Two parameter-tuned meta-heuristics for a discounted inventory control problem in a fuzzy environment

Seyed Mohsen Mousavi¹, M.Sc.
Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran, Phone: +98 281 3665275, Fax: +98 281 3665277, E-mail: Mousavi.mohsen8@gmail.com

Javad Sadeghi, M.Sc.
Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran, E-mail: Sadeghi@qiau.ac.ir

Seyed Taghi Akhavan Niaki, Ph.D.
Department of Industrial Engineering, Sharif University of Technology, P.O. Box 11155-0414 Azadi Ave, Tehran 145889694 Iran E-mail: Niaki@Sharif.edu

Najmeh Alikar, M.Sc.
Department of Science and Technology, Universiti Kebangsaan Malaysia, 43600, Bangi, Selangor, Malaysia Email: Najme.alikar@gmail.com

Ardeshir Bahreininejad, Ph.D.
Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia, E-mail: Bahreininejad@um.edu.my

Hendrik Simon Cornelis Metselaar, Ph.D.
Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia, E-mail: h.metselaar@um.edu.my

Abstract
In this paper, a nearly real-world multi-product, multi-period inventory control problem under budget constraint is investigated, where shortages in combination with backorders and lost sales are considered for each product. The ordered quantities of products are delivered in batch sizes with a known number of boxes, each containing a pre-specified number of products. Some products are purchased under an all unit discount policy, and others are purchased under an incremental quantity discount with fuzzy discount rates. The goal is to find the optimal ordered quantities of products such that not only the total inventory cost but also the required storage space (considered as a fuzzy number) to store the products is minimized. The weighted linear sum of objectives is applied to generate a single-objective model for the bi-objective problem at hand and a harmony search algorithm is developed to solve the complex inventory problem. As no benchmarks are available to validate the obtained results, a particle-swarm optimization algorithm is employed to solve the problem in addition to validate the results given by the harmony search method. The parameters of both algorithms are tuned using both Taguchi and response surface methodology (RSM). Finally, to assess the performance of the proposed algorithms some numerical examples are generated, and the results are compared statistically.

Keywords: Multi-product multi-period inventory; Fuzzy discount rates; Fuzzy storage space; Harmony search algorithm; Particle swarm optimization; Taguchi; RSM

¹ Corresponding author
1. Introduction and literature review

Most real-world problems, in industries and commerce, are studied using a single-objective optimization model. The assumption that organizations always seek to minimize cost or maximize profit rather than make trade-offs among multiple objectives has been used extensively in the literature. In this regards, classical inventory models have been developed under the basic assumption that a single product is purchased or produced. However, in many real-life situations, this assumption does not hold. Instead of a single item, many firms, enterprises or vendors are motivated to store a number of products to enhance their business profitability. They are also motivated to attract customers to purchase several items simultaneously.

Ben-Daya and Raouf [1] considered a multi-product, single-period inventory control problem with stochastic demand, for which the multi-periodic inventory control problem was investigated in depth for multiple seasonal products. Recently, Wang and Xu [37] studied a multi-period, multi-product inventory control problem having several inventory classes that can be substituted for one another to satisfy the demand for a given reservation class. Chiang [4] investigated a periodic review model in which the period is partly long. The important aspect of his study was to introduce emergency orders to prevent shortage. A dynamic programming approach was employed to model the problem. Das et al. [6] developed a multi-item inventory model with constant demand and infinite replenishment under restrictions on storage area, average shortage, and inventory investment cost. Mohebbi and Posner [17] studied an inventory system based on periodic review, multiple replenishment, and multilevel delivery. They assumed that the demand follows a Poisson distribution, that shortages were allowed and that the lost sale policy could be employed. Lee and Kang [13] developed a model to manage the inventory of a single product in multiple periods, whereas Padmanabhan and Vrat [24] developed a multi-item/objective inventory model having deteriorating items
with stock dependent demand using a goal programming method. Moreover, Taleizadeh et al. [35] proposed a multi-product inventory control problem with a stochastic replenishment period in which the demands were fuzzy numbers, and shortages were allowed to occur with a combination of backorders and lost sales.

Quantity discount has attracted further attention because of its practical importance in purchasing and control of a product. It derives better marginal cost of purchase/production availing the chances of cost savings through bulk purchase/production. In supply chains, quantity discounts can be considered as an inventory coordination mechanism between a buyer and a supplier (Shin and Benton [31]). Benton [2] considered an inventory system having quantity discount for multiple price breaks and alternative lot-sizing policy. Maiti and Maiti [15] developed a model for multi-item inventory control system based on breakable items, taking into account all units (AUD) and incremental quantity discount (IQD) policies. In this paper, a combination of AUD and IQD is used, which is rather similar to the work by Sana and Chaudhuri [29]; however the deterministic discount rates were used in the latter study of the research in this paper. Sana and Chaudhuri [29] extended an economic order quantity (EOQ) model based on discounts through the relaxation of the pre-assumptions associated with payments. Furthermore, Taleizadeh et al. [34] considered a mixed integer, nonlinear programming for solving multi-product multi-constraint inventory control systems having stochastic replenishment intervals and incremental discounts for which a genetic algorithm was employed to find the near-optimum order quantities of the products. Recently, Mousavi et al. [20] improved the solution of a discounted multi-item multi-period inventory control problem for seasonal items, in which shortages are allowed, and the costs are calculated under inflation and time value of money.

In general, many of the variables, resources, and constraints used in a decision-making problem are considered to be either deterministic or stochastic. However, real-life
scenarios demand them to be imprecise, i.e., uncertainty is to be imposed in a non-stochastic sense. According to Maity [14], a business may start with some warehouse space in an inventory control problem. However, due to unexpected demands, some additional storage spaces may need to be added. These added spaces are normally imprecise and fuzzy in nature. Moreover, one of the weaknesses of some current inventory models is the unrealistic assumption that all items are purchased under a crisp discount policy. Nevertheless, in many inventory control systems, fuzzy discount rates are natural phenomena. Maity and Maiti [16] formulated an optimal production strategy for an inventory control system of deteriorating multi-items under a single owner based on resource constraints under inflation and discounting in a fuzzy environment. Recently, Chen and Ho [3] investigated a newsboy inventory problem in a fuzzy environment by analyzing the optimal inventory policy for the single-order newsboy problem considering fuzzy demand and quantity discounts. Moreover, Guchhait et al. [10] modeled an inventory control problem under discount based on fuzzy production rate and demand.

