory to the overseas market before their products expire at discounted prices (Lee et al., 2014). On the one hand, hospitals and healthcare systems tend to buy fresh products to mitigate the risk of inventory obsolescence. On the other hand, they can benefit from the manufacturers' discounts by purchasing products at better prices while at the same time, if properly managed, minimizing the risk of inventory obsolescence. Another challenge in managing pharmaceutical products in the healthcare supply chain is to minimize the risk of product shortage. A shortage arises when a product is not available at the time when a patient needs it, which may be life-threatening to the

* Corresponding author.
E-mail address: ahmadi_e@mercer.edu (E. Ahmadi).

https://doi.org/10.1016/j.cor.2022.105968
Received 3 June 2021; Received in revised form 14 June 2022; Accepted 20 July 2022
Available online 25 July 2022

patient. Landis (2002) have reported some observations about the consequences of medicine shortage in hospitals.

The inventory control system plays a critical role in decreasing the risks of product shortage and expiration, as well as decreasing inventory-related costs. In an supply chain environment where demand is deterministic and unsatisfied demand is penalized, the optimal ordering policy for each supply chain member is shown to be a "one for one" policy, in which each member places an order with its upstream supplier whenever an order is received from its downstream customer (Kimbrough et al., 2002). However, the uncertainty in demand for pharmaceutical products at hospitals, which is tied to the number of patients and their treatment procedures, makes it much harder to find the optimal ordering policy. Such uncertainty has inspired researchers to incorporate the stochastic nature of patient arrivals into the traditional inventory control models.

Among the available inventory policies in the healthcare setting, the (R,S), i.e., the periodic automatic replenishment level, inventory control policy is widely used due to its simplicity (Ahmadi, Masel et al., 2018). Under the (R,S) policy, an order is placed to increase the inventory position up to the level S in every R periods. The periodic review (R,s,S) model is also used by many researchers in which the inventory position is reviewed every R periods and an order is placed to bring the inventory position up to the level S if it is below the reorder point s when reviewed (Ahmadi et al., 2020, 2022; Pauls-Worm et al., 2014; Qiu et al., 2017).

In order to deal with the challenge of demand uncertainty, different approaches, including stochastic programming (Ahmadi et al., 2020; Azaron et al., 2008; Fattahi & Govindan, 2018; Kim et al., 2015), dynamic programming (Chen & Wei, 2012; Qiu et al., 2017), fuzzy logic (Dai & Zheng, 2015; Niakan & Rahimi, 2015; Soleimani et al., 2017), robust optimization (Ahmadi et al., 2019; Qiu, Hou et al., 2021, Qiu, Sun & Sun, 2021, Qiu, Yu & Sun, 2021, Qiu et al., 2020, 2017; Soleimani et al., 2017; Sun et al., 2022; Zahiri et al., 2018) and Markov decision process (Giannoccaro & Pontrandolfo, 2002; Guerrero et al., 2013; Vila-Parrish et al., 2008), have been developed in the literature. These approaches, however, often rely on restrictive assumptions that real-world problems can hardly meet. For example, in the stochastic programming approaches, one assumption is that the random demand has a finite set of possible realizations with a known probability of occurrence for each realization. Furthermore, most of these approaches are not specifically developed for perishable pharmaceutical products in a healthcare environment.

In the era of artificial intelligence, with the recent advancements in machine learning techniques, inventory management systems are shifting from the traditional models toward more intelligent systems. In this regard, machine learning techniques can be utilized to mitigate the challenges in inventory systems by providing more flexible solutions to resemble real-world situations. With this consideration, this work strives to determine the optimal inventory policy for a healthcare supply chain, where a central warehouse is responsible for making purchasing decisions from external sources and distributing the products among regional hospitals. The demands for the products in the hospitals are assumed to be uncertain. To this end, intelligent inventory management (IIM) approaches are developed by utilizing two reinforcement learning (RL) techniques to find the near-optimal inventory policies for the problem of interest. A periodic review stochastic mixed integer programming (SMIP) model is also developed by adopting the commonly used (R,s,S) inventory policy, solved by CPLEX®[1] and a genetic algorithm (GA), as benchmarks to measure the performance of the IIM approaches. The solutions obtained from the IIM approaches and the SMIP model are evaluated under a simulated inventory system.

The rest of this paper is structured as follows. In Section 2, the literature relevant to periodic review inventory control models and applications of RL techniques to inventory management is reviewed. In Section 3, the periodic review (R,s,S) SMIP model is presented. The proposed IIM approaches are described in detail in Section 4. Section 5 presents the computational results for the IIM and the (R,s,S) policies using test instances with different sizes and an example from the literature. The managerial insights derived from this study are presented in Section 6. Finally, conclusions are provided and future research avenues are discussed in Section 7.

2. Literature review

The literature relevant to periodic review inventory control models of perishable products is reviewed first. The literature on the applications of RL techniques to inventory management is then reviewed.

2.1. Periodic review inventory control for perishable products

Numerous studies have developed optimization models for inventory management of perishable products (e.g., Guan et al., 2022; Johansson and Olsson, 2018; Karaesmen et al., 2011; Maihami et al., 2021; Maihami and Karimi, 2014), but not all of them deployed to pharmaceutical products in multi-echelon healthcare systems. Guerrero et al. (2013) developed and solved a Markov chain model for an order-up-to-level inventory control policy for infusion solutions in a hospital by considering stochastic demands. Vila-Parrish et al. (2008) used the Markov decision process to determine the inventory ordering policies for perishable medicines for a hospital pharmacy. Kelle et al., (2012) developed a (R, s, S) inventory control policy in the pharmaceutical supply chain context with space constraints. Saedi et al. (2016) proposed a stochastic inventory optimization model to mitigate the pharmaceutical industry shortage costs. Akbarpour et al., (2020) proposed a bi-objective model for designing a supply chain for perishable pharmaceutical items with demand uncertainty. Wu et al. (2013) and Wang et al. (2015) incorporated system dynamic approaches to cope with the high shortage of medicines and surgical supplies in hospitals. Some researchers have particularly investigated the distribution of blood, as a perishable product, from transfusion centers to hospitals (Brodheim & Prastacos, 1979; Katsaliaki & Brailsford, 2007; Prastacos, 1984).

In a broader context, the body of the literature for inventory management of perishable products can be partitioned into two main categories of, i.e., periodic review and continuous review, inventory control models. Since this study is positioned in the periodic review inventory models, only this category of papers is surveyed.

Lagodimos and Koukoumialos (2008) proposed a periodic review inventory control policy for non-perishable items in a two-echelon supply chain. Broekmeulen and van Donselaar (2009) proposed a periodic review inventory policy for perishable products by incorporating the age distribution of the products in the model. van Donselaar and Broekmeulen (2012) then used simulation to extend the model of Broekmeulen and van Donselaar (2009) for estimating the quantity of inventory that are expiring. Qiu et al. (2017) considered demand uncertainty for a single-product and developed a periodic review (R, s, S) inventory policy utilizing robust optimization and dynamic programming. Kaya and Ghahroodi (2018) used dynamic programming to formulate an inventory and pricing decision model for perishable products with age and price-dependent demand. While most of the studies discussed above considered fixed shelf lives of the products in their models, Kouki and Jouini (2015) considered random shelf lives in

[1] https://www.ibm.com/analytics/cplex-optimizer.

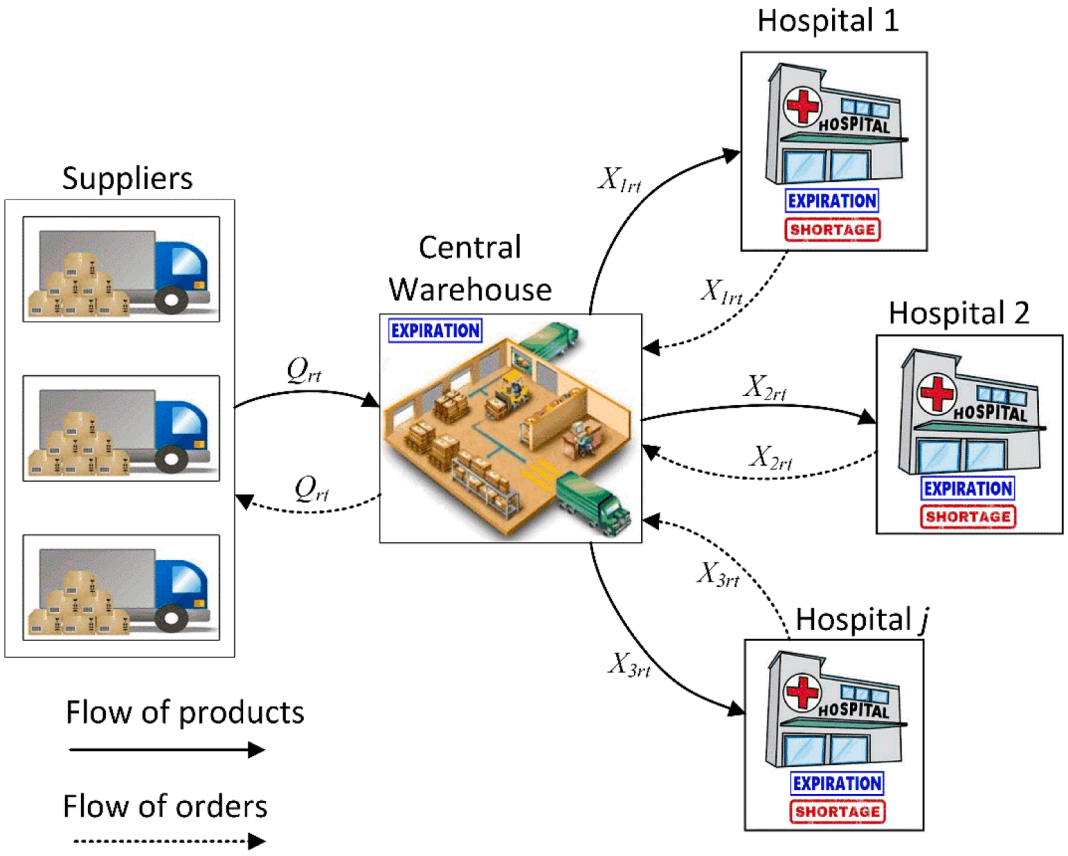

Fig. 1. General scheme of the considered healthcare supply chain.

developing a periodic review inventory control model for the perishable products. Qiu Yu and Sun (2021) and Guan et al. (2022) incorporated the behavior of the decision makers into the optimization of the inventory policy for perishable products. The readers are referred to the comprehensive reviews of inventory management for perishable products by Bakker et al. (2012), Chaudhary et al. (2018), Goyal and Giri (2001), Janssen et al. (2016), Pahl and Voß (2014), Perez and Torres (2020) and Raafat (1991).

2.2. Applications of RL techniques in inventory management

In recent years, with significant advancements in machine learning techniques, the traditional and easy-to-implement inventory control policies have shifted toward intelligence based inventory models. Giannoccaro and Pontrandolfo (2002) pioneered in this area by modeling the inventory decision in a supply chain using a semi-Markov process and the RL method to determine a near-optimal inventory ordering policy. Chaharsooghi et al. (2008) extended the model of Giannoccaro and Pontrandolfo (2002) by proposing a different action space, a reward function and a solution method, and showed that a GA could be as effective as the RL method for optimizing the inventory ordering policy. Qiu et al., (2020) used a data-driven approach to construct an uncertainty set for robust optimization using the available historical demand data to optimize inventory policies. Guan et al. (2022) used a hybrid robust stochastic optimization approach to optimize inventory policies for perishable products.

Sun and Zhao (2012) used Q-learning (QL), one type of RL techniques, for specifying ordering policies of a single product in a multi-echelon supply chain. Rana and Oliveira (2015) utilized a QL approach to determine the optimal pricing policy for perishable interdependent products where the demand for one product is affected by the prices of other products. Oroojlooyjadid et al. (2021) developed a Deep Q-network (DQN), another type of RL techniques, algorithm along with a transfer-learning approach to address the ordering decision in the beer game environment. Kara and Dogan (2018) employed QL, a variation of QL called SARSA, and GAs to address ordering decisions for perishable products in a single-product single-echelon inventory system and showed that QL outperformed two other algorithms. Bharti et al. (2020) modeled a sequential four-echelon supply chain using QL to determine the appropriate order quantities.

This study extends the stochastic (R, s, S) inventory control policy for a perishable product with age dependent purchase prices in a healthcare supply chain (Ahmadi et al., 2020, 2022) by proposing IIM approaches. Therefore, the main contribution of this work is designing and implementing two RL, i.e., QL and DQN, algorithms to determine the near-optimal inventory policies and comparing their performances with traditional methods for a healthcare system. For easy reference, the inventory policy derived from the QL algorithm is referred to as *IIM_QL*, and that derived from the DQN algorithm is referred to as *IIM_DQN*, while *IIM_QL* and *IIM_DQN* are collectively referred to as IIM policies. The IIM policies are compared to (R, s, S) inventory control policies obtained with a SMIP model solved by CPLEX, referred to as (R, s, S)*_CPLEX*, and solved by a GA, referred to as (R, s, S)*_GA*, to benchmark their performance. The DQN algorithm and the GA are used to deal with the computational challenge for large instances of the problem. All the (R, s, S) and IIM policies are evaluated under the same simulation

conditions with respect to inventory-related costs, service level, and the quantity of expired products.

3. Problem description and the SMIP model

In the problem studied in this work, multiple regional hospitals are operated and managed by a healthcare system. Each hospital replenishes its stocks from the central warehouse, and the central warehouse is responsible for ordering and stocking items from various suppliers. Many regional healthcare systems have supply chains with a three-echelon structure including suppliers, a central warehouse, and local healthcare providers (Ash et al., 2022; Nasrollahi and Razmi, 2021). The underlying assumption in this healthcare system is that the suppliers, the central warehouse, and the hospitals share information about the age distribution of their inventories. Therefore, the hospitals and the central warehouse are able to place orders for the product with r periods of remaining life. The purchase price of a product is assumed to be age dependent, more specifically, a linear function of its remaining life, and the price of the product is 0 at expiration. Researchers usually assume that the purchase price of an item decreases linearly as the product ages (Abdel-Malek and Ziegler, 1988; Akbarpour et al., 2020). All products with different remaining lives are consolidated and shipped together from the central warehouse to the hospitals at the beginning of each period.

