Applied Soft Computing 43 (2016) 57–72

Contents lists available at ScienceDirect

Applied Soft Computing

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

A bi-objective inventory optimization model under inflation and discount using tuned Pareto-based algorithms: NSGA-II, NRGA, and MOPSO

Seyed Mohsen Mousavi [a,b,*], Javad Sadeghi [b], Seyed Taghi Akhavan Niaki [c], Madjid Tavana [d,e]

[a] Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia
[b] Young Researchers and Elite Club, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[c] Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran
[d] Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, PA 19141, USA
[e] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, D-33098 Paderborn, Germany

ARTICLE INFO

Article history:
Received 2 October 2014
Received in revised form 7 February 2016
Accepted 8 February 2016
Available online 17 February 2016

Keywords:
Multiproduct inventory control
Inflation
Discounts
Shortages
NSGA-II
NRGA
MOPSO
Taguchi method

ABSTRACT

This study presents a seasonal multi-product multi-period inventory control model with inventory costs obtained under inflation and all-unit discount policy. The products are delivered in boxes of known quantities and both backorder and lost-sale quantities are considered in case of shortage. The goal is to find a representative set of Pareto optimal solutions (including the ordering quantities) in different periods and to minimize both the total inventory cost (i.e. ordering, holding, shortage, and purchasing costs) and the total storage space, simultaneously. Three multi-objective optimization algorithms of non-dominated sorting genetic algorithm (NSGA-II), non-dominated ranked genetic algorithm (NRGA), and multi-objective particle swarm optimization (MOPSO) are proposed to solve the problem. The Taguchi approach with a novel metric (based on the coefficient of variation) is utilized to model the response variable and compare the performances of the algorithms. Three numerical examples are used to demonstrate the applicability and exhibit the efficacy of the procedures and algorithms. The results of statistical analyses show significant differences in the performance metrics for all three algorithms and in all three numerical examples.

1. Introduction

In inventory control problems, determining the ordering times and the order quantities of products are the two strategic decisions either to minimize total costs or to maximize total profits. In this regard, numerous research works were performed in the past decade. Some of these works are connected to the well-known economic order or production quantity models (EOQ and EPQ) to achieve maximum profit or minimum cost (see for example [1–7]).

Inventory problems dealing with seasonal and fashion products are usually modeled in a known number of periods. To name a few research works on multi-periodic inventory control problems, Ahmed et al. [8] investigated a multi-period single-item inventory problem with linear cost, where the objective function was a coherent risk measure. Sepehri [9] formulated an integrated flow network and expanded it to a multi-period multi-product inventory control problem with the possibility of holding inventories in a multi-stage multi-member cooperative supply chain. Zhang et al. [10] presented some convex stochastic programming models for multi-period inventory control problems where the market demand was random and order quantities needed to be decided before the demand was realized. Chio et al. [11] proposed a solution scheme for a periodic review multi-period inventory problem under a mean-variance (MV) framework.

In addition to managing multiple items, real-world inventory control systems have many limitations in warehouse space, budget, shortage, and the like. Zhou [12] developed a deterministic

* Corresponding author at: Department of Mechanical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia.
E-mail addresses: mousavi@siswa.um.edu.my (S.M. Mousavi), keeyarash@gmail.com (J. Sadeghi), niaki@sharif.edu (S.T.A. Niaki), tavana@lasalle.edu (M. Tavana).
URL: http://www.tavana.us/ (M. Tavana).

http://dx.doi.org/10.1016/j.asoc.2016.02.014
1568-4946/© 2016 Published by Elsevier B.V.

replenishment model with multiple warehouses (one an owned warehouse and others rented) possessing limited capacity, in which the replenishment rate was infinite and the demand rate was increasing at a decreasing rate with respect to time. Taleizadeh et al. [13] considered a multiproduct multi-constraint inventory control system with budget and space constraints where replenishment intervals were stochastic and the items were sold under discount.

Nowadays, due to high volume financial turnover between or inside companies, inflation is an important factor to model inventory costs. The effects of inflation on inventory control decisions have been considered in many works since 1975 when Buzacott [14] proposed an EOQ model with inflation subject to different types of pricing policies. Dey et al. [15] considered two- storage inventory problems with dynamic demand and interval valued lead-time over finite time horizon under inflation and time-value of money. Sarkar and Moon [16] proposed a production inventory model for stochastic demand with the effect of inflation where due to real-life constraints (labor problems, machine breakdown, etc.), a certain percentage of products are of imperfect quality. Recently, Mirzazadeh et al. [17] introduced an inventory model with stochastic inflation rate for multiple items with budget constraint.