In this research, a multi-periodic inventory control problem is modeled for seasonal and fusion products in which the replenishment process begins at a pre-specified time (or season) and ends at another one. Binary variables are used in this research to model the purchasing cost based on discounted prices (similar to the work by Lee and Kang [13] that models price break points). However, an AUD policy for some products with crisp discount rates and an IQD policy for other items based on fuzzy discount rates are also considered in this research. Das et al. [6] that considered the shortage to be fully backordered; however, in this paper, shortages include a combination of backorders and lost sales. The constraints of the proposed model are budget, truck space, and the order quantities of the products in a given period. The problem will be investigated in two single-objective and bi-objective optimization approaches. In the first approach, the goal is to find the ordered quantities of the
products in different periods such that the total inventory cost of ordering, holding, shortage, and purchasing is minimized. In the second approach, the objective is to minimize both the total inventory cost and the required fuzzy storage space. The problems will be formulated into mixed binary nonlinear programming based on three binary variables to demonstrate the ordering, shortage, and purchasing costs. The weighted linear sum (WLS) method is used to transform the bi-objective inventory problem into a single-objective one. A harmony search (HS) algorithm is developed to solve the complex inventory problem. As no benchmarks are available to validate the obtained results, a particle-swarm optimization (PSO) algorithm is employed to solve the problem as well as the validation of the performance of HS algorithm. The parameters of both algorithms are tuned using both Taguchi and RSM methods.

In recent decades, scientists have been mimicking natural phenomena to propose methods and algorithms for solving complex optimization problems. Based on the complexity of real-life optimization problems, one may not be able to use exact algorithms. Therefore, typically, meta-heuristic methods are frequently used to find a near optimum solution in an acceptable period of time. Genetic algorithms (Mousavi et al. [19]), simulating annealing (Taleizadeh et al. [32]), particle swarm optimization (Sadeghi et al. [28]), and harmony search (Taleizadeh et al. [34]) are among the most popular meta-heuristic algorithms used for search and optimization.

Geem and Kim [7] developed a HS algorithm that was conceptualized using the musical process of searching for a perfect state of harmony. Harmony in music is analogous to the optimization solution vector, and the musician’s improvisations are analogous to local and global search schemes in optimization techniques. The HS algorithm does not require initial values for the decision variables. Lee and Geem [12] proposed a new HS model for solving engineering optimization problems for continuous design variables. Moreover, in
inventory control problems, Taleizadeh et al. [34] employed a HS algorithm to solve a complicated inventory problem.

The particle swarm optimization (PSO) algorithms are also considered in the category of meta-heuristic optimization methods, as developed by Kennedy and Eberhart [11]. PSO was inspired by the social behavior of birds flocking or fish schooling. Taleizadeh et al. [33] and Tsou [36] utilized PSO to optimize inventory control problems. Furthermore, Yildiz [41] compared the results of PSO and HSA for multi-pass turning problems in manufacturing industries, where a Taguchi method was employed to calibrate the parameters. Yildiz also proposed a hybrid PSO to optimize problems in both the design and manufacturing areas [40]. Moreover, the seminal work of Coello Coello et al. [5] is based on multi-objective PSO (MOPSO).

The remainder of the paper is organized as follows. In Section 2, the problem, along with its assumptions, is defined. In Section 3, the defined problem of Section 2 is modeled. Furthermore, the parameters and the variables of the problem are first introduced. Then, the proposed HSA and PSO algorithms are presented in Section 4. Section 5 contains some numerical examples to both calibrate the parameters of the two algorithms and to compare their performances. Finally, the conclusion and recommendations for future research are presented in Section 6.

2. Problem definition

Consider a multi-product, multi-period inventory control problem in which the products are purchased based on two policies of AUD with crisp discount rates for some products and IQD with fuzzy discount rates for others. The total required budget is constrained and based on some physical limitations (e.g., truck capacity), and the order quantities of the products in all periods are considered as limited. Moreover, an upper bound
is defined for the ordered quantity of a product in a period based on some manufacturing constraints. The demands of the products for different periods are considered constant and independent. When a shortage occurs, a fraction is considered backordered and a fraction as lost sales. The costs involved in the inventory control system are ordering, holding, shortages (backorder and lost sale), and purchasing. Furthermore, in addition to minimizing the total inventory cost, the minimization of the total required fuzzy storage space is also considered. This is based on some adopted manufacturing policies such as setting up a new manufacturing line or creating a new storage. Therefore, the objective is to identify the ordered quantities of the products in each period so that the two objective functions of the problem, i.e., the total inventory costs and the total required storage space, are minimized simultaneously.

3. Modeling

To model the problem, a number of inventory assumptions are first introduced. Then, the parameters and the variables of the model are defined. Based on these definitions, the objectives and the constraints are derived in the following sections.

The inventory assumptions are as follows:

- Replenishment is instantaneous, i.e., the delivery time is assumed negligible. Based on other inventory research works, this assumption is made for the sake of simplicity and will not impose considerable influence on the modeling aspects.
- Demand rates of all products are independent of one another and are constants in a period. This is a general assumption made in many inventory problems.
- In each period, at most one order can be placed for a product, which is the case that usually occurs in real-world situations.
• All products are delivered in special boxes of pre-specified capacities, i.e., the ordered quantity of products is a multiple of a fixed-sized batch.
• In case of shortage, a fraction of demand is considered as backorders and a fraction as lost sales.
• The initial inventory level of all products is considered as to be zero.
• To satisfy the demands as much as possible, the ordered quantity of each product in a period is considered as equal to the shortage quantity of the product in the previous period.
• The planning horizon, a future time period during which departments that support production will plan production work and determine material requirements, is finite and known. In the planning horizon, there are $T$ periods of an equal length, the case that happens for seasonal and fashion products.
• The total available budget to purchase products in a period is considered as limited.
• The AUD policy for some products and the IQD for the others are employed by the vendor in which a fuzzy discount rate is considered for the products that are purchased under the IQD policy.
• The required storage space for each product is considered as fuzzy numbers. This assumption helps to consider the condition in which a decision maker wants to provide a new building warehouse and does not know the right storage size.

3.1. Parameters, variables, and decision variables

For a specific product $i$, $i = 1, 2, ..., m$; a specific time period $t$, $t = 0, 1, ..., T$; and a specific price break point $j$, $j = 1, 2, ..., J$; the parameters and the variables of the model are defined as follows:
Parameters and variables:

\( a_{ij} \): A binary variable that is set to 1 if item \( i \) is purchased at price break point \( j \) in period \( t \) and set to 0 otherwise

\( D_{it} \): Demand of \( i^{th} \) product in period \( t \)

\( E_{it} \): Total time elapsed up to and including the \( t^{th} \) replenishment cycle of the \( i^{th} \) product

\( E_i' \): The time in period \( t \) at which the inventory of product \( i \) reaches zero

\( f \): An upper bound for ordered quantities of all products

\( h_{it} \): Inventory holding cost per unit time of \( i^{th} \) product in period \( t \)

\( I_{it} \): Inventory position of \( i^{th} \) product in period \( t \)

\( k_i \): The fixed batch size of \( i^{th} \) product

\( \tilde{m}_{ij} \): The fuzzy discount rate of \( i^{th} \) product in \( j^{th} \) price break point (\( \tilde{m}_{ij} = 0 \))

\( n_{it} \): A binary variable that is set to 1 if a shortage occurs for product \( i \) in period \( t \) and set to 0 otherwise

