A Hybrid Method of Pareto, TOPSIS and Genetic Algorithm to Optimize Multi-Product Multi-Constraint Inventory Control Systems with Random Fuzzy Replenishments

Ata Allah Taleizadeh¹, Seyed Taghi Akhavan Niaki², and Mir-Bahador Aryanezhad³

Abstract

Multi-periodic inventory control problems are mainly studied employing one of two assumptions. The first is the continuous review, where depending on the inventory level orders can be placed at any time, and the other is the periodic review, where orders can be placed only at the beginning of each period. In this paper, we relax these assumptions and assume that the time-periods between two replenishments are random fuzzy variables. While in the model of the problem at hand the decision variables are of integer-type and there are space and service level constraints, for the shortages we consider a combination of back-order and lost-sales. We show the model of this problem to be an integer-nonlinear-programming type and in order to solve it, a hybrid method of Pareto, TOPSIS and Genetic Algorithm approach is used. At the end, a numerical example is given to demonstrate the applicability of the proposed methodology.

Keywords: Inventory Control; Replenishment; Fuzzy Random Replenishment; Integer Nonlinear Programming; Pareto, Genetic Algorithm, TOPSIS.

¹M.Sc., Department of Industrial Engineering, Iran University of Science and Technology, Email: ata.taleizadeh@gmail.com
²Corresponding author, Professor of Industrial Engineering, Sharif University of Technology, Email: Niaki@Sharif.edu
³Professor of Industrial Engineering, Iran University of Science and Technology, Email: Mirarya@iust.ac.ir
1. Introduction and Literature Review

In multi-periodic inventory control models, the continuous review and the periodic review are the major vastly used policies. However, the underlying assumptions of the proposed models restrict their correct usage and utilization in real-world environments. In continuous review policy, the user has the freedom to act at anytime and replenish orders based upon the available inventory level. While in the periodic review policy, the user is allowed to replenish the orders only in specific and predetermined times.

The multi-period inventory control problems have been investigated in depth in different research. Chiang [1] considered a periodic review model in which the period was partly long, discount was considered, and the costs associated with his modeling were the purchasing, holding, and fixed order costs. He employed a dynamic programming approach to model the problem. Mohebbi and Posner [2] investigated an inventory system with periodic review, multiple replenishment, and multi-level delivery. Chiang [3] analyzed a periodic review problem in two cases of back-order and lost sales and employed the (R,T) policy. Qu et al. [4] investigated a transportation model integrated with an inventory model with a periodic review policy. Eynan and Kropp [5] have propounded the assumption of stochastic demand and variant warehousing costs on a periodic review system; while assuming nonzero lead-time and safety stock.

For a continuous review inventory model, assuming stochastic lead-time, Mohebbi [6] considered demand as a compound Poisson random variable. Furthermore, Taleizadeh et al. [7] investigated a stochastic replenishment multi-product inventory model and proposed two models for two cases of uniform and exponential distribution of the time between two replenishments. They showed that the models were integer-nonlinear programming problems and proposed a Simulated Annealing algorithm to solve it.

In the literature of fuzzy demands, Hsieh [8] introduced two fuzzy inventory models with fuzzy parameters for crisp and fuzzy production quantities. Yao et al. [9] considered an
inventory problem without backorder where both the order and the total demand quantities were triangular fuzzy numbers. In order to maximize the average profit, Mandal and Roy [10] formulated a multi-item displayed inventory problem under shelf-space constraint in a fuzzy environment. They considered the demand rate of an item as a function of the displayed inventory level. In another research in this area, Liu [11] developed a solution method to derive the fuzzy profit of the inventory model where the demand quantity and the unit cost were fuzzy numbers.

For the literature on the multi-objective optimization of inventory models, Mandal et al. [12] developed a multi-item multi-objective inventory model with shortages and demand-dependent unit-cost along with storage space, number of orders and production cost restrictions. In this research, the cost parameters, the objective functions and constraints were imposed in fuzzy environment and the model was solved by a Geometric Programming method. Chen and Lee [13] developed a multi-product multi-stage multi-period scheduling model that was proposed to deal with multiple incommensurable goals for a multi-echelon supply chain network with uncertain market demands and product prices. In order to achieve a compensatory solution among all participants of the supply chain, they presented a two-phase fuzzy decision-making method. Wang and Liang [14] developed a fuzzy multi-objective linear programming model to solve the multi-product aggregate production planning decision problem in a fuzzy environment. Not only the proposed model attempted to minimize the total production, carrying, and backordering costs, but also minimized the rates of changes in labor levels considering inventory level, labor levels, capacity, warehouse space and the time value of money. Rong et al. [15] developed a single wholesaler and multi retailer mixture inventory distribution model for a single item involving controllable lead-time with backorder and lost sales. The retailers purchased their items from the wholesaler in lots at some intervals throughout the year to meet the customers’
demand. In this research, a mathematical analysis had been made for global Pareto-optimal solutions of the multi-objective optimization problem.

In summary, while there has been separate emphasis on the stochastic nature of the demands and lead-time, the real-world constraints of the systems have not been completely investigated simultaneously. Specifically, there is no research in which both the demands and lead-time are considered to be probabilistic. Furthermore, some constraints such as budget and space have been partially studied or the decision variable has been considered integer.

Three main specifications of the proposed model of this research that have led to its novelty are 1) the replenishment period length being random fuzzy variable, 2) the allowance of multi-products with multi-constraints, and 3) the fact that the decision variables are integer. By deploying these conditions simultaneously, the proposed model is different from the other models in the periodic review literature. Moreover, we propose a hybrid Genetic, Pareto, and TOPSIS algorithm to solve the model.