In many real-world inventory problems, managers desire to optimize multiple objectives of different types simultaneously. As a result, many authors extended inventory models to involve multiple objectives. As an example, Mahapatra and Maiti [18] extended a multi-objective multi-item production/inventory system in which demand was a function of inventory level and selling price of the commodity. Maiti and Maiti [19] formulated an inventory control system of deteriorating multi-items under a single management with resource constraints under inflation and discounting in fuzzy environment. Zhang et al. [20] studied a two-stage, multi-item inventory system where stochastic demand occurs at stage 1, and nodes at stage 1 replenish their inventory from stage 2.

In order to solve multi-objective inventory models, Pareto dominance solutions have been proposed by several authors. While the amount of research in this area is limited, some works are Tsou [21] and Tsou et al. [22] who employed a multi-objective particle swarm optimization (MOPSO), Rezaei and Davoodi [23], and Ojha et al. [24] who applied a multi-objective genetic algorithm (MOGA) to find Pareto dominance solutions.

The non-dominated sorting genetic algorithm (NSGA-II), initially proposed by Deb et al. [25], is a multi-objective evolutionary algorithm (MOEA) that has been applied to find Pareto front solutions in different fields of studies. For example, Liao and Hsieh [26] and Bhattacharya and Bandyopadhyay [27] employed it in a facility location problem, Yuan and Quanfeng [28] and Kang et al. [29] used it in a scheduling problem, Amodeo et al. [30] applied it in an inventory control problem and Mousavi et al. [31] solved a multi-state redundancy allocation problem using NSGA-II and CE-NRGA. In addition, Sadeghi and Niaki [32] applied NSGA-II and NRGA to solve a multi-objective vendor-managed inventory problem with trapezoidal fuzzy demand. Non-dominated ranked genetic algorithm (NRGA) is another MOEA proposed by Al Jaddan et al. [33] to solve multi-objectives optimization models. While the implementation of NRGA is limited in the literature, Moradi et al.'s [34] work is the one that uses it to find solutions of a multi-objective assembly line sequencing model. They compared solutions obtained via NRGA against the ones obtained via total enumeration scheme in small problems and against the ones obtained by four other search heuristics in small, medium and large problems as well. Experimental results showed that NRGA was competitive with the investigated algorithms in terms of quality and diversity of solutions. Rahmani et al. [35] proposed a new mathematical model with nonlinear terms and integer variables that could not be solved efficiently for medium and large-sized problems. In their research, a method combining a new ranked-based roulette wheel selection algorithm with Pareto-based population ranking algorithm, named non-dominated ranking genetic algorithm (NRGA), was presented to find non-dominated solutions in a reasonable time. They compared solutions obtained via NRGA against solutions obtained via ε-constraint method in small-sized problems. Moreover, Kayvanfar et al. [36] used two meta-heuristic algorithms (NRGA and NSGA-II) for solving an economic lot scheduling problem regarding deteriorating items and allowable shortage using extended basic period approach under Power-of-Two policy. They showed that there was no significant difference between the performances of the two algorithms in small-sized problems. However, NRGA performed better that NSGA-II for medium and large-sized problems.

Since meta-heuristic algorithms are sensitive to their parameter settings, in order to improve the quality of solutions obtained the Taguchi method is usually applied. Some recent works that employed this method are in location-allocation problem [37], in scheduling problem [38–42], and in statistical quality control problem [43].

While in the past decade, a considerable amount of research works have been devoted to modeling, development, and solution procedures of the multi-objective inventory control problem, the main contribution of this paper is two-folds. First, the modeling area in which a novel bi-objective mathematical formulation for seasonal multi-product multi-period EOQ problem with inflation, discount, and shortage under limitations on budget, number of orders, and available number of transportation boxes is proposed. Although this type of inventory-control formulation that involves various features has many real-world applications, no research has been devoted to it yet. Second, the solving methodologies for the proposed optimization model in which three Pareto-based multi-objective meta-heuristic algorithms called NSGA-II, NRGA and MOPSO are proposed. Since meta-heuristic algorithms are sensitive to their parameters, a Taguchi procedure is employed to tune the parameters of both algorithms, where a novel metric that takes into account both the convergence and diversity is introduced as the response variable. Finally, several performance metrics are used to compare the efficiency of the proposed algorithms using various graphical and statistical methods.

To be more specific, the present paper involves a multi-product multi-period EOQ problem with shortages in which the planning horizon is finite and the demand rates of the items are different in various periods. The problem is modeled under realistic constraints of budget and truck-space, where the vendor sells the products in boxes of known number of items under all-unit discount, where inflation plays an important role affecting the inventory costs. Further, three binary variables are used to model the shortage, the ordering, and the purchasing costs; a novel approach that distinguishes this work with others. The objective is to find the optimum order quantities of the products in different periods such that both the total inventory cost (including ordering, holding, shortage, and purchasing) and the required storage space are minimized simultaneously. The proposed bi-objective inventory model can be used in situations in which production managers desire to produce a new product that requires an extended storage space to locate their purchased items with limited budget. They must consider inflation, discounts, and other mentioned realistic limitations.

