Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

Research Article

Inventory Policy by Dynamic Order

Ahmad A. Abubaker

Faculty of Computer Studies, Arab Open University, Saudi Arabia
E-mail: a.abubaker@arabou.edu.sa

Received: 30 April 2025; **Revised:** 26 June 2025; **Accepted:** 21 July 2025

Abstract: This paper proposes a novel inventory control policy called the dynamic order policy. The proposed policy dynamically adjusts the reorder quantity based on two threshold levels (s^*, s, S, S^*), enhancing its flexibility compared to the traditional policy (s, S). The new policy allows placing a larger order when the inventory drops below a threshold of s^* and places a standard order when the inventory falls between the thresholds of s^* and s. A genetic algorithm is employed to optimize the inventory decision parameters due to its ability to solve complex, nonlinear discrete problems. The model is tested under various simulation scenarios by demand rate, lead time, and customer volume. The results demonstrate that the proposed policy reduces the total average cost by up to 6.7% compared to the traditional policy. This dynamic framework presents a promising alternative for managing uncertain inventory environments with backlogs and variable lead times.

Keywords**:** inventory policy, inventory system, backlog, genetic algorithm

MSC: 90C59, 90C90, 68W40

1. Introduction

Inventory control refers to techniques that minimize costs associated with holding inventory while avoiding stockouts and related problems. Inventory control plays a vital role in managing trading items, stocks, or products in any enterprise. Poor inventory management can increase operational costs and lead to customer dissatisfaction due to shortages, potentially resulting in a loss of clientele.

Maintaining an optimal inventory level is a balancing act. An excessive supply increases holding costs, whereas an insufficient supply may lead to lost sales, reduced revenue, and increased shipping expenses. The timing and volume of reorders are crucial to inventory control. Delays in reordering can lead to shortages, whereas frequent small orders can increase costs.

Furthermore, inventory decisions are increasingly complicated by dynamic information delays, particularly in systems with pending orders. These delays, often caused by system integration problems or manual handling, result in inaccurate inventory visibility. Inaccuracy of inventory information can result in over- or understocking, missed demand fulfillment, elevated costs, and delayed orders. Addressing such delays is critical in modern inventory systems, especially when lead times and demand are uncertain.

This work develops a general policy (s, S), where an order is placed to raise the inventory to level S once it drops below a threshold s. This work proposes a novel adaptive inventory policy based on a dynamic order, specifically (s^*,

DOI: https://doi.org/10.37256/cm.6520257151

s, S, S^*). Compared with classical and adaptive inventory policies, the proposed policy (s^*, s, S, S^*) introduces a novel structure by incorporating two reorder thresholds, a primary threshold s and a critical threshold s^*, alongside two order-up-to levels, S and S^*. This dual-threshold, dual-target mechanism enables the system to respond differently depending on the severity of the inventory depletion. If the inventory falls between s^* and s, the system orders up to level S. If the inventory drops below s^*, the system reacts more aggressively by ordering up to S^*, a higher replenishment level.

In contrast, the classical (s, S) policy has a single threshold and target, triggering uniform ordering behavior regardless of how critically low the inventory level is; therefore, it does not structurally differentiate between moderate and severe inventory shortages as the proposed adaptive policy does. This distinction enhances the adaptive inventory policy flexibility and cost responsiveness by incorporating dynamic decision-making into the policy structure, especially in environments with stochastic lead times and highly variable demand. Furthermore, a specially designed Genetic Algorithm (GA) is employed to optimize the proposed policy parameters via simulation-based evaluation. The effectiveness of the approach, as demonstrated via multiple experiments and sensitivity analyses, indicates consistent cost savings across different scenarios. Together, these contributions distinguish this work and its applicability in complex and realistic inventory systems.

The remainder of this paper is structured as follows. Section 2 presents a literature review on inventory management policies. Next, section 3 describes the inventory problem and optimization objective. Then, section 4 presents the proposed inventory model, and section 5 details the GA used for parameter optimization. Finally, section 6 discusses the simulation results and sensitivity analysis, and section 7 provides conclusions and future research directions.

2. Literature review

Inventory control policies have long attracted the attention of operations management scholars, primarily due to their critical role in optimizing stock levels while minimizing holding and shortage costs. Among various strategies, the (s, S) policy has emerged as a foundational framework for managing inventories under uncertainty. This review synthesizes key contributions in the literature related to the optimization of (s, S) policies, recent advancements, and the emergence of more complex dynamic approaches.

The foundational contributions of the (s, S) policy have been thoroughly investigated in earlier studies, most notably by [1], who presented an efficient algorithm for computing an optimal (s, S) policy. Their work highlights the feasibility and effectiveness of these types of policies in various contexts, including diverse demand patterns. Similarly, foundational contributions include the study [2] analyzing the efficacy of (s, S) types of policies under fixed holding and shortage costs. Another study [3] further simplified the problem, demonstrating that finding optimal (s, S) policies can be computationally efficient, indicating a shift toward practical implementation in inventory management. Another study [4] provided insight into the optimality of (s, S) policies in stochastic environments characterized by penalty costs, reinforcing the foundational principles established in earlier work.

In the realm of algorithmic advancements, another study [5] introduced new algorithms for optimal (s, S) policies in stochastic single-item inventory systems. Concurrently, advances in simulation optimization have been made [6, 7], introducing annealing techniques to navigate complex objective landscapes. Multi-objective simulation optimization has also been explored [8]. Moreover, another study [9] advocated for clustering techniques as valuable tools in inventory control, enriching the methods available for researchers and practitioners.

Numerous extensions have enriched this framework. For example, dual supply modes with distinct order-up-to levels have been introduced [10]. Johansen and Thorstenson [11] combined (r, Q) and (s, S) policies for regular and emergency orders. In further developments, studies examined systems with piecewise-linear concave ordering costs [12] and straightforward policies for managing perishable inventory [13]. The latter study emphasized the need for adaptable strategies in sectors where product life cycles are short. Queueing and Markov Chain Decomposition (QMCD) methodology [14] was introduced for perishability models, incorporating lead time considerations, which added another dimension to the (s, S) policy framework.