The rest of the paper is organized as follows. In section 2, a brief background required for this research is presented. In section 3, the problem along with its assumptions is defined. In section 4, we model the problem of section 3. To do this we first introduce the parameters and the variables of the problem. Then, a single product problem with random replenishment is modeled, and finally the multi-product problem with random fuzzy replenishment is formulated. In the fifth section of the paper, we introduce a hybrid method of Pareto, TOPSIS and GA to solve the model at hand and analyze it under special conditions. Incorporating a numerical example, the solution method is investigated in section 6. The conclusion and recommendations for future research are given in section 7.
2. A Brief Background

In this section, brief backgrounds on Fuzzy environments, Genetic Algorithm, Evolutionary Algorithms, and TOPSIS are given.

2.1. Some Definitions in Fuzzy Environment

In this paper, we adopt the concepts of the fuzzy variable, random fuzzy variable and possibility theory of fuzzy event as defined by Liu [16]:

2.1.1. Possibility

In order to present the axiomatic definition of possibility, it is necessary to assign to each event $A$, a number $\text{Pos} \{ A \}$ which indicates the possibility that $A$ will occur. Let $\Theta$ be a nonempty set representing the sample space, and $P[\Theta]$ the power set of $\Theta$. In order to ensure that $\text{Pos} \{ A \}$ has certain mathematical properties, we intuitively expect the possibility to have the following properties:

**Axiom 1:** $\text{Pos} \{ \Theta \} = 1$

**Axiom 2:** $\text{Pos} \{ \emptyset \} = 0$, where $\emptyset$ is an empty event.

**Axiom 3:** $\text{Pos} \{ \bigcup_i A_i \} = \sup \text{Pos} \{ A_i \}$ for any collection $\{ A_i \}$ in $P[\Theta]$.

**Axiom 4:** let $\Theta_i$ be nonempty sets on which $\text{Pos}_i \{ \bullet \}$ satisfy the first three axioms, $i = 1, 2, \ldots, n$, respectively, and $\Theta = \Theta_1 \times \Theta_2 \times \cdots \times \Theta_n$. Then for each $A \in P(\Theta)$

$$\text{Pos} \{ A \} = \sup_{(\Theta_1 \times \Theta_2 \times \cdots \times \Theta_n) \in \Theta} \text{Pos}_1 \{ \Theta_1 \} \wedge \text{Pos}_2 \{ \Theta_2 \} \wedge \cdots \wedge \text{Pos}_n \{ \Theta_n \}$$

(1)

The first three axioms were introduced to define a possibility measure and the fourth one was introduced to define the product possibility measure.

**Definition 1:** Let $\Theta$ be a nonempty set and $P[\Theta]$ the power set of $\Theta$. Then, $\text{Pos}$ is called a possibility measure if it satisfies the first three axioms.
**Definition 2:** Let $\theta$ be a nonempty set, and $P(\theta)$ the power set of $\theta$ and $Pos$ a possibility measure. Then the triplet $(\theta,P(\theta),Pos)$ is called possibility space.

**Definition 3:** A random fuzzy variable is a function from the possibility space $(\theta,P(\theta),Pos)$ to the set of random variables.

**Example 1.** Let $\xi \approx N(\rho,1)$, where $\rho$ is a fuzzy variable with membership function $\mu_\rho(x) = \left[1 - |x - 2|\right] \vee 0$. Then $\xi$ is a random fuzzy variable taking normally distributed variable $N(\rho,1)$ values.

### 2.2. Genetic Algorithm

The fundamental principal of genetic algorithms (GA) was first introduced by Holland [17]. Since then, many researchers have applied and expanded this concept in different fields of study. Genetic algorithm was inspired by the concept of survival of the fittest. In genetic algorithms, the optimal solution is the winner of the genetic game and any potential solution is assumed to be a creature that is determined by different parameters. These parameters are considered as genes of chromosomes that could be assumed to be binary strings. In this algorithm, the better chromosome is the one that is nearer to the optimal solution. In applied applications of genetic algorithms, populations of chromosomes are created randomly. The number of these populations is different in each problem. Some hints about choosing the proper number of population exist in different reports [18].

In Genetic algorithms, new candidates for the solution are generated with two mechanisms of crossover and mutation. However, in some cases (luckily not in the application of this research), the new generated chromosomes may not necessarily be feasible, and some modifications would need to be applied to ensure feasibility of the solution.
In the crossover operation part of the genetic patrimony of each parent is combined and then a random mutation is applied. If the new individual, called child or offspring, inherits good characteristics from his parents the probability of its survival increases. This process continues until a stopping criterion is satisfied. Then, the best offspring is chosen as a near optimum solution.

In a crossover operation, it is necessary to mate pairs of chromosomes to create offspring. There are several types of crossover operations: single-point, multi-point, uniform, cut & slice, half uniform, etc. In a single-point crossover, we break two chromosomes from one point randomly and exchange their broken parts, resulting in two chromosomes. The initial chromosomes are called "parents" and the chromosomes resulted from the exchange are called "offspring." Crossover operates on the parents chromosomes with the probability of $P_c$, meaning that with the probability of $P_c$ the crossover action will occur. If no crossover occurs, the offspring's chromosomes will be the very same as their parents.

Mutation is the second operation in a GA method for exploring new solutions. In mutation, we replace a gene with a randomly selected number within the boundaries of the parameter [19]. We create a random number $RN$ between (0,1) for each gene. If $RN$ is less than a predetermined mutation probability $P_m$, then the mutation occur in the gene. Otherwise, the mutation operation is not performed in that gene. After producing the new chromosomes by crossover and mutation operations, we need to evaluate them.

The last step in a GA method is to check if the algorithm has found a solution that is good enough to meet the user’s expectations. Stopping criterion is a set of conditions such that when satisfied a good solution is believed to be obtained. Different criteria used in literature are as follows: 1) Stopping the algorithm after a specific number of generations,
2) no improvement in the objective function, and 3) Reaching a specific value of the objective function.

