Scientia Iranica (2025) **32**(8): 5174

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions on Industrial Engineering
https://scientiairanica.sharif.edu

Inventory control and price discount policies for perishable products with age and price-dependent demand

Farzin Bazrafshan , Saeed Emami *, Hamid Mashreghi

Department of Industrial Engineering, Babol Noshiravani University of Technology, P.O. Box:47148-71167, Babol, Iran.

* Corresponding author: s_emami@nit.ac.ir (S. Emami)

Received 11 December 2020; Received in revised form 20 October 2021; Accepted 16 January 2022

Keywords

Inventory control;
Price discount;
Perishable products;
Particle Swarm Optimization (PSO) algorithm;
Genetic Algorithm (GA).

Abstract

In this paper, the inventory control and price discount problem for perishable products with price and age-dependent demand is investigated. The seller adjusts prices to influence demand and optimize profits through determining discount points, especially discount start time. A nonlinear mathematical model is proposed to find optimal order quantity, discount points, and prices before the product's expiration date to maximize profit. The developed model provides the number of discounts such that the shortage will not be allowed before the expiration date. It is observed that determining a proper discount start time provides an optimized sales plan with higher profit. Moreover, the Particle Swarm Optimization (PSO) and the Genetic Algorithm (GA) are applied to solve the problem. The Taguchi approach is used to find optimum control parameters of PSO and GA. To guarantee the validity of PSO and GA, the nonlinear model is solved by the Branch-And-Reduce Optimization Navigator (BARON) solver in General Algebraic Modeling System (GAMS) software. The performance of the algorithms is evaluated based on the real values of parameters for two perishable products (i.e., Cheese and Mayonnaise Sauce) and some random test problems. The computational results demonstrate that the proposed GA outperforms the PSO algorithm.

1. Introduction

In recent years, the demand for fresh products such as fruits, vegetables, milk, meat, yogurt, seafood, and bread has dramatically increased. The freshness of the perishable products as the main factor of quality is one of the important items for customer satisfaction. According to A. C. Nielson Company report, 88% of consumers always or frequently checks expiry dates [1] because the products become useless after their expiration date. Thus, it is observed that billions of dollars' worth of food are expired and wasted every month [2].

The demand for the perishable products is seriously decreased by increasing products' age and eventually these products become completely obsolete in their expiration date. Therefore, the quantity of the products and their age are two significant factors for the retailers. As the products start to deteriorate, the retailer must decide to change the price of products to control demand and to increase the revenue. As a result, the perishability and the time dependency of demand make the inventory control and pricing for these products much harder [3].

The deterioration rate that reduces the quality of a product, or comparing the duration of consumption of the product by a consumer with the remaining time until expiration date can effect on attracting the customers. Thus, by analyzing these factors in an accurate way, one can find out how many changes in the price can convince the customers to buy the product. In other words, changing the price of a perishable product in the form of discounts can lead to consumer reactions and the tendency to buy the product.

The inventory control problem of perishable products has been extensively researched since the 1970s [4]. This control is not effective with classic models, and high inventory costs are imposed on the system with the passing of time until the

To cite this article:
F. Bazrafshan, S. Emami, H. Mashreghi "Inventory control and price discount policies for perishable products with age and price-dependent demand", *Scientia Iranica* (2025); **32**(8): 5174. https://doi.org/10.24200/sci.2022.57315.5174

end of the period. In recent years, the use of joint inventory control and price discount models for perishable products has grown impressively [5]. Paying attention to the amount of consumption and reaching the appropriate product price to prevent additional costs for vendors and business owners has led to a pricing debate along with inventory control. Consequently, in recent years, most researchers have been trying to present an integrated model for inventory control and price discount decisions.

Burwell et al. [6], Rajan et al. [7], Abad [8,9], Mukhopadhyay et al. [10], You [11], Sana [12], Avinadav et al. [13] and Kaya and Polat [3] are some of influencing studies that considered coordinated pricing and inventory control for perishable products. Perishable items are referred to as rotting, damaged, evaporated, obsolete items, and sometimes dropping the value of goods during the time [14].

The inventory of perishable products was first studied by Wagner and Whitin [15], and they looked at perishable items at the end of the warehouse. Ghare and Schrader [16] concluded that the use of perishable items is relatively proportional to the exponential distribution function of time. They proposed the inventory level of perishable items is as follows:

$$\frac{dI(t)}{dt} + \theta I(t) = -f(t), \quad (1)$$

where θ denotes the deterioration rate of perishable items, $I(t)$ is inventory level at time t, and $f(t)$ is the demand rate at time t. They provided an Economic Order Quantity (EOQ) model for perishable products that had a constant deterioration rate. Ghare and Schrader [16] considered the deterioration rate in this model that is a constant or variable ratio of the in-hand inventory, which is shown as a function. They presented a simple economic order with a constant rate of decline. The deterioration rate has created different scenarios in recent researches. It is linear in time [17], two parameters Weibull distribution [18], three parameters Weibull distribution [19], and other functions of time [8].

Eilon and Mallaya [20] considered demand in the inventory model that is dependent on price. Wee [21] investigated the policy of replenishment and pricing in an environment with price-dependent demand, which declined exponentially over time. Moreover, it is assumed that demands during the stockout period is partially back-ordered. Federgruen and Heching [22] examined the combination of inventory and pricing modeling in a stochastic environment. Demands are independent in successive periods and the price is determined based on a function of the state of the system in any period. Chang et al. [23] introduced the inventory and pricing model by considering price-dependent demand and backorder shortage. They presented a model to determine selling price, replenishment number replenishment schedule with partial backlogging. Moreover, they provided a simple algorithm to solve the model. Chen and Sapra [24] developed an integrated inventory control and pricing model with a constant deterioration rate, and deterministic price-dependent demand for a perishable product a fixed lifetime of two periods. The optimal order quantity and demand management (applying price) decisions for the product is dependent on the freshness and old units. Taleizadeh et al. [25] studied integrated pricing and ordering strategies for a deteriorating item. They considered multiple price discounts during the time when demand decreases, and derived a closed-form solution to determine the optimal values of price and order quantity. Adenso-Díaz et al. [26] considered the mixed-priced inventory model with a demand function that was dependent on price and product age. Note that, a demand function is dependent on several factors that can be controlled by the retailer (e.g. Price, inventory level) and uncontrollable (e.g. Time, product's age). Feng et al. [27] developed an inventory control model that the demand of a perishable product is a multivariate function of unit price, freshness condition, and stock level altogether. They illustrated that product freshness is vitally affected on consumers' purchasing decisions and pricing strategy is appropriate competitive tool to increase sales and profits. Yao [28] studied an inventory model where a product's price and ordering decisions are simultaneously determined. The demand depends on the price and excess demand is backlogged. The objective is to find a pricing strategy and an ordering strategy to maximize the expected average profit. Dobson et al. [29] considered the EOQ model for perishable goods with an age-dependent demand rate. They assumed that the demand rate is a linearly decreasing function of the age of the products. They showed that a perishable product acts similarly to a non-perishable product with unit holding cost equal to the ratio of contribution margin to lifetime. Kaya and Polat [3] examined the inventory model based on the EOQ model with price and time-dependent demand. They calculated the optimal profit by considering different discounts at the same time intervals. The demand function that was used in the paper did not simultaneously consider the impact of price and age, which can lead to an incorrect prediction of demand. Chua et al. [30] studied a periodic review inventory and price discount problem for a perishable product with limited shelf life. They presented four models with different customer features. They found that the optimal discount policy is a threshold policy while the optimal order quantity decreases in the inventory of old products with a significant decrease at the threshold. Kaya and Rahimi Ghahroodi [31] modeled a coordinated inventory control and pricing problem for perishable products by considering a general stochastic demand function that is dependent on both the price and remaining shelf life of the products. They proposed a dynamic programming model to determine the optimal ordering decision policy, optimal order quantity and price that should be charged to the products. Agi and Soni [32] proposed a deterministic model for pricing and inventory control decisions of a perishable product subject to physical deterioration and freshness condition degradation. The demand for the product depends on its price, the current inventory level, and the freshness condition. Khan et al. [33] studied the inventory and price discount problem for a perishable product that is not useable after a certain period. They presented two mathematical models by heeding the demand of the product to be dependent on selling price. Moreover, shortages that depend on the customer waiting time are considered and two solution methods are introduced to solve the problem. Wei et al. [34] suggested a mathematical model for the inventory of deteriorating products with quality and quantity loss. The freshness of products is described based on a time-varying freshness function. The demand is affected by the price and freshness of products in different periods. Dye [35] investigated a pricing, advertising, and inventory control problem for perishable products in a multi-period setting. It is assumed that the demand rate varies simultaneously with the selling price, freshness index of the product, and stock of advertising.

Soni [36] considered the inventory control problem for perishable products that the selling price is dependent on the freshness index. A mathematical model is presented for the problem, and it determines the percentage discount on the price based on the freshness to maximize profit. Moreover, a simple algorithm is applied to find an optimal solution. Kaya and Bayer [37] analyzed the inventory control and pricing for perishable products. They developed a mathematical model to determine the time and the quantity of the order, as well as the price of the products considering their freshness over time.

In this paper, in order to fill the gaps in previous studies, the price and the age of the perishable products are simultaneously considered in the demand function. Therefore, a nonlinear mathematical model is proposed for inventory control and price discount problem to find optimal order quantity, discount points, and prices until the expiration date to maximize profit.

According to previous research and the developed models, the discounts were usually announced from the beginning of the period, and it can lead to lower profit. In this paper, in order to fill the gaps in previous studies, the determination of discount start time is considered. To estimate this time, the difference between the expiration date of a product and the mean consumption time for the product is called the discount start time. Moreover, the developed model provides the number of discounts such that the shortage will not be incurred before the expiration date. It is observed that by determining the proper discount start time can be obtained to a sale plan with more profit.

In addition, the proposed nonlinear mathematical model is computationally intractable, and the Particle Swarm Optimization (PSO) algorithm and the Genetic Algorithm (GA) are developed to solve it. Based on the nature of the problem, the size of the chromosome is dependent on the number of discounts. Therefore, at each population and iteration of the algorithms, the chromosomes have different dimensions. The Taguchi approach is applied to find optimum control parameters of PSO and GA. Furthermore, to guarantee the validity of the PSO and GA, the nonlinear model is solved by the Branch-And-Reduce Optimization Navigator (BARON) solver that is a General Algebraic Modeling System (GAMS) solver for the global solution of Non-Linear Program (NLP) and Mixed-Integer Nonlinear Programs (MINLP) [38]. The performance of the algorithms is evaluated based on the real values of parameters for two perishable products (i.e., Cheese and Mayonnaise Sauce) and some random test problems. The computational results demonstrate that the proposed GA outperforms the PSO algorithm. Specifically, the followings are the significant contributions of this paper:

- The price and the age of the perishable products are simultaneously considered in the demand function;
- A nonlinear mathematical model is proposed for inventory control and price discount problem to find optimal order quantity, discount points, and prices until the expiration date to maximize profit;
- Determination of discount start time is considered, and the proposed model provides the number of discounts such that the shortage will not be incurred before the expiration date;
- The model determines order quantity, discount points, and prices until the expiration date;
- The PSO algorithm and the GA are developed to solve the model. Since the size of the chromosome is dependent on the number of discounts, thus, each population is composed of chromosomes with different dimensions.