Recent literature has significantly expanded the realm of inventory management. Researchers [15] have explored joint pricing and inventory control via nonparametric learning algorithms, revealing innovative approaches to dealing with lost sales and censored demand. Moreover, another study [16] focused on intelligent management techniques for perishable pharmaceutical products in healthcare supply chains, emphasizing the need for customized inventory strategies in specialized sectors. Additionally, another paper [17] explored (s, S) inventory policies for Lindley systems with unbounded costs to minimize the expected discounted total costs. The study also revealed the existence of optimal policies and provided a numerical example for illustration. Another work [18] addressed the ordering of Coronavirus Disease 2019 (COVID-19) vaccines using optimal dynamic policies, highlighting the increasing relevance of (s, S) policies in contemporary environments characterized by rapid changes and heightened demand volatility. Recent surveys [19, 20] have consolidated these developments by reaffirming the optimality of traditional (s, S) and related policies in continuous- and discrete-time frameworks, especially under fixed-cost conditions. Their work, along with that of others, indicates a continuing evolution in inventory control driven by technological and methodological innovations. The literature (e.g., [21, 22]) offers additional information regarding the inventory system.

Despite these advancements, many traditional models do not fully capture dynamic environments characterized by stochastic lead times, variable demand, and information delays. In response, the research community has shifted toward adaptive and hybrid inventory policies, such as the dynamic structure proposed in this paper. The (s^*, s, S, S^*) policy represents a significant step in this direction by enabling differentiated responses to varying degrees of inventory depletion, improving cost-efficiency and service levels in uncertain systems. Table 1 summarizes the literature regarding policies, contributions, optimization methods, and findings.

Table 1. Comparison of related literature with the proposed policy

Ref.	Policy type	Contribution	Optimization method	Findings
[1]	Classical (s, S)	Efficient algorithm for optimal (s, S) policies	Analytical	Effectively determines optimal (s, S) policies with minimal computational effort.
[2]	Classical (s, S)	Analysis under fixed holding and shortage costs	Analytical	Significantly reduces the risk of stock-outs
[3]	Classical (s, S)	Novel algorithm to identify an optimal single (s, S) policy	Analytical	Simplifies the process of calculating optimal (s, S) policies.
[4]	Stochastic (s, S)	Include proportional/lump-sum penalty cost	Analytical	Strengthened classical policies in uncertainty and penalty costs
[5]	Stochastic (s, S)	New algorithms for stochastic environments	Heuristic	Obtaining optimal inventory levels in stochastic environments
[6]	Stochastic (s, S)	Simulated annealing for inventory optimization	Metaheuristic	Enhanced search performance
[7]	Stochastic (s, S)	Simulation-based optimization with Multi-objective	Simulated annealing	The algorithm converges rapidly
[8]	Stochastic (s, S)	Combined annealing with stochastic models	Metaheuristic	Effectively solve complex multi-objective inventory systems
[11]	Stochastic (s, S)	Combined regular and emergency ordering	Heuristic	Useful under urgent ordering conditions
[12]	Stochastic (s, S)	Inventory control policy under piecewise-linear concave ordering cost structures	Analytical	(s, S) policy remains optimal under complex cost conditions
[14]	Stochastic (s, S)	QMCD methodology with perishability and delays	Analytical	Lead time variability leads to higher costs
[16]	Intelligent inventory management IIM	AI-driven control in healthcare logistics	AI-based	Effective in specialized supply environments
[17]	Stochastic (s, S)	Development of policy that addresses the complexities of lost sales in a stochastic environment	Numerical	Minimizes total costs while accommodating the stochastic nature of demand and the occurrence of lost sales
This Study	Stochastic (s^*, s, S, S^*)	Proposes a dual-threshold, dual-target adaptive policy	Genetic Algorithm (GA)	Reduces cost up to 6.7%; outperforms (s, S) policy under variable demand/lead time

3. Problem description

Inventory management is crucial for ensuring operational continuity and customer satisfaction in supply-driven businesses. The primary challenge lies in determining the optimal quantities and timing of reordering under uncertain demand, fluctuating delivery times, and limited storage capacity. This problem is formally embodied in the Inventory Policy Optimization Problem (IPOP), where the goal is to minimize total inventory costs, including ordering, holding, and shortage penalties, while maintaining a service level that avoids excess and out-of-stock inventory. Traditional models, such as the (s, S) policy, aid in inventory management but lack the precision to address practical complexities, including handling extremely low inventory levels or variable demand frequency.

The proposed policy (s^*, s, S, S^*) addresses this problem by introducing a dual-threshold adaptive strategy: one that responds differently to moderate versus severe stockouts. This flexibility enables improved cost-control decisions. From a practical perspective, the IPOP model addresses challenges faced in diverse sectors (e.g., retail, pharmaceuticals, and logistics), where dynamic consumer behavior and supply chain disruptions lead to insufficient fixed inventory rules. Formulating this problem and solving it using a sophisticated algorithm contributes to the development of a robust and easily implementable decision-making policy.

The goal of this section is to outline the IPOP. Terminology from [23] was applied. Subsections 3.1, 3.2, and 3.3 provide the nomenclature, notation and assumptions, and objective functions, respectively.

3.1 *Terminology*

1. ***Item*** refers to a product ordered by a company from a supplier, which is held in inventory and sold to customers.
2. ***Ordering*** indicates the stage of shipping items from the supplier to the warehouse based on company requests.
3. ***Ordering cost*** refers to expenses incurred when ordering items and processing the order.
4. ***Setup cost*** is defined as the expense incurred for ordering and preparing an order.
5. ***Lead time*** refers to the time it takes for a supplier to deliver an order to the warehouse.
6. ***Demand*** indicates the number of items customers purchase from a company.
7. ***Maximum demand*** refers to the maximum number of items demanded by customers.
8. ***Demand time*** is the time interval between successive customer demands.
9. ***Backlog*** refers to the number of items that cannot be delivered to the customer because they are not in the warehouse until the items become available in stock, which is known as a full backlog.
10. ***Holding cost*** refers to the expense incurred by a company for storing an item. This cost includes insurance, taxes, warehouse rent, and maintenance.
11. ***Shortage cost*** refers to the expense incurred by a company for every item in a backlog. This expense includes fines that the company must pay for delaying customer demand.
12. ***Inventory level*** refers to the number of items, which is positive when it includes the number of items in the warehouse or negative when it includes the number of backlog items.

3.2 *Notations and assumptions*

This section identifies the key parameters and variables employed in the model, followed by the assumptions that support the inventory policy optimization framework.

Notation

- s^*: Critical reorder point (emergency threshold)
- s: Primary reorder point
- S: Target inventory level when inventory drops below s but above s^*
- S^*: Higher target inventory level when inventory drops below s^*
- L_t: Inventory level at time t (may be negative)
- L_t^+: Positive inventory level at time t, $\max(0,\ L_t)$
- L_t^-: Backlogged items at time t, $\max(0,\ -L_t)$

- n_i: Number of items ordered in the ith month
- M: Planning horizon in months
- ST: Fixed setup cost per order
- β: Cost per unit item ordered
- HC: Holding Cost per unit per month
- SC: Shortage (backlog) cost per unit per month
- IC: Inventory Capacity
- D: Random variable representing customer demand per order
- DT: Time between successive demands (exponentially distributed)
- MS: Maximum demand per customer
- p_k: Probability of a customer demanding k items
- lt: Lead time, uniformly distributed over $[v, w]$
- μ: Mean of the exponential distribution governing DT
- OC_i: Ordering Cost in month i, $OC_i = ST + \beta n_i$
- AOC: Average Ordering Cost
- AHC: Average Holding Cost
- ASC: Average Shortage Cost
- AC: Total Average Cost = $AOC + AHC + ASC$.