There is a risk of product expiry in the central warehouse and in the hospitals. Both the central warehouse and the hospitals are assumed to adopt the first expiry first out (FEFO) strategy. According to the FEFO strategy, products that are to expire sooner should have a higher priority for consumption. This strategy is commonly used for chemical and pharmaceutical products (Sazvar et al., 2016). Furthermore, the orders made by the hospitals are assumed to be fulfilled immediately, i.e., the ordering lead time is assumed to be near zero. Because the healthcare system operates locally, most of the products can be delivered on the same day of ordering. Because the planning time in this work is on a weekly basis, a significant lead time is not considered in the ordering process. The central warehouse receives orders from the hospitals and then ships the product to the hospitals immediately at the beginning of each period. Each hospital receives the product from the central warehouse at the beginning of the period; realizes the demand and consumes the product during the period; and reviews the inventory position, removes the expired products from the stock, records the shortage as a backorder and places a replenishment order with the central warehouse at the end of the period. Fig. 1 is a graphical representation of the structure of the healthcare supply chain in the problem addressed in this study.

Since patients are assumed to arrive at the hospitals at random, a two-stage SMIP model is adopted to account for demand uncertainties for the products in the central warehouse and the hospitals. A stage refers to a point in time when the values of the decision variables are determined. The first-stage decision variables are those whose values are determined before the random variables are realized, and the second-stage decision variables are those whose values are determined after the random variables are realized (Birge & Louveaux, 1997; King & Wallace, 2012). When solving the two-stage SMIP, a set of scenarios representing the possible realizations of the random variables is generated and used. In the following, a two-stage SMIP model is formulated to find the near-optimal (R, s, S) policy.

3.1. Notations

Parameters:

J	Number of hospitals
T	Number of periods
N	Number of scenarios
E	Longest shelf life in the number of periods
WI_{r0}	Initial inventory in the central warehouse with r periods of remaining life
D_{jt}^{ξ}	Demand for the product in hospital j at period t under scenario ξ
$Prob_{j}^{\xi}$	Probability of scenario ξ in hospital j
$DelCost_j$	Fixed delivery cost from the central warehouse to hospital j
$OrCost$	Fixed ordering cost from the suppliers by the central warehouse
$ShCost_j$	Unit product shortage cost in hospital j
$HHCost_j$	Unit holding cost in hospital j per period
$WHCost$	Unit holding cost in the central warehouse per period
$ExpCost$	Unit expiration cost of the product
$Price$	Purchase price of the product with remaining life equal to the longest shelf life (fresh product)
$MaxHS_j$	Capacity, i.e., the maximum available storage space for the product, in hospital j
$MaxWS$	Capacity, i.e., the maximum available storage space for the product, in the central warehouse
M	A big number

Variables:

HI_{jrt}^{ξ}	Inventory level in hospital j with r periods of remaining life at the end of period t under scenario ξ
WI_{rt}	Inventory level in the central warehouse with r periods of remaining life at the beginning of period t
X_{jrt}	Quantity of the product with r periods of remaining life ordered by hospital j from the central warehouse at period t
Q_{rt}	Quantity of the product with r periods of remaining life ordered by the central warehouse from the suppliers at period t
Sh_{jt}^{ξ}	Product shortage in hospital j at period t under scenario ξ
Y_{jt}	Binary variable: 1 if the product is delivered to hospital j at the beginning of period t, and 0 otherwise
A_{rt}	Binary variable: 1 if the inventory level of the product with r periods of remaining life at the central warehouse at the beginning of period t is greater than 0, and 0 otherwise
B_t	Binary variable: 1 if the inventory level of the product at the central warehouse at the beginning of period t is less than s, and 0 otherwise
V_{jrt}^{ξ}	Binary variable: 1 if the product with r periods of remaining life in hospital j at period t under scenario ξ is not completely consumed, and 0 otherwise
$ExpWar_t$	Quantity of expired product in the central warehouse at the end of period t
$ExpHos_{jt}^{\xi}$	Quantity of expired product in hospital j at the end of period t under scenario ξ
d_{jrt}^{ξ}	Usage of the product in hospital j with r periods of remaining life during period t under scenario ξ
s	Reorder point in the (R, s, S) policy
S	Maximal inventory, i.e., the order-up-to-level, allowed for the product in a (R, s, S) policy

It is necessary to make warehouse decisions, i.e., determining s, S and X_{jrt}, before realizing hospital demands. Consequently, the warehouse decision variables are not scenario dependent and act as the first-stage decision variables. Conversely, decisions made in hospitals such as Sh_{jt}^{ξ} and $ExpHos_{jt}^{\xi}$ are scenario dependent and are second-stage decision variables.

3.2. Objective function

The objective function represents the total cost of the supply chain to be minimized. The different components of the objective function are presented as follows:

$$Min\ Z = \sum_{t=1}^{T}\left(Z_1(t)+Z_2(t)+Z_3(t)+Z_4(t)+\sum_{j=1}^{h}Z_5(t,j)+\sum_{j=1}^{h}Z_6(t,j)+\sum_{j=1}^{h}Z_7(t,j)+\sum_{j=1}^{h}Z_8(t,j)\right) \quad (1)$$

Purchase cost $Z_1(t) : Z_1(t) = \sum_{r=1}^{E} Q_{rt} \bullet Price\left(\frac{r}{E}\right)$ (1.1)

Ordering cost $Z_2(t) : Z_2(t) = B_t \bullet OrCost$ (1.2)

Warehouse expiration cost $Z_3(t) : Z_3(t) = ExpWar_t \bullet Price$ (1.3)

Warehouse holding cost $Z_4(t) : Z_4(t) = \sum_{r=1}^{E} WI_{rt} \bullet WHCost$ (1.4)

Delivery cost between central warehouse and hospitals $Z_5(t,j) : Z_5(t,j) = Y_{jt} \bullet DelCost_j$ (1.5)

Expected hospital expiration cost $Z_6(t,j) : Z_6(t,j) = \sum_{\xi=1}^{N} Prob_j^{\xi} \bullet ExpHos_{jt}^{\xi} \bullet ExpCost$ (1.6)

Expected hospital holding cost $Z_7(t,j) : Z_7(t,j) = \sum_{\xi=1}^{N} Prob_j^{\xi} \bullet \sum\nolimits_{r=1}^{E} HI_{jrt}^{\xi} \bullet HHCost_j$ (1.7)

Expected shortage cost $Z_8(t,j) : Z_8(t,j) = \sum_{\xi=1}^{N} Prob_j^{\xi} \bullet Sh_{jt}^{\xi} \bullet ShCost_j$ (1.8)

3.3. Warehouse constraints

The constraints for the central warehouse are:

$$WI_{rt} = WI_{(r+1)(t-1)} + Q_{rt} - \sum_{j=1}^{J} X_{jrt} \quad \forall r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\} \quad (2)$$

$$WI_{rt} = Q_{rt} - \sum_{j=1}^{J} X_{jrt} \quad \forall r = E,\ t \in \{1,\cdots,T\} \quad (3)$$

$$ExpWar_t = WI_{1t} \quad \forall t \in \{1,\cdots,T\} \quad (4)$$

$$\sum_{r=1}^{E} WI_{rt} - ExpWar_t - s \le M \bullet (1-B_t) \quad \forall t \in \{1,\cdots,T\} \quad (5)$$

$$\sum_{r=1}^{E} WI_{rt} - ExpWar_t - s \ge -M \bullet B_t + 1 \quad \forall t \in \{1,\cdots,T\} \quad (6)$$

$$\sum_{r=1}^{E} WI_{r(t-1)} - ExpWar_t + \sum_{r=1}^{E} Q_{rt} - S \ge -M \bullet (1-B_t) \quad \forall t \in \{1,\cdots,T\} \quad (7)$$

$$\sum_{r=1}^{E} WI_{r(t-1)} - ExpWar_t + \sum_{r=1}^{E} Q_{rt} - S \le M \bullet (1-B_t) \quad \forall t \in \{1,\cdots,T\} \quad (8)$$

$$Q_{rt} \le M \bullet B_t \quad \forall r \in \{1,\cdots,E\},\ t \in \{1,\cdots,T\} \quad (9)$$

$$s+1 \le S \quad (10)$$

$$\sum_{r=1}^{E} X_{jrt} \le M \bullet Y_{jt} \quad \forall j \in \{1,\cdots,J\},\ t \in \{1,\cdots,T\} \quad (11)$$

$$WI_{rt} \ge -M \bullet (1-A_{rt}) + 1 \quad \forall r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\} \quad (12)$$

$$WI_{rt} \le M \bullet A_{rt} \quad \forall r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\} \quad (13)$$

$$\sum_{r'=r+1}^{E} X_{jr't} \ge -M \bullet (1-A_{rt}) \quad \forall j \in \{1,\cdots,J\},\ r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\} \quad (14)$$

$$\sum_{r'=r+1}^{E} X_{jr't} \le M \bullet (1-A_{rt}) \quad \forall j \in \{1,\cdots,J\},\ r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\} \quad (15)$$

$$\sum_{r=1}^{E-1} WI_{(r+1)(t-1)} + \sum_{r=1}^{E} Q_{rt} \le MaxWS \quad \forall t \in \{1,\cdots,T\} \quad (16)$$

Constraints (2) and (3) are the inventory balancing, also called the conservation of flow, constraints for the central warehouse. Constraints (4) model the quantity of the products that are expired at the end of each period in the central warehouse. In the (R, s, S) policy, after each review for every R periods, the central warehouse places an order with the suppliers to bring the inventory level up to S if the inventory level is below the reorder point s. Constraints (5)–(9) are used to model these requirements. Constraint (10) ensures that the reorder point s is not greater than the order-up-to-level S. Constraints (11) record whether or not the product is shipped from the central warehouse to the hospitals at the beginning of each period. Constraints (12)–(15) are used to model the FEFO strategy. Constraints (16) restrict the space used by the inventory not to exceed the capacity of the central warehouse in each period.

3.4. Hospital constraints

The set of constraints for the individual hospitals are:

$$HI^{\xi}_{jrt} = HI^{\xi}_{j(r+1)(t-1)} + X_{jrt} - d^{\xi}_{jrt} \quad \forall j \in \{1,\cdots,J\},\ r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\},\ \xi \in \{1,\cdots,N\} \tag{17}$$

$$HI^{\xi}_{jrt} = X_{jrt} - d^{\xi}_{jrt} \quad \forall j \in \{1,\cdots,J\},\ r = E,\ t \in \{1,\cdots,T\},\ \xi \in \{1,\cdots,N\} \tag{18}$$

$$ExpHos^{\xi}_{jt} = HI^{\xi}_{j1t} \quad \forall j \in \{1,\cdots,J\},\ \xi \in \{1,...,N\} \tag{19}$$

$$\sum_{r=1}^{E} d^{\xi}_{jrt} + Sh^{\xi}_{jt} = D^{\xi}_{jt} + Sh^{\xi}_{j(t-1)} \quad \forall j \in \{1,\cdots,J\},\ t \in \{1,\cdots,T\},\ \xi \in \{1,...,N\} \tag{20}$$

$$HI^{\xi}_{jrt} \geq -M \bullet \left(1 - V^{\xi}_{jrt}\right) + 1 \quad \forall j \in \{1,\cdots,J\},\ r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\},\ \xi \in \{1,...,N\} \tag{21}$$

$$HI^{\xi}_{jrt} \leq M \bullet \left(1 - V^{\xi}_{jrt}\right) \quad \forall j \in \{1,\cdots,J\},\ r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\},\ \xi \in \{1,...,N\} \tag{22}$$

$$\sum_{r'=r+1}^{E} d^{\xi}_{jr't} \geq -M \bullet \left(1 - V^{\xi}_{jrt}\right) \quad \forall j \in \{1,\cdots,J\},\ r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\},\ \xi \in \{1,...,N\} \tag{23}$$

$$\sum_{r'=r+1}^{E} d^{\xi}_{jr't} \leq M \bullet \left(1 - V^{\xi}_{jrt}\right) \quad \forall j \in \{1,\cdots,J\},\ r \in \{1,\cdots,E-1\},\ t \in \{1,\cdots,T\},\ \xi \in \{1,...,N\} \tag{24}$$

$$\sum_{r=1}^{E-1} HI^{\xi}_{j(r+1)(t-1)} + \sum_{r=1}^{E} X_{jrt} \leq MaxHS_j \quad \forall j \in \{1,\cdots,J\},\ t \in \{1,\cdots,T\} \tag{25}$$

X_{jrt}, Q_{rt}, WI_{rt}, HI^{ξ}_{jrt}, Sh^{ξ}_{jt}, $ExpWar_t$, $ExpHos^{\xi}_{jt}$, $d^{\xi}_{jrt} \geq 0$ and integer $\forall j \in \{1,\cdots,J\}$,

$$r \in \{1,\cdots,E\},\ t \in \{1,\cdots,T\},\ \xi \in \{1,2,..,N\} \tag{26}$$

Constraints (17) and (18) are the inventory balancing, or the conservation of flow, constraints for the hospitals. Constraints (19) model the quantity of the product expired in the hospitals at the end of each period. Constraints (20) restrict that the shortage is backordered. Constraints (21)–(24) are used to model the FEFO strategy in the hospitals. Constraints (25) guarantee that the space used by the inventory in each hospital does not exceed the corresponding capacity. Constraints (26) restricts the decision variables, for both hospitals and warehouse, to be in their respective domains.

4. The intelligent inventory management approaches

With the advancements in artificial intelligence, machine learning techniques pave the way for designing better inventory management policies. Machine learning approaches can be categorized into four groups, i.e., *supervised learning*, *unsupervised learning*, *semi-supervised learning* and *reinforcement learning*. Techniques in different groups are used for different purposes. Supervised learning techniques are used for classification and, therefore, need a label for each observation in training dataset to discover the underlying patterns in the data (Ghanadian & Ghanbartehrani, 2021). Unlike supervised learning, techniques in unsupervised learning are for clustering and can find the hidden patterns in the data even the observations in the dataset are unlabeled. Techniques in semi-supervised learning can learn from datasets with both labeled and unlabeled observations (Albalate & Minker, 2013). RL techniques are derived from supervised learning and dynamic programming, and can learn through constant interaction with the environment (Kulkarni, 2012). The RL algorithms can also be described as simulation-based stochastic methods that have been proven to be very efficient for large-size Markov decision processes (Firdausiyah et al., 2019; Giannoccaro & Pontrandolfo, 2002).

One type of RL methods is the QL algorithm, which has been successfully applied by many researchers in making optimal decisions in a stochastic environment (Ahmadi, Goldengorin et al., 2018; Mosadegh et al., 2020). In the QL algorithm, an agent constantly interacts with the environment by taking actions through two mechanisms, i.e., explorations and exploitations, to learn the optimal policy. The environment responds to the agent by returning an *immediate reward*, and then the agent moves to a new *state* (Mosadegh et al., 2020; Sutton & Barto, 2018). The goodness of an action is assessed by recording the *Q-value* in a so-called *Q-table* for each state-action pair, where a good action will be awarded by receiving a high reward while a poor action will be punished by receiving a low reward. A good action would result in a high immediate reward and/or transition to another state with a high earned reward (Giannoccaro & Pontrandolfo, 2002). The agent learns to maximize the values of the rewards, leading to the best action in each state. The agents represent the central warehouse and the hospitals of the healthcare system in the inventory problem addressed in this study.