The remainder of this paper is organized as follows: In Section 2, the notations and assumptions are presented. In Section 3, the problem is formulated. Solution algorithms are introduced in Section 4. Section 5 provides experimental results along with discussions. Conclusion and future works are given in Section 6.

2. Notations and assumptions

To develop the mathematical model of the inventory control problem at hand, the indices, parameters, and decision variables adopted in this article are as follows:

Indices

i: An index for a product; $i = 1, 2, \ldots, m$

j: An index for a replenishment cycle; $j = 0, 1, \ldots, N$

k: An index for a price break point; $k = 1, 2, \ldots, K$

Parameters

N: Number of replenishment cycles in the planning horizon

m: Number of products

K: Number of price break points

$A_{i,j}$: Replenishment cost per order of product i in period j

P_i: Purchasing cost per unit of the ith product

$P_{i,k}$: Purchasing cost per unit of the ith product at the kth price break point

h_i: Inventory holding cost per unit per unit time of the ith product

$\pi_{i,j}$: Backorder cost per unit of the ith product in period j

$\pi'_{i,j}$: Lost sale cost per unit of the ith product in period j

C: Total available budget

$I_{i,j}(t)$: Inventory position of ith product at time t in period j

$D_{i,j}$: Demand of ith product in period j

B_i: The fixed batch size of ith product

$q_{i,k}$: kth price discount point for ith product ($q_{i1} = 0$)

S_i: Required storage space per unit of ith product

$T_{i,j}$: Total time elapsed up to and including the jth replenishment cycle of the ith product

$T'_{i,j}$: The time in period j at which the inventory of item i reaches zero

$U_{i,j,k}$: A binary variable; set equal 1 if item i is purchased at price break point k in period j, and 0 otherwise

$W_{i,j}$: A binary variable; set equal 1 if a purchase of product i is made in period j, and 0 otherwise

$L_{i,j}$: A binary variable; set equal 1 if a shortage occurs for product i in period j, and 0 otherwise

β_i: Percentage of unsatisfied demand of ith product that is back ordered

M_1: An upper bound for order quantity of ith product in period j

M_2: An upper bound for the available number of boxes

Decision variables

$V_{i,j}$: Number of the boxes for ith product order in period j

$b_{i,j}$: Shortage quantity for ith product in period j ($b_{i,0} = 0$)

$Q_{i,j}$: Ordering quantity of ith product in period j

$X_{i,j}$: Beginning positive inventory of the ith product in period j (in $j = 0$, the beginning inventory of all items is zero)

The assumptions involved in the inventory problem at hand are:

1. Replenishment is instantaneous.
2. Demand rates of all products are independent from each other and are constant in a period.
3. In each period, at most one order can be placed for a product.
4. All products are delivered in special boxes, i.e. the order quantity of the products is a multiple of a fixed-sized batch.
5. When a shortage occurs, a fraction is considered backorder and a fraction lost sale.
6. The initial inventory level of all products is zero (i.e. $X_{i,1} = 0$).
7. The order quantity of each product in each period is at least equal to the shortage quantity of the product in the previous period (i.e. $Q_{i,j+1} \geq b_{i,j}$ for $i = 1, 2, \ldots, m$; $j = 0, 1, \ldots, N-1$, where $b_{i,0} = 0$).
8. Planning horizon is finite and known. In the planning horizon, there are N periods of equal length.
9. The total available budget to purchase products is limited.
10. The order quantity of a product in a certain period is limited.
11. The number of available boxes to deliver products in different periods is limited.
12. The holding cost of an item is assumed independent of its required storage space.

3. Mathematical formulation

A graphical representation of the inventory control problem at hand with five replenishment cycles for product i is shown in Fig. 1, where some possible inventory scenarios are given. Based on Fig. 1, the inventory planning starts in period T_0 and ends in period T_5, where shortages occur in some periods in between. Moreover, an order that is at least equal to the shortage quantity of the previous period is placed in a period.