\( O_{it} \): Ordering cost per unit of \( i^{th} \) product in period \( t \)

\( P_i \): Purchasing cost per unit of \( i^{th} \) product

\( P_{ij} \): Purchasing cost per unit of \( i^{th} \) product at \( j^{th} \) price break point

\( q_{ij} \): \( j^{th} \) price break-point for \( i^{th} \) product (\( q_{ij} = 0 \))

\( S_i \): The fuzzy required storage space per unit of \( i^{th} \) product

\( TB \): Total available budget in a period

\( TMF \): The combined objective function (the weighted combination of the total inventory cost and the total available storage space)
\( ub : \) An upper bound for the available number of boxes

\( w_1 : \) The weight associated with the total inventory cost \((0 \leq w_1 \leq 1)\)

\( w_2 : \) The weight associated with the total required storage space \((0 \leq w_2 \leq 1)\)

\( x_{it} : \) A binary variable that is set to 1 if a purchase of product \(i\) is made in period \(t\) and set to 0 otherwise

\( Z_1 : \) The total inventory costs

\( Z_2 : \) The total required storage space

\( \pi_{it} : \) Backorder cost per unit of \(i^{th}\) product in period \(t\)

\( \hat{\pi}_{it} : \) Lost sale cost per unit of \(i^{th}\) product in period \(t\)

\( \alpha_i : \) Percentage of unsatisfied demand of \(i^{th}\) product that is backordered

\textit{Decision variables:}

\( M_{it} : \) Number of boxes for \(i^{th}\) product ordered in period \(t\)

\( b_{it} : \) Shortage quantity for \(i^{th}\) product in period \(t\)

\( Q_{it} : \) Ordering quantity of \(i^{th}\) product in period \(t\)

\( Y_{it} : \) The initial positive inventory of \(i^{th}\) product in period \(t\) (in \(t = 1\), the beginning inventory of all items is zero)

3.1. The inventory graph

A graphical representation of some possible situations for an inventory control problem of product \(i\) is shown in Fig. 1. According to Fig. 1, the inventory planning starts in period \(E_0\) and ends in period \(E_T\), where a shortage in combination of backorders and lost sales is possible in some periods. In addition, to satisfy the demand in a period in which a
shortage occurs for a product, the order quantity is considered greater than or equal to its shortage quantity in the previous period (i.e., $Q_{i,t} \geq b_i$).

Please insert Fig 1 about here

In Figure 1, an order quantity of $Q_{i,t-1}$ is received at the end of period $t-2$. This quantity satisfies $D_{i,t}$ with a remaining $Q_{i,t-1}$ shown by $Y_{i,t}$ to satisfy the current demand in the time interval $[t-1, t]$. The inventory $Y_{i,t}$ and the order quantity received in period $t-1$, i.e., $Q_{i,t}$, are not enough to satisfy demand $D_{i,t}$. Thus, a shortage $b_{i,t}$ occurs with $\alpha_i$ percentage as backorders and $(1-\alpha_i)$ percentage as lost sales. When the order quantity $Q_{i,t+1}$ is received, the shortage $b_{i,t}$ is first satisfied, and the rest, $(Q_{i,t+1} - b_{i,t})$, is used to satisfy $D_{i,t+1}$.

3.2. The total inventory cost

The total inventory cost considered in this study consists of four costs of ordering, holding, shortages, and purchasing that are modeled as follows.

Given that, at most, one order can be placed for a product in a period, a binary variable $x_{it}$ is used that is set to 1 if a purchase of a unit of product $i$ is made in period $t$ and set to 0 otherwise. As a result, the total ordering cost ($O$) is obtained by

$$O = \sum_{i=1}^{m} \sum_{t=1}^{T-1} O_{it} x_{it}$$

The holding cost of a product in a period is equal to the area of a trapezoid above the horizontal line of Fig. 1. Therefore, for $E_{it-1} + \epsilon_{it} < t < E_{it} + E_{it} n_{it}$, the holding cost ($HC$) is given by

$$HC = \int_{E_{it-1}}^{E_{it} + E_{it} n_{it}} I_i(t) \, dt$$

11
where $I_i(t)$ is the inventory of $i^{th}$ product in time $t$. In Eq. (2), if a shortage occurs for product $i$, it is clear that $n_i = 1$. Hence, the term $E_i (1-n_i) + E_i n_i$ will be equal to $E_i'$. Otherwise, $n_i = 0$, and the term $E_i (1-n_i) + E_i n_i$ will be equal to $E_i'$. As the inventory position of product $i$ in period $t$ is $I_{i+1} = I_i + Q_i - D_i$ and that at time $E_i'$ in period $t$ the inventory becomes zero, the total holding cost is given by

$$HC = \sum_{i=1}^{m} \sum_{t=1}^{T-1} \left( \frac{Y_{it} + Q_{it} + Y_{it+1}}{2} \right) (E_i (1-n_i) + E_i n_i - E_i') h_i$$

(3)

The shortage cost consists of two parts: backorders and lost sales. As the $\alpha_i$ percentage of the demand of product $i$ represents the backorders, and the $(1-\alpha_i)$ percentage represents lost sales, the partial backorder (PB) and partial lost sale (PL) costs are obtained by (see Fig. 1)

$$PB = \sum_{i=1}^{m} \sum_{t=1}^{T-1} \left( \frac{\pi_i b_i}{2} (E_i - E_i') \alpha_i \right)$$

(4)

$$PL = \sum_{i=1}^{m} \sum_{t=1}^{T-1} \left( \frac{\pi_i b_i}{2} (E_i - E_i')(1-\alpha_i) \right)$$

(5)

where $(E_i - E_i') = \frac{b_i}{D_i}$.

To formulate the purchasing cost under the all-unit discount policy, let the price break points be considered as

$$P_i = \begin{cases} P_{i_1} & 0 < Q_{it} \leq q_{i_2} \\ P_{i_2} & q_{i_2} < Q_{it} \leq q_{i_3} \\ \vdots \\ P_{i_j} & q_{i_j} < Q_{it} \end{cases}$$

(6)

Then, the total purchasing cost under the AUD policy ($TPCA$) is given by

$$TPCA = \sum_{i=1}^{m} \sum_{t=1}^{T-1} \sum_{j=1}^{J} Q_{it} P_{ij} a_{ij}$$

(7)
In the IQD policy, the purchasing cost for each unit of \(i^{th}\) product depends on its ordered quantity, where in each price discount-point, it is obtained by

\[
P_i q_{i,1} + P_i (Q_{i,1} - q_{i,1})(1 - \bar{m}_{i,1}) + \cdots + P_i (Q_{i,J} - q_{i,J})(1 - \bar{m}_{i,J}) \quad q_{i,J} < Q_{i,J} < q_{i,J-1} \quad \cdots \quad q_{i,2} < Q_{i,2} \leq q_{i,1} \quad 0 < Q_{i,1} \leq q_{i,1}
\]

(8)

Note that in Eq. (8) the vendor offers a fuzzy discount rate of \(\bar{m}\) to the buyer based on the vendor's purchased quantity. Therefore, the total purchasing cost under this policy (TPCI) is obtained by

\[
TPCI = \sum_{i=1}^{m} \sum_{t=1}^{T-1} \left\{ (Q_{i,t} - q_{i,t})P_i q_{i,t}(1 - \bar{m}_{i,t}) + \sum_{j=t}^{T-1} (q_{i,j+1} - q_{i,j})P_i (1 - \bar{m}_{i,j}) \right\}
\]

(9)

Hence, with the supposition that \(P=TPC+TPCI\), the first objective function is to minimize the total inventory cost \(Z_1 = O + HC + PB + PL + P\) obtained by Equations (1), (3), (4), (5), (7), and (9).