While the QL algorithm offers many advantages, the curse of dimensionality limits its applicability to large practical problems. With real-world problems, the state space and/or action space grow exponentially, making it intractable for the QL algorithm to explore all state-action pairs in a reasonable amount of time. Therefore, some of the state-action pairs remain unvisited during the exploration process, ending up with a local optimal policy. To address this challenge, researchers proposed the use of linear and non-linear approximators to estimate the *Q-values* without the need to visit all state-action pairs (Li, 2017; Sutton & Barto, 2018). The non-linear approximators such as artificial neural

networks (ANNs) have shown superior performance over the linear ones, but are known to be unstable due to the correlation in the sequence of observations. Mnih et al. (2015) tackled this problem by developing the *experience replay memory* and *target network* mechanisms in a proposed DQN algorithm. The DQN algorithm utilizes an ANN, with the state variables being in the input layer and the action variables being in the output layer, to estimate the Q-values.

4.1. RL algorithm design

The different components of the RL algorithms are described in this subsection. After the state variables and the action variables are defined, the reward mechanism and the action strategies are discussed.

4.1.1. State variables

For the problem addressed in the study, an agent represents a hospital or the central warehouse, the state of each agent is defined as the inventory position of the product of that agent at a certain time, and the state of an agent at time t is represented by the value of the state variable. Thus, the discrete state space for the agents representing the hospitals can be represented by the set $HS_j(t) = \left\{\sum_{r=1}^{E} HI_{jrt}\right\}, \forall j \in \{1, \cdots, J\}$, $t \in \{1, \cdots, T\}$, and the discrete state space for the agent representing the central warehouse can be represented by the set $WS(t) = \left\{\sum_{r=1}^{E} WI_{rt}\right\}$, $t \in \{1, \cdots, T\}$. In $HS_j(t)$, HI_{jrt} is the inventory position of the product with r periods of remaining life in hospital j at time t. Likewise in $WS(t)$, WI_{rt} is the inventory position of the product with r periods of remaining life in the central warehouse at time t. It is clear that $0 \leq HS_j(t) \leq MaxHS_j$ and $0 \leq WS(t) \leq MaxWS$ hold due to the capacity constraints.

4.1.2. Action variables

The action of each agent at time t is defined as a vector describing the age distribution of the ordered product. Therefore, $HA_j(t) = \{X_{jrt} | r \in \{1, \cdots, E\}\}$ and $WA(t) = \{Q_{rt} | r \in \{1, \cdots, E\}\}$ represent the discrete action space of hospital j and the central warehouse, respectively, where X_{jrt} and Q_{rt} are the quantities of the product with r periods of remaining life ordered by hospital j from the central warehouse and ordered by the central warehouse from the suppliers, respectively, at time t. The central warehouse is assumed to have enough capacity to handle all orders received from all hospitals, i.e., shortage is not allowed in the central warehouse. Thus, the constraints $Q_{rt} + WI_{(r+1)(t-1)} \geq \sum_{j=1}^{J} X_{jrt}$, $\forall r \in \{1, \cdots, E\}, t \in \{1, \cdots, T\}$, are included in the QL algorithm when the agent representing the central warehouse is trained. In the IIM approaches, the agents representing hospitals are first trained to optimize their ordering decisions, X_{jrt}, and the agent representing the central warehouse is then trained to optimize its ordering decision, Q_{rt}, by considering the optimal values of X_{jrt}. For the (R, s, S) policies, however, the optimal values for X_{jrt} and Q_{rt} are determined simultaneously, and constraints (2) ensure that all orders received from the hospitals are fulfilled in the central warehouse.

4.1.3. Reward mechanism

At each time t, an agent visits a state i, takes an action represented by a, and then moves to state i'. The agent receives an immediate reward as a consequence of the action taken. The insights from the work of Sun and Zhao (2012) are used to define the immediate rewards for the agents representing the hospitals and the central warehouse, respectively, as follows,

$$HIR_j(t) = C_j - \left(Z_5(t,j) + Z_6(t,j) + Z_7(t,j) + Z_8(t,j) + \sum_{r=1}^{E} X_{jrt} \cdot Price\left(\frac{r}{E}\right) \right), \tag{27}$$

$$WIR(t) = C - (Z_1(t) + Z_2(t) + Z_3(t) + Z_4(t)), \tag{28}$$

where C_j is a constant for hospital j and C is a constant for the central warehouse. The values for C_j and C are set experimentally large enough to be an upper bound for the cost components.

The QL algorithm records a Q-value, denoted by $Q(i, a)$, for each state-action pair. At every time t, the Q-value is updated according to (29) in the following.

$$Q(i,a) \leftarrow (1-\alpha) Q(i,a) + \alpha \left[IR + \gamma \max_{a'} Q(i', a') \right], \tag{29}$$

where $IR = HIR_j(t)$ and $IR = WIR(t)$ for the hospitals and the central warehouse, respectively, $\alpha \in [0, 1]$ is a learning rate and $\gamma \in [0, 1]$ is a discount factor that gives a weight to future rewards. The Q-values are recorded in the Q-table which maps the states to the actions. Given an optimal Q-table, the optimal policy π^* for the agent in state i can be derived by.

$$\pi^* = \underset{a}{argmax} Q^*(i, a). \tag{30}$$

4.1.4. Action selection strategies

A wide range of selection strategies is examined for the QL and DQN algorithms, including ε-greedy, roulette-wheel, ε-greedy with linearly and exponentially decayed parameters, and combinations of them. In the QL algorithm, actions are selected based on the ε-greedy and the roulette-wheel mechanisms, in which an agent takes a random action with probability ε, a greedy action with probability η, and a roulette-wheel selection with probability θ, with $\varepsilon + \eta + \theta = 1$. In a given state, the greedy action is the one that has the greatest Q-value. For the roulette-wheel mechanism, the probability of each action for being selected is proportional to its Q-value. With the DQN algorithm, the ε-greedy strategy, where ε decays exponentially with parameter μ, has shown good performance.

4.2. Training processes

The agents representing the hospitals act independently and are trained to optimize their own policies. The agent representing the central warehouse aggregates the optimized replenishment polices received from the hospitals and is trained to optimize its ordering policy with the suppliers. The goal of training the warehouse agent is to find the optimal ordering policy that respects all hospital replenishment polices.

4.2.1. QL training

During every training episode k, the agents run for T periods. At every time period, an agent reviews its inventory position to determine the current state i and then takes an action a, i.e., a decision on the order quantity and the ages of the product, to receive an immediate reward according to (27) or (28). The corresponding value of $Q(i, a)$ is updated based on (29). In each time period t, the set of state-actions $SA(t) = \{(i, a)\}$ leading to a change in the current policy π is recorded. Within each training episode k, after the agent runs for T periods, the current policy π is extracted from the updated Q-table and the performance of the policy is evaluated in terms of the cost imposed on the inventory

system. Finally, before proceeding to the next episode, the Q-values corresponding to the set of state-actions $SA(t)$ for the agents representing the hospitals and warehouse are updated. The Q-value for hospital j is updated according to (31) in the following:

$$Q(i,a) \leftarrow Q(i,a) + F_j - cost_j(\pi), \quad \forall\, i \text{ and } a \in SA(t), \tag{31}$$

where $cost_j(\pi) = \sum_{t=1}^{T}\left(Z_5(t,j) + Z_6(t,j) + Z_7(t,j) + Z_8(t,j) + \sum_{r=1}^{E} X_{jrt} \cdot Price\left(\frac{r}{E}\right)\right)$ is the cost for hospital j with an experimentally determined upper bound F_j. The Q-value for the central warehouse is updated according to (32) in the following:

$$Q(i,a) \leftarrow Q(i,a) + F - cost(\pi), \quad \forall\, i \text{ and } a \in SA(t), \tag{32}$$

where $cost(\pi) = \sum_{t=1}^{T}(Z_1(t) + Z_2(t) + Z_3(t) + Z_4(t))$ is the cost for the central warehouse with an upper bound F. The idea behind this procedure is that the actions leading to better policies in the learning process receive higher rewards while the actions not leading to policy improvement receive lower rewards as penalties. This approach speeds up the convergence of the QL algorithm to the optimal Q-table. The Pseudo-codes of the QL algorithms for training the agents representing the hospitals and the central warehouse are presented in Algorithms 1 and 2, respectively.

Algorithm 1. Pseudo-code of the QL algorithm for training the agents representing the hospitals

Input: parameters of the model: J, E, T, N, $DelCost_j$, $ShCost_j$, $HHCost_j$, $ExpCost$, $MaxHS_j$, $Price$; and parameters of the QL algorithm: α, γ, ε, η, F, C_j, num_eps;

Initialize the Q-table;
For $k = 1$ **To** num_eps **Do**
  $i \leftarrow$ *initial inventory position*;
  $Q\_values \leftarrow 1$;
  For $t = 1$ **To** T **Do**
    Sample *demand* from the specified distribution for each hospital j;
    $rnd \leftarrow rand()$;
    If $\{(rnd \le \varepsilon)$ **or** $(t = 0$ **and** $k = 0)\}$ **Then**
      $a \leftarrow random\_action()$;
    Else
      If $\{(\varepsilon < rnd \le \varepsilon + \eta$ **and** $t > 0)$ **or** $(\varepsilon < rnd \le \varepsilon + \eta$ **and** $k > 0)\}$ **Then**
        $a \leftarrow greedy\_action()$;
      Else
        $a \leftarrow roulette\_wheel()$;
      End If
    End If
    Compute the immediate reward *HIR* and *inventoryposition*;
    $i' \leftarrow inventoryposition$;
    $\pi \leftarrow \underset{a}{\operatorname{argmax}} Q(i,a)$;
    $Q(i,a) \leftarrow (1-\alpha)Q(i,a) + \alpha\left[HIR + \gamma \max_{a'} Q(i', a')\right]$;
    $\pi' \leftarrow \underset{a}{\operatorname{argmax}} Q(i,a)$;
    If $\pi \neq \pi'$ **Then**
      $SA(t) \leftarrow (i,a)$;
    End If
    $i \leftarrow i'$;
  End For
  $Q(i,a) \leftarrow Q(i,a) + F - cost(\pi'), \forall ian da \in SA(t)$;
End For
$\pi^* \leftarrow \underset{a}{\operatorname{argmax}} Q(i,a)$;

Output: π^*

Algorithm 2. Pseudo-code of the QL algorithm for training the agent representing the central warehouse

Input: parameters of the model: J, E, T, N, $OrCost$, $WHCost$, $ExpCost$, $MaxWS$, $Price$; and parameters of the QL algorithm: α, γ, ε, η, F, C, num_eps, π_j^*;

Initialize the Q-table
For $k = 1$ **To** num_eps **Do**
  $i \leftarrow initialinventoryposition$;
  $Q\_values \leftarrow 1$;
  For $t = 1$ **To** T **Do**
    Sample *demand* from the specified distribution for each hospital j;
    Determine the current state for each hospital j and select the optimal policy π_j^* using Algorithm 1;
    Demand for the warehouse $\leftarrow \sum_{j=1}^{J} a_j | a_j \in \pi_j^*$;
    $rnd \leftarrow rand()$;
    If $\{(rnd \le \varepsilon)$ **or** $(t = 0$ **and** $k = 0)\}$ **Then**
      $a \leftarrow random\_action()$;
    Else
      If $\{(\varepsilon < rnd \le \varepsilon + \eta$ **and** $t > 0)$ **or** $(\varepsilon < rnd \le \varepsilon + \eta$ **and** $k > 0)\}$ **Then**
        $a \leftarrow greedy\_action()$;
      Else
        $a \leftarrow roulette\_wheel()$;
      End If
    End If
    Compute the immediate reward *WIR* and *inventoryposition*;
    $i' \leftarrow inventoryposition$;
    $\pi \leftarrow \underset{a}{\operatorname{argmax}} Q(i,a)$;
    $Q(i,a) \leftarrow (1-\alpha)Q(i,a) + \alpha\left[WIR + \gamma \max_{a'} Q(i', a')\right]$;
    $\pi' \leftarrow \underset{a}{\operatorname{argmax}} Q(i,a)$;
    If $\pi \neq \pi'$ **Then**
      $SA(t) \leftarrow (i,a)$;
    End If
    $i \leftarrow i'$;
  End For
  $Q(i,a) \leftarrow Q(i,a) + F - cost(\pi'), \forall ian da \in SA(t)$;
End For
$\pi^* \leftarrow \underset{a}{\operatorname{argmax}} Q(i,a)$;

Output: π^*

4.2.2. DQN training

The DQN algorithm uses an ANN to predict the Q-values corresponding to each action in a given state. The loss function in the ANN is the mean square error between the predicted Q-values, represented by y_n, derived from the target network with parameters, i.e., weights, θ^-, and the Q-values obtained from the Q-network with parameters θ. The parameters of the target network, θ^-, are only updated after c iterations. The pseudocodes of the DQN algorithms for training the agents representing the hospitals and the central warehouse are presented in Algorithms 3 and 4, respectively.

Algorithm 3. Pseudo-code of the DQN algorithm for training the agents representing the hospitals

Input: parameters of the model: J, E, T, N, $DelCost_j$, $ShCost_j$, $HHCost_j$, $ExpCost$, $MaxHS_j$, $Price$; and parameters of the DQN algorithm: $\gamma, \varepsilon, \mu, C_j$, num_eps, M;

Initialize the *experience replay memory D*, Q-network weights θ, and target network weights θ^-;
For $k = 1$ **To** num_eps **Do**
  $i \leftarrow initialinventoryposition$;
  For $t = 1$ **To** T **Do**
    Sample *demand* from the specified distribution for each hospital j;
    $rnd \leftarrow rand()$;
    If $rnd \le \varepsilon$ **Then**
      $a \leftarrow random\_action()$;

(continued on next page)

(*continued*)

Algorithm 3. Pseudo-code of the DQN algorithm for training the agents representing the hospitals

```
Else
    a←greedy_action();
End If
Execute action a and compute the immediate reward HIR and inventoryposition;
i' ←inventoryposition;
Store experience (i, a, HIR, i') in D;
Sample a mini-batch of experience (i_n, a_n, s'_n, HIR_n) with the size of M from D;
Set y_n = HIR_n + γ max_a Q(i'_n, a; θ⁻);
Run the ANN with loss function (1/M) Σ_n (y_n − Q(i_n, a_n; θ))²;
Update the target network parameters θ⁻ ←θ for every c iteration;
End For
ε← μ × ε;
End For
π* ←argmax_a Q(i, a);
```

Output: π^*

Algorithm 4. Pseudo-code of the DQN algorithm for training the agents representing the central warehouse

Input: parameters of the model: J, E, T, N, $DelCost_j$, $ShCost_j$, $HHCost_j$, $ExpCost$, $MaxHS_j$, $Price$; and parameters of the DQN algorithm: $\gamma, \varepsilon, \mu, C_j$, *num_eps*, M, π_j^*;

```
Initialize the experience replay memory D, Q-network weights θ, and target network weights θ⁻;
For k = 1 To num_eps Do
    i←initial inventory position;
    For t = 1 To T Do
        Sample demand from the specified distribution for each hospital j;
        Determine the current state for each hospital j and select the optimal policy π*_j using Algorithm 3;
        Demand for the warehouse ←Σ_{j=1}^{J} a_j | a_j ∈ π*_j
        rnd←rand();
        If rnd ≤ ε Then
            a← random_action();
        Else
            a← greedy_action();
        End If
        Execute action a and compute the immediate reward WIR and inventoryposition;
        i' ←inventoryposition;
        Store experience (i, a, WIR, i') in D;
        Sample a mini-batch of experience (i_n, a_n, s'_n, HIR_n) with the size of M from D;
        Set y_n = HIR_n + γ max_a Q(i'_n, a; θ⁻);
        Run the ANN with loss function (1/M) Σ_n (y_n − Q(i_n, a_n; θ))²;
        Update the target network parameters θ⁻ ←θ for every c iterations;
    End For
    ε← μ × ε;
End For
π* ←argmax_a Q(i, a);
```

Output: π^*

5. Numerical experiments

In this section, the performances of the IIM policies found by the QL and DQN algorithms are evaluated against the (R, s, S) policies found with the SMIP model using different test instances with different sizes presented in Table 1.