Assumptions

- The company deals with a single item and reviews inventory monthly.
- Inventory replenishment decisions are made at the beginning of each month.
- The lead time lt is a random variable that follows a uniform distribution on $[v, w]$.
- The customer demand per order follows a discrete distribution p_k up to MS.
- The interarrival time of demand DT follows an exponential distribution with mean μ.
- Partial backlogging is allowed, and the backlog is cleared upon the arrival of new inventory.
- The inventory capacity is finite and bound by IC.
- The cost function to minimize costs includes average ordering, holding, and shortage costs.
- The GA optimizes over the four-tuple (s^*, s, S, S^*), subject to ordering constraints.

3.3 *Objective function*

The IPOP aims to decrease the sum of the three components. The three components are the average ordering cost, average holding cost, and average shortage cost.

Average Ordering Cost (AOC):

If the company places an order for n_i items in the ith month, where $i = 1, \ldots, M$, the ordering cost OC_i can be determined as follows: $OC_i = ST + \beta n_i$, where β denotes the cost of one item, and ST indicates the setup cost. Hence, within M months, the average ordering cost (i.e., AOC) each month is estimated as $AOC = \frac{1}{M} \sum_{i=1}^{M} OC_i$.

Average Holding Cost (AHC):

The AHC for M months is determined as $AHC = HC \cdot \bar{L}^+$, where $\bar{L}^+$ represents the average number of items each month in the inventory for M months; thus, $\bar{L}^+ = \frac{1}{M} \int_0^M L_t^+ dt$, where L_t^+ indicates the number of items in the company warehouse during time t, $L_t^+ = max\{0, L_t\}$, and L_t denotes the inventory levels during time t, where $0 \leqslant t \leqslant M$, where HC represents the holding cost.

Average Shortage Cost (ASC):

The ASC each month is determined as follows: $ASC = SC \cdot \bar{L}^-$, where $\bar{L}^-$ represents the average number of items each month that are backlog for M months; hence, $\bar{L}^- = \frac{1}{M} \int_0^M L_t^- dt$, where L_t^- indicates the number of backlogged items during time t, and $L_t^- = max\{0, -L_t\}$, where SC represents the shortage cost.

Every company strives to decrease its Average Cost (AC). The average cost refers to the sum of the three minimizing components: AOC, AHC, and ASC, that is, $AC = AOC + AHC + ASC$. To achieve this objective, the following constraints are incorporated into the inventory policy framework:

1. Customer demand: The number of items purchased by customers, which could be less than or equal to the maximum demand.

2. Order arrival: When the order arrives, the company must clear its backlog, and afterward, additional items, if any, can be added to the inventory.

3. Order quantity: Using the inventory level as a guide, the order quantity is determined at the start of every month to ensure that it does not surpass the inventory capacity plus the number of backlogged items.

These constraints are essential to maintaining system feasibility while enabling the optimization algorithm to minimize the overall average cost.

4. Model

If a company sells a single product, it must estimate the number of items that it must maintain in inventory every month for the next M months. The inventory level is examined at the start of every month. Depending on the number of items in the inventory, the company determines the number of items it must order from the supplier. Thus, the OC_i can be estimated for $i = 1, \ldots, M$. The lead time (lt) refers to a random variable with a uniform distribution between v and w months, where v and w are real numbers and $0 < v < w$, which assumes that any value within a specified range is equally likely to occur. If the lead time is $lt \sim U(v, w)$, then lt has an equal probability of taking any value between v and w, where v and w denote the lower and upper bounds of the lead time. Hence, the lead times are estimated using equation (1):

$$lt = v + \text{rand} \cdot (w - v) \tag{1}$$

where *rand* denotes a random number between 0 and 1.

Any company can apply the proposed model to adapt the inventory policy for dynamic ordering (s^*, s, S, S^*) to determine the number of items that must be ordered, where s^*, s, S, and S^* are integer numbers. Moreover, $0 < s < S < S^* \leq IC$ and $-IC < s^* < s$, and IC represents the inventory capacity. The inventory policy (s^*, s, S, S^*) indicates that, at the start of the ith month, when L_i decreases below s but is above s^*, $S - L_i$ items must be ordered. However, if L_i decreases below s^*, then $S^* - L_i$ items must be ordered. Thus, the item order quantity can be calculated using equation (2):

$$n_i = \begin{cases} S - L_i & s^* < L_i \leq s \\ S^* - L_i & L_i \leq s^* \\ 0 & L_i > s \end{cases} \tag{2}$$

where DT denotes the demand time, which is presumed to be an IID exponential random variable with a mean of μ months. Equation (3) can estimate the DT as follows:

$$DT = -\mu \ln(\text{rand}) \tag{3}$$

where D represents the value of the discrete random variable distributed as demonstrated in equation (4).

$$D = \begin{cases} 1 & \text{with probability } p_1 \\ 2 & \text{with probability } p_2 \\ \vdots & \vdots \\ MS & \text{with probability } p_{MS} \end{cases} \tag{4}$$

The maximum size of customer demand for the item is denoted by MS. In addition, p_k indicates the probability of the customer demand for k items, where $k = 1, \ldots, MS$, and $\sum_{k=1}^{MS} p_k = 1$. Two scenarios exist for the demand process. The demand can be less than the number of items defined in the inventory level, in which case the demand is executed. Alternatively, the demand can be higher than the number of items described in the inventory level, in which case some of the demand is implemented, and the remaining backlog is completed later after the order arrives from the supplier.