The second objective function corresponds to the minimization of the total required storage space \((Z_2)\) for which the required storage space per unit of a product, \(\bar{S}_i\), are considered as fuzzy numbers. It is obtained using

\[
Z_2 = \sum_{i=1}^{m} \sum_{t=1}^{T-1} (Y_{i,t} + Q_{i,t})\bar{S}_i
\]

(10)

The ordered quantities of an item in a given period \(Q_{i,t}\), in addition to its remaining inventory in the previous period \(Y_{i,t}\), are allocated to storage. A unit of product \(i\) requires \(\bar{S}_i\) units of storage, and the holding cost is assumed to be independent of the storage space. A significant part of the holding cost is based on the capital investment and hence the part corresponding to maintaining the warehouse is assumed to be negligible.
Therefore, the combined objective function (the fitness function) of the problem using the weighted linear sum approach is given by

\[ TMF = w_1 \cdot Z_1 + w_2 \cdot Z_2 \]  

(11)

3.3. The constraints

The inventory of product \( i \) in period \( t \), i.e., \( I_{it} \), can be either positive, as denoted by \( Y_{it+1} \) (the beginning inventory of period \( t+1 \)), or negative, as denoted by \( b_{it} \) (the shortage quantity in period \( t \)). In other words,

\[ I_{it} = \begin{cases} Y_{it+1} &; \quad I_{it} \geq 0 \\ b_{it} &; \quad I_{it} < 0 \end{cases} \]  

(12)

Moreover, the starting inventory of product \( i \) in period \( t+1 \) is equal to its beginning inventory in the previous period \( t \) plus the ordered quantity minus the demand. Or

\[ Y_{it+1} = Y_{it} + Q_{it} - D_{it} \left( E_{it} n_{it} + E_{it} \left( 1 - n_{it} \right) - E_{it-1} \right) \]  

(13)

As the ordered quantity of product \( i \) in period \( t \), \( Q_{it} \), is delivered in \( M_{it} \) boxes, each containing \( k_{it} \) products, the next constraint is given by

\[ Q_{it} = k_{it} M_{it} \]  

(14)

Based on the real-world limitations of truck space, the order quantity of all products in all periods cannot be greater than a given fixed number \( f \). In other words,

\[ \sum_{i=1}^{m} \sum_{t=1}^{T} Q_{it} \leq f \]  

(15)

Furthermore, the number of available boxes to deliver product \( i \) in period \( t \) is limited. Thus,

\[ Q_{it} \leq ub \]  

(16)

In addition, the purchasing price per unit of product \( i \) is \( P_i \); the order quantity of product \( i \) in period \( t \) is \( Q_{it} \); the total budget is \( TB \). As a result the budget constraint is
Finally, as, at most, one order can be placed for a product in a period, and it can be purchased at one price break point, we have
\[
\sum_{j=1}^{J} a_{ij} = 1 \quad (18)
\]

Therefore, the complete mathematical model of the problem is given as

\[
\begin{align*}
\text{Min } Z_1 \\
\text{Min } Z_2 \\
\text{s.t.}:
\end{align*}
\]

\[
I_{it} = \begin{cases} 
Y_{it} + 1 & ; I_{it} \geq 0 \\
b_i & ; I_{it} < 0 
\end{cases}
\]

\[
Y_{it+1} = Y_{it} + Q_a - D_a \left( E_{it} + n_{it} + E_{a} (1 - n_{it}) - E_{a-1} \right)
\]

\[
Q_a = k_i M_a 
\]

\[
\sum_{i=1}^{m} \sum_{t=1}^{T} Q_{it} \leq f 
\]

\[
Q_{it} \leq ub 
\]

\[
\sum_{i=1}^{m} \sum_{t=1}^{T} Q_{it} P_{it} \leq TB 
\]

\[
\sum_{j=1}^{J} a_{ij} = 1 
\]

\[i = 1, 2, ..., m \quad ; \quad t = 0, 1, ..., T \quad ; \quad j = 1, 2, ..., J \quad (19)\]

4. Solution methods

In this section, two solution approaches, namely, a harmony search (HSA) and a particle-swarm optimization (PSO) algorithm, are employed to solve the complex mixed
binary nonlinear programming problem in Eq. (19). As no benchmark was available in the literature, two optimization methods have been adopted in this paper for validation purposes.

4.1. A harmony search algorithm

The steps involved in the HS algorithm of this research are detailed in the following subsections.

4.1.1. Parameter initialization

The parameters of HSA to solve the problem in (19) as follows:
1) the harmony memory size (HMS), which is the number of solution vectors in harmony memory (HM),
2) the harmony memory considering rate (HMCR),
3) the pitch adjusting rate (PAR), and
4) the stopping criterion.

The parameters HMCR and PAR are used to improve the solution vector. In this research, the values for HM, HMCR, PAR, and stopping criterion are tuned using both the Taguchi and RSM methods, explained in detail in Mousavi et al. [20], Pasandideh et al. [26], Najafi et al. [22], Yildiz [39, 41, 42], and Pasandideh et al. [25]. Eq. (20) shows a HM matrix of the order quantities of the products where $M$ is a decision matrix shown in Fig. 2.

$$HM = \begin{bmatrix} M_1 \\ M_2 \\ \vdots \\ M_{HMS} \end{bmatrix}$$  \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
4.1.2. New harmony improvisation

Random selection, HM consideration, and pitch adjustment are three rules of the new harmony improvisation.

In the random selection rule, as a musician plays any pitch within the instrument range, the new value of each decision variable, $M^t_i : i=1,2,...,m; t=1,2,...,T-1$, is randomly generated within the range $[0, ub]$, where the new solution matrix is denoted by

$$
M' = 
\begin{pmatrix}
M'_{11} & M'_{12} & \ldots & M'_{1,T-1} \\
M'_{21} & M'_{22} & \ldots & M'_{2,T-1} \\
\vdots & \vdots & \ddots & \vdots \\
M'_{m1} & M'_{m2} & \ldots & M'_{m,T-1}
\end{pmatrix}
$$

(21)

In the HM consideration rule, as a musician plays any pitch out of the preferred pitches in his/her memory in HM, $\{(M'_1, M'_2, ..., M'_{HM})\}$, with a probability of $P_{HMCR}$, it is randomly chosen with a probability of $(1-P_{HMCR})$ in the random selection process [8]. Fig. 3 presents the pseudo-code for the choosing and the selection processes [7]. In pitch adjustment, every component obtained by the memory consideration is examined to determine whether it should be pitch adjusted or not. The value of the decision variable is changed using Eq. (22) with probability of $P_{PAR}$, and is kept without any change with probability $(1 - P_{PAR})$. In Eq. (22), the selection to increase or decrease each vector component are carried out with the same probability [26].