### 2.3. Evolutionary Algorithms

From the several emergent research areas, Evolutionary Algorithms (EAs), due to their properties, have become increasingly popular in solving multi-objective optimization problems [20]. For the past 15 years or so, evolutionary multi-objective optimization (EMO) methodologies have adequately demonstrated their usefulness in finding a well-converged and well distributed set of near Pareto-optimal solutions [20 & 21]. Due to these extensive studies and with source codes available both commercially and freely, the EMO procedures have been popularly applied in various problem-solving tasks and have received a great deal of attention even by the classical multi-criteria optimization and decision-making communities.

There are several possible ways to classify Multi-Objective Evolutionary Algorithms (MOEAs). The following taxonomy is perhaps the most simple and is based on the type of selection mechanism adopted:

1. Aggregating Functions Approaches
2. Population-based Approaches
3. Pareto-based Approaches

Under the Pareto-based category, we consider MOEAs that incorporate the concept of Pareto optimality in their selection mechanism. A wide variety of Pareto based MOEAs have been proposed in the last few years. A review is available in [20]. The most common algorithms are MOGA (Multi Objective Genetic Algorithm) of Fonseca and Fleming [22], VEGA (Vector Evaluated Genetic Algorithm) of Schaffer and Grefenstette [23], NSGA (Non-Dominated Sorting Genetic Algorithm) of Srinivas and Deb [24], NSGA II of Deb et al. [25], PAES (Pareto Archived Evolution Strategy) of Knowles and
Corne [26], and SPEA (Strength Pareto Evolutionary Algorithm) of Zitzler and Thiele [27].

2.3.1. Pareto Dominance

A vector \( \bar{u} = (u_1, ..., u_k) \) is said to dominate \( \bar{v} = (v_1, ..., v_k) \) (denoted by \( \bar{u} \preceq \bar{v} \)) if and only if \( u \) is partially less than \( v \), i.e., \( \forall i \in \{1, ..., k\}, u_i \leq v_i \land \exists i \in \{1, ..., k\} : u_i < v_i \).

2.3.2. Pareto Optimality

A solution \( x \in \Omega \) is said to be Pareto optimal with respect to \( \Omega \) if and only if there is no \( x' \in \Omega \) for which \( \bar{v} = F(x') = (f_1(x'), ..., f_k(x')) \) dominates \( \bar{u} = F(x) = (f_1(x), ..., f_k(x)) \). The phrase "Pareto optimal" is taken to mean with respect to the entire decision variable space unless otherwise specified.

2.3.3. Pareto Optimal Set

For a given multi-objective problem \( F(x) \), the Pareto optimal set \( (P^*) \) is defined as:

\[
P^* := \{x \in \Omega | \neg \exists x' \in \Omega \text{ s.t. } F(x') \leq F(x) \}.
\]

2.3.4. Pareto Front

For a given \( MOP \ F(x) \) and Pareto optimal set \( P^* \), the Pareto front \( (PF^*) \) is defined as:

\[
PF^* := \{\bar{u} = F(x) = (f_1(x), ..., f_k(x)) | x \in P^* \}.
\]

The Pareto optimal solutions are the ones within the search space whose corresponding objective vector components cannot all be improved simultaneously. These solutions are also known as non-inferior, admissible, or efficient solutions, with the entire set represented by \( P^* \). Their corresponding vectors are known as non-dominated. Figure (1) (selecting a vector(s) from this vector set (the Pareto Front set \( PF^* \)) implicitly indicates acceptable Pareto optimal solutions (genotypes). These are the set of all solutions whose vectors are non-dominated; these solutions are classified based on their phenotypical
expression. Their expression (the non-dominated vectors), when plotted in criterion (phenotype) space, is known as the Pareto.

![Figure (1): Illustration of the concept of Pareto optimality](image)

2.3.5. Basic Operation of a MOEA

Generally speaking, a MOEA is an extension on an EA in which two main issues are considered:

- How to select individuals such that non-dominated solutions are preferred over those which are dominated.

- How to maintain diversity as to be able to maintain in the population as many elements of the Pareto optimal set as possible.

Regarding selection, most current MOEAs use some form of Pareto ranking. This approach, which was originally proposed by Goldberg [28], sorts the population of an EA based on Pareto dominance, such that all non-dominated individuals are assigned the same rank (or importance). The idea is that all non-dominated individuals get the same probability to reproduce and that such probability is higher than the one corresponding to individuals which are dominated. Although conceptually simple, several possible ways exist to implement a MOEA using Pareto ranking [20 & 21].
The issue of how to maintain diversity in an EA has been addressed by an extensive number of researchers [21 & 29]. The proposed approaches include fitness sharing and niching [30], clustering, and geographically-based schemes to distribute solutions [26]. In addition, some researchers have adopted a mating restriction scheme [31]. More recently, the use of relaxed forms of Pareto dominance has been adopted as a mechanism to encourage more exploration and, therefore, to provide more diversity. From these mechanisms, $\epsilon$-dominance has become increasingly popular, not only because of its effectiveness, but also because of its sound theoretical foundation [32].

3. Problem Definition

Consider a periodic inventory control model for one provider in which the times required to order each of several available products are stochastic in nature. Let the time-periods between two product-replenishments be random fuzzy variables; the demands of the products to be constant and in case of shortage, a fraction is considered back-order and the rest as lost-sale. The costs associated with the inventory control system are holding, back-order, lost-sales, and purchase costs. Furthermore, the warehouse space along with the service level of each product are considered as constraints of the problem, and the decision variables are integer digits. We need to identify the inventory levels in each cycle such that the expected profit is maximized. In short, the assumptions involved in the problem are:

1. The time-periods between replenishments are random fuzzy variables.
2. There are several products.
3. There are service level and space constraints.
4. The decision variables are integer.
5. There are holding, shortage, and purchase costs.
6. A fraction of a shortage is back-ordered.
7. Only one provider exists.