The (R, s, S) policies are obtained by an exact method, CPLEX, and an evolutionary algorithm, GA, introduced in Ahmadi et al. (2020). The values of the problem parameters of the benchmark instances are listed in Table 2. The SMIP problem is solved using IMB ILOG CPLEX Version 12.6 on a PC with an Intel® Core™ i7 CPU @1.1 GHz and with 16 GB RAM. The QL and DQN algorithms are developed using Python Version 3.8, and GA is developed with MATLAB R2020a, all running on the same PC.

Table 1
Designed test instances.

Test instance	Number of hospitals	*MaxWS*
TI1	3	70
TI2	5	100
TI3	10	200
TI4	15	260
TI5	20	340

Table 2
Problem parameters adopted from Ahmadi et al. (2020) and Kouki et al. (2015).

Index	Parameter	Value
E	Maximum shelf life	3, 4, 5, 6 weeks
T	Periods	10 weeks
N	Number of scenarios for hospitals	5, 10, 20, 40
$DelCost_j$	Fixed delivery cost per delivery from the central warehouse to hospital j	Randomly selected from [20, 100]
$OrCost$	Ordering cost per order by the central warehouse	\$100
$ShCost_j$	Shortage cost in hospital j	\$40 for all hospitals
$HHCost_j$	Unit holding cost per period in hospital j	\$2 for all hospitals
$WHCost$	Unit holding cost per period in the central warehouse	\$1/week/unit
$ExpCost$	Unit expiration cost of the product	\$13
$MaxHS_j$	Capacity, i.e., the maximum available space, for the product in hospital j	$MaxHS_1 = 30$, $MaxHS_2 = 23$, $MaxHS_j = 15 \forall j \in \{3, \cdots, 20\}$
$Price$	Purchase price for a fresh product	$Price$ = \$13

The initial inventories in the hospitals and in the central warehouse are assumed to be 0. Demands in hospitals 1 and 2 are assumed to follow uniform distributions $U(0, 20)$ and $U(0, 15)$, respectively. In all other hospitals, demand follows $U(0,10)$. For each hospital, a large number of initial scenarios, e.g., 1000, with equal probabilities are generated randomly from the prescribed uniform distributions. Each scenario reflects the realization of demand over $T = 10$ consecutive periods. A scenario backward reduction approach developed by Wu et al. (2007) is then employed in order to reduce the number of scenarios and compute their associated probabilities.

5.1. Parameter settings

The Taguchi method was used to tune the values of the parameters of the QL and DQN algorithms. The designed QL algorithm has four parameters, i.e., ε, η, α and γ. Four levels, i.e., values, for each parameter were considered as shown in Table 3. Similarly, the DQN algorithm also has four parameters, i.e., M, c, γ and μ, each with four levels, also shown in Table 3. Thus, a Taguchi design of $L_{16}(4^4)$ is implemented to determine the best value for each parameter of the algorithms.

The Taguchi method calibrates the values for the parameters through maximizing the signal-to-noise (S/N) ratio. According to the S/N ratios obtained by executing the Taguchi method using MINITAB 18, the selected parameter values for the agents representing the hospitals in the QL algorithm are $\varepsilon = 0.10$, $\eta = 0.10$, $\alpha = 0.15$ and $\gamma = 0.20$. These values indicate that an equal chance of 10 % is given to selecting the greedy or the random actions, while a chance of 80 % is given to selecting actions with the roulette-wheel selection mechanism. Furthermore, 15 % of the immediate rewards is used as a learned value and a weight of 20 % is given to the estimate of future rewards.

The parameter values for the agent representing the central ware-

Table 3
The values of the parameters for the QL and DQN algorithms.

Level	QL				DQN			
	ε	η	α	γ	M	c	γ	μ
1	0.10	0.10	0.15	0.20	50	50	0.20	0.995
2	0.20	0.25	0.35	0.40	100	100	0.40	0.997
3	0.30	0.35	0.55	0.60	500	150	0.60	0.998
4	0.45	0.55	0.75	0.80	2,000	200	0.80	0.999

Table 4
Results of the (R,s,S) and IIM policies across different test instances for a product with $E = 3$ periods of maximum shelf life.

Test instance	(R,s,S)						IIM				Improvement (%)
	CPLEX			GA			RL		DQN		
	Total cost ($)	Total cost of simulated solution ($)	Time (s)	Total cost ($)	Total cost of simulated solution ($)	Time (s)	Total cost ($)	Time (s)	Total cost ($)	Time (s)	
TI1	7,197	8,119	2,346	7,689	8,511	75	8,034	1,447	**7,497**	1,316	7.7
TI2	9,735	11,375	10,800	10,783	11,818	164	11,343	2,090	**10,195**	2,132	10.4
TI3	18,214	22,874	10,800	22,805	26,677	308	19,826	2,774	**19,106**	2,550	16.5
TI4	27,727	33,370	10,800	31,623	41,225	1,053	28,214	3,866	**27,676**	3,956	17.1
TI5	33,351	40,123	10,800	40,895	55,037	1,912	32,471	9,244	**32,359**	9,850	19.4

house in the QL algorithm are $\varepsilon = 0.20$, $\eta = 0.35$, $\alpha = 0.75$ and $\gamma = 0.20$. This agent performs the roulette-wheel with a 45 % chance and selects the random and greedy actions with 20 % and 35 % chances, respectively. The learning rate of $\alpha = 0.75$ indicates that a higher weight is given to the most recently learned knowledge, as opposed to the hospital agents, in which a higher weight is given to the prior knowledge. Overall, the agent representing the central warehouse emphasizes the exploration during the learning process, while the agents representing the hospitals stress exploitation rather than exploration for knowledge discovery.

The DQN algorithm also exhibits the same characteristics, where the warehouse agent prioritizes exploration and the hospital agents prioritize exploitation. The selected parameter values in the DQN algorithm are $M = 500$, $c = 200$, $\gamma = 0.2$ and $\mu = 0.997$ for the hospital agents, and $M = 500$, $c = 150$, $\gamma = 0.2$ and $\mu = 0.998$ for the central warehouse agent. The implemented ANN consists of two fully connected hidden layers, each containing 100 neurons, using rectified linear activation functions.

The agents for both the hospitals and the central warehouse in the DL and DQN algorithms are allowed to run $num\_eps = 5,000$ training episodes in the training process. The GA parameters are set in accordance with those in Ahmadi et al. (2020).

5.2. Experimental results

The performances of *IIM_QL* and *IIM_DQN* are compared with those of (R,s,S)*_CPLEX* and (R,s,S)*_GA* using different test instances. In Table 4, the results of the (R,s,S) policies are obtained with $N = 10$ scenarios, and the obtained stochastic, also known as *here-and-now* (HN), solutions are simulated for 1000 times. Due to their randomness, the results obtained with *IIM_QL*, *IIM_DQN* and (R,s,S)*_ GA* are the averages over $N = 10$ independent replications.

A time limit of 10,800 s is set for CPLEX in solving the SMIP models. CPLEX output the best, instead of the optimal, solution that it could find with an optimality gap when stopped prematurely after reaching this time limit. The QL and DQN algorithms run in parallel for training the agents representing the hospitals, and use the optimal policies of the hospitals as inputs for training the agent representing the central warehouse. Thus, the computational time for an IIM approach is the sum of the average computational time of training the agents representing the hospitals and the computational time of training the agent representing the central warehouse. The results reported in Table 4 indicate that the (R,s,S) policy generated with CPLEX produced solutions 23.7 % better on average than those generated with GA. However, GA runs 12.9 times faster on average than CPLEX, which is significant for large instances of the problem. *IIM_DQN* consistently generates better solutions than all other policies do. The *IIM_DQN* solutions are computed 2.3 times faster, and at the same time are 16.4 % better in quality, on average than the (R,s,S)*_CPLEX* solutions. In Table 4, the last column shows the percentage of improvement in total cost achieved with *IIM_DQN* policy over (R,s,S)*_CPLEX*. The SMIP model faces a considerable challenge in both the solution quality and computational time when the size of the problem increases.

In order to evaluate the impacts of the number of scenarios N on the solution quality and computational time for the (R,s,S)*_CPLEX* policy, each test instance is solved with different values of N. The expected value of perfect information (EVPI) and performance loss are computed for each test instance. The EVPI may be interpreted as the highest fee a decision maker is willing to pay for the perfect demand data and is the difference between the average total costs of the optimal solution obtained under each demand scenario, known as the *wait-and-see* (WS) solution, and the optimal stochastic, i.e., HN, solution. The performance loss is defined as the difference between the total costs of the simulated HN solution and the HN solution. A time limit of 10,800 s is set for CPLEX.

The results in Table 5 indicate that CPLEX obtains better solutions, but takes much more computation time, by increasing the number of scenarios. In addition, for a given test instance, a larger number of scenarios results in a larger EVPI, because the more scenarios are used, the more accurate the demand information is, and the more costly to obtain the solutions. With the TI2 instance, the performance loss is smaller with 10 scenarios than with 20 scenarios, because the HN solution has a 10 % optimality gap for 20 scenarios, but has a 2 % optimality gap for 10 scenarios. Therefore, a portion of the performance loss is due to the poor quality solutions obtained by CPLEX when more scenarios are used. Performance measures for TI4 with 40 scenarios and

Table 5
Impacts of the number of scenarios N on the solutions of (R,s,S)_CPLEX for the product with $E = 3$ periods of maximum shelf life.

Test instance	# Scenarios	Time (s)	HN	Optimality gap (%)	WS	Simulated HN	EVPI	Performance loss
TI1	5	249	5,492	0	3,590	7,787	1,902	2,295
	10	2,346	7,197	0	3,601	8,119	3,596	922
	20	10,800	7,442	9	3,582	8,291	3,860	850
	40	10,800	7,682	12	3,376	8,241	4,306	560
TI2	5	8,064	8,848	0	4,018	12,775	4,830	3,927
	10	10,800	9,735	2	4,161	11,375	5,575	1,640
	20	10,800	10,536	10	4,388	12,338	6,148	1,802
	40	10,800	10,612	13	4,032	12,440	6,580	1,828
TI3	5	10,800	14,602	8	6,736	23,989	7,866	9,387
	10	10,800	18,214	12	7,176	22,874	11,038	4,660
	20	10,800	20,170	17	7,182	23,279	12,988	3,109
	40	10,800	20,633	20	7,625	23,316	13,007	2,684
TI4	5	10,800	22,955	8	12,400	36,099	10,555	13,144
	10	10,800	27,727	11	13,000	33,370	14,728	5,643
	20	10,800	30,945	14	12,728	33,763	18,217	2,818
	40	10,800	135,438	80	—	—	—	—
TI5	5	10,800	26,830	8	15,675	42,618	11,155	15,789
	10	10,800	33,351	17	15,476	40,123	17,875	6,772
	20	10,800	194,049	87	—	—	—	—
	40	10,800	250,547	90	—	—	—	—

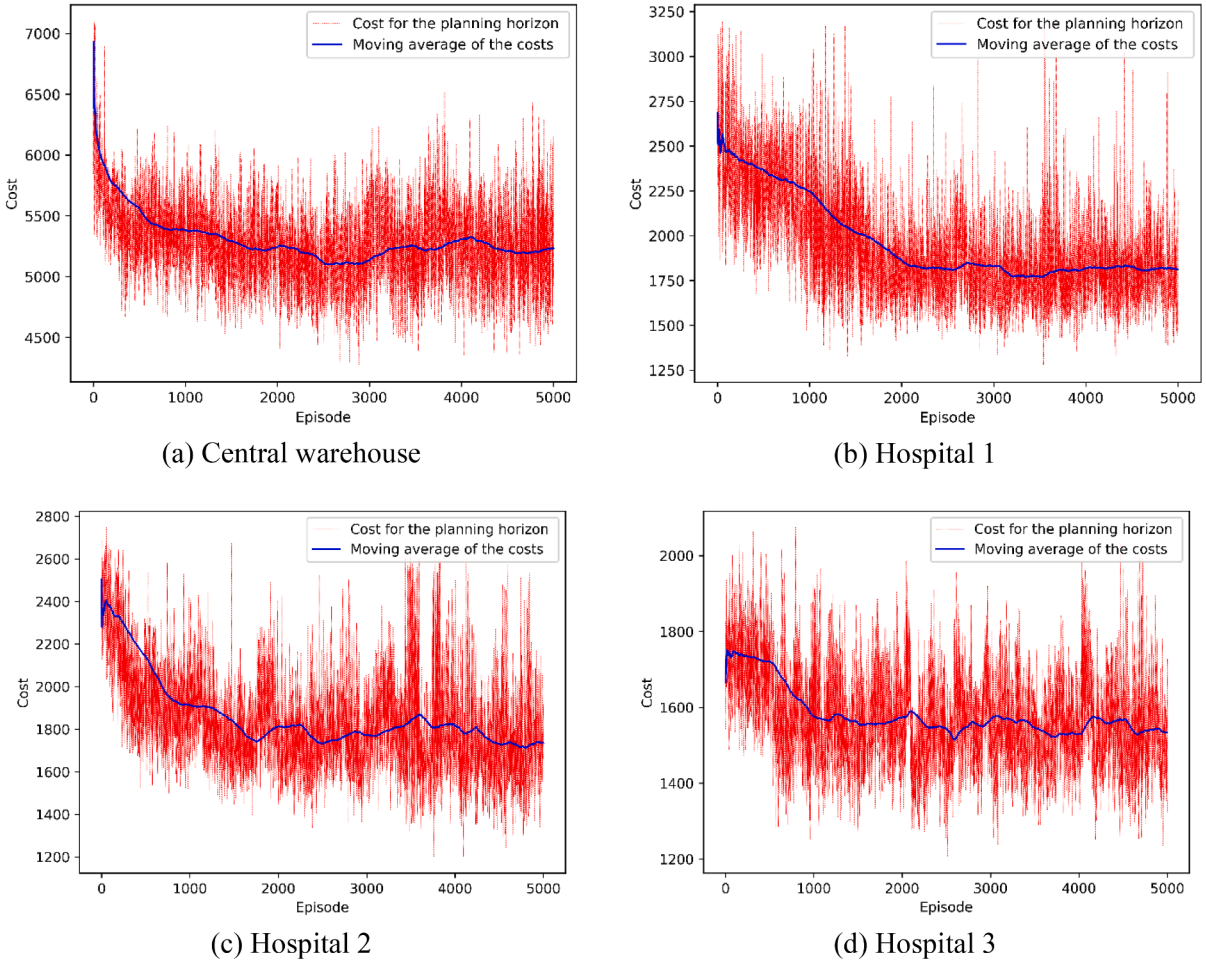

Fig. 2. Convergence of the agents using the QL algorithm for the central warehouse and the hospitals for the product with $E = 3$ weeks of maximum shelf life.