Please insert Fig 3 about here

$$
M' = M' \pm (Rand)(BWC); \quad Rand \sim (0,1)
$$

(22)

where $BWC$ denotes the amount of changes for pitch adjustment.
4.1.3. Harmony memory update

The penalty function approach is employed in this research to ensure that the randomly generated decision variables are feasible, i.e., they satisfy all the constraints of the model in Eq. (19). To do this, consider a typical constraint of the form given in Eq. (23) is satisfied by all $x$ values.

$$G(x) \leq B$$  \hspace{1cm} (23)

Then a penalty of zero is considered. Otherwise, the penalty given in Eq. (24) is added to the objective functions, where $N$ is a sufficiently large number [38].

$$Penalty(x) = N \times \text{Max} \left( \frac{G(x)}{B} - 1, 0 \right)$$ \hspace{1cm} (24)

In other words, the cost function $R(x)$ corresponding to a solution $x$ is given by

$$R(x)\begin{cases} R(x); & x \in \text{Feasible region} \\ R(x) + \text{Penalty}(x); & \text{otherwise} \end{cases}$$ \hspace{1cm} (25)

As a result, if the fitness value obtained by a new solution is better than the worst case in the HM, the worst harmony vector is replaced by the new solution vector.

4.2. Particle swarm optimization (PSO)

The important steps of the PSO algorithm are the initialization and the position and velocity updates of the particles. The structure of the proposed PSO to find a near-optimum solution of the inventory control problem in (19) is described as follows.

4.2.1. Generation and initialization of the positions and velocities of the particles

Assuming a $d$-dimensional search space, PSO is first initialized by a group of random particles (solutions) and then searches for optima by updating generations. To randomly generate $Pop$ particles (solutions), let $x_{k}^j = \{x_{k,1}^j, x_{k,2}^j, ..., x_{k,d}^j \}$ and $v_{k}^j = \{v_{k,1}^j, v_{k,2}^j, ..., v_{k,d}^j \}$,
respectively, be the position and the velocity of particle $i$ in time $k$, in the $d$-dimensional search space. Then, Equations (26) and (27) are applied to initial particles, in which $x_{\text{min}}$ and $x_{\text{max}}$ are the lower and the upper bounds of the design variables, and $R\text{AND}$ is a random number between 0 and 1.

$$x_i^0 = x_{\text{min}} + R\text{AND} (x_{\text{max}} - x_{\text{min}})$$  \hfill (26)\hfill

$$v_i^0 = x_{\text{min}} + R\text{AND} (x_{\text{max}} - x_{\text{min}})$$  \hfill (27)\hfill

An important note for the generation and initializing step is that solutions must be feasible and satisfy the constraints. As a result, if a solution vector does not satisfy a constraint, the related vector solution is penalized by a large penalty on its fitness given in (25).

4.2.2. Selecting the best position and velocity vectors

For every particle, denote the best solution (fitness) achieved thus far as

$$\text{pBest}^i_k = \{pBest^i_{k,1}, pBest^i_{k,2}, \ldots, pBest^i_{k,d}\}$$  \hfill (28)\hfill

$$\text{gBest}^i_k = \{gBest^i_{k,1}, gBest^i_{k,2}, \ldots, gBest^i_{k,d}\}$$  \hfill (29)\hfill

where Eq. (28) shows the best position already found by particle $i$ until time $k$, and Eq. (29) is the best position already found by a neighbor until $k$.

4.2.3. Velocity and position updating

The new velocities and the positions of the particles for the next fitness evaluation are calculated using the following two equations

$$v_{i,k+1,d} = w \cdot v_{i,k,d} + C_1 \cdot r_1 \cdot (pBest_{i,k,d} - x_{i,k,d}) + C_2 \cdot r_2 \cdot (gBest_{i,k,d} - x_{i,k,d})$$  \hfill (30)\hfill

$$x_{i,k+1,d} = x_{i,k,d} + v_{i,k+1,d}$$  \hfill (31)\hfill
where $r_1$ and $r_2$ are uniform random numbers between 0 and 1. Coefficients $C_1$ and $C_2$ are given acceleration constants towards $p\text{Best}$ and $g\text{Best}$, respectively, and $w$ is the inertia weight. Introducing a linearly decreasing inertia weight into the original PSO significantly improves its performance through the parameter study of inertia weight [23, 30]. The linear distribution of the inertia weight is expressed as in [23]:

$$w = w_{\text{max}} - \frac{w_{\text{max}} - w_{\text{min}}}{Gen} \cdot \text{count}$$

(32)

where $Gen$ is the maximum number of iterations, and $\text{count}$ is the current number of iterations. Eq. (32) presents how the inertia weight is updated, considering $w_{\text{max}}$ and $w_{\text{min}}$ are the initial and final weights, respectively. The related results of the two parameters $w_{\text{max}} = 0.9$ and $w_{\text{min}} = 0.4$ were investigated by Shi and Eberhart [30] and Naka et al. [23].

4.2.4. Stopping criterion

Various stopping criterion may be adopted such as:

- achieving a predetermined solution,
- obtaining no significant change in the average and standard deviations of the solution in several consecutive generations,
- stopping at a predetermined (CPU) time, or
- reaching a predetermined number of iterations.

As numerous works in the literature have employed the last type of stopping criterion (see for example [9]), in this research, the algorithm is repeated until the maximum number of iterations is reached. Furthermore, the number of iterations along with other parameters of the algorithms will be calibrated using both Taguchi and RSM methods.

In the next section, some numerical examples are provided to illustrate both the applications and the performances of the two solution methods.
5. Performance evaluation

In this section, simulation experiments are presented to evaluate the performances of the two solution algorithms. In the performance evaluation, a single objective form of the problem is first considered for both algorithms, where the objective is minimizing the total inventory cost \( Z_i \). The total storage space is considered a constraint in this case, where the required storage spaces for each product are expressed using fuzzy numbers. This constraint is shown in Eq. (33) in which \( \tilde{S} \) is the total available storage space, which is also a fuzzy number.

\[
\sum_{i=1}^{m} \sum_{t=1}^{T-1} (Q_{it} + Y_{it}) \tilde{S}_i \leq \tilde{S}
\]  

(33)

Then, a bi-objective inventory control problem is investigated in which the WLS method is used to generate a single-objective form of the two objectives \( Z_i \) and \( Z_j \).