8. The demands are constant.

9. All of the purchased product will be sold.

10. Lead time is zero.

4. Modeling

For the problem at hand, since the time-periods between two replenishments are random fuzzy variables, in order to maximize the expected profit of the planning horizon we need to consider only one period. Furthermore, since we assumed that the costs associated with the inventory control system are holding and shortage (back-order and lost-sale), we need to calculate the expected inventory level and the expected required storage space in each period. Before doing this, let us define the parameters and the variables of the model.

4.1. The Parameters and the Variables of the Model

For \( i = 1, 2, ..., n \), let us define the parameters and the variables of the model as

- \( R_i \): The inventory level of the \( i^{th} \) product at the start of a cycle.
- \( T_i \): A random variable denoting the time-period between two replenishments (cycle length) of the \( i^{th} \) product.
- \( T_{\text{Max}} \): The upper interval limit of a probability distribution for \( T_i \).
- \( T_{\text{Min}} \): The lower interval limit of a probability distribution for \( T_i \).
- \( f_{T_i}(t_i) \): The Probability density functions of \( T_i \) with fuzzy parameter.
- \( \tilde{\theta}_i \): Fuzzy parameter of the probability density functions of \( T_i \) for the \( i^{th} \) product.
- \( h_i \): The holding cost per unit inventory of the \( i^{th} \) product in each period.
- \( \pi_i \): The back-order cost per unit demand of the \( i^{th} \) product.
$W_i$: The purchasing cost per unit of the $i^{th}$ product.

$SL_i$: The lower limit of the service level for the $i^{th}$ product.

$P_i$: The sale price per unit of the $i^{th}$ product.

$D_i$: The constant demand rate of the $i^{th}$ product.

$t_{Di}$: The time at which the inventory level of the $i^{th}$ product reaches zero.

$\beta_i$: The percentage of unsatisfied demands of the $i^{th}$ product that is back-ordered.

$I_i$: The expected amount of the $i^{th}$ product inventory per cycle multiplied by the cycle time.

$B_i$: The expected amount of the $i^{th}$ product back-order in each cycle.

$Q_i$: The expected amount of the $i^{th}$ product order in each cycle.

$f_i$: The required warehouse space per unit of the $i^{th}$ product.

$F$: Total available warehouse space.

$C_h$: The expected holding cost per cycle of the $i^{th}$ product.

$C_s$: The expected shortage cost in back-order state.

$C_p$: The expected purchase cost of the $i^{th}$ product.

$r_i$: The expected revenue obtained from sales of the $i^{th}$ product.

$Z_i$: The expected profit obtained in each cycle of the $i^{th}$ product.

For sake of simplicity, in section 4.2 we first consider a single-product problem. Then, we extend the modeling to the multi-product modeling in section 4.3. We introduce a pictorial representation of the single-product problem in section 4.2.
4.2. Inventory Diagram

According to Ertogal and Rahim [33] and considering the fact that the time-periods between replenishments are stochastic variables, two cases may occur. In the first case the time-period between replenishments is less than the amount of time required for the inventory level to reach zero (see Figure 2), and in the second case, it is greater (see Figure 3). Figure (4) depicts the shortages in both cases.

4.3. Single Product Model–Back Order and Lost Sales Cases with Random Replenishment

In this section, we first model the costs, the profit, and the constraint of a single-product inventory problem with constant demand where stochastic replenishments, back-order, and lost-sales are allowed. Then, we model the multi-product inventory problem with fuzzy demands in section 4.4.

Figure (2): Presenting the inventory cycle when $T_{Min} \leq T \leq T_D$.
4.3.1. Calculating the Costs and the Profit

In order to calculate the expected profit in each cycle, we need to evaluate all of the terms in equation (2) [33].

\[ Z_i = r_i - C_{p_i} - C_{h_i} - C_{h_t} = p_i Q_i - W_i Q_i - h_i I_i - \pi_i B_i \]  

(2)

Based on Figure (4), \( L_i, B_i, I_i, \) and \( Q_i \) are evaluated by the following equations:

\[ B_i = \beta_i \int_{t_0}^{T_{Max}} (D_i T_i - R_i) f_{ri}(t) dt_i \]  

(3)
\[ I_i = \int_{T_{min}}^{T_{max}} \left( R_i T_i - \frac{D_i T_i^2}{2} \right) f_{T_i}(t_i) dt_i + \int_{t_N}^{T_{max}} \left( \frac{R_i^2}{2D_i} \right) f_{T_i}(t_i) dt_i \]  \hspace{1cm} (4)

\[ Q = \int_{T_{min}}^{T_{max}} (D_i T_i) f_{T_i}(t_i) dt_i + \int_{t_N}^{T_{max}} \left( R_i + \beta (D_i T_i - R_i) \right) f_{T_i}(t_i) dt_i \]  \hspace{1cm} (5)

4.3.2. Presenting the Constraints

As the total available warehouse space is \( F \), the space required for each unit of the \( i^{th} \) product is \( f_i \), and the inventory level of the \( i^{th} \) product is \( R_i \), the space constraint will be

\[ f_i R_i \leq F \]  \hspace{1cm} (6)

Since the shortages only occur when the cycle time is more than \( t_{D_i} \) and that the lower limit for the service level is \( SL_i \), then

\[ P( T_i > t_{D_i} ) = \int_{T_{min}}^{T_{max}} f_{T_i}(t_i) dt_i \leq 1 - SL_i \]  \hspace{1cm} (7)