Therefore, the mathematical IPOP model is as follows:

$$\begin{aligned}
\min \quad & AC = AOC + AHC + ASC \\
\text{s.t.} \quad & 1 \le D \le MS, \\
& p_k \le 1, && \forall k = 1, \ldots, MS \\
& \sum_{k=1}^{MS} p_k = 1, \\
& 0 \le n_i \le IC - L_i, && \forall i = 1, \ldots, M \\
& v \le lt \le w, \\
& L_t \le IC, && 0 \le t \le M \\
& 0 < s < S < S^* \le IC, \\
& -IC < s^* < s, \\
& D,\ MS,\ IC,\ M,\ s,\ S,\ S^* \in \mathbb{N}, \\
& s^* \in \mathbb{Z}, \\
& n_i \in \mathbb{N}, && \forall i = 1, \ldots, M \\
& L_t \in \mathbb{Z}, && 0 \le t \le M
\end{aligned} \tag{5}$$

Figure 1 describes L_t, L_t^+, and L_t^- for four months, which can be used as an example of an inventory policy $(s^*,\ s,\ S,\ S^*)$. After the first month, the inventory level was lower than s, $L_1 < s$. Thereafter, the company must reorder $n_1 = S - L_1$ items. After a fixed lead time, the order must be added to the existing inventory. After the second month, the inventory level increases above s, $L_2 > s$. Therefore, the company does not have to reorder, as $n_2 = 0$. During the second month, the inventory level is less than s^*. Furthermore, an inventory shortage exists owing to the variations in L_t^+ and L_t^-, where L_t^- changes from 0 to a positive number, whereas the positive L_t^+ value changes to 0. Hence, after the third month, the number of ordered items is calculated as $n_3 = S^* - L_3$. If the items are received after lead time, the inventory shortage can be covered, and the remaining items are added to the inventory level. Thus, over time, after the fourth month, the inventory level decreases below s, $L_4 < s$, and the order quantity ranges are estimated as $n_4 = S - L_4$.

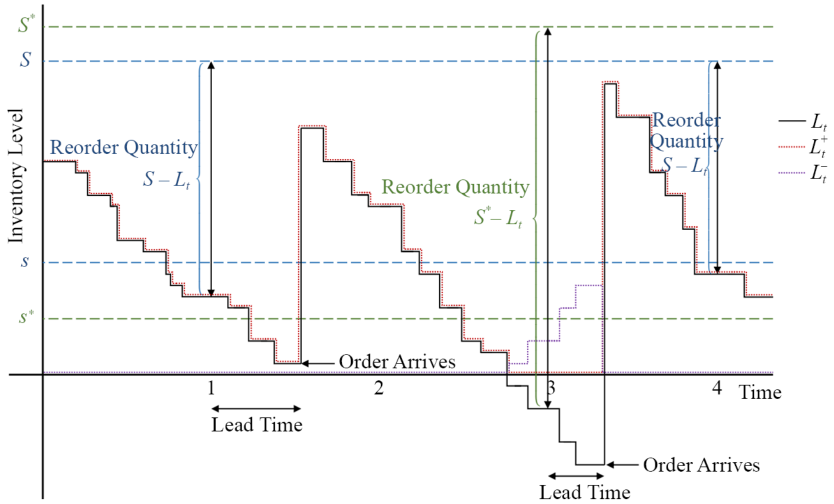


Figure 1. Example of the inventory policy for four months $(s^*,\ s,\ S,\ S^*)$, presents the L_t, L_t^+, and L_t^-

The IPOP model solutions are discrete, nonlinear, and situated in a stochastic simulation framework. The GA is well-suited to address such challenges because it offers global search capabilities, resilience in uncertain environments, and adaptability in managing multiple policy parameters simultaneously. Consequently, the GA was selected for this study due to its proven effectiveness in exploring feasible solutions. Section 5 introduces the GA as the chosen optimization technique, given its capacity to effectively address the stochastic, nonlinear, and discrete characteristics of the IPOP conditions under which conventional optimization methods are often inadequate.

5. Algorithm

The GA method is called the "intelligent" probability search algorithm [24], which is based on the theory of biological evolution. It depends on two critical components: chromosomes and genes. This method can be used to solve a variety of combinatorial optimization problems. The primary objective of the GA simulation is the survival of the fittest. The GA employs several variables to model optimization problems. A chromosome is a representation of the solution to an optimization problem, which includes a vector of genes, where every gene is a variable in the optimization problem (for further information regarding the GA, see [25, 26]).

In this problem, chromosomes can be denoted by $(s^*,\ s,\ S,\ S^*)$, and the gene can be represented as a variable $\{s^*,\ s,\ S,\ S^*\}$. The GA uses the three major operators of selection, crossover, and mutation to improve the chromosomes in each generation. The following subsections cover each of these steps and the initial population.

5.1 *Chromosome and gene representation*

As described in the earlier section, every chromosome offers a candidate solution to the IPOP problem. A chromosome, $\mathscr{C} = (s^*,\ s,\ S,\ S^*)$, refers to a vector comprising many genes, which stimulate variables in the IPOP. The parameters included in the chromosome are ascending, offering an effective solution (i.e., $s^* < s < S < S^*$). The three parameters (i.e., s, S, and S^*) are positive integers, and s^* is an integer. For example, when $0 < s < S < S^* < 5$ and $-2 < s^* < 3$, 22 candidate solutions are presented in Table 2.

Table 2. Example for all candidate solutions, where $0 < s < S < S^* < 5$ and $-2 < s^* < 3$

s^*	s	S	S^*	s^*	s	S	S^*
-2	1	2	3	-1	3	4	5
-2	1	2	4	0	1	2	3
-2	1	2	5	0	1	2	4
-2	2	3	4	0	1	2	5
-2	2	3	5	0	2	3	4
-2	3	4	5	0	2	3	5
-1	1	2	3	0	3	4	5
-1	1	2	4	1	2	3	4
-1	1	2	5	1	2	3	5
-1	2	3	4	1	3	4	5
-1	2	3	5	2	3	4	5

5.2 *Initial population*

The GA initiates the optimization to determine effective solutions using the initial population $\mathscr{P}_0$ of random chromosomes. The following Eqs. (6)-(9), describe the mechanism of the jth chromosome, where $\mathscr{C}_{0j} = (s^*_{0j},\ s_{0j},\ S_{0j},\ S^*_{0j})$ refers to the initial population as the candidate solutions:

$$s^*_{0j} = \lfloor (b_1 - a + 1) \cdot r \rfloor + a \tag{6}$$

$$s_{0j} = \left\lfloor \left(b_2 - s^*_{0j}\right) \cdot r \right\rfloor + s^*_{0j} \tag{7}$$

$$S_{0j} = \left\lfloor \left(b_3 - s_{0j}\right) \cdot r \right\rfloor + s_{0j} \tag{8}$$

$$S^*_{0j} = \left\lfloor \left(b_4 - S_{0j}\right) \cdot r \right\rfloor + S_{0j} \tag{9}$$

where a denotes the minimal value of s^*, r represents a random number in $(0,\ 1)$, and b_1, b_2, b_3, and b_4 are maximal values of s^*, s, S, and S^* respectively. In the proposed IPOP, the population size $|\mathscr{P}_i| = 10$, and $\mathscr{P}_i = \{\mathscr{C}_{i1},$ where $\mathscr{C}_{i2},\ \ldots,\ \mathscr{C}_{i10}$ indicates the population at iteration i.