The parameters of the two solution methods are calibrated using both the Taguchi and RSM methods, where number of populations \( \text{Pop} \), the maximum number of generations \( \text{MaxGen} \), \( C_j \), and \( C_j \) are PSO parameters, and HMS, HMCR, PAR and number of generations \( \text{NG} \) are HSA parameters.

To specify the optimal levels, the Taguchi method establishes the relative significance of individual factors in terms of their main effects on the objective function. The Taguchi method generates a transformation of the repetition data to another value called the measure of variation. The transformation is the signal-to-noise \( (S/N) \) ratio, which explains why this type of parameter design is called robust design [27]. Here, the term “signal” denotes the desirable value (mean response variable), and “noise” indicates the undesirable value (standard deviation). Therefore, the \( S/N \) ratio indicates the amount of variation in the response variable. The aim is to maximize the \( S/N \) ratio. The Taguchi method categorizes objective functions into three groups: the smaller-is-better, the larger-is-better, and nominal-
is-best. In this research, a three-level orthogonal array is used for an objective function of a minimization type. Hence, the smaller-is-the-better objective function with its corresponding $S/N$ ratio given in Eq. (34) is suitable, where $\log$ is the logarithm function [21].

$$S/N \text{ ratio} = -10 \log(\text{Objective function})^2$$  \hfill (34)

As an alternative to tune the parameters, RSM is employed in this research as well. RSM is a collection of mathematical and statistical techniques useful for the modeling and analysis of problems in which a response of interest is influenced by several variables, and the objective is to optimize this response [18]. In most RSM problems, it is not easy to determine the relationship between the independent variables (the parameters) and the response variable (the solution obtained) in advance. To develop a proper approximation of the response variable, one usually starts with a low-order polynomial in some small region. If the response can appropriately be approximated by a linear function of independent variables, then the approximating function is a first-order model [42]. If there is a curvature in the response surface, then a higher degree polynomial that is called the second-order model should be utilized [18].

In this paper, the Taguchi method is used to tune the four parameters of PSO and HSA, each at three levels with nine observations represented by L9. In the RSM application, the codes "-1", "0", and "+1" are usually used for the low, medium (center), and high levels of the independent variables, respectively. To find the optimum levels of the parameters, a fractional factorial design of $2^{4-1}$ with 4 central points is selected to perform the experiments. Tables 1 and 2 show the parameter levels of the proposed PSO and HSA, respectively. The optimum values of the parameters of the two proposed algorithms using the Taguchi method and RSM are shown in Table 3.

Please insert Table 1 about here
The decision variables in the inventory problem are \( Q, Y, M \) and \( b \). However, determination of the optimal number of boxes for each product in each time period (i.e., \( M \)) automatically results in the determination of the other decision variables. Thus, based on Eq. (17), the values of \( M \) are randomly generated in \([0, ub]\). In addition, 15 problems of different sizes are used to compare the performances of the parameter-tuned algorithms. Tables 4 and 5 contain general data on these problems along with near optimal solutions in the single and bi-objective versions. The number of products in these tables is a variable in the range 1 to 15 where this range for the number of periods is between 3 and 15. Moreover, Tables 6 and 7 depict the general data of problem No. 5, where the products in the first two price break points use an AUD policy and the others use an IQD policy. In addition, the required storage space and the discount rate for all of the products are assumed to be trapezoidal fuzzy numbers. Fig 4 shows a trapezoidal fuzzy number \( \tilde{u} \) defined as \( \tilde{U} = (u_1, u_2, u_3, u_4) \), where its membership function is given by

\[
\mu_u = \begin{cases} 
0 & 0 \leq X \leq u_1 \\
\frac{X - u_1}{u_2 - u_1} & u_1 \leq X \leq u_2 \\
1 & u_2 \leq X \leq u_3 \\
\frac{u_4 - X}{u_4 - u_3} & u_3 \leq X \leq u_4 \\
0 & u_4 \leq X 
\end{cases} \quad (35)
\]

In Eq. (35), \( X \) is a fuzzy variable. Moreover, the centroid defuzzification method is employed to transform the fuzzy trapezoidal numbers to crisp values.
To combine the two objectives into one, the weights in the WLS method are chosen to be \( w_1 = w_2 = 0.5 \). Furthermore, the upper bound for the available number of boxes, i.e., \( ub \), is 550 for all products. The near optimal number of boxes based on the two algorithms under the Taguchi and RSM methods for single objective version of problem No. 5 are shown in Figs. 5 and 6, respectively. Moreover, the convergence paths of the best result of the two algorithms are presented in Figs. 7 to 10 for the single-objective version. Figs. 11 and 12 depict the optimal number of boxes obtained by the two algorithms under the Taguchi and RSM methods for the bi-objective version of problem No. 5. In this case, the convergence paths of the best result of the two algorithms are given in Figs. 13 to 16.

Figures 7 to 16 show how the proposed algorithms perform to reach the best solution. In these figures, the state of solutions obtained in each iteration under the Taguchi and RSM methods for problem No. 5 are shown. Furthermore, the trend of convergence declines sharper in the earlier iterations compared with iterations that come later.

Although the results in Tables 4 and 5 show better performances of HSA in terms of the objective function for both single and bi-objective versions of the problem, a t-test at confidence level of 95% indicates no significant difference on the average. To better demonstrate this, the trends of the RSM and Taguchi objective functions for single-objective and bi-objective versions are shown in Figs. 17 and 18. These trends demonstrate that the proposed algorithms that are calibrated using the Taguchi method have better performances than the RSM-tuned performances. Figs. 19 and 20 show the box-plots of the objective
function values of the two parameter-tuned algorithms for single and bi-objective versions of the problem, respectively. These plots show that HSA has better performance than PSO under both the Taguchi and RSM methods for single and bi-objective versions.

Please insert Figures 17-20 about here

6. Conclusion and recommendations for future research

In this paper, both the single- and bi-objective versions of a multi-item multi-period inventory control problem with a budget constraint under all unit discount policies for some products and incremental quantity discount for others with fuzzy discount rates were investigated. The quantities of the products were assumed to be ordered in batch sizes of pre-specified boxes. For the single-objective version, the total inventory cost was minimized where the total available storage space was limited, and the required space for products was considered to be a fuzzy number. In the bi-objective version, in addition to minimization of the total inventory cost, the total required storage space was also minimized, whereas the WLS method was employed to generate a single-objective problem. In both cases, shortages were allowed in combinations of backorders and lost sales. As the derived mathematical formulations of the two problems were too complicated to be solved analytically, two meta-heuristic algorithms, harmony search and particle swarm optimization, were developed to find near optimum solutions of both problems. Both algorithms were calibrated using the Taguchi and RSM methods. Finally, some numerical examples were generated to investigate the performances of the algorithms. The results indicate that while both HSA and PSO were able to solve the single and bi-objective inventory control problem efficiently, HSA offered better performance than PSO under both the Taguchi and RSM for single and bi-objective versions.
Some recommendations for future works include expanding the model to be applicable in supply chain environments. Moreover, the demands may also be considered stochastic, and the problem can be investigated in the context of inflation and the time value of money.