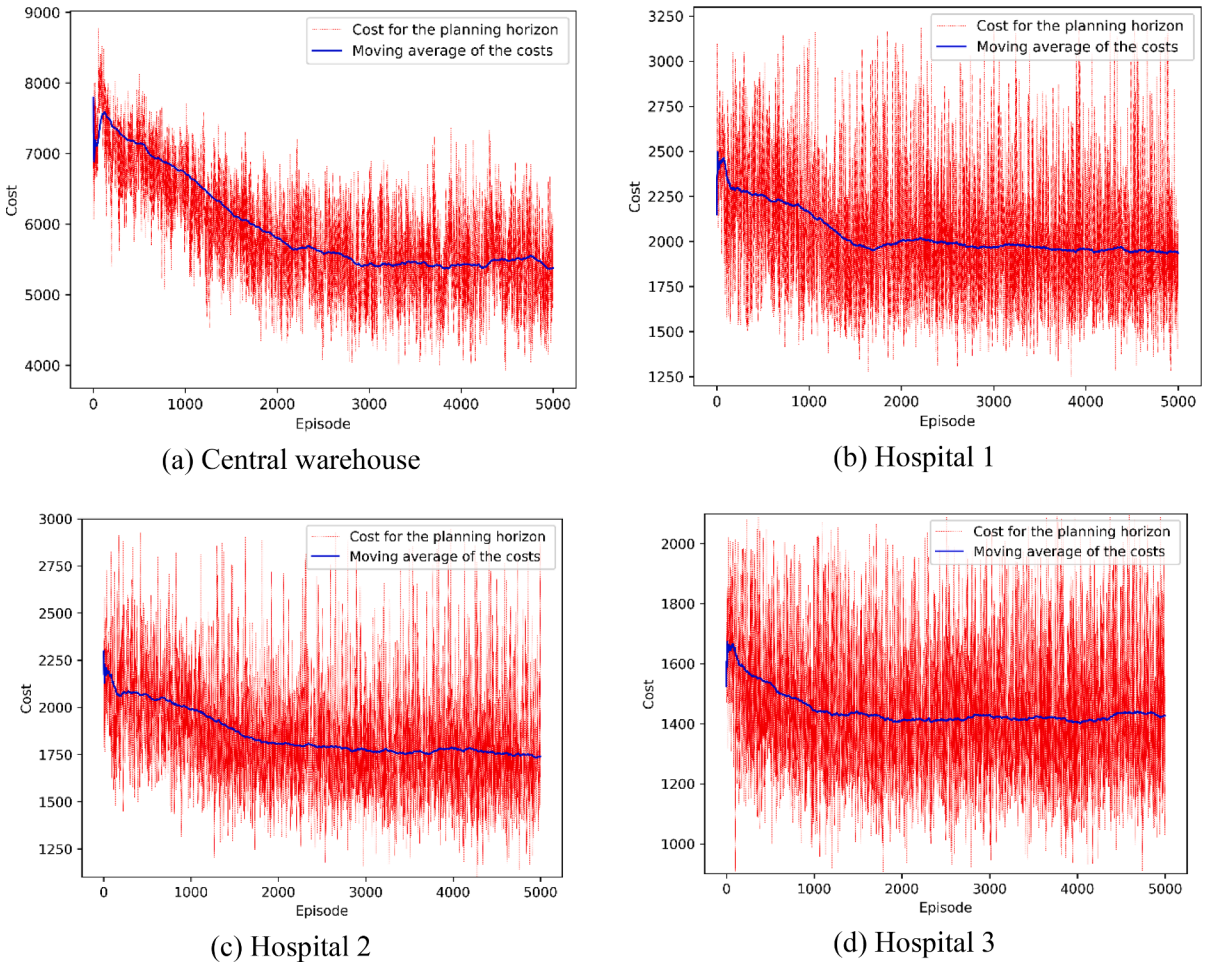

Fig. 3. Convergence of the agents using the DQN algorithm for the central warehouse and the hospitals for the product with $E = 3$ weeks of maximum shelf life.

for TI5 with 20 and 40 scenarios are not computed as the feasible solutions reported by CPLEX within the 10,800 s time limit have optimality gaps of over 80 %.

5.3. Solution analyses

In this section, the TI1 instance is analyzed in more detail for different values of the maximum shelf life. Figs. 2 and 3 show the performance of the agents of the central warehouse and the hospitals in learning the optimal ordering policies throughout the training process using the DL and DQN algorithms, respectively. The convergence condition for the algorithms is defined as a point when no significant improvement can be made in the last 500 training episodes. Thus, the plots in Figs. 2 and 3 show the total cost for the planning horizon for each training episode, as well as the moving average of the costs for the last 500 consecutive episodes.

The agents of the central warehouse and the hospitals were initially allowed to run for 5,000 training episodes. However, with the QL algorithm, the agents for both the central warehouse and the hospitals converged and no significant improvements in the moving average of the costs can be made after approximately 2500 episodes, as shown in Fig. 2. However, the convergence speed is different depending on the size and structure of the problem. With the DQN algorithm, agent of the central warehouse converged after approximately 2500 episodes, but the agents of the hospitals converged after approximately 1500 episodes. These results indicate that the DQN algorithm converged faster than the QL algorithm. The cost components for the IIM and (R, s, S) policies are presented in Table 6. Due to their randomness, the results obtained by *IIM_QL*, *IIM_DQN* and (R, s, S)_ *GA* are the average values over ten independent replications.

The results in Table 6 show that *IIM_DQN* generated solutions with consistently lower costs, on average $258 lower, than *IIM_QL* for products with different shelf lives. At the steady states, *IIM_DQN* yielded an average cost of $6,955, which is approximately 7.52 %, 3.58 % and 13.63 % lower than that of (R, s, S)_*CPLEX*, *IIM_QL* and (R, s, S)_*GA*, respectively. Fig. 4 graphically depicts the results for the products with different periods of shelf lives.

The frequency polygons of the shortage cost of each policy over 1,000 observations, shown in Fig. 5, reveal that (R, s, S)_*CPLEX* and (R, s, S)_*GA* have high risks of product shortage in the hospitals. However, *IIM_QL* performs better in securing the availability of the products for improving the service level for patients. For products with a maximum shelf life of $E = 3$ periods, the *IIM_DQN* and *IIM_QL* results are comparable. For products with $E > 3$, *IIM_QL* offers a better service level than the other policies. The average shortage costs over the studied products for *IIM_QL* is $224, followed by *IIM_DQN*, (R, s, S)_*GA* and (R, s, S)

Table 6
Comparison of costs of the IIM and (R, s, S) policies for the TI1 instance.

Cost components	$E = 3$				$E = 4$				$E = 5$				$E = 6$			
	(R,s,S)		IIM		(R,s,S)		IIM		(R,s,S)		IIM		(R,s,S)		IIM	
	CPLEX	GA	QL	DQN	CPLEX	GA	QL	DQN	CPLEX	GA	QL	DQN	CPLEX	GA	QL	DQN
Z_1: Purchase cost	$2,398	$2,661	$3,030	$3,041	$2,174	$2,459	$2,843	$2,206	$1,903	$2,778	$2,506	$1,935	$1,594	$2,373	$2,156	$2,184
Z_2: Order cost	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000	$1,000
Z_3: Warehouse expiration cost	$0	$0	$987	$849	$0	$0	$858	$812	$0	$0	$1,184	$485	$0	$0	$1,441	$481
Z_4: Warehouse holding cost	$35	$58	$110	$87	$81	$53	$146	$121	$85	$70	$190	$154	$113	$97	$232	$181
Z_5: Delivery cost	$1,069	$1,338	$1,400	$1,149	$1,041	$1,527	$1,381	$1,115	$938	$1,515	$1,278	$943	$992	$1,454	$1,178	$1,322
Z_6: Expected hospitals expiration cost	$1,815	$945	$570	$435	$1,662	$1,418	$427	$183	$1,708	$1,115	$437	$118	$1,589	$951	$407	$182
Z_7: Expected hospitals holding cost	$120	$242	$212	$227	$231	$375	$327	$176	$339	$694	$341	$176	$289	$718	$313	$293
Z_8: Expected shortage cost	$1,683	$2,267	$725	$710	$1,428	$1,359	$125	$1,346	$1,413	$813	$37	$2,026	$1,381	$930	$11	$884
Total expected cost	$8,119	$8,511	$8,034	**$7,497**	$7,618	$8,191	$7,107	**$6,959**	$7,387	$7,985	$6,973	**$6,837**	$6,958	$7,523	$6,738	**$6,527**

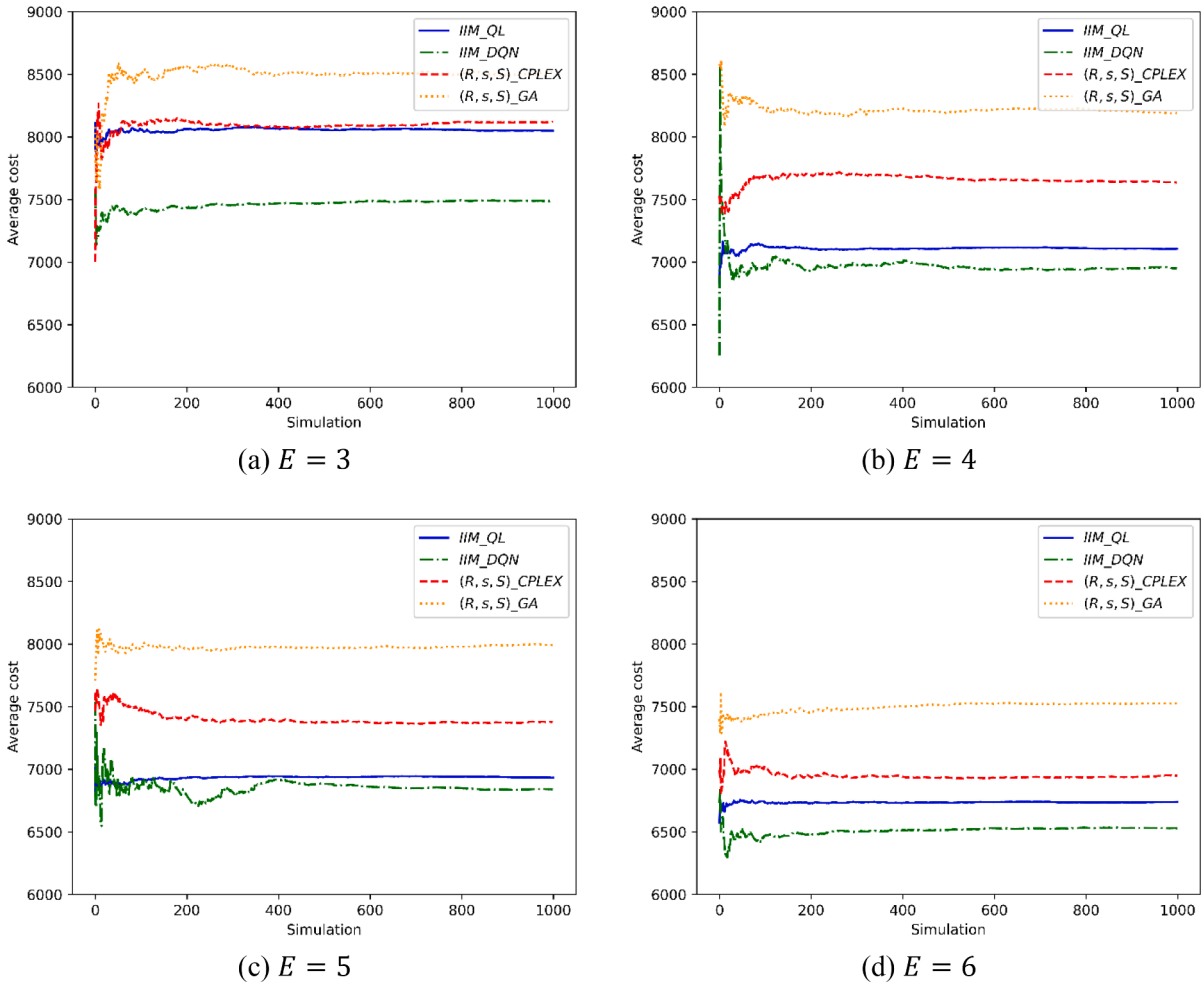

Fig. 4. The average costs of the whole inventory system obtained with the IIM and (R, s, S) policies.

_CPLEX with \$1,241, \$1,378 and \$1,342, respectively.

The average total expiration cost of *IIM_DQN* for the central warehouse and the hospitals is estimated to be \$886, which is 44 % less than that of *IIM_QL*. The average expiration cost of (R, s, S)*_GA* is \$1,107, that is 35 % less than that of (R, s, S)*_CPLEX*. Using *IIM_DQN*, a 20 % cost reduction in the expired products is achieved as compared to the best (R, s, S) policy, indicating that *IIM_DQN* distributed the products more efficiently among hospitals. Fig. 6 shows the frequency polygons of the expiration costs for the whole inventory system of both the IIM and the (R, s, S) policies over 1,000 observations. In the central warehouse, on average, 5.05 units of the product are expired each period with *IIM_DQN*, while none of the product is expired with the (R, s, S) policies. However, on average, 6.8 more units of the products are expired with (R, s, S)*_GA* than with *IIM_DQN* each period in the hospitals.