The structure of the paper is as follows: The problem definition and the proposed mathematical model are presented in Section 2. The PSO and GA algorithms that are used to solve the model are discussed in Section 3. Section 4 gives experimental results and efficiency analysis of PSO and GA algorithms. Finally, in Section 5, a discussion and some suggestions for future works are offered.

2. Problem description and formulation

In this paper, the inventory control and price discount decisions of perishable products (e.g., Cheese, sauce, fruits, and vegetables) are considered. The products are useless after the expiration date. Moreover, the demand for the products is dependent on the remained time until the expiration date. In other words, the demand decreases over time until the expiration date. Therefore, we can motivate customers to purchase products by changing the price.

The purpose of the problem is to find optimal order quantity, the number of discounts that are needed between the discount start time and the expiration date, so that the inventory level at the end of the product's life cycle is equal to zero. This will be done by reducing selling prices through discounts and motivating customers to purchase products. The basis of the proposed model is inspired by the EOQ model, in which the profit is equal to revenue minus total cost, i.e., inventory holding cost, ordering cost, and purchasing cost.

2.1. Assumptions

The proposed mathematical model in this paper is based on the following assumptions:

- Demand is deterministic and dependent on the price and age of the product;
- The deterioration rate is constant;
- The model is a single-period model, and the expiration date of the product is the end of the period;
- The discount rate is constant;
- Backorders are not allowed;
- The inventory level reaches zero at the end of the period.

2.2. *Notation*

The following notations are used to formulate the proposed model.

Index

i	The index of discount interval

Parameters

α	Price elasticity of demand
β	The influence of ageing on demand
L	Expiration date
DC	Discount rate
h	Holding cost per unit time
P_0	Initial selling price
$P_i = (1 - DC) \times P_{i-1}$	The selling price at the ith discount interval
D_0	Initial demand
θ	Deterioration rate
A	Ordering cost
C	The purchase price of a product

Decision variables

N	Number of discounts
Q	Optimal order quantity
t_i	The start time of ith discount interval
X	Discount start time

2.3. *The mathematical model*

In this subsection, an MINLP model is presented for the problem. An EOQ-based approach is applied. Since the customers for perishable products will not be willing to wait to get these products, therefore, backorders are not allowed.

Figure 1 represents the inventory level in a period. In this figure, $t_1, t_2, t_3, \dots, t_N$ are the discount start times and $I(t_1), I(t_2), I(t_3), \dots, I(t_N)$ are the inventory levels at the times, respectively. By considering the deterioration rate and age and price-dependent demand rate $\left(D(p,t)\right)$, that the demand rate is as a function of both price (denoted p) and age (denoted t), with $(t \le L)$, the inventory level is described at time $t\left(I(t)\right)$ by the derivative equation.

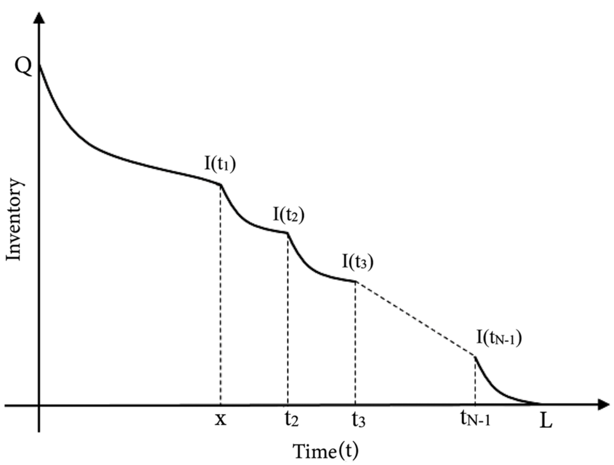


Figure 1. Illustrations of the inventory process.

$$\frac{\partial I(t)}{\partial t} = -D(p,t) - \theta I(t). \tag{2}$$

Let $I_i(t)$ denotes the inventory level for any time $t \in [t_{i-1}, t_i)$ in ith discount interval, also let $I(t_i)$ denotes the inventory level at the end of the period $t \in [t_{i-1}, t_i)$. Eq. (2) is a first-order differential equation with variable coefficients. By multiplying the factor $e^{\theta t}$ in Eq. (2), Eq. (3) will be obtained:

$$e^{\theta t}\frac{\partial I_i(t)}{\partial t} + \theta e^{\theta t} I_i(t) = -e^{\theta t} D(p_i, t), \tag{3}$$

$$\frac{\partial\left(e^{\theta t} I_i(t)\right)}{\partial t} = -e^{\theta t} D(p_i, t). \tag{4}$$

By integrating from both sides of Eq. (4), the following equation is obtained:

$$e^{\theta t} I_i(t) = \int_t^{t_i} e^{\theta s} D(p_i, s) ds + c, \tag{5}$$

where c is an arbitrary constant. The general solution of Eq. (5) is as follows:

$$I_i(t) = e^{-\theta t}\int_t^{t_i} e^{\theta s} D(p_i, s) ds + c e^{-\theta t}. \tag{6}$$

Then, for any time $t \in [t_{i-1}, t_i)$, by applying the differential Eq. (6), the following instantaneous inventory equation for $I_i(t)$ is obtained:

$$I_i(t) = \int_t^{t_i} D(p,s) e^{\theta(s-t)} ds + I(t_i) e^{\theta(t_i - t)}. \tag{7}$$

In this paper, the proposed nonlinear demand function by Adenso-Díaz et al. [26] is applied, which describes a demand function based on the price and age of the product. The demand function is as bellows:

$$D_i(p,t) = D_{i-1}\left(\frac{p_i}{p_{i-1}}\right)^{-\alpha}\left(1 - \left(\frac{t}{L}\right)^{\beta}\right). \tag{8}$$

By using this demand function, the inventory equation is obtained as follows:

$$I_i(t) = \left(D_{i-1}\left(\frac{p_i}{p_{i-1}}\right)^{-\alpha}\right)\left[\frac{1}{\theta}\left(e^{\theta(t_i - t)} - 1\right)\right.$$
$$\left. - \frac{1}{L^{\beta}}\left[\frac{\Gamma(\beta + 1. \theta t_i) e^{-\theta t}}{-\theta^{\beta+1}} - \frac{\Gamma(\beta + 1, \theta t) e^{-\theta t}}{-\theta^{\beta+1}}\right]\right]$$
$$+ I(t_i) e^{\theta(t_i - t)}. \tag{9}$$

Given that at the end of the period, the inventory reaches zero, $I(t_N) = 0$, and the inventory level at the beginning of the period is equal to $Q, I(t_0) = Q$ then the value $I(t_i)$ for the ith discount interval will be calculated as follows:

$$I(0) = (D_{(i-1)}(\frac{p_i}{p_{i-1}})^{-\alpha})[\frac{1}{\theta} e^{\theta(t_1)} - 1)$$
$$-\frac{1}{L^{\beta}}[\frac{\Gamma(\beta + 1, \theta t_1)}{-\theta^{\beta+1}} - \frac{\Gamma(\beta + 1,0)}{-\theta^{\beta+1}}] + I(t_1) e^{\theta(t_1)}, \tag{10}$$

And the Eq. (11) as shown in Box I.

$$I(t_i)=\sum_{j=i+1}^{N}\left[\begin{array}{l}\left(D_{j-1}\left(\frac{p_j}{p_{j-1}}\right)^{-\alpha}\right)\\ \times\left[\frac{1}{\theta}\left(e^{\theta(t_j-t_i)}-1\right)-\frac{1}{L^{\beta}}\left[\frac{\Gamma(\beta+1,\theta t_j)e^{-\theta t_{i-1}}}{-\theta^{\beta+1}}-\frac{\Gamma(\beta+1,\theta t_{j-1})e^{-\theta(t_j-t_i)}}{-\theta^{\beta+1}}\times e^{\theta(t_{j-1}-t_i)}\right]\right]\end{array}\right]. \quad (11)$$

Box I

$$\pi_H=h\times\sum_{i=1}^{N}\int_{t_{i-1}}^{t_i}I_i(t)dt=$$
$$h\sum_{i=1}^{N}\left(\begin{array}{l}D_{i-1}(\frac{p_i}{p_{i-1}})^{-\alpha}\left(\frac{e^{-\theta t_{i-1}}\left((\theta t_{i-1}-\theta t_i-1)e^{t_{i-1}}+e^{\theta t_i}\right)}{\theta^2}\right.\\ \left.-\frac{1}{L^{\beta}}\left(\frac{\Gamma(\beta+1,\theta t_i)}{-\theta^{\beta+1}}\times\left(\frac{e^{-\theta t_{i-1}}}{\theta}-\frac{e^{\theta t_i}}{\theta}\right)+\frac{1}{\theta^{\beta+1}}\int_{t_{i-1}}^{t_i}\Gamma(\beta+1,\theta t)\times e^{-\theta t}\right)\right)\end{array}+\frac{I(t_i)}{\theta}\left(1-e^{\theta(t_i-t_{i-1})}\right)\right). \quad (14)$$

Box II

The sales amount of the product until the beginning of the discount (S_1), as well as the sales amount in the ith discount interval (S_i), are calculated as follows:

$$S_1=\int_0^{t_1}D_0\left(\frac{p_1}{p_0}\right)^{-\alpha}\left(1-\left(\frac{t}{L}\right)^{\beta}\right)dt$$
$$=D_0\left(\frac{p_1}{p_0}\right)^{-\alpha}\left[t_1-\frac{1}{L^{\beta}}-\frac{t_1^{\beta+1}}{\beta+1}\right], \quad (12)$$

and:

$$S_i=\int_{t_{i-1}}^{t_i}\left(D_{i-1}\left(\frac{p_i}{p_{i-1}}\right)^{-\alpha}\left(1-\left(\frac{t}{L}\right)^{\beta}\right)\right)$$
$$=D_{i-1}\left(\frac{p_i}{p_{i-1}}\right)^{-\alpha}\left[(t_i-t_{i-1})-\frac{1}{L^{\beta}}\left(\frac{t_i^{\beta+1}}{\beta+1}-\frac{t_{i-1}^{\beta+1}}{\beta+1}\right)\right]. \quad (13)$$

The total holding cost, π_H, is calculated by taking the area under the inventory curve in Figure 1 and then multiply it with the unit holding cost (h). Therefore, by using Eq. (9), we can write as shown in Box II.