5.3 *Selection*

The GA implements the selection operator for each iteration to choose chromosomes from the initial population that must be recombined. The selection operator increases in accordance with fitness and provides more opportunities to select chromosomes. Their genes are passed on to succeeding generations owing to effective chromosomal selection. In IPOP, chromosomes were selected to generate offspring using two stages in iteration i.

First stage: A subpopulation $\mathscr{P}_i^* = \left\{\mathscr{C}_{i1}^*, \mathscr{C}_{i2}^*, \ldots, \mathscr{C}_{i\eta}^*\right\} \subseteq \mathscr{P}_i$ was presented with a size of η, where $3 \leq \eta \leq |\mathscr{P}_i|$ from the initial population using the ith iteration at a probability of $\rho = 0.7$. This setup indicates that every chromosome in the initial population could be selected (probability of 0.7), so that it could be defined as a chromosome in the subpopulation. If the subpopulation size is less than three, the probability of choosing a chromosome not included in the subpopulation is repeated. Figure 2 depicts the flowchart describing the generation of the subpopulation set. Weak chromosomes can generate new offspring because they may contain beneficial genes.

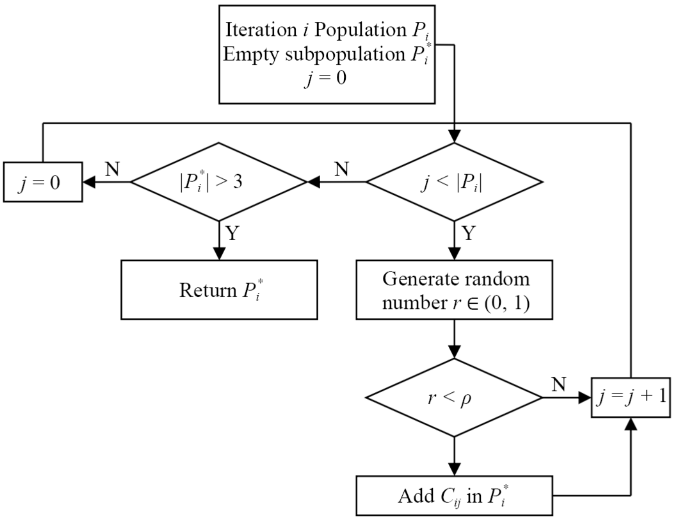


Figure 2. Flowchart to generate the subpopulation $\mathscr{P}_i^*$

Second stage: The roulette wheel technique [27] was employed to select η^* chromosomes randomly from subpopulation $\mathscr{P}_i^*$, such that the likelihood of selecting high-performance chromosomes is high. The following steps explain the mechanisms of selecting η^* chromosomes using the roulette wheel technique:

1. Evaluate the average cost AC_{ij} of every chromosome $\mathscr{C}_{ij}^*$ in the $\mathscr{P}_i^*$ subpopulation.
2. Calculate the selection probability ρ_{ij} of every chromosome $\mathscr{C}_{ij}^*$ using equation (10).

$$\rho_{ij} = \frac{AC_{ij}}{\sum_{j=1}^{\eta^*} AC_{ij}} \tag{10}$$

3. Calculate the cumulative probability $\rho_{ij} = \sum_{k=1}^{j} \rho_{ik}$.
4. Determine a random number $r \in (0,\ 1]$.
5. Select chromosome $\mathscr{C}_{ij}^*$, if $\rho_{ij-1} < r \leq \rho_{ij}$, where $\rho_{i0} = 0$.
6. Repeat Steps 4 and 5, η^* times to select η^* chromosomes (candidate solution).

The IPOP model assumes that $\eta^* = 3$, which could be used to generate three to six new offspring. If $\eta^* = 2$, only one to two new offspring are available, decreasing the speed of GA.

5.4 *Crossover*

The gene exchange across chromosomes (parents) produces new offspring (children). Children receive genes from both parents via gene exchanges from the two chromosomes. The crossover is conducted via a randomly chosen crossover point to generate new candidate solutions using parent genes (single point [28]). Every chromosome in this problem includes three crossover points: τ_1^*, τ_2^*, and τ_3^*. The selected crossover point is dependent on the random number $\tau \in [0,\ 1]$, wherein the algorithm defines the crossover point using equation (11):

$$\text{crossover point} = \begin{cases} \tau_1^*, & 0 \leq \tau < \dfrac{1}{3} \\ \tau_2^*, & \dfrac{1}{3} \leq \tau < \dfrac{2}{3} \\ \tau_3^*, & \dfrac{2}{3} \leq \tau \leq 1 \end{cases} \tag{11}$$

Figure 3 presents crossover steps used in the GA with an example.

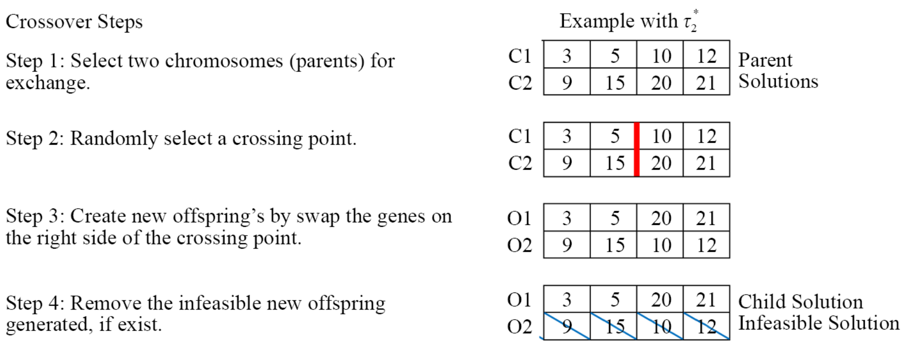


Figure 3. Crossover steps to generate the child solutions and example with τ_2^*

As mentioned, the number of chromosomes selected from the subpopulation $\mathscr{P}_i^*$ to produce new solutions is three, and the above steps are performed on each pair of these selected chromosomes. Thus, three to six new candidate solutions are created. Before these solutions are added to the population, some of the new solutions may undergo genetic mutations. The following section describes the mutation process occurring on new solutions.