Since *IIM_QL* has considerably lower shortages and lower product expiries than (R, s, S)*_CPLEX* does, this policy is expected to carry over more inventory in each period and, consequently, to incur a higher inventory holding cost. The average total inventory holding cost of *IIM_QL* is \$145 more than that of (R, s, S)*_CPLEX* for the whole inventory system over the products with different shelf lives. This higher cost is caused by carrying, on average, 11.8 more units of the product each period. The average inventory holding cost of *IIM_DQN* is estimated to be \$354, which is about the same as that of (R, s, S)*_CPLEX* and is \$114 less than that of *IIM_QL*.

Fig. 7 shows the average age distribution of the product ordered by the central warehouse for the TI1 instance when placing orders under the different inventory policies. For the product with $E = 3$ periods of maximum shelf life, as shown in Fig. 7(a), (R, s, S)*_CPLEX*, (R, s, S)*_GA* and *IIM_QL* tend to order the product with the shortest remaining life. The product with $r = 1$ period of remaining life constitutes the highest portion of the orders. For the product with $E > 3$ periods of maximum shelf life as shown in Fig. 7(b), (c) and (d), the (R, s, S) policies still recommend buying the cheaper products with shorter remaining lives. The IIM policies, however, propose to mainly order a product with $r = 2$ periods of remaining life. In general, the IIM policies advocate the ordering of fresher products than the (R, s, S) policies do. (R, s, S)*_CPLEX* has a considerably higher risk of product shortage because approximately 64 % of the purchased product becomes obsolete and thus no longer useable if not used within one period. For (R, s, S)*_GA*, *IIM_QL* and *IIM_DQN*, this risk is about 41 %, 24 % and 25 %, respectively.

Fig. 8 shows the maximum inventory positions for the IIM and the (R, s, S) policies across different periods. In a (R, s, S) policy, the optimal order-up-to-level S is the inventory position after receiving an order in each period. In an IIM policy, the sum of the average inventory positions, i.e., the state of the inventory system, of the 1000 episodes for each period, and the order quantity of that period are used to obtain the inventory position. As Fig. 8 illustrates, (R, s, S)*_CPLEX* has the highest inventory position followed by *IIM_QL*, (R, s, S)*_GA* and *IIM_DQN*.

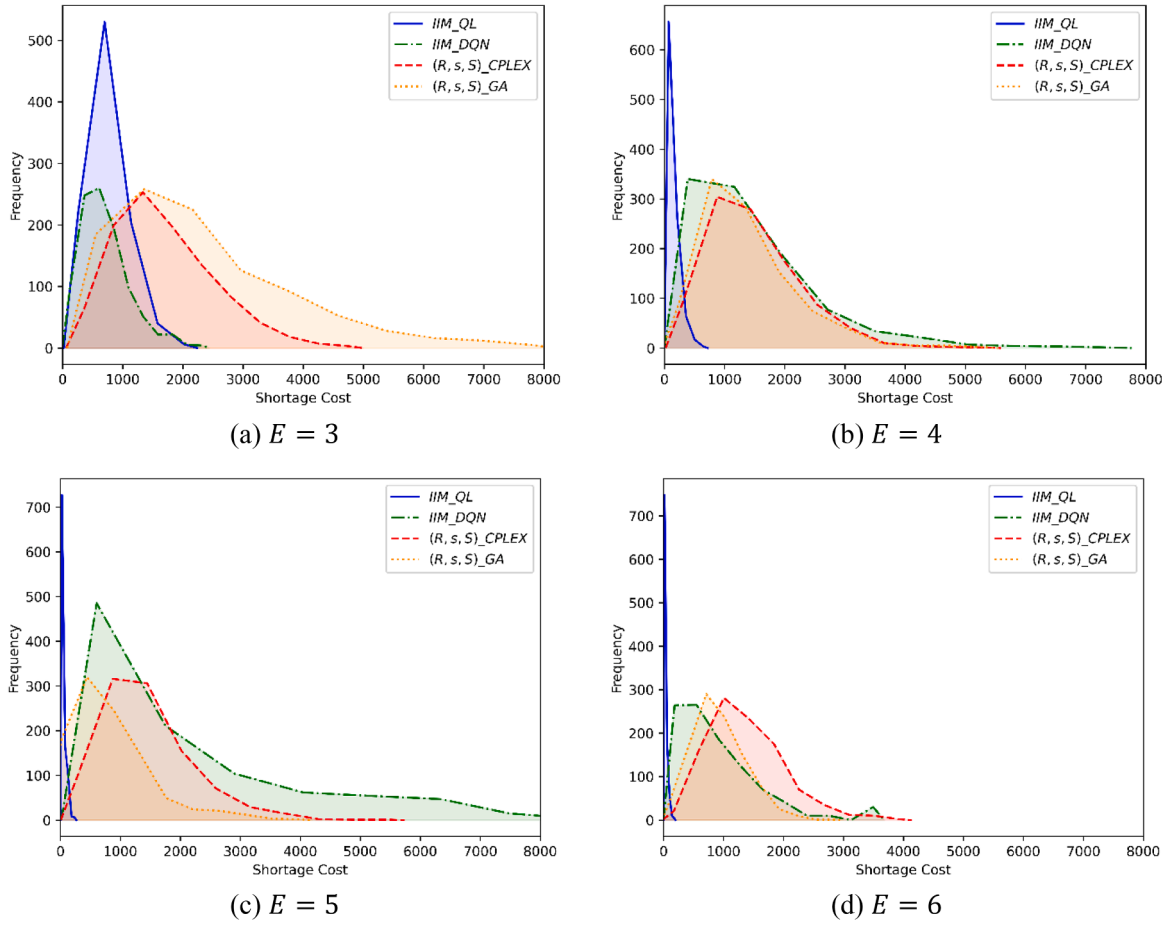

Fig. 5. The frequency polygons of the shortage costs of the whole inventory system for the IIM and the (R, s, S) policies.

Compared with other policies, *IIM_DQN* has a lower maximum inventory position, meaning that it can save storage space.

The traditional and widely used (R, s, S) policy determines s and S using a limited number of predetermined scenarios without taking into account the dynamic nature of the inventory system and their influence on future decisions. The underlining assumption of the (R, s, S) policy is that, after hitting the reorder point s, the inventory level must increase to the order-up-to-level S, regardless of the future events that may occur. The (R, s, S) policy can take advantage of the product with shorter remaining lives, as depicted in Fig. 7, that is cheaper and can offset other costs of the system by assuming future demands can be precisely predicted through scenarios at the time of a decision. In contrast, the optimal inventory of an IIM approach is determined in a dynamic environment, where an agent learns to make optimal decisions under repeated observations, considering its impact on future decisions. In making a decision, an agent interacts with the environment and is rewarded for its actions. The QL algorithm maximizes the expected value of the sum of future rewards, and the DQN algorithm maximizes the prediction accuracy of future rewards. Consequently, an IIM policy is more responsive to the uncertainties in the inventory system than a (R, s, S) policy. This responsiveness results in a variation in the maximum inventory for the IIM policies, as shown in Fig. 8.

5.4. Sensitivity analyses

Determining the correct values for the cost parameters is often a difficult and challenging task due to the lack of detailed information. Therefore, inventory managers often set the values of the cost parameters at reasonable levels. Sensitivity analyses can then be carried out to evaluate the effects of each cost parameter on the performance of the resulting inventory policies. In this study, sensitivity analyses are conducted for two key cost parameters, i.e., the expiration cost *ExpCost*, and the shortage cost *ShCost*, for the product with $E = 3$ periods of maximum shelf life for the TI1 instance. Note that *ShCost* instead of $ShCost_j$ is used because $ShCost_j$ is the same for all $J = 3$ hospitals in the example. To this end, the values of these two cost parameters are changed by –60 %, –40 %, –20 %, +20 %, +40 %, and +60 % of their initial values, one cost parameter at a time, while keeping the values of other parameters unchanged. The results of the sensitivity analyses are graphically depicted in Fig. 9.

Fig. 9 shows that the total cost with the IIM policies, unlike that with the (R, s, S) policies, is highly robust against the changes in the values of *ExpCost* and *ShCost*. The total cost remains relatively stable with the IIM policies, and is highly volatile with the (R, s, S) policies, as the values of *ExpCost* and *ShCost* change. A 20 % increase in the value of *ShCost*

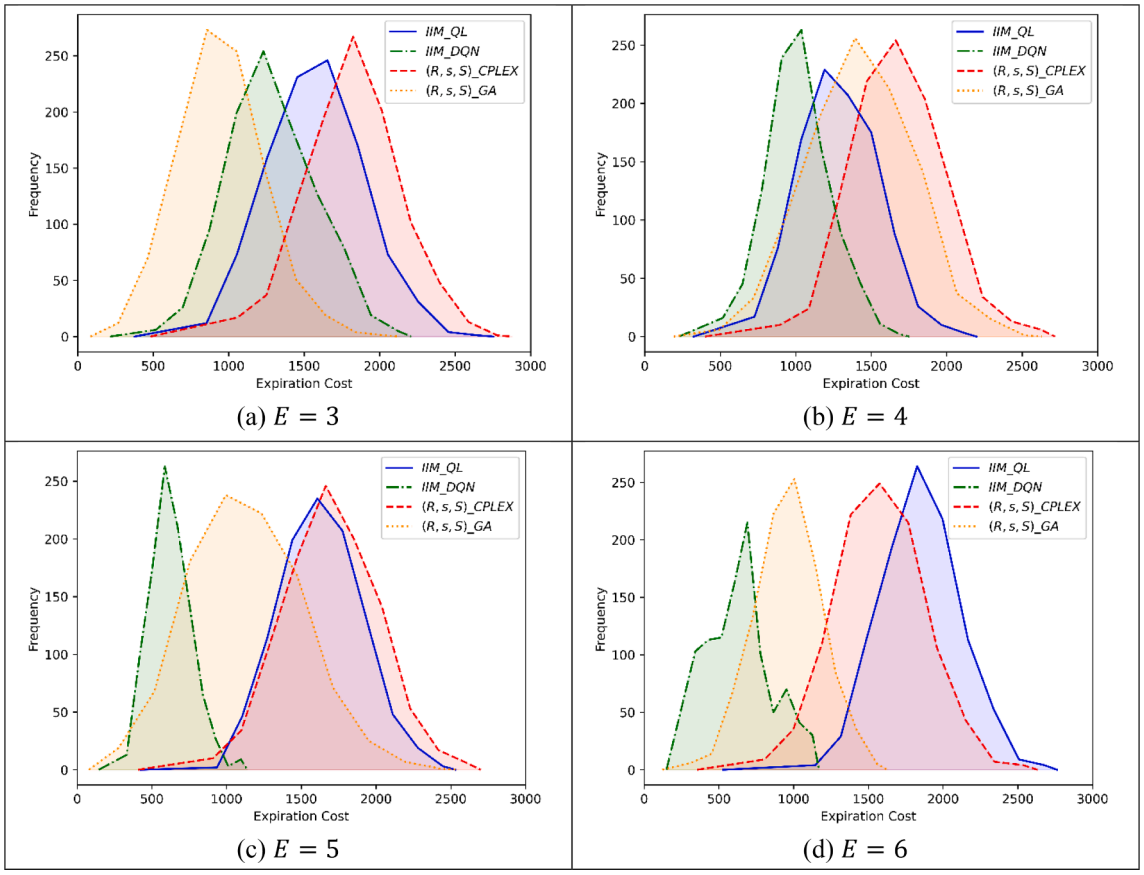

Fig. 6. The frequency polygons of the expiration costs of the whole inventory system of the IIM and the (R, s, S) policies.

results in 1.79 % and 2.30 % increases with *IIM_QL* and *IIM_DQN*, respectively, as compared to 5.19 % and 2.06 % increases with (R, s, S) *_CPLEX* and (R, s, S)*_GA*, respectively, in the total cost. When *ShCost* decreases, the same trend is observed. If *ShCost* is reduced by 20 %, the total cost is reduced by 0.74 % and 0.72 % with *IIM_QL* and *IIM_DQN*, respectively, but by 5.87 % and 6.54 % with (R, s, S)*_CPLEX* and (R, s, S)*_GA*, respectively. Therefore, more careful and accurate estimates of the cost parameters are needed when using the SMIP model than using the IIM approaches for setting up the inventory policies.

6. Managerial insights

By applying the proposed IIM approaches, the healthcare system can benefit from significant reductions in the inventory-related cost for products with different expiration dates and different prices. Not only fresher products with longer remaining shelf lives, which of course are healthier to the patients, are purchased, but also smaller storage area is required to hold the inventory. In addition, the IIM approaches provide a higher service level for patients by reducing the risk of inventory shortage. These advantages of the IIM approaches are quite beneficial for the general public and for the reputation of the healthcare system. In particular, as an example of application of the IIM approaches, blood banks have a continuous challenge with limited cold storage facilities that have to provide fresh blood packages to patients across the supply chain. This challenge can be fairly moderated by applying the proposed IIM approaches.

The IIM approaches enable inventory managers to determine the near-optimal inventory policies either by using the Q-table or by using the predictive model. More in detail, a large body of optimal state-action pairs is available in the hands of the inventory managers who can roughly evaluate, analyze, and decide in a dynamic inventory system. Even without continuously running the time-consuming simulations, especially when cost parameters do not change for a long time, the healthcare system can benefit from the Q-tables and/or the predictive model obtained from the QL and DQN algorithms and simulated based on different scenarios. As a result, the proposed IIM approaches are applicable to the highly dynamic environment of the healthcare systems and can provide useful decision support for managing inventories. The model can be utilized in real-time once it has been trained even though an IIM approach can be time-consuming in training a model for large-scale problems.