Eventually, the profit per unit time function is acquired by considering the difference between total revenue and total costs in an inventory cycle and dividing that value by the duration of an inventory cycle. As a result, the mathematical model for the considered problem is as shown in Box III.

The proposed model is an MINLP model, which includes the objective function that derives from the difference in revenue and total costs. On the other hand, constraint set Eq. (16) ensures the discount times are increasing order, which is not allowed to issue double discounts at a time. Constraint set Eq. (17) provides the inventory level of the ith discount interval is less than the previous interval. Constraint set Eq. (18) guarantees that the inventory level of the ith discount interval does not exceed the optimal order quantity. Constraints Eqs. (19) and (20) show the start time of the period and the inventory level in the start time, respectively. Constraints Eqs. (21) and (22) define the value range of variables N and Q.

3. Solution methodology

The proposed mathematical model is computationally intractable. There are the gamma function and integral expression in the model. Therefore, in this paper, GA and PSO algorithms are used to obtain a good solution. Moreover, to guarantee the validity of the PSO and GA, the relaxed MINLP model is solved by the BARON solver in GAMS software by considering a constant value for N. The PSO and GA are described as follows.

3.1. PSO algorithm

PSO algorithm is one of the optimization algorithms based on random population generation. This algorithm is based on modeling and simulation of birds' flying behavior or collective movement of fish. This algorithm was first defined for continuous variables but was also developed for issues with discrete variables. PSO algorithm in discrete mode is introduced with Binary Particle Swarm Optimization (BPSO). In the most common implementations of PSO, particles move through the search space. The particles use a combination of an attraction to the best solution that they individually have found, and an attraction to the best solution that any particle in their neighborhood has found [39].

Each group member is defined by a velocity vector and a position vector in the search space. In each iteration, a new particle position is updated according to the current velocity vector. The best position is found by the particle and by the best particle in the group. This updating and searching process in the solution space is performed as below.

Consider a D-dimensional search space, an individual particle l is composed of three vectors: the position vector $Y_l=(y_{l1},y_{l2},\dots,y_{lD})$, the best position of the lth particle that it has found $P_l=(p_{l1},p_{l2},\dots,x_{lD})$, and its velocity vector $V_l=(v_{l1},v_{l2},v_{l3},\dots,v_{lD})$. These particles move throughout the search space by a set of update equations. The Eqs. (23) and (24) are used to update the velocity and the position of particle l, respectively.

$$\max Z=\left(\frac{1}{L}\right)\left\{\left[\left(p_0\left(\left(D_0\left(\frac{p_1}{p_0}\right)^{-\alpha}\right)\cdot\left(t_1-\frac{1}{L^\beta}\cdot\frac{t_1^{\beta+1}}{\beta+1}\right)\right)\right)\right.\right.$$

$$+\left(\sum_{i=1}^{N}p_i\cdot\left(D_{i-1}\left(\frac{p_i}{p_{i-1}}\right)^{-\alpha}\cdot\left((t_i-t_{i-1})-\frac{1}{L^\beta}\left(\frac{t_i^{\beta+1}}{\beta+1}-\frac{t_{i-1}^{\beta+1}}{\beta+1}\right)\right)\right)\right)\Bigg]$$

$$-\ h\sum_{i=1}^{N}\left\lfloor\ \left\lfloor(D_{i-1}(\frac{p_i}{p_{i-1}})^{-\alpha})\ \left\lfloor\left(\frac{e^{-\theta t_i\left((\theta t_{i-1}-\theta t_i-1)e^{\theta t_{i-1}+e^{\theta t_i}}\right)}}{\theta}\right)-\frac{1}{L^\beta}\left(\frac{\Gamma(\beta+1,\ \theta t_i)}{-\theta^{\beta+1}}\cdot\left(\frac{e^{-\theta t_i}}{\theta}-\frac{e^{-\theta t_{i-1}}}{\theta}\right)\right.\right.\right.\right.$$

$$\left.\left.\left.+\left(\frac{1}{\theta^{\beta+1}}\int_{t_{i-1}}^{t_i}\Gamma(\beta+1,\theta t_i)\cdot e^{-\theta t}dt\right)\right)\right]+\frac{I(t_i)}{\theta}\left(1-e^{\theta(t_i-t_{i-1})}\right)\Bigg]\Bigg]-[A+CQ]\Big\},\tag{15}$$

s.t.:

$$t_i-t_{i-1}>0\,,\quad\forall i\ \in\{1,2,\dots,N\},\tag{16}$$

$$I(t_i)\le I(t_{i-1})\,,\quad\forall i\ \in\{1,2,\dots,N\},\tag{17}$$

$$0\le I(t_i)\le Q,\qquad\forall i\ \in\{1,2,\dots,N\},\tag{18}$$

$$t_0=0,\tag{19}$$

$$I(t_0)=Q,\tag{20}$$

$$N\ \in\{1,2,3,\dots\},\tag{21}$$

$$Q\ge 0.\tag{22}$$

Box III

Start a discount	Number of discounts				Expiration date
(X)	t2	t3	t4	t5	(L)

$\longleftarrow\quad x<t_i<L\quad\longrightarrow$

Figure 2. Solution representation in PSO algorithm with $N=6$.

$$v_{ld}=w\cdot v_{ld}+c_1r_1\cdot(p_{ld}-y_{ld})+c_2r_2\cdot(p_{gd}-y_{ld}),\tag{23}$$

$$y_{ld}=y_{ld}+v_{ld},\tag{24}$$

where, the symbols are as follows:

w	The inertial weighting factor, which shows the effect of the velocity vector of the previous iteration on the velocity vector in the current iteration.
c_1,c_2	Learning coefficients.
r_1,r_2	Uniformly random numbers between 0 and 1.
$\vec{P}_g$	The best position funded by any neighbor of the particle [39].

3.2. *PSO for the proposed model*

As stated above, N is the number of discounts. So, a feasible solution of the PSO algorithm (Y) is an N-dimensional vector. The first dimension indicates the beginning of the discount, the last dimension introduces the expiration date of the product, and the $(N-2)$ mid-dimension denote $(N-2)$ discount intervals between first and last discount intervals. The start time of the discount intervals is randomly generated and sorted in ascending order. Figure 2 shows a feasible solution with six discount intervals as that $X<t_2<\dots<t_5<L$. According to Figure 2, the minimum number of discounts is equal to 2, which is composed of the first and last (expiration date) discount intervals. Moreover, the maximum number of discounts is obtained from the difference

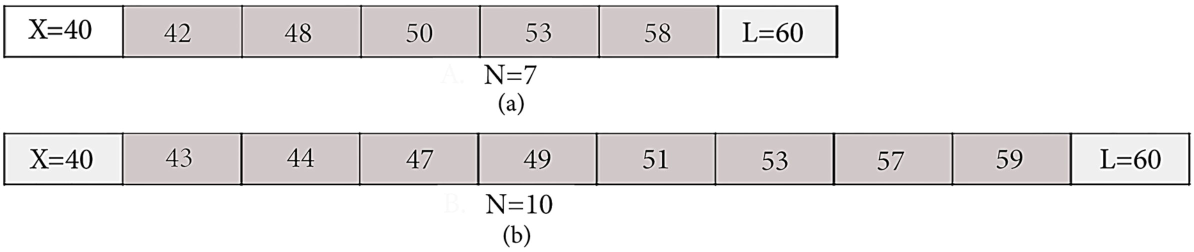


Figure 3. Two chromosomes with dimensions (a) 7 and (b) 10.

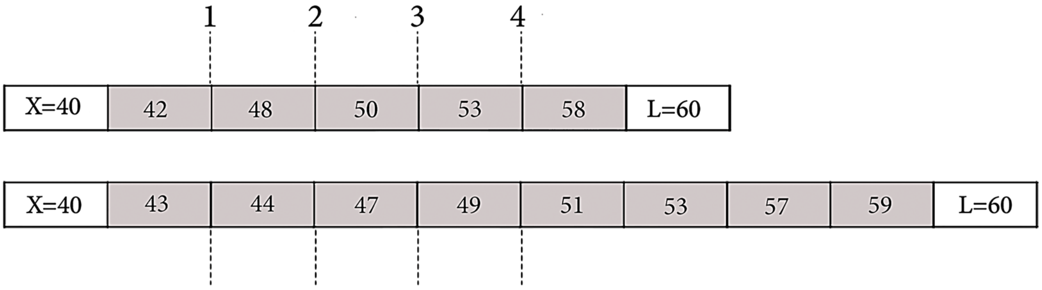


Figure 4. All modes to perform the crossover operator.

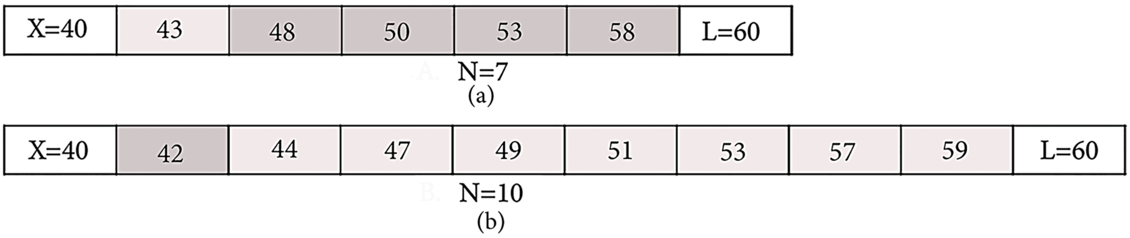


Figure 5. The obtained two offspring from crossover operator.

of the expiration date from the beginning of the discount plus one $(L - X + 1)$.

After calculating the upper and lower bounds of N, the PSO algorithm is implemented based on the possible values for N in the range 2 to $L - X + 1$. The position of each solution is calculated based on random t_i and N. The new position that is created by Eqs. (23) and (24) should be in ascending order, and the repetitive values of t_i are not allowed. The value of the objective function in each iteration is calculated based on the values of t_i and N. The original PSO algorithm is presented to solve the problems in the continuous solution space. In this paper, Eq. (25) is applied to convert the continuous solution to a discrete one.

$$t_i = \min([r.(L - X + 1) + (X + 1)],\ L - 1), \qquad (25)$$

where, r is a random number between 0 and 1.

3.3. Genetic Algorithm (GA)

GA is an effective optimization process and a strong search that is commonly used in complex search spaces. GA imitates the biological principles for solving optimization problems. It contains a set of individual solutions or chromosomes that is called a population. In this algorithm, some natural-inspired operators create a new population from the previous population. According to evolutionary theory, only the superior individuals in the population can generate a better generation; therefore, the superior genetic information is transferred to the new generation [40].

3.4. GA for the proposed model

In this section, GA is developed to solve the considered problem.