In short, the complete mathematical model of the single product inventory is:

\[ \text{Max} \quad Z = (P_i - W_i) \left[ \int_{T_{min}}^{T_{max}} (D_i T_i) f_{T_i}(t_i) dt_i + \int_{t_N}^{T_{max}} \left( R_i + \beta (D_i T_i - R_i) \right) f_{T_i}(t_i) dt_i \right] \]

\[ -h_i \left[ \int_{T_{min}}^{T_{max}} \left( R_i T_i - \frac{D_i T_i^2}{2} \right) f_{T_i}(t_i) dt_i + \int_{t_N}^{T_{max}} \frac{R_i^2}{2D_i} f_{T_i}(t_i) dt_i \right] \]

\[ -\pi_i \beta_i \left[ \int_{T_{min}}^{T_{max}} \left( D_i T_i - R_i \right) f_{T_i}(t_i) dt_i \right] \]

s.t.:

\[ f_i R_i \leq F \]

\[ \int_{t_N}^{T_{max}} f_{T_i}(t_i) dt_i \leq 1 - SL_i \]

\[ R_i \geq 0, \text{Integer} \]  \hspace{1cm} (8)
In the next section, we extend the single-product model in (8) to a multi-product model
with random fuzzy replenishment.

4.4. Multi-Product Model-Back Order & Lost Sales with Random Fuzzy Replenishment

The single-product inventory model of section 4.3 can be easily extended to a multiple-
product model with random fuzzy replenishments as follows:

\[
\text{Max } Z(R_i, \tilde{\theta}) = \sum_{i=1}^{n} \left[ (P_i - W_i)Q_i - h_iI_i - \pi_iB_i - (P_i - W_i)L_i \right]
\]

\[
= \sum_{i=1}^{n} \left[ (P_i - W_i) \left[ \int_{D_i}^{R_i} (D_i T_i) f_i(t_i, \tilde{\theta}) dt_i + \int_{D_i}^{\min} (R_i + \beta_i(D_i T_i - R_i)) f_i(t_i, \tilde{\theta}) dt_i \right] \right]
\]

\[
- \sum_{i=1}^{n} h_i \left[ \int_{D_i}^{\min} \left( R_i T_i - \frac{D_i^2}{2} \right) f_i(t_i, \tilde{\theta}) dt_i + \int_{D_i}^{\min} \frac{R_i^2}{2D_i} f_i(t_i, \tilde{\theta}) dt_i \right]
\]

\[
- \sum_{i=1}^{n} \pi_i \beta_i \left[ \int_{\min}^{\max} (D_i T_i - R_i) f_i(t_i, \tilde{\theta}) dt_i \right]
\]

s.t.:

\[
\sum_{i=1}^{n} f_i R_i \leq F
\]

\[
\int_{D_i}^{\max} f_i(t_i) dt_i \leq 1 - SL_i \quad \forall \ i = 1, 2, \ldots, n
\]

\[
R_i \geq 0, \text{Integer} \quad \forall \ i = 1, 2, \ldots, n
\] (9)

In what follows, we consider exponential probability density functions for \( T_i \).

4.4.1. The Model with an Exponential Distribution for \( T_i \)

If \( T_i \) follows an exponential distribution with fuzzy parameters \( \tilde{\lambda}_i \), then the probability
density function of \( T_i \) is \( f_i(t_i) = \tilde{\lambda}_i e^{-\tilde{\lambda}_i t_i} \), in which \( \tilde{\lambda}_i \) is a triangular fuzzy variable and \( T_{\min} = 0 \) and \( T_{\max} = +\infty \). In this case, the model becomes:
\[
\begin{align*}
\text{Max } Z(R_j, \tilde{\lambda}_i) &= \sum_{i=1}^{n} \left[ \frac{1}{\tilde{\lambda}_i} \left( D_i (1 - \beta_i) (W_i - P_i) - \pi_i \beta_i D_i \right) e^{-\frac{R_i}{\tilde{\lambda}_i}} ight]^{\frac{1}{\lambda_i}} \\
&+ \frac{1}{\tilde{\lambda}_i} \left[ D_i (P_i - W_i) - h_i R_i \right] + \frac{h_i D_i}{\tilde{\lambda}_i^2} \left( 1 - e^{-\frac{R_i}{\tilde{\lambda}_i}} \right)
\end{align*}
\]

subject to:

\[
\begin{align*}
\sum_{i=1}^{n} f_i R_i &\leq F \\
\left( \frac{R_i}{D_i} \right) \tilde{\lambda}_i &\leq 1 - SL_i \quad \forall \, i = 1, 2, \ldots, n \\
R_i &\geq 0, \text{Integer} \quad \forall \, i = 1, 2, \ldots, n
\end{align*}
\]

(10)

In section 4.4.2 we utilize the defuzzified version of (10).

### 4.4.2. Defuzzification

As Klir and Yoan [34] showed, a fuzzy mathematical model with fuzzy benefit vector \( C_1, C_2, \ldots, C_n \) and fuzzy coefficient matrix \( \tilde{\alpha}_{ij} \) given by expression (11) (all of the fuzzy variables are triangular):

\[
\begin{align*}
\text{Max } Z &= \sum_{i=1}^{n} \tilde{C}_i X_i \\
\text{s.t.}: \quad &\sum_{i=1}^{n} \tilde{a}_{ij} X_i \leq b_j \quad \forall j: \, j = 1, 2, \ldots, m \\
&X_i \geq 0 \\
\tilde{a}_{ij} &= (\tilde{a}_{ij1}, \tilde{a}_{ij2}, \tilde{a}_{ij3}), \quad \tilde{C}_i = (C_{i1} C_{i2} C_{i3})
\end{align*}
\]

(11)

can be defuzzified to the mathematical model in (12).