7. Conclusions

In this work, two IIM approaches are developed for pharmaceutical perishable products in a healthcare supply chain. The supply chain consists of multiple regional hospitals whose stocks are replenished from a central warehouse. The central warehouse itself replenishes stocks from suppliers. The age distribution of the product in the inventory is assumed to be available at the time of placing replenishment orders. As

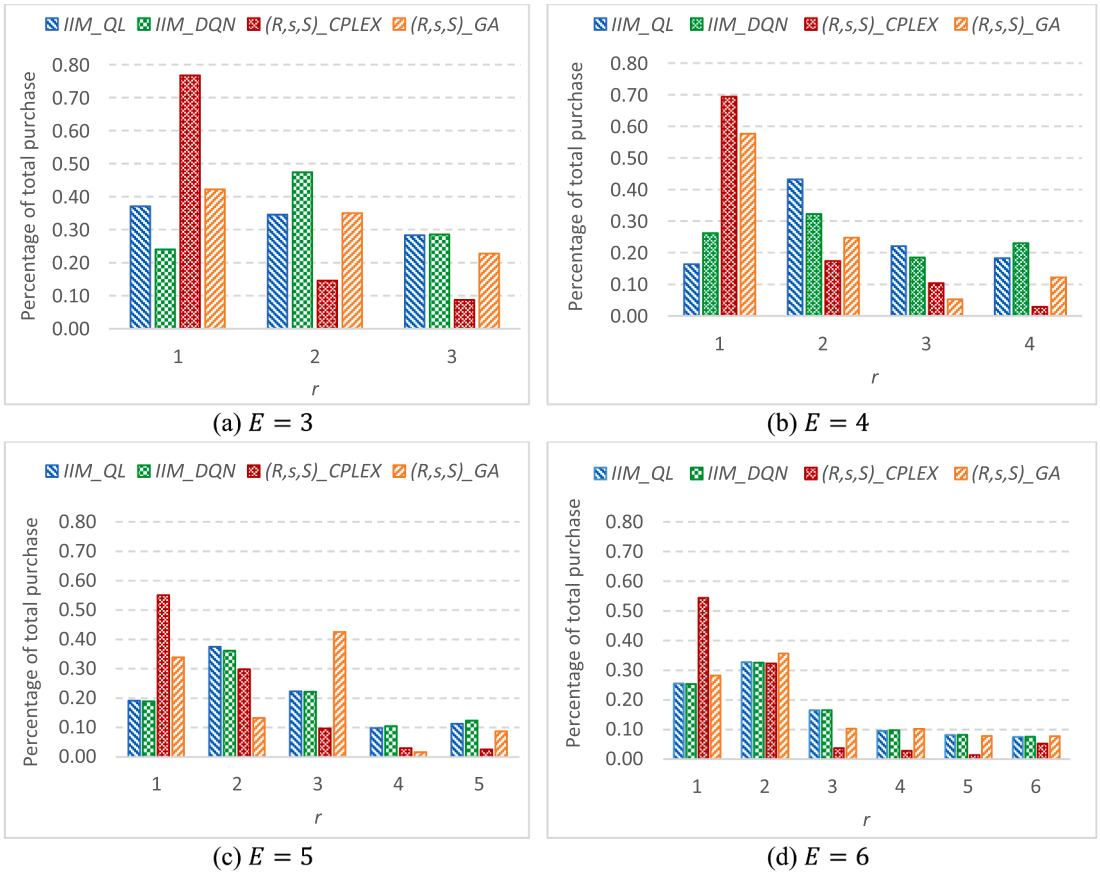

Fig. 7. Average age distributions of the purchased product in the central warehouse for the IIM and the (R, s, S) policies for the TI1 instance.

discounts are given to the products that are not fresh, the purchase prices of the product are assumed to be age dependent. The IIM approaches use two RL techniques, specifically the QL and DQN algorithms, to recommend the near-optimal inventory policy for each agent representing the members of the supply chain. The DQN algorithm was developed to overcome the problem of the curse of dimensionality in the QL algorithm. Periodic review (R, s, S) inventory policies obtained by a SMIP model, solved with CPLEX and GA, are used as benchmarks to compare the performance of the IIM approaches. Test instances with different sizes and an example from the literature composed of three regional hospitals and a central warehouse are used to show the applicability and effectiveness of the proposed approaches.

The computational results show that the IIM policies are superior to the (R, s, S) policies in leading to a lower total cost. In addition, the solutions offered by the IIM approaches not only have a lower risk of inventory shortage, and thereby a higher service level for patients, but also have a lower risk of product expiry. *IIM_QL* and *IIM_DQN* led to 4.09 % and 7.52 % cost reduction, respectively, as compared to (R, s, S)*_CPLEX* for the example. The risk of product shortage of *IIM_QL* is considerably lower than those of (R, s, S)*_CPLEX* and (R, s, S)*_GA*, while at the same time the expiration cost of *IIM_QL* is $116 lower than that of (R, s, S) *_CPLEX* for the example. *IIM_DQN* has the lowest expiration cost among all the policies, being $221 less than the best obtained by (R, s, S)*_GA* for the example. In addition, the IIM approaches have the potential to generate policies that offer additional opportunities for cost savings through reducing the required inventory storage space. However, (R, s, S)*_GA* shows a speed advantage of approximately-six times when compared to the IIM approaches.

As the performances of the resulting policies can be affected by the values of some key cost parameters, sensitivity analyses are conducted to derive managerial insights on the effects of these important cost parameters because their accurate values are hard to estimate. Specifically, sensitivity analyses are performed for the effects of the expiration and shortage cost parameters on the total cost for each inventory policy. The results show that the IIM policies are relatively stable, and the (R, s, S) policies are highly volatile, when the values of these cost parameters change. Thus, the inventory managers should pay considerable attention to the accurate estimation of the values of the shortage and expiration costs if the SMIP model is used to find the (R, s, S) policies.

Healthcare systems can adopt the IIM approaches to derive the most cost-effective policies for their inventory control decisions. Future studies may consider extending the IIM approaches to multi-product inventory systems with budget constraints. In addition, designing other learning-based inventory management models using other RL techniques such as policy gradient methods, actor-critic methods, and other variants of QL such as double QL and delayed QL, and evaluating their performance against the proposed IIM approaches are also a future research avenues. Another possibility for future work would be to extend the model to allow interactions between hospitals and train all agents simultaneously. Thus, the product expiry risk can be reduced because products with near-expiration dates can be transferred to other hospitals with higher demands.

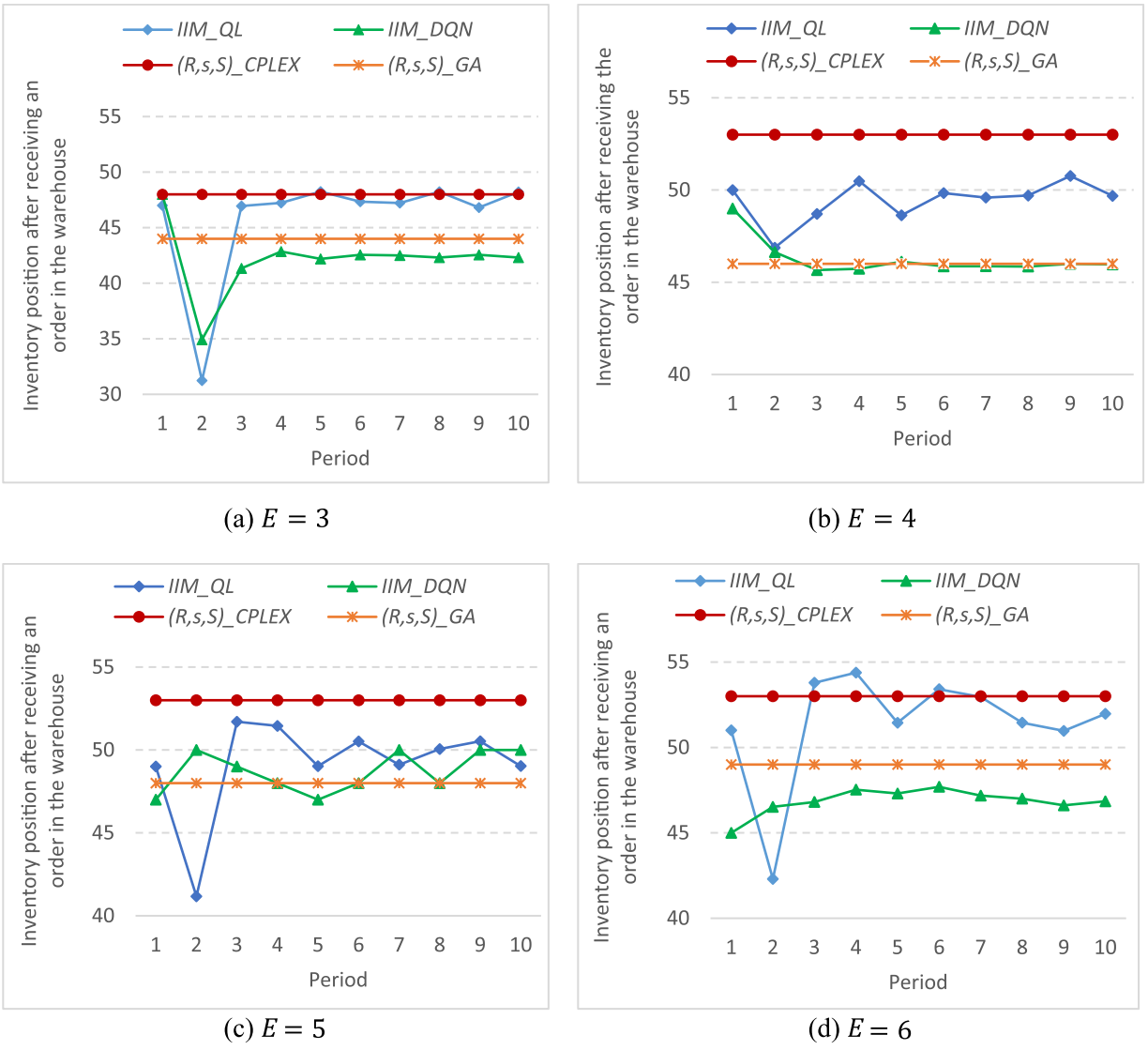

Fig. 8. Inventory positions after receiving an order at the central warehouse in the IIM and the (R, s, S) policies.

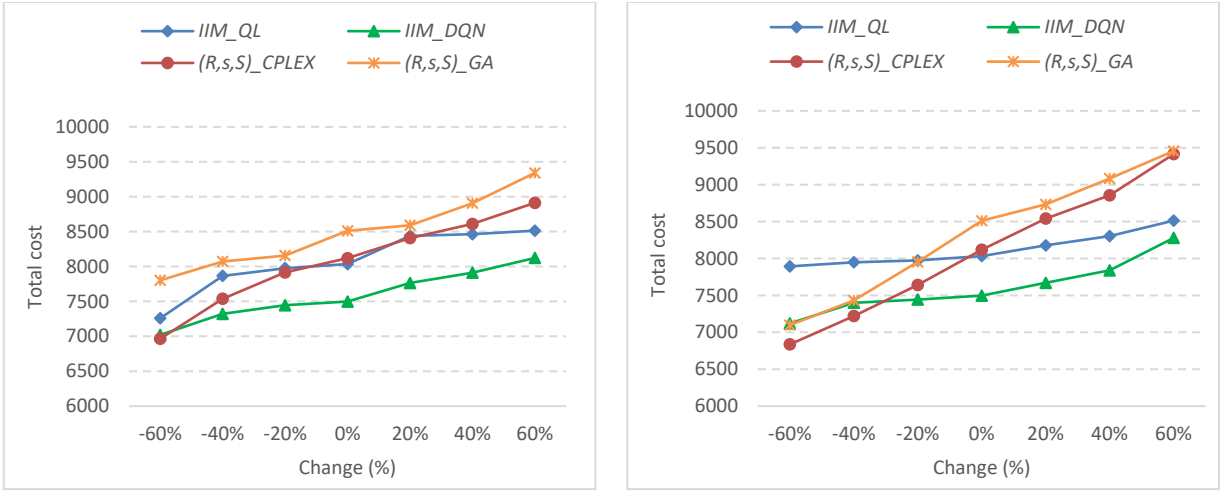

Fig. 9. Impacts of the expiration and shortage costs on the performance of the policies for the product with $E = 3$ periods of maximum shelf life.

CRediT authorship contribution statement

Ehsan Ahmadi: Conceptualization, Methodology, Software, Writing – original draft, Supervision, Project administration. **Hadi Mosadegh:** Conceptualization, Methodology, Writing – original draft. **Reza Maihami:** Formal analysis, Investigation, Writing – original draft. **Iman Ghalehkhondabi:** Visualization, Investigation, Writing – review & editing. **Minghe Sun:** Writing – review & editing, Validation, Supervision. **Gürsel A. Süer:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abdel-Malek, L.L., Ziegler, H., 1988. Age dependent perishability in two-echelon serial inventory systems. Comput. Oper. Res. 15, 227–238. https://doi.org/10.1016/0305-0548(88)90035-4.

Ahmadi, E., Goldengorin, B., Süer, G.A., Mosadegh, H., 2018a. A hybrid method of 2-TSP and novel learning-based GA for job sequencing and tool switching problem. Appl. Soft Comput. 65, 1–16. https://doi.org/10.1016/j.asoc.2017.12.045.

Ahmadi, E., Masel, D., Metcalf, A., Schuller, K., 2018b. Inventory management of surgical supplies and instruments in hospitals: A literature review. Heal. Syst. 1–18 https://doi.org/10.1080/20476965.2018.1496875.

Ahmadi, E., Maihami, R., Sun, M., Masel, D., Cagle, C., 2022. Periodic review multi-period inventory control models for perishable pharmaceutical products in hospitals. Proceedings of the IISE Annual Conference & Expo 2022 in press.

Ahmadi, E., Masel, D., Hostetler, S., 2019. A robust stochastic decision-making model for inventory allocation of surgical supplies to reduce logistics costs in hospitals: A case study. Oper. Res. Heal. Care 20, 33–44. https://doi.org/10.1016/J.ORHC.2018.09.001.

Ahmadi, E., Masel, D.T., Hostetler, S., Maihami, R., Ghalehkhondabi, I., 2020. A centralized stochastic inventory control model for perishable products considering age-dependent purchase price and lead time. TOP 28, 231–269. https://doi.org/10.1007/s11750-019-00533-1.

Akbarpour, M., Ali Torabi, S., Ghavamifar, A., 2020. Designing an integrated pharmaceutical relief chain network under demand uncertainty. Transp. Res. Part E Logist. Transp. Rev. 136, 101867 https://doi.org/10.1016/j.tre.2020.101867.

Albalate, A., Minker, W., 2013. Semi-Supervised and Unervised Machine Learning: Novel Strategies. John Wiley & Sons.

Ash, C., Diallo, C., Venkatadri, U., VanBerkel, P., 2022. Distributionally robust optimization of a Canadian healthcare supply chain to enhance resilience during the COVID-19 pandemic. Comput. Ind. Eng. 168, 108051 https://doi.org/10.1016/j.cie.2022.108051.

Azaron, A., Brown, K.N., Tarim, S.A., Modarres, M., 2008. A multi-objective stochastic programming approach for supply chain design considering risk. Int. J. Prod. Econ. 116, 129–138. https://doi.org/10.1016/j.ijpe.2008.08.002.

Bakker, M., Riezebos, J., Teunter, R.H., 2012. Review of inventory systems with deterioration since 2001. Eur. J. Oper. Res. 221, 275–284. https://doi.org/10.1016/j.ejor.2012.03.004.

Bharti, S., Kurian, D.S., Pillai, V.M., 2020. Reinforcement learning for inventory management, in: Innovative Product Design and Intelligent Manufacturing Systems. Springer, pp. 877–885. 10.1007/978-981-15-2696-1_85.

Birge, J.R., Louveaux, F., 1997. Introduction to Stochastic Programming. Springer-Verlag, New York.

Boerma, T., AbouZahr, C.L., Ho, J., 2009. World health statistics. World health organization.