5.5 *Mutation*

During the crossover stage, genes are exchanged from both parents, and no new genes are added to the offspring [26]. Thus, new genes that differ from the parents are not generated. Consequently, the mutation stage is crucial to prevent early convergence. The mutation process leads to the onset of random variations in the genes. The parameter that leads to mutation is known as the mutation probability ω, where $0 \leq \omega \leq 1$, which can be applied for every gene that could generate a child solution developed during the crossover stage. Random numbers, such as $\mathfrak{g}_i \in [0,\ 1]$, where $i = 1,\ 2,\ 3,\ 4$, could be generated for the ith gene in the offspring. If $\mathfrak{g}_i < \omega$, then the ith gene could be replaced using a random integer between the upper and lower limits of the gene. A random integer was generated for the offspring chromosome, $\mathscr{C} = (s^*,\ s,\ S,\ S^*)$ using equation (12):

$$\begin{aligned} s^* &\leftarrow \lfloor (s - (a+1)) * r \rfloor + a && \text{if } \mathfrak{g}_1 < \omega \\ s &\leftarrow \lfloor (S - (s^*+1)) * r \rfloor + s^* + 1 && \text{if } \mathfrak{g}_2 < \omega \end{aligned}$$

$$
\begin{aligned}
S &\leftarrow \lfloor (S^* - (s+1)) * r \rfloor + s + 1 \quad \text{if } \mathfrak{g}_3 < \omega \\
S^* &\leftarrow \lfloor (b_4 - (S+1)) * r \rfloor + S + 1 \quad \text{if } \mathfrak{g}_4 < \omega
\end{aligned}
\tag{12}
$$

where a denotes the minimum value of s^*, b_4 indicates the maximum value of S^*, and r represents a random number in $(0, 1)$. Figure 4 displays the mutation steps implemented by GA for the offspring chromosome, $\mathscr{C} = (s^*, s, S, S^*)$:

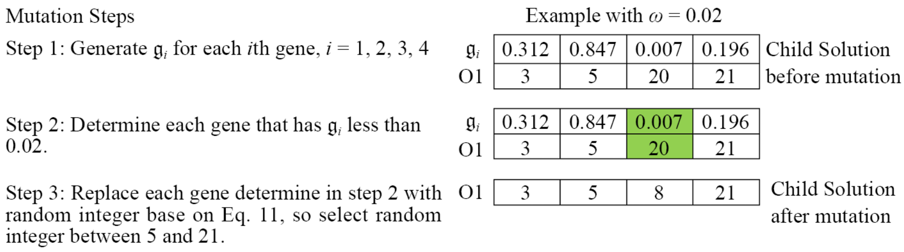


Figure 4. Mutation steps on the child solutions and example with $\omega = 0.02$

The presence of variations in a few genes after crossover can transfer the search for optimized solutions to new areas far from the area of local solutions.

5.6 *Updating the population*

This step is regarded as significant in developing a new population for the next iteration. All solutions in the $\mathscr{P}_i$ population and new offspring solutions, acquired after the crossover and mutation stages, were included in the $\hat{\mathscr{P}}_i$ set. Thereafter, the *AC* value is estimated for every solution in the $\hat{\mathscr{P}}_i$ set. Then, the $\hat{\mathscr{P}}_i$ set is updated after eliminating weak solutions from the sets to generate $\hat{\mathscr{P}}_i$ with a size of η. The new set has a population of $\mathscr{P}_{i+1} \leftarrow \hat{\mathscr{P}}_i$.

5.7 *Stopping criterion*

The algorithm is halted if the number of iterations reaches a threshold. Thereafter, the best solution from the last population is designated as the best solution.

Although the GA offers considerable flexibility in solving complex, nonlinear optimization problems, it has limitations. The performance of the GA can be affected by parameter settings (e.g., the population size and mutation rate). Further, convergence to the global optimum is not guaranteed, especially in highly rugged search spaces. Moreover, the GA often requires numerous evaluations, which can be computationally intensive when combined with simulation models. Therefore, this work conducts extensive experiments to verify the effectiveness of the proposed policy. Section 6 presents the simulation results, evaluating the performance of the optimized (s^*, s, S, S^*) policy under various conditions using the GA. The following section also presents a sensitivity analysis and comparisons with traditional inventory policies to demonstrate the robustness and cost-effectiveness of the proposed approach.

6. Numerical results

The proposed GA technique (Section 5), which can be employed to address the IPOP, was set up using a computer with a Hewlett-Packard Intel® core™ i7-4790 CPU with a 3.60 GHz processor and 16 GB RAM and MATLAB (R2021a). Various experiments with different maximal demands, demand times, and lead times were implemented using the GA to verify the efficiency of the dynamic order approach (s^*, s, S, S^*) inventory policy compared with the traditional inventory policy (s, S).

The experiments were conducted on three cases with varying lead times, $lt \in (0,\ 0.1)$, $(0.03,\ 0.5)$, and $(0.7,\ 0.9)$, representing a fluctuating lead time, rapid lead time, and significant lead time, respectively. The experiments used three distinct maximum demand values ($MS = 5,\ 50,\ 100$), indicating low, medium, and high demand, respectively. In addition, three distinct mean demand times ($\mu = 0.001,\ 0.01,\ 0.1$) were used, indicating that the number of customers was frequent and high, acceptable, or low, respectively. Table 3 displays the IPOP parameter values for this study, and Table 4 displays the GA parameter values.

Table 3. IPOP parameters

Description	Parameter
The cost of item	$\beta = 5\$$
The cost of setup	$ST = 20\$$
The holding cost per item per month	$HC = 1\$$
The shortage cost per item per month	$SC = 0.5\$$
The initial inventory level	$L_t = \text{ round } \left(\frac{S+s}{2}\right)$
Number of months	$M = 120$
The inventory capacity	$IC = 1,000$
The minimum value of s^*	$a = -50$
The minimum value of s, S, and S^*	0
The maximum value of and	$b_1,\ b_2,\ b_3,\ b_4 = 1,000$
The probability of the customer demand k items	Equations (13), (14), (15), and $\sum_{k=1}^{MS} p_k = 1$
Number of iterations	$it = 100$

Table 4. Parameter values of GA algorithm

Description	Parameter
The size of population in the ith iteration	$\lvert\mathscr{P}_i\rvert = 10$
Probability value for generating a subpopulation	$\rho = 0.7$
Number of chromosomes that are selected from a subpopulation to generate new offspring	$\eta^* = 3$
Mutation probability	$\omega = 0.02$
Number of iterations	$iter = 10^5$

As $\sum_{k=1}^{MS} p_k = 1$, the likelihood that the customer demand k for items p_k is based on the MS value. Equations (13)-(15) present p_k, where $k = 1,\ 2,\ \ldots,\ MS$, which is used in all experiments conducted in this study.