Acknowledgements

This work has been financially supported by Ministry of High Education (MOHE) of Malaysia, grants number UM.C/HIR/MOHEENG/ 21-(D000021-16001) and Malaysian FRGS national grant (FP007/2013A). The author thanks the Bright Sparks unit (University of Malaya) for additional financial support.

References

[6]. Das K., Roy T., Maiti M. Multi-item inventory model with quantity-dependent inventory costs and demand-dependent unit cost under imprecise objective and restrictions: a geometric programming approach, Production Planning & Control, 11 (2000) 781-788.


[14]. Maiti A., Maiti M. Discounted multi-item inventory model via genetic algorithm with Roulette wheel selection, arithmetic crossover and uniform mutation


[38]. Yeniay O. Penalty function methods for constrained optimization with genetic algorithms, Mathematical and Computational Applications, 10 (2005) 45-56.


Table 1: The PSO parameter levels

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>1.5</td>
<td>2</td>
<td>2.5</td>
</tr>
<tr>
<td>$C_2$</td>
<td>1.5</td>
<td>2</td>
<td>2.5</td>
</tr>
<tr>
<td>$NP$</td>
<td>50</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>$MaxGen$</td>
<td>100</td>
<td>500</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 2: The HSA parameter levels

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HMS$</td>
<td>20</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td>$HMCR$</td>
<td>0.90</td>
<td>0.95</td>
<td>0.98</td>
</tr>
<tr>
<td>$PAR$</td>
<td>0.2</td>
<td>0.9</td>
<td>2</td>
</tr>
<tr>
<td>$NG$</td>
<td>200</td>
<td>500</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 3: Optimum values of the parameters of the two algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>Optimum value (Taguchi)</th>
<th>Optimum value (RSM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>$C_1$</td>
<td>1.5</td>
<td>2.5</td>
</tr>
<tr>
<td>PSO</td>
<td>$C_2$</td>
<td>2.5</td>
<td>2</td>
</tr>
<tr>
<td>PSO</td>
<td>$NP$</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>PSO</td>
<td>$MaxGen$</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>HSA</td>
<td>$HMS$</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>HSA</td>
<td>$HMCR$</td>
<td>0.98</td>
<td>0.95</td>
</tr>
<tr>
<td>HSA</td>
<td>$PAR$</td>
<td>0.9</td>
<td>2</td>
</tr>
<tr>
<td>HSA</td>
<td>$NG$</td>
<td>200</td>
<td>200</td>
</tr>
</tbody>
</table>
Table 4: Computational results comparing the performances of the two algorithms for the single-objective version

<table>
<thead>
<tr>
<th>Problem</th>
<th>$m$</th>
<th>$N$</th>
<th>$f$</th>
<th>$TB$</th>
<th>Taguchi</th>
<th>RSM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HSA</td>
<td>PSO</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>10000</td>
<td>40000</td>
<td>18510</td>
<td>18510</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>13000</td>
<td>85000</td>
<td>44622</td>
<td>44943</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>15000</td>
<td>10000</td>
<td>85830</td>
<td>86689</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>18000</td>
<td>11000</td>
<td>149910</td>
<td>151002</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4</td>
<td>22000</td>
<td>127000</td>
<td>248664</td>
<td>251022</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>4</td>
<td>25000</td>
<td>150000</td>
<td>283620</td>
<td>286510</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>4</td>
<td>29500</td>
<td>190000</td>
<td>325020</td>
<td>328230</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>4</td>
<td>32000</td>
<td>220000</td>
<td>340185</td>
<td>344825</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>5</td>
<td>37000</td>
<td>290000</td>
<td>402583</td>
<td>407235</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>6</td>
<td>40000</td>
<td>340000</td>
<td>414484</td>
<td>418645</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>8</td>
<td>45000</td>
<td>400000</td>
<td>432034</td>
<td>438720</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>10</td>
<td>53000</td>
<td>440000</td>
<td>457011</td>
<td>465982</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>15</td>
<td>70000</td>
<td>530000</td>
<td>510167</td>
<td>518472</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>10</td>
<td>83000</td>
<td>650000</td>
<td>534200</td>
<td>541093</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td>95000</td>
<td>810000</td>
<td>603661</td>
<td>613781</td>
</tr>
</tbody>
</table>

P-Value 0.949 0.965
T-Value -0.06 -0.04

Table 5: Computational results comparing the performances of the two algorithms for the bi-objective version

<table>
<thead>
<tr>
<th>Problem</th>
<th>$m$</th>
<th>$N$</th>
<th>$f$</th>
<th>$TB$</th>
<th>Taguchi</th>
<th>RSM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HSA</td>
<td>PSO</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>10000</td>
<td>40000</td>
<td>11342</td>
<td>11342</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>13000</td>
<td>85000</td>
<td>28992</td>
<td>29219</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>15000</td>
<td>10000</td>
<td>48689</td>
<td>49340</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>18000</td>
<td>110000</td>
<td>82874</td>
<td>84217</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4</td>
<td>22000</td>
<td>127000</td>
<td>171794</td>
<td>174339</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>4</td>
<td>25000</td>
<td>150000</td>
<td>190620</td>
<td>193807</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>4</td>
<td>29500</td>
<td>190000</td>
<td>214325</td>
<td>216539</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>4</td>
<td>32000</td>
<td>220000</td>
<td>233105</td>
<td>236871</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>5</td>
<td>37000</td>
<td>290000</td>
<td>292909</td>
<td>296410</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>6</td>
<td>40000</td>
<td>340000</td>
<td>327819</td>
<td>331063</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>8</td>
<td>45000</td>
<td>400000</td>
<td>342127</td>
<td>345732</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>10</td>
<td>53000</td>
<td>440000</td>
<td>359072</td>
<td>364289</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>15</td>
<td>70000</td>
<td>530000</td>
<td>421985</td>
<td>430328</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>10</td>
<td>83000</td>
<td>650000</td>
<td>442800</td>
<td>451020</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td>95000</td>
<td>810000</td>
<td>518945</td>
<td>525437</td>
</tr>
</tbody>
</table>

P-Value 0.953 0.958
T-Value -0.06 -0.05
Table 6: The general data for a problem with 5 items and 4 periods