3.4.1. Solution representation

By considering Figure 2 , the upper and lower bounds of the number of discounts (N) are calculated similarly to the PSO algorithm. Solution representation in GA is similar to the PSO algorithm, and each chromosome is an N-dimensional vector. Since the size of the chromosome is dependent on N, thus, each population is composed of chromosomes with different dimensions. Figure 3 depicts two chromosomes with different dimensions.

3.4.2. Crossover

The single point operator is used to perform the crossover operator to generate two offspring from two parents. Since the elements of the chromosomes are time points, thus, the ascending order is necessary. Therefore, the ascending order must be observed in the crossover operator.
Consider two chromosomes in Figure 3; there are four modes to perform the crossover operator on the two parents. But, only the cross-action 1 can led to a feasible solution (see Figure 4).

By using a single-point crossover and choosing the cross-action 1, the obtained two offspring are shown in Figure 5.

3.4.3. Mutation

The mutation operator is applied to rearrange the structure of a chromosome. It helps to increase the searching power in the solution space. In this paper, random changes of N are

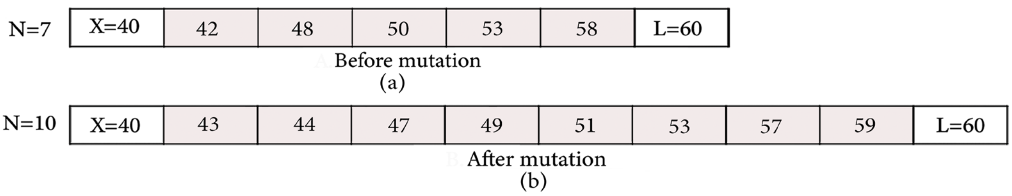
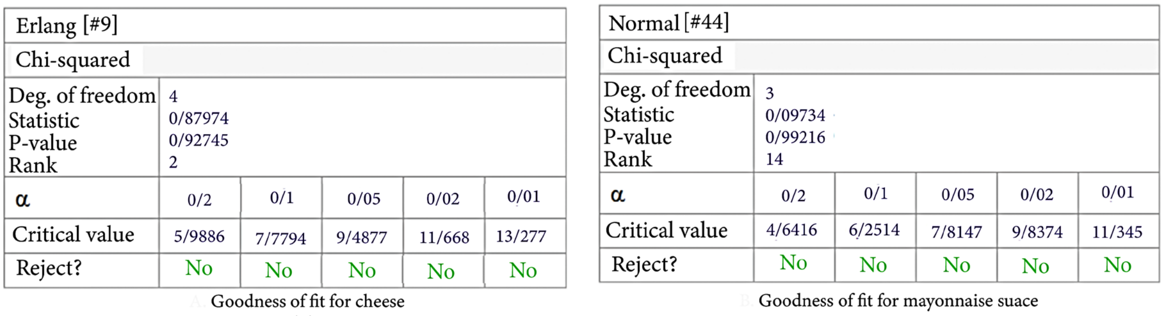


Figure 6. The mutation operator (a) before mutation and (b) after mutation



Erlang [#9]					
Chi-squared					
Deg. of freedom	4				
Statistic	0/87974				
P-value	0/92745				
Rank	2				
α	0/2	0/1	0/05	0/02	0/01
Critical value	5/9886	7/7794	9/4877	11/668	13/277
Reject?	No	No	No	No	No

Goodness of fit for cheese

(a)

Normal [#44]					
Chi-squared					
Deg. of freedom	3				
Statistic	0/09734				
P-value	0/99216				
Rank	14				
α	0/2	0/1	0/05	0/02	0/01
Critical value	4/6416	6/2514	7/8147	9/8374	11/345
Reject?	No	No	No	No	No

Goodness of fit for mayonnaise suace

(b)

Figure 7. Goodness-of-fit test for the consumption period, (a) Cheese and (b) Mayonnaise Sauce.

considered for mutation operator. For example, if the upper bound of the number of times the discount is equal to 20, the appropriate value for N should be in the interval [2,20]. The mutation attempts to obtain the number of random discounts in each iteration by randomly selecting a number from this interval. Figure 6 shows the mutation operator for a chromosome.

3.4.4. Create successive generations

The roulette wheel is used to generate a new population. The fitness of a chromosome determines the size of its segment on the roulette wheel. The roulette wheel is then 'spun' repeatedly to produce a new population of the same size as the previous population.

4. Computational results

In this section, the obtained results from solving the model by GA and PSO algorithms are analyzed. In Section 4.1, first, the discount start time (X) for the two perishable products is estimated. In Section 4.2, the performance of the proposed algorithms to solve the considered model is investigated.

4.1. Estimation of the discount start time

The discount start time (X) is estimated according to the customer's consumption pattern for different products. To estimate this time, the difference between the expiration date and the average consumption period can be called the discount start time. To calculate the average, the consumption period of each product is obtained from the buyers in the form of an oral questionnaire. In this paper, the consumption period of two perishable products (Cheese and Mayonnaise Sauce) is gathered from 50 buyers. Then Easy Fit software was used to fit a probability distribution on the data. Figure 7 shows the output of the software by performing the goodness-of-fit test.

As can be seen, the Erlang distribution with parameters $m = 21$ and $\beta = 0.646$ and Normal distribution with parameters $\mu = 44.5$ and $\sigma = 8.22$ are correctly fitted on gathered data about the consumption period of Cheese and Mayonnaise Sauce, respectively. Therefore, the average consumption period of the Cheese and Mayonnaise Sauce is estimated equal to be 14 ($= 21 \times 0.646$) and 44 days, respectively. The value of X will be obtained from the value of L minus the average consumption period of the perishable product, which is 56 for Mayonnaise Sauce and 45 for Cheese.

Table 1. The input parameters of the model for Cheese and Mayonnaise Sauce.

Parameter	Cheese	Mayonnaise sacuce
α	0.8	0.6
β	2	2
DC	0.1	0.1
L	60	100
h	50	50
X	46	56
p_0	3000	6000
D_0	100	100
θ	0.05	0.007
A	50000	50000
C	2900	3500

4.2. Results analysis

In this section, we analyze the model and evaluate the performance of the proposed GA and PSO algorithms. In Subsection 4.2.1, the performance of the algorithms is evaluated based on the real values of parameters for two perishable products (i.e., Cheese and Mayonnaise Sauce). In Subsection 4.2.2, the proposed algorithms are compared together based on the random test problems. The sensitivity analysis is performed on the value of X in Subsection 4.2.3. Moreover, in this subsection, we compare two strategies for discount start time, (a) from the beginning of the period, (b) from point X. Finally, in Subsection 4.2.4, to guarantee the validity of the proposed algorithms, the results of the algorithms are compared to the results of the relaxed model that is solved by BARON solver in GAMS software.

4.2.1. The results of solving the model for two considered products

The real values of the parameters for the two perishable products (Cheese and Mayonnaise Sauce) are shown in Table 1.

Table 2. The parameter values of GA and PSO algorithms.

Parameters	**PSO**			**Parameters**	**GA**		
	Level				**Level**		
	1	**2**	**3**		**1**	**2**	**3**
$MaxIt$	200	300	**700**	$MaxIt$	200	**300**	700
$Npop$	20	40	**60**	$Npop$	20	**40**	60
w	1	2	**3**	P_c	0.6	**0.7**	0.8
c_1	**2**	4	6	P_m	0.4	**0.3**	0.2
c_2	3	**5**	7	$Tour\ Size$	3	4	**5**
				SP	6	**7**	8

$MaxIt$: Maximum number of iterations, $Npop$: Population size, w: Coefficient of inertia, c_1 and c_2: Learning factors, P_c: Crossover probability, P_m: Mutation probability, Tour Size: Tournament Size, SP: Selection pressure

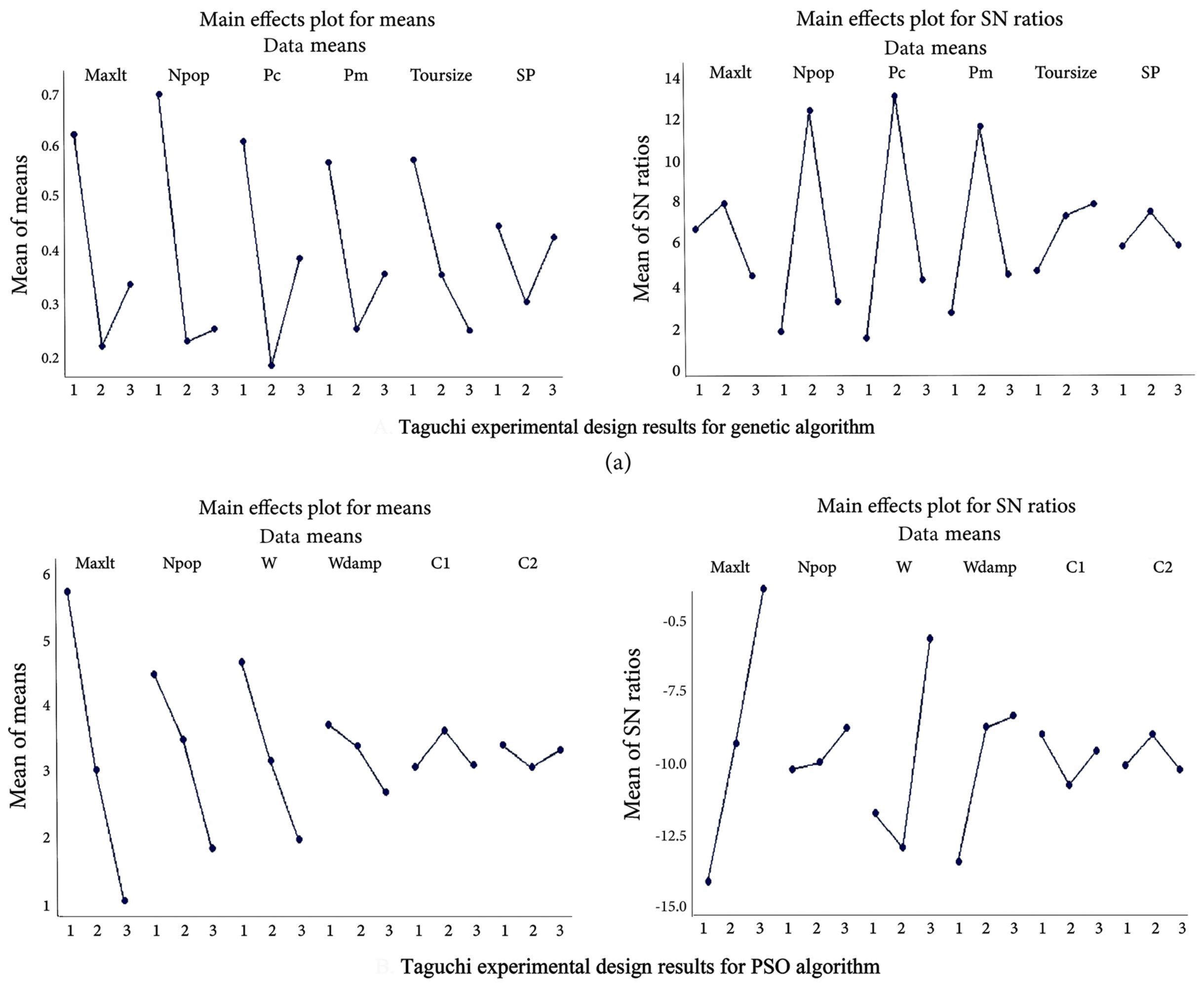


Figure 8. The results of Taguchi experimental design (a) GA algorithm and (b) PSO algorithm

To tune the parameters of GA and PSO algorithms, the Taguchi experimental design approach is performed in the MINITAB software. The parameter values of GA and PSO algorithms are shown in Table 2. Moreover, Figure 8 depicts the results of the Taguchi experimental design approach.