\[
\begin{align*}
\text{Min } Z_1 &= \sum_{i=1}^{n} (C_{i2} - C_{i1}) X_i \\
\text{Max } Z_2 &= \sum_{i=1}^{n} C_{i2} X_i \\
\text{Max } Z_3 &= \sum_{i=1}^{n} (C_{i3} - C_{i2}) X_i
\end{align*}
\]
s.t.: \[ \sum a_{ij}x_i \leq b_j \quad \forall j : j = 1, 2, \ldots, m \]

\[ \sum a_{ij}x_i \leq b_j \quad \forall j : j = 1, 2, \ldots, m \]

\[ \sum a_{ij}x_i \leq b_j \quad \forall j : j = 1, 2, \ldots, m \]

\[ X_i \geq 0 \]  \hspace{1cm} (12)

Taking advantages of (11) and (12), the model in (10) is defuzzified to the model in (13).

\[
\begin{align*}
\text{Min } Z_1 &= \sum_{i=1}^{n} \left[ \left( \frac{hD_i}{\lambda_i} + \frac{D_i(1-\beta)(P_i-W_i) + \pi_i \beta D_i}{\lambda_i} \right) - \left( \frac{hD_i}{\lambda_i} + \frac{D_i(1-\beta)(P_i-W_i) + \pi_i \beta D_i}{\lambda_i} \right) \right] \frac{\lambda_i}{\lambda_i} R_i \\
\text{Max } Z_2 &= \sum_{i=1}^{n} \left[ \left( \frac{hD_i}{\lambda_i} + \frac{D_i(1-\beta)(P_i-W_i) + \pi_i \beta D_i}{\lambda_i} \right) - \left( \frac{hD_i}{\lambda_i} + \frac{D_i(1-\beta)(P_i-W_i) + \pi_i \beta D_i}{\lambda_i} \right) \right] \frac{\lambda_i}{\lambda_i} R_i \\
\text{Max } Z_3 &= \sum_{i=1}^{n} \left[ \left( \frac{hD_i}{\lambda_i} + \frac{D_i(1-\beta)(P_i-W_i) + \pi_i \beta D_i}{\lambda_i} \right) - \left( \frac{hD_i}{\lambda_i} + \frac{D_i(1-\beta)(P_i-W_i) + \pi_i \beta D_i}{\lambda_i} \right) \right] \frac{\lambda_i}{\lambda_i} R_i \\
\text{s.t.:} \hspace{1cm} & \sum_{i=1}^{n} f_i R_i \leq F \\
& -\left( \frac{\lambda_i}{D_i} \right) R_i \leq \ln(1 - SL_i) \quad \forall i : i = 1, 2, \ldots, n \\
& -\left( \frac{\lambda_i}{D_i} \right) R_i \leq \ln(1 - SL_i) \quad \forall i : i = 1, 2, \ldots, n \\
& -\left( \frac{\lambda_i}{D_i} \right) R_i \leq \ln(1 - SL_i) \quad \forall i : i = 1, 2, \ldots, n \\
& R_i \geq 0 \text{ integer} \quad \forall i : i = 1, 2, \ldots, n \\n\end{align*}
\]

(13)

Since the model in (13) is an integer-nonlinear-programming type with multiple objective functions, a hybrid intelligent search algorithm is proposed in section 5 to solve it.
5. A Hybrid Intelligent Algorithm

Reaching an analytical solution (if any) to the multi-objective integer-nonlinear-programming in (13) is difficult [19]. Accordingly, in this section we develop a hybrid intelligent algorithm of Genetic Algorithm, Pareto, and TOPSIS to solve it.

For each chromosome of the GA algorithm, three objective functions are required to be evaluated, and the Pareto selecting approach of section 5.1 is used to evaluate different combinations of the three evaluated objective functions. Finally, the TOPSIS methodology is used to rank different solutions for the Pareto selecting approach.

The chromosomes of the proposed GA are strings of the inventory levels of the products ($R_j$), and 10, 100, and 1000 are chosen as different population sizes. The single point crossover operation with $P_c$ (the probability of performing the crossover operation) values of 0.8, 0.85, and 0.9 are used in the proposed GA. Figure (5) depicts a sample single-point crossover operation of the proposed GA in which $R_j$ shows the chromosome containing the starting inventory levels of the products and the break point is randomly chosen as $M=7$. By this selection, the 7th and the 8th genes of the two chromosomes are exchanged.

\begin{align*}
    &R_j \begin{bmatrix}
        120 & 78 & 130 & 105 & 78 & 140 & 86 & 90 \\
    \end{bmatrix} \\
    \mapsto
    &M=7 \\
    &R_j \begin{bmatrix}
        105 & 75 & 130 & 100 & 66 & 140 & 91 & 84 \\
    \end{bmatrix} \\
    \mapsto
    &R_j \begin{bmatrix}
        120 & 78 & 130 & 105 & 78 & 140 & 91 & 84 \\
    \end{bmatrix} \\
    \mapsto
    &R_j \begin{bmatrix}
        105 & 75 & 130 & 100 & 66 & 140 & 86 & 90 \\
    \end{bmatrix}
\end{align*}

Figure (5): The single-point crossover operation
In the mutation operation, we assume that for a specific gene such as $a_j$ in a chromosome $R_j$ the generated random number is less than $P_m$ and hence the gene is selected for mutation. Then, we change the value of $a_j$ to the new value $a_j^*$ according to equations (14) and (15), randomly and with the same probability:

\[
a_j^* = a_j + (u_j - a_j) \times r \times (1 - \frac{i}{\text{max gen}}) \tag{14}
\]

\[
a_j^* = a_j - (a_j - l_j) \times r \times (1 - \frac{i}{\text{max gen}}) \tag{15}
\]

where $l_j$ and $u_j$ are the lower and upper limits of the specified gene, $r$ is a uniform random variable between 0 and 1, $i$ is the number of current generation, and $\text{max gen}$ is the maximum number of generations. Note that the value of $a_j$ is transferred to its right or left randomly by equations (14) and (15) respectively and $r$ is this percentage. Furthermore, $1 - \frac{i}{\text{max gen}}$ is an index with a value close to one in the first generation and close to zero in the last generation that makes large mutations in the early generations and almost no mutation in the last generations. In this paper, 0.076, 0.098, and 0.2 are employed as different values of the $P_m$ parameter, $l_j = 0$, $u_j = 800$, $i = 100$, $\text{max gen} = 500$. Figure (6) depicts a mutation operation in which $P_m$ is chosen 0.5.