Brodheim, E., Prastacos, G.P., 1979. The Long Island blood distribution system as a prototype for regional blood management. Interfaces (Providence) 9, 3–20. https://doi.org/10.1287/inte.9.5.3.

Broekmeulen, R.A.C.M., van Donselaar, K.H., 2009. A heuristic to manage perishable inventory with batch ordering, positive lead-times, and time-varying demand. Comput. Oper. Res. 36, 3013–3018. https://doi.org/10.1016/j.cor.2009.01.017.

Chaharsooghi, S.K., Heydari, J., Zegordi, S.H., 2008. A reinforcement learning model for supply chain ordering management: an application to the beer game. Decis. Support Syst. 45, 949–959. https://doi.org/10.1016/j.dss.2008.03.007.

Chaudhary, V., Kulshrestha, R., Routroy, S., 2018. State-of-the-art literature review on inventory models for perishable products. J. Adv. Manag. Res. 15, 306–346. https://doi.org/10.1108/JAMR-09-2017-0091.

Chen, L.T., Wei, C.C., 2012. Multi-period channel coordination in vendor-managed inventory for deteriorating goods. Int. J. Prod. Res. 50, 4396–4413. https://doi.org/10.1080/00207543.2011.592159.

Dai, Z., Zheng, X., 2015. Design of close-loop supply chain network under uncertainty using hybrid genetic algorithm: a fuzzy and chance-constrained programming model. Comput. Ind. Eng. 88, 444–457. https://doi.org/10.1016/j.cie.2015.08.004.

Fattahi, M., Govindan, K., 2018. A multi-stage stochastic program for the sustainable design of biofuel supply chain networks under biomass supply uncertainty and disruption risk: A real-life case study. Transp. Res. Part E Logist. Transp. Rev. 118, 534–567. https://doi.org/10.1016/j.tre.2018.08.008.

Firdausiyah, N., Taniguchi, E., Qureshi, A.G., 2019. Modeling city logistics using adaptive dynamic programming based multi-agent simulation. Transp. Res. Part E Logist. Transp. Rev. 125, 74–96. https://doi.org/10.1016/j.tre.2019.02.011.

Ghanadian, S.A., Ghanbartehrani, S., 2021. Evaluating Supply Chain Network Designs: An Approach Based on SNA Metrics and Random Forest Feature Selection. J. Oper. Manag. Res. 15–35.

Giannoccaro, I., Pontrandolfo, P., 2002. Inventory management in supply chains: a reinforcement learning approach. Int. J. Prod. Econ. 78, 153–161. https://doi.org/10.1016/S0925-5273(00)00156-0.

Goyal, S.K., Giri, B.C., 2001. Recent trends in modeling of deteriorating inventory. Eur. J. Oper. Res. 134, 1–16. https://doi.org/10.1016/S0377-2217(00)00248-4.

Guan, Z., Mou, Y., Sun, M., 2022. Hybrid robust and stochastic optimization for a capital-constrained fresh product supply chain integrating risk-aversion behavior and financial strategies. Comput. Ind. Eng. 108224 https://doi.org/10.1016/j.cie.2022.108224.

Guerrero, W.J., Yeung, T.G., Guéret, C., 2013. Joint-optimization of inventory policies on a multi-product multi-echelon pharmaceutical system with batching and ordering constraints. Eur. J. Oper. Res. 231, 98–108. https://doi.org/10.1016/j.ejor.2013.05.030.

Janssen, L., Claus, T., Sauer, J., 2016. Literature review of deteriorating inventory models by key topics from 2012 to 2015. Int. J. Prod. Econ. 182, 86–112. https://doi.org/10.1016/j.ijpe.2016.08.019.

Johansson, L., Olsson, F., 2018. Age-based inventory control in a multi-echelon system with emergency replenishments. Eur. J. Oper. Res. 265, 951–961. https://doi.org/10.1016/j.ejor.2017.08.057.

Kara, A., Dogan, I., 2018. Reinforcement learning approaches for specifying ordering policies of perishable inventory systems. Expert Syst. Appl. 91, 150–158. https://doi.org/10.1016/j.eswa.2017.08.046.

Karaesmen, I.Z., Scheller–Wolf, A., Deniz, B., 2011. Managing perishable and aging inventories: review and future research directions, in: Kempf, K.G., Keskinocak, P., Uzsoy, R. (Eds.), . Springer US, New York, NY, pp. 393–436. 10.1007/978-1-4419-6485-4_15.

Katsaliaki, K., Brailsford, S.C., 2007. Using simulation to improve the blood supply chain. J. Oper. Res. Soc. 58, 219–227. https://doi.org/10.1057/palgrave.jors.2602195.

Kaya, O., Ghahroodi, S.R., 2018. Inventory control and pricing for perishable products under age and price dependent stochastic demand. Math. Methods Oper. Res. 1–35 https://doi.org/10.1007/s00186-017-0626-9.

Kelle, P., Woosley, J., Schneider, H., 2012. Pharmaceutical supply chain specifics and inventory solutions for a hospital case. Oper. Res. Heal. Care 1, 54–63. https://doi.org/10.1016/j.orhc.2012.07.001.

Kim, G., Wu, K., Huang, E., 2015. Optimal inventory control in a multi-period newsvendor problem with non-stationary demand. Adv. Eng. Informatics 29, 139–145. https://doi.org/10.1016/j.aei.2014.12.002.

Kimbrough, S.O., Wu, D.J., Zhong, F., 2002. Computers play the beer game: Can artificial agents manage supply chains? Decis. Support Syst. 33, 323–333. https://doi.org/10.1016/S0167-9236(02)00019-2.

King, A.J., Wallace, S.W., 2012. Modeling with Stochastic Programming. Springer Science & Business Media.

Kouki, C., Jouini, O., 2015. On the effect of lifetime variability on the performance of inventory systems. Int. J. Prod. Econ. 167, 23–34. https://doi.org/10.1016/j.ijpe.2015.05.007.

Kouki, C., Jemaï, Z., Minner, S., 2015. A lost sales (r, Q) inventory control model for perishables with fixed lifetime and lead time. Int. J. Prod. Econ. 168, 143–157. https://doi.org/10.1016/j.ijpe.2015.06.010.

Kulkarni, P., 2012. Reinforcement and systemic machine learning for decision making. John Wiley & Sons. https://doi.org/10.1002/9781118266502.

Lagodimos, A.G., Koukoumialos, S., 2008. Service performance of two-echelon supply chains under linear rationing. Int. J. Prod. Econ. 112, 869–884. https://doi.org/10.1016/j.ijpe.2007.07.007.

Landis, N., 2002. Provisional observations on drug product shortages: effects, causes, and potential solutions. Am. J. Heal. Pharm. 59, 2173–2182. https://doi.org/10.1093/ajhp/59.22.2173.

Lee, Y.-M., Mu, S., Shen, Z., Dessouky, M., 2014. Issuing for perishable inventory management with a minimum inventory volume constraint. Comput. Ind. Eng. 76, 280–291. https://doi.org/10.1016/j.cie.2014.08.007.

Li, Y., 2017. Deep reinforcement learning: An overview. arXiv Prepr. arXiv1701.07274.

Maihami, R., Ghelehkhondabi, I., Ahmadi, E., 2021. Pricing and inventory planning for non-instantaneous deteriorating products with greening investment: a case study in beef industry. J. Clean. Prod. 295, 126368 https://doi.org/10.1016/j.jclepro.2021.126368.

Maihami, R., Karimi, B., 2014. Optimizing the pricing and replenishment policy for non-instantaneous deteriorating items with stochastic demand and promotional efforts. Comput. Oper. Res. 51, 302–312. https://doi.org/10.1016/j.cor.2014.05.022.

McKone-Sweet, K.E., Hamilton, P., Willis, S.B., 2005. The ailing healthcare supply chain: a prescription for change. J. Supply Chain Manag. 41, 4–17. https://doi.org/10.1111/j.1745-493X.2005.tb00180.x.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518, 529–533. https://doi.org/10.1038/nature14236.

Mosadegh, H., Fatemi Ghomi, S.M.T., Süer, G.A., 2020. Stochastic mixed-model assembly line sequencing problem: Mathematical modeling and Q-learning based simulated

annealing hyper-heuristics. Eur. J. Oper. Res. 282, 530–544. https://doi.org/10.1016/j.ejor.2019.09.021.
Nasrollahi, M., Razmi, J., 2021. A mathematical model for designing an integrated pharmaceutical supply chain with maximum expected coverage under uncertainty. Oper. Res. 21, 525–552. https://doi.org/10.1007/s12351-019-00459-3.
Niakan, F., Rahimi, M., 2015. A multi-objective healthcare inventory routing problem; a fuzzy possibilistic approach. Transp. Res. Part E Logist. Transp. Rev. 80, 74–94. https://doi.org/10.1016/j.tre.2015.04.010.
Oroojlooyjadid, A., Nazari, M., Snyder, L., Takáč, M., 2021. A deep Q-network for the beer game: a deep reinforcement learning algorithm to solve inventory optimization problems. Manuf. Serv. Oper. Manag. 10.1287/msom.2020.0939.
Pahl, J., Voß, S., 2014. Integrating deterioration and lifetime constraints in production and supply chain planning: a survey. Eur. J. Oper. Res. 238, 654–674. https://doi.org/10.1016/j.ejor.2014.01.060.
Pauls-Worm, K.G.J., Hendrix, E.M.T., Haijema, R., Van Der Vorst, J.G.A.J., 2014. An MILP approximation for ordering perishable products with non-stationary demand and service level constraints. Int. J. Prod. Econ. 157, 133–146. https://doi.org/10.1016/j.ijpe.2014.07.020.
Perez, F., Torres, F., 2020. Inventory Models for Managing Deteriorating Products: A Literature Review. Authorea Prepr.
Prastacos, G.P., 1984. Blood inventory management: an overview of theory and practice. Manage. Sci. 30, 777–800. https://doi.org/10.1287/mnsc.30.7.777.
Qiu, R., Sun, M., Lim, Y.F., 2017. Optimizing (s, S) policies for multi-period inventory models with demand distribution uncertainty: Robust dynamic programming approaches. Eur. J. Oper. Res. 261, 880–892. https://doi.org/10.1016/j.ejor.2017.02.027.
Qiu, R., Sun, Y., Fan, Z.P., Sun, M., 2020. Robust multi-product inventory optimization under support vector clustering-based data-driven demand uncertainty set. Soft Comput. 24, 6259–6275. https://doi.org/10.1007/s00500-019-03927-2.
Qiu, R., Hou, L., Sun, Y., Sun, M., Sun, Y., 2021a. Joint pricing, ordering and order fulfillment decisions for a dual-channel supply chain with demand uncertainties: A distribution-free approach. Comput. Ind. Eng. 160, 107546 https://doi.org/10.1016/j.cie.2021.107546.
Qiu, R., Sun, Y., Sun, M., 2021b. A distributionally robust optimization approach for multi-product inventory decisions with budget constraint and demand and yield uncertainties. Comput. Oper. Res. 126, 105081 https://doi.org/10.1016/j.cor.2020.105081.
Qiu, R., Yu, Y., Sun, M., 2021c. Joint pricing and stocking decisions for a newsvendor problem with loss aversion and reference point effect. Manag. Decis. Econ. 42, 275–288. https://doi.org/10.1002/mde.3233.
Raafat, F., 1991. Survey of literature on continuously deteriorating inventory models. J. Oper. Res. Soc. 42, 27–37. https://doi.org/10.1057/jors.1991.4.
Rana, R., Oliveira, F.S., 2015. Dynamic pricing policies for interdependent perishable products or services using reinforcement learning. Expert Syst. Appl. 42, 426–436. https://doi.org/10.1016/j.eswa.2014.07.007.
Saedi, S., Erhun Kundakcioglu, O., Henry, A.C., 2016. Mitigating the impact of drug shortages for a healthcare facility: An inventory management approach. Eur. J. Oper. Res. 251, 107–123. https://doi.org/10.1016/j.ejor.2015.11.017.
Sazvar, Z., Mirzapour Al-e-hashem, S.M.J., Govindan, K., Bahli, B., 2016. A novel mathematical model for a multi-period, multi-product optimal ordering problem considering expiry dates in a FEFO system. Transp. Res. Part E Logist. Transp. Rev. 93, 232–261. https://doi.org/10.1016/j.tre.2016.04.011.
Soleimani, H., Govindan, K., Saghafi, H., Jafari, H., 2017. Fuzzy multi-objective sustainable and green closed-loop supply chain network design. Comput. Ind. Eng. 109, 191–203. https://doi.org/10.1016/j.cie.2017.04.038.
Sun, Y., Qiu, R., Sun, M., 2022. Optimizing decisions for a dual-channel retailer with service level requirements and demand uncertainties: A Wasserstein metric-based distributionally robust optimization approach. Comput. Oper. Res. 138, 105589 https://doi.org/10.1016/j.cor.2021.105589.
Sun, R., Zhao, G., 2012. Analyses about efficiency of reinforcement learning to supply chain ordering management. IEEE Int. Conf. Ind. Informatics 124–127. https://doi.org/10.1109/INDIN.2012.6301163.
Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction, Second. ed. The MIT Press.
van Donselaar, K.H., Broekmeulen, R.A.C.M., 2012. Approximations for the relative outdating of perishable products by combining stochastic modeling, simulation and regression modeling. Int. J. Prod. Econ. 140, 660–669. https://doi.org/10.1016/j.ijpe.2012.02.023.
Vila-Parrish, A.R., Ivy, J.S., King, R.E., 2008. A simulation-based approach for inventory modeling of perishable pharmaceuticals, in: 2008 Winter Simulation Conference. IEEE, pp. 1532–1538. 10.1109/WSC.2008.4736234.
Wang, L., Cheng, C., Tseng, Y., Liu, Y., 2015. Demand-pull replenishment model for hospital inventory management: a dynamic buffer-adjustment approach. Int. J. Prod. Res. 53, 7533–7546. https://doi.org/10.1080/00207543.2015.1102353.
WHO, 2021. Global Expenditure on Health: Public Spending on the Rise? World Health Organization.
Wu, D., Rossetti, M.D., Tepper, J.E., 2013. Possibility of inventory pooling in China's public hospital and appraisal about its performance. Appl. Math. Model. 39, 7277–7290. https://doi.org/10.1016/j.apm.2015.02.042.
Wu, L., Shahidehpour, M., Li, T., 2007. Stochastic security-constrained unit commitment. IEEE Trans. Power Syst. 22, 800–811. https://doi.org/10.1109/TPWRS.2007.894843.
Zahiri, B., Jula, P., Tavakkoli-Moghaddam, R., 2018. Design of a pharmaceutical supply chain network under uncertainty considering perishability and substitutability of products. Inf. Sci. (Ny) 423, 257–283. https://doi.org/10.1016/j.ins.2017.09.046.