The final values of the parameters of GA and PSO algorithms based on the Taguchi experimental design approach are presented in Table 3.

Table 3. The final values of the parameters of the algorithms.

GA					
SP	$Tour\ Size$	P_m	P_c	$Npop$	$MaxIt$
7	5	0.3	0.7	40	300
PSO					
C_2	C_1	w	$Npop$	$MaxIt$	
5	2	3	60	700	

The algorithms are implemented in MATLAB software and tested on a computer with a 3.1 GHz CPU and 8 GB of RAM. The results of the implementation of the algorithms are shown in Table 4. The obtained results for Cheese with a 60-day expiration date and the discounts start time from the 46th day show that both algorithms are presented similar profit and the number of discounts.But GA outperforms PSO by considering the run time and Number Function Evaluation (NFE). For Mayonnaise Sauce, by considering

Table 4. The results of the implementation of GA and PSO algorithms.

	Cheese				Mayonnaise sauce			
	Profit	**N**	**Run time (s)**	**NFE**	**Profit**	**N**	**Run time (s)**	**NFE**
GA	340176	7	5.35	12040	35141	14	11.97	12040
PSO	340176	7	17.91	42060	23122	16	43.57	42060

Table 5. The range of parameter values.

Parameters	**Range**	**Parameters**	**Range**
α	[0.5, 0.9]	P_0	[100, 3000]
β	[0.01, 2]	D_0	[10, 200]
DCR	[0.01, 0.5]	θ	[0.005, 0.9]
h	[50, 200]	A	[100, 50000]
L	[50, 200]	C	[90, 2900]
X	[10, 290]		

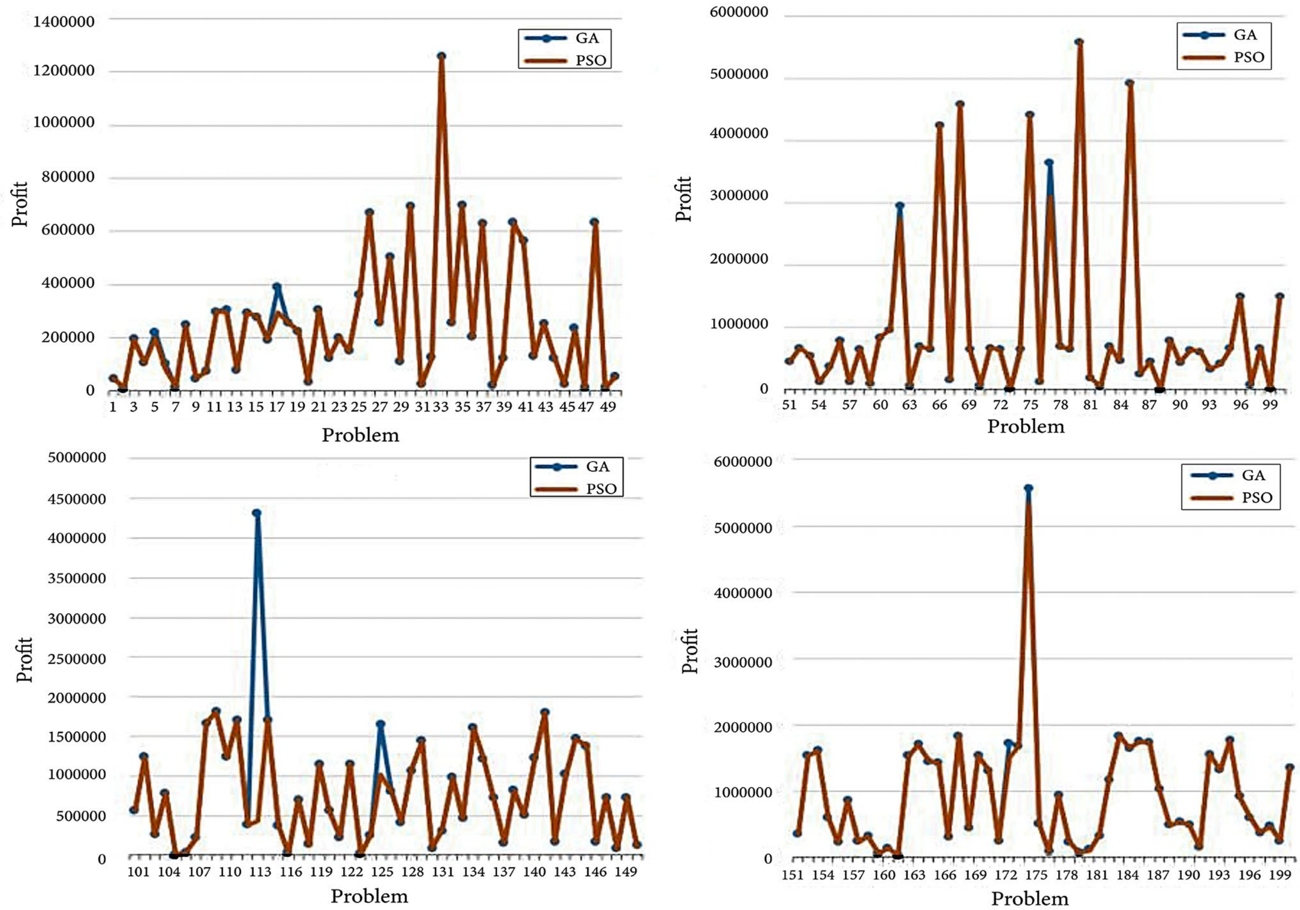


Figure 9: The profit for each problem.

expiration date and discount start time equal to 100-day and 57th day, respectively, GA outperforms PSO.

4.2.2. Comparison of the algorithms for random test problems

In this subsection, the performance of GA and PSO algorithms is investigated based on some random test problems. The parameter values of the model were uniformly generated in the ranges that are given in Table 5.

200 random test problems are generated to assess GA and PSO algorithms. Each of them is executed five times on a computer with a 3.1 GHz CPU and 8 GB of RAM. The best solution of each algorithm for any test problem is reported in Table 6.

Table 6 depicts the profit, the number of discounts and the run time of each problem. Moreover, the Relative Percentage Deviation (RPD) is used to compare the algorithms on the problems. The value of RPD is calculated as follows:

$$RPD = \frac{|\,Method\ Sol - Best\ Sol\,|}{Best\ Sol} * 100. \tag{26}$$

Based on the values of RPD, GA and PSO algorithms presented similar profit in the 178 random test problems. Moreover, GA presented a better solution than PSO in the 22 random test problems. The average computational results of GA and PSO algorithms for all 200 random test problems can be seen in Table 7.

The obtained results in Table 6 and Table 7 show that GA out performs PSO in profit, run time and RPD. Figures 9 and 10 depict the profit and the number of discounts, that the algorithms are presented for each test problem, respectively.

Table 6. The best solution of GA and PSO algorithms for 200 random test problems.

	GA			PSO			
Problem	Profit	*N*	Run time (s)	Profit	*N*	Run time (s)	RPD (%)
1	47592.45	7	7.65	47592.45	7	28.26	0
2	11459.36	6	5.79	11459.59	7	25.77	0
3	194639.81	7	5.59	194639.81	8	20.44	0
4	105652.69	8	7.14	105652.69	8	26.6	0
5	220388.2	4	4.01	196704.32	5	17.96	10.74
6	102742.59	7	4.89	82882.99	8	23.34	19.32
7	14466.48	6	6.41	14466.48	7	25.39	0
8	247524.54	5	4.97	247524.54	5	20.58	0
9	46815.59	7	5.56	46815.59	7	25.56	0
10	73315.35	9	6.57	73315.35	9	29.36	0
11	297499.02	6	5.9	297499.02	6	5.9	0
12	307140.4	11	13.58	296213.18	9	12.53	3.5
13	78702.26	9	11.78	78702.26	8	15.31	0
14	295276.14	9	17.75	295276.14	9	20.65	0
15	277176.86	14	27.57	277176.86	12	29.1	0
16	192100.16	9	19.64	192100.16	9	20.46	0
17	392181.62	8	10.18	295053.54	8	14.33	24.7
18	258003.35	7	18.13	258003.35	7	21.06	0
19	225523.45	10	18.08	225523.45	9	19.47	0
20	33318.72	6	13.35	33318.72	6	15.9	0
21	305979.29	6	10.68	305979.29	6	12.21	0
22	123713.73	8	20.93	123713.73	8	23.25	0
23	201701.7	10	13.07	201701.7	10	16.22	0
24	149563.53	8	13.95	149563.53	7	14.66	0
25	362772.52	10	13.85	362772.52	7	12.03	0
26	669532.36	9	11.23	669532.36	9	13.23	0
27	256116.69	11	13.02	256116.69	11	14.02	0
28	503694.25	12	14.45	503694.25	12	16.45	0
29	112695.32	6	11.05	112695.32	6	13.05	0
30	695362.02	12	15.01	695362.02	12	17.01	0
31	25362.55	6	6.35	20569.31	7	8.35	18.9
32	125630.12	10	13.01	125630.12	10	14.65	0
33	1256890.22	10	18.02	1256890.22	10	20.55	0
34	256388.08	11	14.02	256388.08	11	17.02	0
35	698603.65	12	15.02	698603.65	12	17.02	0
36	203695.02	11	13.10	203695.02	11	15.10	0
37	630125.95	6	7.01	630125.95	6	9.01	0
38	23623.78	9	10.25	23623.78	9	12.25	0
39	122236.02	9	11.03	122236.02	9	14.03	0
40	632502.36	10	12.95	632502.36	11	15.95	0
41	563152.12	8	13.05	563152.02	8	14.05	0
42	129369.05	7	9.25	129369.05	7	10.39	0
43	253620.05	7	8.08	253620.05	7	10.95	0
44	123605.08	11	13.85	123605.08	11	15.78	0
45	25364.06	11	15.05	25364.06	11	16.59	0
46	236156.09	7	9.85	226667.61	8	10.65	4.01
47	12552.03	8	10.25	10762.74	9	10.65	14.2
48	632953.15	9	12.36	632953.15	9	12.35	0
49	12963.06	10	10.65	12963.06	10	11.03	0
50	55369.09	10	11.25	55369.09	10	12.66	0
51	446392.06	14	13.95	446392.06	14	14.06	0
52	673216.98	10	11.36	673216.98	10	12.14	0
53	553216.02	19	19.65	553216.02	19	20.65	0
54	126593.78	18	16.28	126593.78	18	16.20	0
55	369563.41	14	13.25	369563.41	14	14.69	0
56	785962.02	9	8.05	785962.02	9	10.25	0

Table 6. The best solution of GA and PSO algorithms for 200 random test problems (continued).