<table>
<thead>
<tr>
<th>$R_j$</th>
<th>120</th>
<th>78</th>
<th>130</th>
<th>105</th>
<th>78</th>
<th>140</th>
<th>86</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RN$</td>
<td>0.573</td>
<td>0.131</td>
<td>0.485</td>
<td>0.684</td>
<td>0.743</td>
<td>0.972</td>
<td>0.371</td>
<td>0.824</td>
</tr>
</tbody>
</table>

![Figure (6): A sample of mutation operation](image.png)
For the fitness value evaluation, since the model in (13) is a multi-objective problem, for each chromosome three objective functions are required to be evaluated. The Pareto selecting approach of section 5.1 is used to evaluate different combinations of the three evaluated objective functions. In the GA, we stop when a predetermined number of consecutive generations is reached. The number of sequential generations depends on the specified problem and the expectations of the user.

5.1. Pareto Selecting Approach

The steps involved in the hybrid GA and Pareto algorithm used in this research are:

1. Setting the parameters $P_c$, $P_m$, and $N$.
2. Initializing the population randomly.
3. Evaluating Multi-objective fitness for each chromosome.
4. Sorting by dominated selecting best ranking individual for mating pool according to Pareto selecting approach.
5. Applying the crossover operation for each pair of chromosomes with probability $P_c$.
6. Applying mutation operation for each chromosome with probability $P_m$.
7. Replacing the current population by the resulting mating pool.
8. Evaluating the objective function.
9. If stopping criterion is met, then stop. Otherwise, go to step 5.

5.2. TOPSIS for Ranking

Having used Pareto and Genetic Algorithm to synthesize a range of solutions, a single solution must be selected by the decision maker. Ranking methods can be used to reduce the non-dominated solution set to a single design of the inventory system at hand. In this
paper, a ranking process called TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) is used for solving this kind of Multi-Attribute Decision Making (MADM) problem.

Wang and Liang [35] developed TOPSIS based upon the concept that the chosen (best) alternative should have the shortest distance from the positive-ideal solution and the longest distance from the negative-ideal solution. The reasons of using TOPSIS are its rational and comprehensible concept and the simplicity of its computations [36]. In this approach, by assuming that each attribute takes a monotonically increasing (or decreasing) utility; it is easy to locate both the ideal solution, which is composed of all best attribute values, and the negative ideal solution composed of all worst attribute values attainable. One approach is to take an alternative which has the (weighted) minimum Euclidean distance to the ideal solution in a geometrical sense. It is argued that this alternative should be farthest from the negative ideal solution at the same time. TOPSIS considers the distances to both the ideal and the negative ideal solutions simultaneously. A MADM problem with \( p \) alternatives each having \( q \) attributes can be mapped into a \( q \)-dimensional space with \( p \) points.

In this research, we treat the non-dominated solutions as alternatives and use the three objectives \([Z_1, Z_2, Z_3]\) of equation (13) as attributes. The data matrix \( A \) in Figure (7), generated from Pareto and genetic algorithm is used to perform the computation of TOPSIS.

\[
A = \begin{bmatrix}
Z_1 & Z_2 & Z_3 \\
A_1 & X_{11} & X_{12} & X_{13} \\
A_2 & X_{21} & X_{22} & X_{23} \\
\vdots & \vdots & \vdots & \vdots \\
A_p & X_{p1} & X_{p2} & X_{p3}
\end{bmatrix}
\]

Figure (7): Data Matrix for TOPSIS
In Figure (7) \(X_{ij}\) is the \(j^{th}\) attribute value (or objective function value) of alternative (or non-dominated solution) \(i\). For the detailed steps of TOPSIS, please refer to Wang and Liang [35].

In order to demonstrate the proposed Hybrid intelligent algorithm and evaluate its performance, we solve a numerical example used in Ertogal and Rahim [33] in the next section. However, since they modeled a single objective inventory problem without constraints, the numerical example is extended to contain the required information for the proposed constraint multi-objectives inventory model.

6. Numerical Example

Consider a multi-product inventory control problem with eight products and general data given in Table (1). Table (2) shows the parameters of the exponential distribution used for the time-period between two replenishments as a triangular fuzzy variable. The total available warehouse space is 15000. Table (3) shows different values of the parameters of the GA method. In this research all of the possible combinations of the parameters of GA \((P_c, P_m, N)\) methods are employed and using the \(\max(\max)\) criterion, the best combination of the parameters was selected.

Table (4) shows the generated results of Pareto and GA that are sorted by the TOPSIS approach. The first part of this table shows the inventory levels of the products at the start of a cycle. The second part contains the values of the three objective functions derived for different rows of the starting inventory levels given in the first part. Finally, the third part of Table (4) shows the TOPSIS scores and ranks. We note that the solution with the score of 0.9929 is ranked first.
The best combinations of the GA algorithms are shown in Tables (5). This combination is obtained by using 27 different combinations of the GA parameters shown in Table (3) and selecting the best based on the \textit{max} (max) criterion.