<table>
<thead>
<tr>
<th>Product</th>
<th>( D_{i1} )</th>
<th>( D_{i2} )</th>
<th>( D_{i3} )</th>
<th>( \pi_{i1} )</th>
<th>( \pi_{i2} )</th>
<th>( \pi_{i3} )</th>
<th>( \hat{\pi}_{i1} )</th>
<th>( \hat{\pi}_{i2} )</th>
<th>( \hat{\pi}_{i3} )</th>
<th>( k_i )</th>
<th>( h_{i1} )</th>
<th>( h_{i2} )</th>
<th>( h_{i3} )</th>
<th>( O_{i1} )</th>
<th>( O_{i2} )</th>
<th>( O_{i3} )</th>
<th>( \alpha_i )</th>
<th>( S_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1400</td>
<td>1000</td>
<td>1500</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>20</td>
<td>14</td>
<td>17</td>
<td>0.5</td>
<td>(2,5,6,7)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1500</td>
<td>1100</td>
<td>1600</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>15</td>
<td>19</td>
<td>18</td>
<td>0.5</td>
<td>(4,5,7,9)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1000</td>
<td>1100</td>
<td>1300</td>
<td>24</td>
<td>23</td>
<td>22</td>
<td>8</td>
<td>12</td>
<td>11</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>25</td>
<td>20</td>
<td>16</td>
<td>0.8</td>
<td>(3,5,9,10)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1200</td>
<td>1300</td>
<td>1500</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>8</td>
<td>12</td>
<td>11</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>18</td>
<td>20</td>
<td>15</td>
<td>0.8</td>
<td>(2,3,7,8)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1300</td>
<td>1500</td>
<td>1700</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>10</td>
<td>12</td>
<td>12</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>16</td>
<td>13</td>
<td>17</td>
<td>0.6</td>
<td>(5,6,7,9)</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: The general data of the purchasing cost for a problem with 5 items and 4 periods

<table>
<thead>
<tr>
<th>Price break point</th>
<th>( q_{i1} )</th>
<th>( q_{i2} )</th>
<th>( q_{i3} )</th>
<th>( q_{i4} )</th>
<th>( \mu_{i1} )</th>
<th>( \mu_{i2} )</th>
<th>( \mu_{i3} )</th>
<th>( \mu_{i4} )</th>
<th>( P_{i1} )</th>
<th>( P_{i2} )</th>
<th>( P_{i3} )</th>
<th>( P_{i4} )</th>
<th>( P_{i5} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>([0,500))</td>
<td>([0,450))</td>
<td>([0,600))</td>
<td>([0,500))</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>([500,1000)) ([450,\infty))</td>
<td>([600,1000))</td>
<td>([500,1100))</td>
<td>([1000,2100))</td>
<td>([0.02,0.06,0.09,1))</td>
<td>([0.07,0.08,12,13))</td>
<td>([0.04,11,13,15))</td>
<td>3</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>([1000,\infty))</td>
<td>-</td>
<td>([1000,1500))</td>
<td>([1100,\infty))</td>
<td>([2100,\infty))</td>
<td>([0.04,0.09,1,13))</td>
<td>([1.1,12,18,2))</td>
<td>([1.1,19,21,24))</td>
<td>4</td>
<td>-</td>
<td>5</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>([1500,\infty))</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Fig. 1: The representation of some possible situations for the inventory of product \( i \)

\[
M = \begin{pmatrix}
M_{11} & M_{12} & \cdots & M_{1T-1} \\
M_{21} & M_{22} & \cdots & M_{2T-1} \\
\vdots & \vdots & \ddots & \vdots \\
M_{m1} & M_{m2} & \cdots & M_{mT-1}
\end{pmatrix}
\]

Fig. 2: A representation of the decision variable
For $i = 1 : m$

For $t = 1 : T - 1$

Generate $c$ between $(0,1)$;

If $c \leq P_{HMCR}$

$M_t \leftarrow M_t \in [M_1, M_2, \ldots, M_{HMS}]$

Else

$M'_t \leftarrow$ generate a new one within the range

End If

End For

End For

Fig. 3: Pseudo code for the choosing and selection processes of the HM algorithm

Fig. 4: A trapezoidal fuzzy number $\tilde{u}$

\[
M_{HSA} = \begin{bmatrix} 521 & 329 & 3 \\ 206 & 216 & 111 \\ 45 & 400 & 84 \\ 85 & 173 & 203 \\ 32 & 124 & 206 \end{bmatrix} \quad M_{PSO} = \begin{bmatrix} 310 & 514 & 412 \\ 187 & 116 & 189 \\ 112 & 310 & 331 \\ 27 & 184 & 164 \\ 146 & 18 & 238 \end{bmatrix}
\]

Fig. 5: The optimal number of boxes using the Taguchi method for single-objective problem No. 5
\[
\begin{bmatrix}
137 & 750 & 558 \\
83 & 136 & 175 \\
54 & 303 & 197 \\
106 & 109 & 229 \\
61 & 153 & 119
\end{bmatrix}
\]

\[
\begin{bmatrix}
425 & 383 & 127 \\
154 & 129 & 159 \\
61 & 487 & 15 \\
56 & 216 & 207 \\
41 & 119 & 240
\end{bmatrix}
\]

Fig. 6: The optimal number of boxes using the RSM method for single-objective problem No. 5

Fig. 7: The convergence path of the best result by HSA using the Taguchi method for single-objective problem No. 5
Fig. 8: The convergence path of the best result by PSO using the Taguchi method for single-objective problem No. 5

Fig. 9: The convergence path of the best result by HSA using the RSM method for single-objective problem No. 5
**Fig. 10:** The convergence path of the best result by PSO using the RSM method for single-objective problem No. 5

\[
\begin{bmatrix}
228 & 300 & 780 \\
60 & 229 & 217 \\
119 & 83 & 171 \\
37 & 197 & 153
\end{bmatrix}
\] \quad \begin{bmatrix}
523 & 374 & 228 \\
90 & 161 & 220 \\
73 & 276 & 28 \\
152 & 208 & 10
\end{bmatrix}

**Fig. 11:** The optimal number of boxes using the Taguchi method for multi-objective problem No. 5

\[
\begin{bmatrix}
270 & 532 & 282 \\
88 & 287 & 19 \\
175 & 72 & 349 \\
40 & 232 & 154 \\
81 & 145 & 240
\end{bmatrix}
\] \quad \begin{bmatrix}
120 & 421 & 733 \\
286 & 21 & 120 \\
15 & 289 & 333 \\
163 & 53 & 216 \\
56 & 165 & 109
\end{bmatrix}

**Fig. 12:** The optimal number of boxes using the RSM method for bi-objective problem No. 5
Fig. 13: The convergence path of the best result by HSA using the Taguchi method for bi-objective problem No. 5

Fig. 14: The convergence path of the best result by PSO algorithm using the Taguchi method for bi-objective problem No. 5
Fig. 15: The convergence path of the best result by HSA using the RSM method for bi-objective problem No. 5

Fig. 16: The convergence path of the best result by PSO algorithm using the RSM method for bi-objective problem No. 5
Fig. 17: The trend of objective functions by HSA and PSO using both the RSM and Taguchi methods for the single-objective version
Fig. 18: The trend of objective functions by HSA and PSO using both the RSM and Taguchi methods for the bi-objective version
Fig. 19: Box-plots of the objective-function values of the algorithms for the single-objective version
Fig. 20: Box-plots of the objective-function values of the algorithms for the bi-objective version