GA				PSO			
Problem	Profit	*N*	Run time (s)	Profit	*N*	Run time (s)	RPD (%)
57	123956.63	10	12.09	123956.63	10	13.10	0
58	659623.45	11	13.36	659623.45	11	13.05	0
59	99632.77	8	8.95	99632.77	8	9.05	0
60	845632.65	12	10.09	845632.65	12	11.26	0
61	965369.35	18	17.09	965369.35	18	17.25	0
62	2963516.44	16	62	2744536.09	15	15.09	7.4
63	69326.15	11	11.25	69326.15	11	12.33	0
64	698453.49	6	8.09	698453.49	6	10.10	0
65	653946.99	7	7.96	653946.99	7	7.56	0
66	4256325.63	9	10.86	4256325.63	9	11.08	0
67	163259.45	10	12.11	163259.45	10	11.33	0
68	4596326.11	11	13.06	4596326.11	11	14.37	0
69	656362.55	12	10.95	656362.55	12	10.25	0
70	69123.37	12	13.65	69123.37	12	14.78	0
71	669432.89	9	11.39	669432.89	9	12.96	0
72	651395.91	6	9.32	651395.91	6	10.06	0
73	15632.04	6	10.25	15632.04	6	11.02	0
74	653956.14	12	13.69	603971.55	13	15.52	7.6
75	4416596.35	18	16.84	4416596.35	18	17.55	0
76	126630.83	5	6.45	126630.83	5	7.69	0
77	3659302.14	9	9.56	3100617.96	10	9.88	15.27
78	695123.77	10	11.63	695123.77	10	12.59	0
79	659360.39	11	12.95	659360.39	11	12.03	0
80	5596120.09	13	12.09	5596120.09	13	13.66	0
81	199623.79	8	9.65	199623.79	8	10.02	0
82	65643.22	8	8.99	65643.22	8	8.06	0
83	694361.26	7	8.05	694361.26	7	9.67	0
84	462251.13	7	7.36	462251.13	7	8.65	0
85	4936252.33	9	10.98	4936252.33	9	11.81	0
86	255503.61	10	12.23	255503.61	10	13.62	0
87	452362.12	11	13.05	452362.12	11	13.25	0
88	12653.63	10	12.11	12653.63	10	12.84	0
89	796608.01	9	9.95	796608.01	9	9.65	0
90	443630.81	7	8.65	443630.81	7	8.99	0
91	642359.14	8	7.05	642359.14	8	7.06	0
92	614953.22	8	9.26	614953.22	8	10.99	0
93	335129.26	9	10.25	335129.26	9	11.28	0
94	416239.81	11	13.05	416239.81	11	14.66	0
95	664923.12	10	11.36	664923.12	10	12.03	0
96	1496235.11	12	12.96	1496235.11	12	13.51	0
97	87563.08	12	11.85	87563.08	12	12.31	0
98	669423.14	6	10.62	669423.14	6	11.88	0
99	16352.11	8	11.95	16352.11	8	12.67	0
100	1496325.99	14	16.06	1496325.99	14	17.68	0
101	569312.66	9	10.55	569312.66	9	11.02	0
102	1236593.22	12	18.81	1236593.22	12	20.28	0
103	259411.01	10	11.36	259411.01	10	11.31	0
104	786123.88	9	12.12	786123.88	9	13.44	0
105	12653.63	6	9.36	12653.63	6	10.65	0
106	36805.23	10	10.22	30221.12	10	12.05	17.9
107	222684.33	16	21.25	222684.33	16	21.36	0
108	1670330.02	18	19.62	1670330.02	18	20.11	0
109	1806860.68	14	20.11	1806860.68	14	21.88	0
110	1236008.74	8	10.02	1236008.74	8	10.36	0
111	1704294.11	12	13.55	1704294.11	12	13.96	0
112	380214.01	17	18.89	380214.01	17	19.02	0
113	4310270.1	13	17.96	431027.01	13	18.61	0

Table 6. The best solution of GA and PSO algorithms for 200 random test problems (continued).

	GA			PSO			
Problem	**Profit**	***N***	**Run time (s)**	**Profit**	***N***	**Run time (s)**	**RPD (%)**
114	1711101.39	14	12.32	1711101.39	14	13.88	0
115	367238.13	16	17.14	367238.13	16	17.55	0
116	32356.91	14	19.58	32356.91	14	19.27	0
117	699445.54	7	12.09	699445.54	7	14.13	0
118	136592.11	11	12.88	122301.01	11	14.02	10.4
119	1141564.19	17	19.99	1141564.19	17	21.05	0
120	563924.42	9	9.05	563924.42	9	10.39	0
121	226889.23	7	8.66	226889.23	7	8.97	0
122	1151918.08	14	22.25	1151918.08	14	26.14	0
123	25332.99	9	10.33	19002.01	10	12.02	25
124	256001.25	13	20.11	256001.25	13	24.77	0
125	1653259.08	6	8.05	1021289.81	8	9.88	38.22
126	802850.61	11	11.35	802850.61	11	12.11	0
127	416312.00	15	16.18	416312.00	15	18.68	0
128	1069890.22	7	9.8	1069890.22	7	10.96	0
129	1444601.11	14	17.09	1444601.11	14	17.01	0
130	90599.35	18	19.72	90599.35	18	20.62	0
131	305978.62	11	10.91	305978.62	11	13.89	0
132	981993.41	12	10.44	981993.41	12	11.22	0
133	468081.69	18	18.29	468081.69	18	18.38	0
134	1608620.42	7	8.18	1608620.42	7	9.11	0
135	1218899.93	14	16.16	1218899.93	14	16.49	0
136	726668.15	9	10.95	726668.15	9	11.66	0
137	160950.31	11	12.64	160950.31	11	12.39	0
138	815707.18	14	16.89	815707.18	14	18.71	0
139	507764.90	13	12.02	507764.90	13	13.29	0
140	1229613.16	15	18.80	1229613.16	15	18.92	0
141	1794466.32	10	11.17	1794466.32	10	11.18	0
142	171373.09	17	19.89	171373.09	17	19.29	0
143	1023417.05	9	10.72	986025.12	11	11.65	3.6
144	1468689.87	13	13.35	1468689.87	13	13.22	0
145	1379728.26	15	16.80	1379728.26	15	17.77	0
146	174167.71	12	18.09	174167.71	12	18.68	0
147	731165.47	8	10.42	731165.47	8	10.71	0
148	81731.57	16	22.14	81731.57	16	22.92	0
149	728184.59	14	11.36	728184.59	14	11.22	0
150	124250.11	11	11.28	101962.08	11	11.99	17.9
151	367109.99	6	9.91	367109.99	6	10.02	0
152	1539982.82	8	10.47	1539982.82	8	12.44	0
153	1616479.35	13	14.97	1616479.35	13	14.55	0
154	605224.90	13	16.64	605224.90	13	16.96	0
155	237106.55	11	12.26	237106.55	11	15.30	0
156	865325.01	10	11.88	851253.02	10	12.86	1.6
157	256314.77	9	11.02	230143.97	10	11.66	1.02
158	326745.79	15	20.90	326745.79	15	22.90	0
159	65160.82	8	9.75	65160.82	8	10.75	0
160	152896.96	9	10.05	152896.96	9	12.05	0
161	32659.10	10	11.02	30115.55	10	11.96	7.7
162	1537587.27	15	18.60	1537587.27	15	19.02	0
163	1710124.37	6	10.07	1710124.37	6	12.22	0
164	1457066.18	8	11.01	1457066.18	8	13.44	0
165	1427833.62	9	13.50	1427833.62	9	14.03	0
166	308897.49	19	22.99	308897.49	19	23.46	0
167	1839053.85	13	16.20	1839053.85	13	16.63	0
168	446149.07	11	12.79	446149.07	11	14.25	0
169	1538733.90	13	14.66	1538733.90	13	14.96	0
170	1315571.45	13	15.23	1315571.45	13	15.05	0

Table 6. The best solution of GA and PSO algorithms for 200 random test problems (continued).

	GA			PSO			
Problem	**Profit**	***N***	**Run time (s)**	**Profit**	***N***	**Run time (s)**	**RPD (%)**
171	247885.07	13	14.51	247885.07	13	15.06	0
172	1728430.91	14	15.36	1502563.11	16	18.60	13.06
173	1683228.34	17	22.36	1683228.34	17	23.67	0
174	5563251.02	14	21.08	5325993.12	16	23.86	4.26
175	509415.80	9	10.59	509415.80	9	12.09	0
176	99104.17	9	11.66	99104.17	9	12.22	0
177	936742.68	16	23.63	936742.68	16	24.35	0
178	244092.79	7	9.00	244092.79	7	11.23	0
179	86325.01	6	8.85	84196.15	7	9.79	2.4
180	126953.12	10	10.33	101563.58	10	12.08	20
181	336928.01	18	22.97	336928.01	18	23.09	0
182	1166452.34	11	13.45	1166452.34	11	15.33	0
183	1838892.92	13	15.50	1838892.92	13	16.56	0
184	1654676.54	11	14.92	1654676.54	11	14.09	0
185	1755249.13	10	10.99	1755249.13	10	12.93	0
186	1734489.21	17	20.69	1734489.21	17	22.77	0
187	1030611.72	12	16.92	1030611.72	12	18.62	0
188	495947.99	16	18.16	495947.99	16	18.42	0
189	536870.80	19	20.31	536870.80	19	21.29	0
190	503085.08	16	19.37	503085.08	16	19.12	0
191	156850.22	13	12.87	156850.22	13	13.23	0
192	1557898.43	12	11.09	1557898.43	12	12.86	0
193	1321429.31	13	14.35	1321429.31	13	15.33	0
194	1766272.28	8	10.79	1766272.28	8	13.08	0
195	922738.55	19	20.03	922738.55	19	22.12	0
196	608951.39	16	16.30	608951.39	16	16.19	0
197	377142.61	13	11.89	377142.61	13	11.33	0
198	486321.05	7	9.45	452663.19	9	10.99	6.9
199	256985.02	9	11.05	256985.02	9	12.98	0
200	1356998.35	11	12.33	1356998.35	11	13.09	0

Table 7. The average computational results.