$$MS = 5 : p_k = \begin{cases} \frac{1}{4}, & 1 \leq k \leq 3 \\ \\ \frac{1}{8}, & 3 < k \leq 5 \end{cases} \tag{13}$$

$$MS = 50: p_k = \begin{cases} \frac{30}{1,000}, & 1 \le k \le 20 \\ \frac{15}{1,000}, & 20 < k \le 40 \\ \frac{10}{1,000}, & 40 < k \le 50 \end{cases} \tag{14}$$

$$MS = 100: p_k = \begin{cases} \frac{20}{1,000}, & 1 \le k \le 25 \\ \frac{10}{1,000}, & 25 < k \le 50 \\ \frac{6}{1,000}, & 50 < k \le 75 \\ \frac{4}{1,000}, & 75 < k \le 100 \end{cases} \tag{15}$$

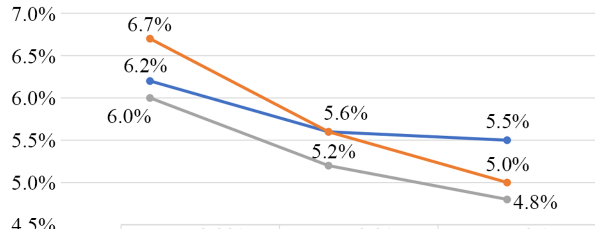
The GA conducts 10^5 iterations, and in every iteration, the GA calculates the objective functions, such as *AC* (related to the solutions in the population and candidate solutions obtained after crossover and mutation), 100 times via simultaneous repetitions. The GA was implemented in 54 experiments, with 27 experiments conducted for both policies, (s^*, s, S^*) and (s, S). Then, optimal solutions were determined for both policies for every experiment. All candidate solutions in the simulation satisfy the feasibility conditions of the proposed policy structure, including that s^*, s, S, and S^* are integers, where $0 < s < S < S^* \le IC$ and $-IC < s^* < s$ and n_i denotes a bounded order quantity. These constraints ensure that the simulated policies are logically sound and practically implementable.

Table 5 presents the *AC* values for the 54 experiments using various *MS*, μ, and *lt* values and two inventory policies, (s^*, s, S, S^*) and (s, S). Thus, the *AC* values were calculated for all experiments, and the *AC* value derived using the (s^*, s, S, S^*) policy was better than the (s, S) policy. Furthermore, the (s^*, s, S, S^*) policy was more effective when many customers were involved. In addition, the (s^*, s, S, S^*) policy was very efficient when lead times varied.

Table 5. The values of *AC* for 54 experiments with different values of the three parameters *MS*, μ, and *lt* for inventory policies $(s^*,\ s,\ S,\ S^*)$ and $(s,\ S)$

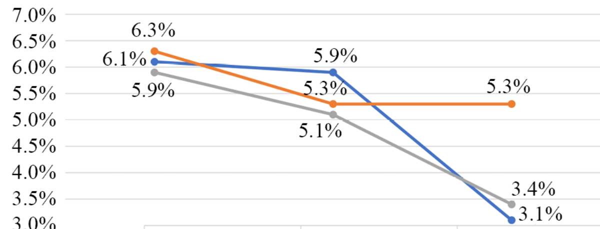
MS	$\mu \setminus lt$	(0, 0.1)		(0.03, 0.5)		(0.7, 0.9)	
		AC for policy $(s^*,\ s,\ S,\ S^*)$	AC for policy $(s,\ S)$	AC for policy $(s^*,\ s,\ S,\ S^*)$	AC for policy $(s,\ S)$	AC for policy $(s^*,\ s,\ S,\ S^*)$	AC for policy $(s,\ S)$
5	0.001	13,396	14,275	13,520	14,497	14,162	15,073
	0.01	1,374	1,456	1,359	1,440	1,345	1,419
	0.1	171	181	170	179	159	167
50	0.001	103,312	110,069	105,371	112,492	110,613	117,501
	0.01	10,009	10,640	10,051	10,617	10,443	11,007
	0.1	1,040	1,073	1,033	1,091	1,030	1,066
100	0.001	177,358	188,409	180,499	191,214	189,656	201,879
	0.01	17,015	17,976	17,326	18,099	18,240	19,177
	0.1	1,747	1,832	1,711	1,791	1,705	1,790

The study results reveal that implementing the new proposed policy (s^*, s, S, S^*) could decrease the monthly cost for the company compared with the traditional (s, S) policy for $MS = 5$, $MS = 50$, and $MS = 100$ by up to 6.7%, 6.3%, and 6.1%, respectively (Figures 5, 6, and 7). The results also demonstrate an increase in the reduction percentages in the experiments where $\mu = 0.001$, compared to the experiments where it was calculated, which increased with a higher number of customers. The results also indicated that the new proposed policy (s^*, s, S, S^*) helped reduce the costs in all cases.



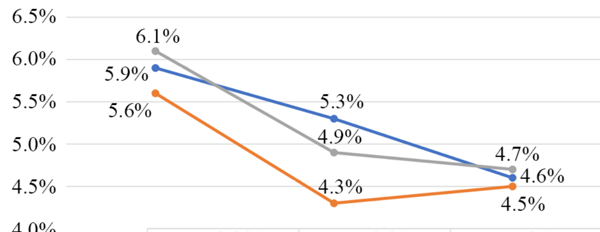
	$\mu = 0.001$	$\mu = 0.01$	$\mu = 0.1$
$lt = (0, 0.1)$	6.2%	5.6%	5.5%
$lt = (0.03, 0.5)$	6.7%	5.6%	5.0%
$lt = (0.7, 0.9)$	6.0%	5.2%	4.8%

Figure 5. Cost reduction percentage using (s^*, s, S, S^*) in experiments $MS = 5$



	$\mu = 0.001$	$\mu = 0.01$	$\mu = 0.1$
$lt = (0, 0.1)$	6.1%	5.9%	3.1%
$lt = (0.03, 0.5)$	6.3%	5.3%	5.3%
$lt = (0.7, 0.9)$	5.9%	5.1%	3.4%

Figure 6. Cost reduction percentage using (s^*, s, S, S^*) in experiments $MS = 50$



	$\mu = 0.001$	$\mu = 0.01$	$\mu = 0.1$
$lt = (0, 0.1)$	5.9%	5.3%	4.6%
$lt = (0.03, 0.5)$	5.6%	4.3%	4.5%
$lt = (0.7, 0.9)$	6.1%	4.9%	4.7%

Figure 7. Cost reduction percentage using (s^*, s, S, S^*) in experiments $MS = 100$

This work presents a sensitivity analysis to assess the effectiveness and robustness of the (s^*, s, S, S^*) inventory policy by varying parameters: maximum customer demand size for the item (MS), mean demand time (m), and interval of lead time (lt), which affect three components: AOC, AHC, and ASC. Each parameter was varied while keeping the others constant to observe the resulting average total cost (AC) changes. Although an increase occurred in AC as the mean demand time dropped, the proposed policy consistently produced a lower average total cost compared to the (s, S) policy. The dual-threshold design enabled the proposed model to adapt more effectively, ensuring service levels while keeping costs low. These findings underscore the flexibility and efficiency of this policy across operating conditions, highlighting its potential value in real-world inventory settings characterized by uncertain demand and fluctuating cost parameters.