Furthermore, the convergence path of the objective functions values of the Hybrid method of Pareto and GA is shown in Figures (8). Each point in this figure shows the 3-coordinates value of the three objective functions. As it can be seen, the hybrid Pareto and GA method converges from right to left and riches the best combination of the three objective functions at the most left-top of the figure.

\section*{6. Conclusion and Recommendations for Future Research}

In this paper, a random fuzzy replenishment multi-product inventory model was investigated. Two mathematical models for two cases of Fuzzy and deterministic programming were developed in the form of multi-objective integer-nonlinear programming. A hybrid intelligent algorithm (Pareto and GA) was proposed to solve the multi-objective integer non-linear problems.

$\begin{array}{|c|cccccccc|}
\hline
\text{Product} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
h_i & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\
\pi_i & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\
P_i & 100 & 100 & 100 & 100 & 150 & 150 & 150 & 150 \\
\beta_i & 0.5 & 0.9 & 0.9 & 0.5 & 0.5 & 0.9 & 0.9 & 0.5 \\
f_i & 3 & 3 & 3 & 3 & 6 & 6 & 6 & 6 \\
W_i & 70 & 70 & 70 & 70 & 70 & 70 & 70 & 70 \\
D_i & 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\
SL_i & 0.5 & 0.6 & 0.6 & 0.5 & 0.5 & 0.6 & 0.6 & 0.5 \\
\hline
\end{array}$
Table (2): Data for the parameter of the Exponential distribution

<table>
<thead>
<tr>
<th>Product</th>
<th>( \lambda_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1/40, 1/30, 1/20)</td>
</tr>
<tr>
<td>2</td>
<td>(1/40, 1/30, 1/20)</td>
</tr>
<tr>
<td>3</td>
<td>(1/70, 1/60, 1/50)</td>
</tr>
<tr>
<td>4</td>
<td>(1/70, 1/60, 1/50)</td>
</tr>
<tr>
<td>5</td>
<td>(1/40, 1/30, 1/20)</td>
</tr>
<tr>
<td>6</td>
<td>(1/40, 1/30, 1/20)</td>
</tr>
<tr>
<td>7</td>
<td>(1/70, 1/60, 1/50)</td>
</tr>
<tr>
<td>8</td>
<td>(1/70, 1/60, 1/50)</td>
</tr>
</tbody>
</table>

Table (3): The parameters of the GA method

<table>
<thead>
<tr>
<th>( P_c )</th>
<th>( P_m )</th>
<th>( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.076</td>
<td>10</td>
</tr>
<tr>
<td>0.85</td>
<td>0.098</td>
<td>100</td>
</tr>
<tr>
<td>0.9</td>
<td>0.2</td>
<td>1000</td>
</tr>
</tbody>
</table>
Table (4): The ranked results of \( R_i \) based on TOPSIS

<table>
<thead>
<tr>
<th>Product</th>
<th>Objectives</th>
<th>TOPSIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>279</td>
<td>369</td>
<td>686</td>
</tr>
<tr>
<td>281</td>
<td>370</td>
<td>687</td>
</tr>
<tr>
<td>279</td>
<td>370</td>
<td>687</td>
</tr>
<tr>
<td>279</td>
<td>369</td>
<td>687</td>
</tr>
<tr>
<td>279</td>
<td>368</td>
<td>687</td>
</tr>
<tr>
<td>278</td>
<td>370</td>
<td>687</td>
</tr>
<tr>
<td>278</td>
<td>369</td>
<td>687</td>
</tr>
<tr>
<td>278</td>
<td>368</td>
<td>687</td>
</tr>
<tr>
<td>278</td>
<td>368</td>
<td>686</td>
</tr>
<tr>
<td>278</td>
<td>369</td>
<td>685</td>
</tr>
<tr>
<td>278</td>
<td>368</td>
<td>685</td>
</tr>
<tr>
<td>278</td>
<td>368</td>
<td>687</td>
</tr>
<tr>
<td>278</td>
<td>368</td>
<td>685</td>
</tr>
<tr>
<td>279</td>
<td>367</td>
<td>685</td>
</tr>
<tr>
<td>278</td>
<td>367</td>
<td>687</td>
</tr>
<tr>
<td>278</td>
<td>367</td>
<td>686</td>
</tr>
<tr>
<td>278</td>
<td>367</td>
<td>686</td>
</tr>
<tr>
<td>279</td>
<td>368</td>
<td>686</td>
</tr>
<tr>
<td>278</td>
<td>368</td>
<td>687</td>
</tr>
<tr>
<td>278</td>
<td>367</td>
<td>686</td>
</tr>
<tr>
<td>278</td>
<td>367</td>
<td>686</td>
</tr>
<tr>
<td>278</td>
<td>367</td>
<td>686</td>
</tr>
<tr>
<td>278</td>
<td>367</td>
<td>685</td>
</tr>
<tr>
<td>278</td>
<td>367</td>
<td>686</td>
</tr>
<tr>
<td>281</td>
<td>367</td>
<td>685</td>
</tr>
</tbody>
</table>
Table (5): The best combination of the GA parameters

<table>
<thead>
<tr>
<th>$P_c$</th>
<th>$P_m$</th>
<th>$N$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>0.098</td>
<td>100</td>
</tr>
</tbody>
</table>

Some recommendations for future works are:

1. In addition to a GA algorithm, some other meta-heuristic algorithms such as Tabu-Search, Ant-Colony Optimization, Simulated Annealing Optimization or Particle Swarm Optimization may be employed to solve the integer non-linear problem.

2. In addition to Pareto, other kinds of selecting approaches may be used to select non-dominate or dominate results.
3. Some other probability density functions such as uniform or normal may be considered for the time between replenishments.

4. The discount factor may be considered in the problem.

7. Acknowledgement

The authors would like to thank the referees for their valuable comments and suggestions that improved the presentation of this paper.

8. References