GA			PSO		
Avg. profit	**Avg. *N***	**Avg. run time (s)**	**Avg. profit**	**Avg. *N***	**Avg. run time (s)**
774798.4	10.76	13.2773	743995.9	10.825	15.0019

Table 8. Sensitivity analysis on X value.

			GA		PSO	
Problem	***L***	***X***	**Profit**	***N***	**Profit**	***N***
1	100	70	83575.99	18	76988.6	16
2	100	75	132490.54	15	128074.96	15
3	100	80		12	189271.05	10
4	100	85	267503.95	10	263690.64	10
5	100	90	324303.89	9	324303.89	9

By examining Figures 9 and 10, it can be concluded that GA presents a solution with better or similar profit and fewer discounts whit respect to PSO.

4.2.3. The sensitivity analysis

The sensitivity analysis on the X value is shown in Table 8. It is observed that by increasing the value of X, profit increased, but the number of discounts decreased. Moreover, by decreasing the time distance between X and L , the time intervals of discounts will be reduced.

Figure 11 depicts the profit values of the considered problems in Table 8 that are obtained by GA and PSO algorithms. As can see, GA presented a better solution with respect to the PSO algorithm.

Moreover, in this subsection, we compare two strategies for discount start time, (a) from the beginning of the period and (b) from X-point. The value of parameters in Table 1 that are related to Cheese is considered as a base case. Profit and the number of discounts (N) are calculated based on different values of parameters that are low and more than the base case. Since the GA outperformed PSO, therefore, we applied it to investigate the strategies. The numerical results of the comparison are illustrated in Table 9.

As seen as in Table 9, considering discount start time from the beginning of the period will always follow less profit than when the discounts are started from X-point. This is due to the greater number of discounts and faster selling price reduction in strategy (a).

Comparison of the base case and other cases show that with increasing α, profit increases and the number of discounts decreases. As β decreases, profit increases and the number of discounts decreases. This is logical with respect to the considered demand function. Moreover, with growth

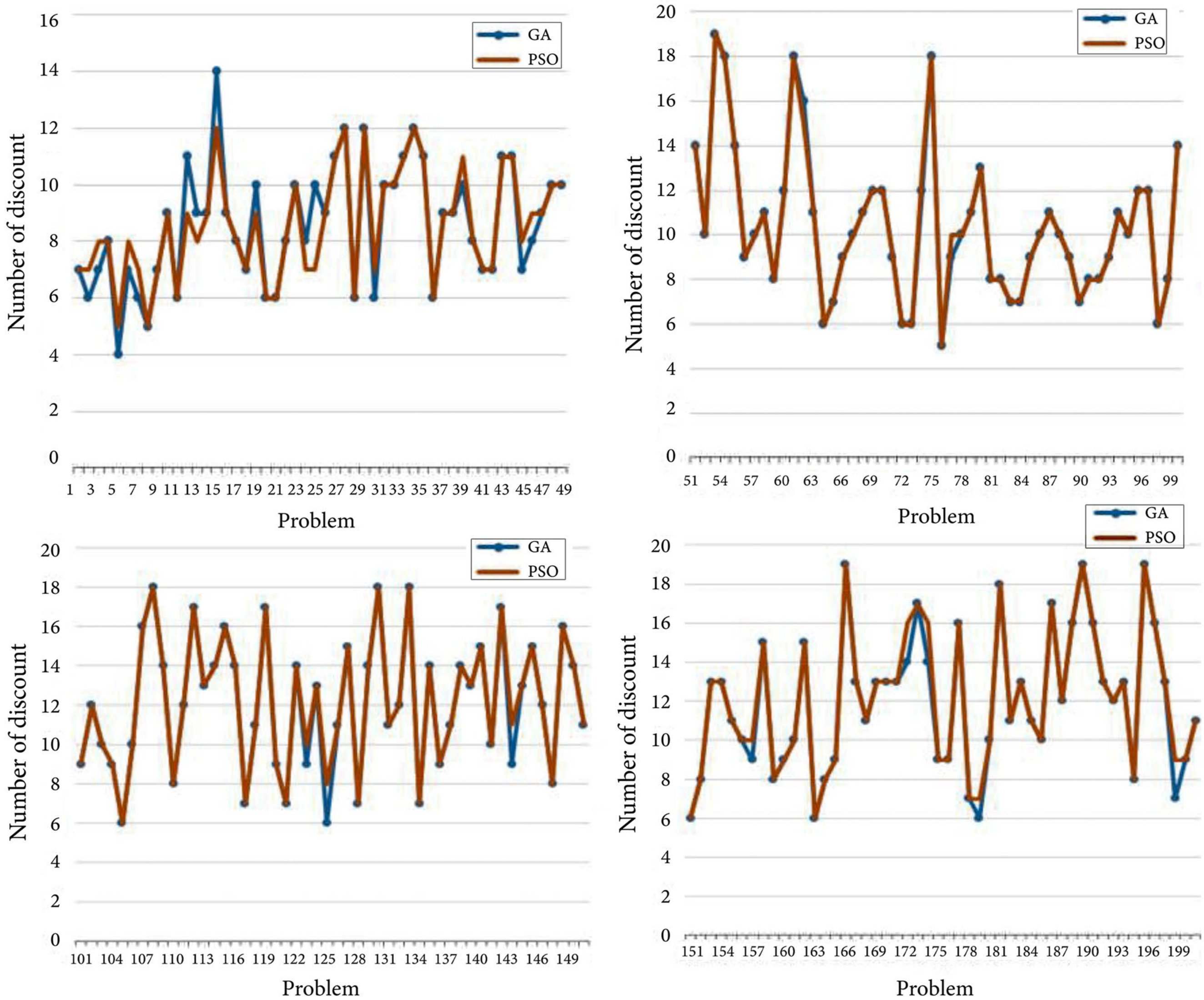


Figure 10. The number of discounts for each problem.

Table 9. Comparison of two strategies for discount start time, (a) from the beginning of the period and (b) from point *X*.

Parameters	From the beginning of the period		From X-point ($X > 0$)	
	Profit	*N*	Profit	*N*
Base case	98102	13	340176	7
$\alpha = 0.6$	90891	14	300122	9
$\alpha = 0.9$	124785	10	541550	4
$\beta = 1$	95148	13	338179	7
$\beta = 3$	90254	14	360778	6
$DC = 0.05$	60451	16	290447	9
$DC = 0.2$	112581	11	380014	6
$L = 30$	70500	16	240154	10
$L = 90$	130147	10	450014	6
$\theta = 0.01$	110487	11	441511	6
$\theta = 0.1$	55177	16	112145	11
$C = 2500$	100128	13	350998	7
$C = 3200$	112478	12	367741	7

DC, demand and profit increase and the number of discounts decreases. As L increases, we observe that the number of discounts decreases and profit increases. If the deterioration rate of a product (θ) is high, the number of discounts will be increased. Moreover, with growth C, profit and the number of discounts will be changed very low.

4.2.4. Verification and validation of GA and PSO algorithms

To guarantee the validity of the PSO and GA, the relaxed MINLP model is solved by the BARON solver in GAMS software. The model is MINLP with complicated variables; therefore, the relaxed model is obtained by considering a constant value for N (the value t_i is calculated by the value

Table 10. Comparison GA and PSO with BARON.

	GA			PSO			BARON		
Problem	**Profit**	**Run time (s)**	**RPD (%)**	**Profit**	**Run time (s)**	**RPD (%)**	**Profit**	**Run time (s)**	**RPD (%)**
1	1105569.59	14.77	1.81	1105569.59	16.01	1.81	1125988.14	6.44	0.00
2	1353491.58	10.36	9.89	1353491.58	19.67	9.89	1501825.88	2.36	0.00
3	1057974.77	14.02	9.54	996025.88	18.26	14.83	1169514.45	2.88	0.00
4	941158.36	14.88	3.53	941158.36	16.79	3.53	975639.07	5.01	0.00
5	971182.71	15.05	1.45	956110.68	18.04	2.98	985562.74	4.41	0.00
6	987714.83	15.44	8.85	951496.83	16.01	12.19	1083638.41	4.58	0.00
7	990984.25	16.78	6.14	990984.25	19.21	6.14	1055861.22	6.47	0.00
8	1069919.03	17.21	0.00	1069919.03	18.99	0.00	1069919.03	5.66	0.00
9	916839.53	14.55	2.23	916839.53	16.99	2.23	937710.88	6.03	0.00
10	1194089.49	14.88	5.15	1154893.11	16.15	8.27	1259056.33	7.99	0.00
Average	1058892.41	14.79	4.86	1043648.88	17.61	6.19	1116471.62	5.18	0.00

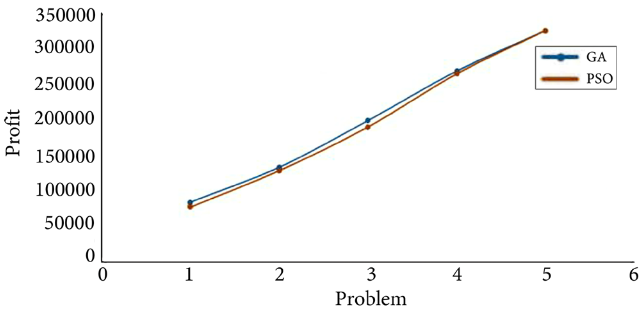


Figure 11. The profit values of the considered problems in Table 7.

N). 10 random test problems are generated based on the range of parameter values in Table 5. The test problems are solved by GA, PSO and BARON solver by considering $N = 8$, $L = 100$, and $X = 85$ and the results are shown in Table 10. The BARON solver is presented a better solution for the relaxed model in all the random test problems. The RPD for each test problem that is solved by GA and PSO algorithms is calculated based on the solution of BARON. The average RPD for GA and PSO is equal to 4.86% and 6.19%. As can see, the obtained solution from the algorithms is very close to the best solution for some test problems.

5. Conclusion

In this paper, the inventory control and price discount problem of perishable products with price and age-dependent demand is investigated. It is assumed that the starting point of discounts will be determined by the seller or the owner of the business, and the end of the period is the same as the expiration date of the product. A nonlinear mathematical model is proposed to maximize profit by determining discount points until the expiration date of the product and achieve to zero level of inventory on the date. The proposed model is computationally intractable. Therefore, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) algorithms are developed to solve the model. Assessments of the proposed algorithms using randomly generated data demonstrated that GA outperforms PSO algorithm. The obtained results are shown that the number of discounts and the discount start time have a high effect on profit. Suggestions for future studies include solving the model with other metaheuristic algorithms, considering other demand functions and stochastic demand. Extension of the proposed model under allowed shortage and considering multi-level supply chain are other avenues for further research.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of 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.

Authors contribution statement

Farzin Bazrafshan: Conceptualization; Data curation; Formal analysis; Methodology; Resources; Software; Validation; Visualization; Roles/Wrting-original draft;

Saeed Emami: Conceptualization; Data curation; Formal analysis; Methodology; Project administration; Software; Supervision; Validation; Writing- review and editing;

Hamid Mashreghi: Data curation; Formal analysis; Resources; Supervision; Visualization; Writing- review and editing.