7. Conclusions

The inventory system poses a significant problem, and many companies strive to control this problem to minimize costs. A novel policy (s^*, s, S, S^*) was proposed for controlling the inventory system of the company that sells a single item. This policy helped determine the number of items ordered based on dynamic ordering. This work compares the proposed policy with the traditional policy (s, S), finding that the new (s^*, s, S, S^*) policy helped improve the AC reduction by 6.7% to 3.1% in the 54 experiments. Thus, the newly proposed policy is very effective in improving the inventory system, regardless of the number of customers, demand for items, or time needed for order arrival.

Although the current study demonstrates the effectiveness of the (s^*, s, S, S^*) policy via extensive simulation experiments, it does not include real-world data. To further develop the proposed model, future research should aim to apply the proposed inventory policy to real experimental datasets in the retail, healthcare, and manufacturing sectors to validate its robustness and operational value in practical conditions.

Acknowledgement

The author thanks the Arab Open University, Saudi Arabia, for supporting this study.

Conflict of interest

The author declares that there is no conflict of interest regarding the publication of this paper.

References

[1] Federgruen A, Zipkin P. An efficient algorithm for computing optimal (s, S) policies. *Operations Research*. 1984; 32(6): 1268-1285.

[2] Ishigaki T, Sawaki K. On the (s, S) policy with fixed inventory holding and shortage costs. *Journal of the Operations Research Society of Japan*. 1991; 34(1): 47-57.

[3] Zheng YS, Federgruen A. Finding optimal (s, S) policies is about as simple as evaluating a single policy. *Operations Research*. 1991; 39(4): 654-665.

[4] Aneja Y, Noori AH. The optimality of (s, S) policies for a stochastic inventory problem with proportional and lump-sum penalty cost. *Management Science*. 1987; 33(6): 750-755.

[5] Feng Y, Xiao B. A new algorithm for computing optimal (s, S) policies in a stochastic single item/location inventory system. *IIE Transactions*. 2000; 32(11): 1081-1090.

[6] Ahmed MA, Alkhamis TM. Simulation-based optimization using simulated annealing with ranking and selection. *Computers & Operations Research*. 2002; 29(4): 387-402.

[7] Alrefaei MH, Diabat AH. A simulated annealing technique for multi-objective simulation optimization. *Applied Mathematics and Computation*. 2009; 215(8): 3029-3035.

[8] Alrefaei MH, Diabat A, Alawneh A, Al-Aomar R, Faisal MN. Simulated annealing for multi-objective stochastic optimization. *International Journal of Science and Applied Information Technology (IJSAIT)*. 2013; 2(2): 18-21.

[9] Balugani E, Lolli F, Gamberini R, Rimini B, Regattieri A. Clustering for inventory control systems. *IFAC-PapersOnLine*. 2018; 51(11): 1174-1179.

[10] Chiang C. Optimal replenishment for a periodic review inventory system with two supply modes. *European Journal of Operational Research*. 2003; 149(1): 229-244.
[11] Johansen SG, Thorstenson A. Emergency orders in the periodic-review inventory system with fixed ordering costs and compound Poisson demand. *International Journal of Production Economics*. 2014; 157: 147-157.
[12] Benjaafar S, Chen D, Yu Y. Optimal policies for inventory systems with concave ordering costs. *Naval Research Logistics (NRL)*. 2018; 65(4): 291-302.
[13] Zhang H, Zhang J, Zhang RQ. Simple policies with provable bounds for managing perishable inventory. *Production and Operations Management*. 2020; 29(11): 2637-2650.
[14] Barron Y, Baron O. QMCD approach for perishability models: The (S, s) control policy with lead time. *IISE Transactions*. 2020; 52(2): 133-150.
[15] Chen B, Chao X, Shi C. Nonparametric learning algorithms for joint pricing and inventory control with lost sales and censored demand. *Mathematics of Operations Research*. 2021; 46(2): 726-756.
[16] Ahmadi E, Mosadegh H, Maihami R, Ghalehkhondabi I, Sun M, Süer GA. Intelligent inventory management approaches for perishable pharmaceutical products in a healthcare supply chain. *Computers & Operations Research*. 2022; 147: 105968.
[17] Blancas-Rivera R, Cruz-Suárez H, Portillo-Ramírez G, López-Ríos R. (s, S) Inventory policies for stochastic controlled system of Lindley-type with lost-sales. *AIMS Mathematics*. 2023; 8(8): 19546-19565.
[18] Xu X, Sethi SP, Chung SH, Choi TM. Ordering COVID-19 vaccines for social welfare with information updating: Optimal dynamic order policies and vaccine selection in the digital age. *IISE Transactions*. 2024; 56(7): 729-745.
[19] Perera SC, Sethi SP. A survey of stochastic inventory models with fixed costs: Optimality of (s, S) and (s, S)-type policies-Continuous-time case. *Production and Operations Management*. 2023; 32(1): 154-169.
[20] Perera SC, Sethi SP. A survey of stochastic inventory models with fixed costs: Optimality of (s, S) and (s, S)-type policies-Discrete-time case. *Production and Operations Management*. 2023; 32(1): 131-153.
[21] Axsäter S. *Inventory Control*. Cham: Springer; 2015.
[22] Feinberg EA. Optimality conditions for inventory control. In: *Optimization Challenges in Complex, Networked and Risky Systems*. Cantonsville, MD: INFORMS; 2016. p.14-45.
[23] Altiok T, Melamed B. *Simulation Modeling and Analysis with Arena*. San Diego, CA: Elsevier; 2010.
[24] Holland JH. Genetic algorithms. *Scientific American*. 1992; 267(1): 66-73.
[25] Katoch S, Chauhan SS, Kumar V. A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*. 2021; 80(5): 8091-8126.
[26] Mirjalili S, Song Dong J, Sadiq AS, Faris H. Genetic algorithm: Theory, literature review, and application in image reconstruction. In: *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*. Cham: Springer; 2020. p.69-85.
[27] Cantú-Paz E. A survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux Et Systems Repartis*. 1998; 10(2): 141-171.
[28] Srinivas M, Patnaik LM. Genetic algorithms: A survey. *Computer*. 1994; 27(6): 17-26.