Reference

1. Harcar, T. and Karakaya, F. “A cross-cultural exploration of attitudes toward product expiration dates”, *Psychology & Marketing,* **22**(4), pp. 353–371 (2005). https://doi.org/10.1002/mar.20063

2. Minner, S. and Transchel, S. “Periodic review inventory-control for perishable products under service-level constraints”, *OR Spectrum,* **32**(4), pp. 979-996 (2010). https://doi.org/10.1007/s00291-010-0196-1

3. Kaya, O. and Polat, A.L. “Coordinated pricing and inventory decisions for perishable products”, *OR Spectrum,* **39**(2), pp. 589-606 (2017). https://doi.org/10.1007/s00291-016-0467-6

4. Nahmias, S. “Perishable inventory theory: a review”, *Operations Research,* **30**(3), pp. 680-708 (1982). https://doi.org/10.1287/opre.30.4.680

5. Janssen, L., Claus, T., and Sauer, J. "Literature review of deteriorating inventory models by key topics from 2012 to 2015", *International Journal of Production Economics,* **182**(1), pp. 86-112 (2016). https://doi.org/10.1016/j.ijpe.2016.08.019

6. Burwell, T.H., Dave, D.S., Fitzpatrick, K.E., et al. "An inventory model with planned shortages and price dependent demand", *Decision Sciences,* **22**(5), pp. 1187-1191 (1991). https://doi.org/10.1111/j.1540-5915.1991.tb01916.x

7. Rajan, A., Rakesh, and Steinberg, R. "Dynamic pricing and ordering decisions by a monopolist", *Management Science,* **38**(2), pp. 240-262 (1992). https://doi.org/10.1287/mnsc.38.2.240

8. Abad, P.L. "Optimal price and order size for a reseller under partial backordering", *Computers & Operations Research,* **28**(1), pp. 53-65 (2001). https://doi.org/10.1016/s0305-0548(99)00086-6

9. Abad, P.L. "Optimal pricing and lot-sizing under conditions of perishability, finite production and partial backordering and lost sale", *European Journal of Operational Research,* **144**(3), pp. 677-685 (2003). https://doi.org/10.1016/s0377-2217(02)00159-5

10. Mukhopadhyay, S., Mukherjee, R.N., and Chaudhuri, K.S. "Joint pricing and ordering policy for a deteriorating inventory", *Computers & Industrial Engineering,* **47**(4), pp. 339-349 (2004). https://doi.org/10.1016/j.cie.2004.06.007

11. You, P.S. "Inventory policy for products with price and time-dependent demands", *Journal of the Operational Research Society,* **56**(7), pp. 870-873 (2005). https://doi.org/10.1057/palgrave.jors.2601905

12. Sana, S.S. "Optimal selling price and lotsize with time varying deterioration and partial backlogging", *Applied Mathematics and Computation,* **217**(1), pp. 185-194 (2010). https://doi.org/10.1016/j.amc.2010.05.040

13. Avinadav, T., Herbon, A., and Spiegel, U. "Optimal inventory policy for a perishable item with demand function sensitive to price and time", *International Journal of Production Economics,* **144**(2), pp. 497-506 (2013). https://doi.org/10.1016/j.ijpe.2013.03.022

14. Wee, H.M. "Economic production lot size model for deteriorating items with partial back-ordering", *Computers & Industrial Engineering,* **24**(3), pp. 449-458 (1993). https://doi.org/10.1016/0360-8352(93)90040-5

15. Wagner, H.M. and Whitin, T.M. "Dynamic version of the economic lot size model", *Management Science,* **5**(1), pp. 89-96 (1958). https://doi.org/10.1287/mnsc.1040.0306

16. Ghare, P.M. and Schrader, G.F. "A model for exponentially decaying inventory", *Journal of Industrial Engineering,* **14**(5), pp. 238-243 (1963).

17. Bhunia, A.K. and Maiti, M. "Deterministic inventory model for deteriorating items with finite rate of replenishment dependent on inventory level", *Computers & Operations Research,* **25**(11), pp. 997-1006 (1998). https://doi.org/10.1016/s0305-0548(97)00091-9

18. Wee, H.M. "A replenishment policy for items with a price-dependent demand and a varying rate of deterioration", *Production Planning & Control,* **8**(5), pp. 494-499 (1997). https://doi.org/10.1080/095372897235073

19. Chakrabarty, T., Giri, B.C., and Chaudhuri, K.S. "An EOQ model for items with Weibull distribution deterioration, shortages and trended demand: an extension of Philip's model", *Computers & Operations Research,* **25**(7), pp. 649-657 (1998). https://doi.org/10.1016/s0305-0548(97)00081-6

20. Eilon, S. and Mallaya, R.V. "Issuing and pricing policy of semi-perishables", in *Proceedings of the 4th International Conference on Operational Research*., Wiley-Interscience (1966).

21. Wee, H.M. "Joint pricing and replenishment policy for deteriorating inventory with declining market", *International Journal of Production Economics,* **40**(2-3), pp. 163-171 (1995). https://doi.org/10.1016/0925-5273(95)00053-3

22. Federgruen, A. and Heching, A. "Combined pricing and inventory control under uncertainty", *Operations Research,* **47**(3), pp. 454-475 (1999). https://doi.org/10.1287/opre.47.3.454

23. Chang, H.J., Teng, J.T., Ouyang, L.Y., et al. "Retailer's optimal pricing and lot-sizing policies for deteriorating items with partial backlogging", *European Journal of Operational Research,* **168**(1), pp. 51-64 (2006). https://doi.org/10.1016/j.ejor.2004.05.003

24. Chen, L.M. and Sapra, A. "Joint inventory and pricing decisions for perishable products with two-period lifetime", *Naval Research Logistics (NRL),* **60**(5), pp. 343-366 (2013).https://doi.org/10.1002/nav.21538

25. Taleizadeh, A.A., Satarian, F., and Jamili, A. "Optimal multi-discount selling prices schedule for deteriorating product", *Scientia Iranica,* E, **22**(6), pp. 2595-2603 (2015).

26. Adenso-Díaz, B., Lozano, S., and Palacio, A. "Effects of dynamic pricing of perishable products on revenue and waste", *Applied Mathematical Modelling,* **45**, pp. 148-164 (2017). https://doi.org/10.1016/j.apm.2016.12.024

27. Feng, L., Chan, Y.L., and Cárdenas-Barrón, L.E. "Pricing and lot-sizing polices for perishable goods when the demand depends on selling price, displayed stocks, and expiration date", *International Journal of Production Economics,* **185**, pp. 11-20 (2017). https://doi.org/10.1016/j.ijpe.2016.12.017

28. Yao, D. "Joint pricing and inventory control for a stochastic inventory system with Brownian motion demand", *IISE Transactions,* **49**(12), pp. 1101-1111 (2017). https://doi.org/10.1080/24725854.2017.1355126

29. Dobson, G., Pinker, E.J., and Yildiz, O. "An EOQ model for perishable goods with age-dependent demand rate", *European Journal of Operational Research,* **257**(1), pp. 84-88 (2017). https://doi.org/10.1016/j.ejor.2016.06.073

30. Chua, G.A., Mokhlesi, R., and Sainathan, A. "Optimal discounting and replenishment policies for perishable products", *International Journal of Production Economics,* **186**, pp. 8-20 (2017). https://doi.org/10.1016/j.ijpe.2017.01.016

31. Kaya, O. and Rahimi Ghahroodi, S. "Inventory control and pricing for perishable products under age and price dependent stochastic demand", *Mathematical Methods of Operations Research,* **88**(1), pp. 1-35 (2018). https://doi.org/10.1007/s00186-017-0626-9

32. Agi, M.A. and Soni, H.N. "Joint pricing and inventory decisions for perishable products with age-, stock-, and price-dependent demand rate", *Journal of the Operational Research Society,* **71**(1), pp. 85-99 (2019). https://doi.org/10.1080/01605682.2018.1525473

33. Khan, M.A.A., Shaikh, A.A., Panda, G.C., et al. "Inventory system with expiration date: Pricing and replenishment decisions", *Computers & Industrial Engineering,* **132**, pp. 232-247 (2019). https://doi.org/10.1016/j.cie.2019.04.002

34. Wei, J., Liu, Y., Zhao, X., and Yang, X. "Joint optimization of pricing and inventory strategy for perishable product with the quality and quantity loss", *Journal of Industrial and Production Engineering,* **37**(1), pp. 23-32 (2020). https://doi.org/10.1080/21681015.2020.1719222

35. Dye, C.Y. "Optimal joint dynamic pricing, advertising and inventory control model for perishable items with psychic stock effect", *European Journal of Operational Research,* **283**(2), pp. 576-587 (2020). https://doi.org/10.1016/j.ejor.2019.11.008

36. Soni, H.N. "Joint pricing and inventory policies for perishable items with price discount based on freshness index", *Turkish Journal of Computer and Mathematics Education,* **12**(11), pp. 1954-1963 (2021). https://doi.org/10.17762/turcomat.v12i11.6150

37. Kaya, O. and Bayer, H. "Pricing and lot-sizing decisions for perishable products when demand changes by freshness", *Journal of Industrial and Management Optimization,* **17**(6), p. 3113 (2021). https://doi.org/10.3934/jimo.2020110

38. Tawarmalani, M. and Sahinidis, N.V. "GAMS/BARON: A tutorial and empirical performance analysis", *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming*, pp. 313-401 (2002). https://doi.org/10.1007/978-1-4757-3532-1_11

39. Bratton, D. and Kennedy, J. "Defining a standard for particle swarm optimization", in *Swarm Intelligence Symposium, SIS 2007,* IEEE (2007). https://doi.org/10.1109/sis.2007.368035

40. Snkar, K.P., *Classification and Learning Using Genetic Algorithms*, Springer (2007). https://doi.org/10.1007/3-540-49607-6

Biographies

Farzin Bazrafshan received his BS and MS degrees in Industrial Engineering from Babol Noshirvani University of Technology, Babol, Iran. His research interests include operations research, pricing and inventory control, and production planning.

Saeed Emami received his BS, MS and PhD degrees in Industrial Engineering from Isfahan University of Technology, Isfahan, Iran, and he is currently an Assistant Professor in the Department of Industrial Engineering at Babol Noshirani University of Technology, Babol, Iran. His research interests include operations research, robust optimization, Inventory control, production planning and scheduling.

Hamid Mashreghi received MS and PhD degrees from Industrial and Systems Engineering Department of Tarbiat Modares University, Tehran, Iran, respectively in 2008 and 2013. He currently works as an Assistant Professor at Industrial Engineering Department, Babol Noshirvani University of Technology, Babol, Iran. His current research interests include supply chain coordination and contracting, pricing optimization and behavioral operations management.