The total inventory cost consists of ordering, holding, shortage, and purchasing costs that are modeled as follows. Since at most one order can be placed for a product in a period, in order to model the ordering cost, a binary variable $W_{i,j}$ is used where it is 1 if a purchase of a unit of product i is made in period j, otherwise 0. As a result, the total ordering cost when inflation is not present is obtained as $\sum_{i=1}^{m}\sum_{j=1}^{N} A_{i,j}W_{i,j}$. In case of an existing inflation with a rate of f, the total ordering cost (A) under continuous compounding policy becomes [15]

$$A = \sum_{i=1}^{m}\sum_{j=1}^{N} A_{i,j}W_{i,j}\mathrm{e}^{-fT_{i,j}} \tag{1}$$

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 $T_{i,j-1} < t < T_{i,j}(1-L_{i,j}) + T'_{i,j}L_{i,j}$ the holding cost under inflation (H) is

$$H = h_i \int_{T_{j-1}}^{T_j(1-L_{i,j})+T'_jL_{i,j}} I_i(t)\,\mathrm{e}^{-ft}\,dt \tag{2}$$

For $T_{i,j-1} < t < T_{i,j}(1-L_{i,j}) + T'_{i,j}L_{i,j}$, since the inventory position of product i at time t in period j is $I_{i,j}(t) = X_{i,j} + Q_{i,j} - D_{i,j}(t - T_{i,j-1})$ and that at time $T_{i,j} - T'_{i,j} = b_{i,j}/D_{i,j}$ in period j the inventory becomes zero, by letting $T_{i,j}(1-L_{i,j}) + T'_{i,j}L_{i,j} = a$, Eq. (2) becomes

$$H = \sum_{i=1}^{m}\sum_{j=1}^{N-1} h_i \left\{ \left(\frac{\mathrm{e}^{-fa} - \mathrm{e}^{-fT_{i,j-1}}}{f} \right) D_{i,j} \left(\frac{\mathrm{e}^{-fa}\left(a + \frac{1}{f}\right) - \mathrm{e}^{-fT_{i,j-1}}\left(T_{i,j-1} + \frac{1}{f}\right)}{\mathrm{e}^{-fa} - \mathrm{e}^{-fT_{o,j-1}}} - \left(\frac{X_{i,j} + Q_{i,j}}{D_{i,j}} \right) - T_{i,j-1} \right) \right\} \tag{3}$$

The shortage cost consists of two parts; backorder and lost sale. Since β_i percentage of the demand of product i is backorder and the $(1-\beta_i)$ percentage is lost sale, based on Fig. 1, the backorder (BO) and lost sale (LS) costs under inflation are obtained as

$$BO = \sum_{i=1}^{m}\sum_{j=1}^{N-1} \pi_{i,j} e^{-fT_{i,j}} \beta_i \int_{T'_{i,j}}^{T_{i,j}} I_{i,j}(t)\mathrm{e}^{-ft}dt \tag{4}$$

$$LS = \sum_{i=1}^{m}\sum_{j=1}^{N-1} \pi'_{i,j}\mathrm{e}^{-fT_{i,j}}(1-\beta_i) \int_{T'_{i,j}}^{T_{i,j}} I_{i,j}(t)\mathrm{e}^{-ft}dt \tag{5}$$

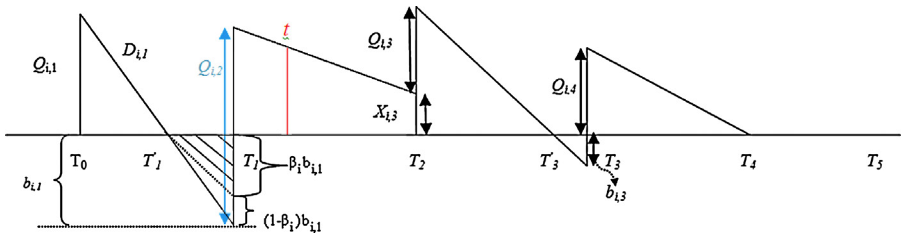

Fig. 1. Some possible scenarios for an inventory problem with five periods.

where for $T'_{i,j} < t < T_{i,j}$ we have $I_{i,j}(t) = D_{i,j}(t - T'_{i,j})$.

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

$$P_i = \begin{cases} P_{i,1} & 0 \le Q_{ij} < q_{i2} \\ & q_{i2} \le Q_{ij} < q_{i3} \\ P_{i,2} & \\ \vdots & \\ P_{i,K} & q_{iK} \le Q_{ij} \end{cases} \tag{6}$$

Then, the purchasing cost (P) becomes

$$P = \sum_{i=1}^{m} \sum_{j=1}^{N-1} \sum_{k=1}^{K} Q_{i,j} P_{i,k} U_{i,j,k} e^{-fT_{i,j}} \tag{7}$$

The first objective function is the minimization of the total inventory cost $Z_1 = A + H + BO + LS + P$ obtained as

$$\begin{aligned} \operatorname{Min} Z_1 = {} & \sum_{i=1}^{m} \sum_{j=1}^{N-1} A_{i,j} W_{i,j} e^{-fT_{i,j}} + \sum_{i=1}^{m} \sum_{j=1}^{N-1} h_i e^{-fT_{i,j-1}} \int_{T_{i,j-1}}^{T_{i,j}(1-L_{i,j})+T'_{i,j}L_{i,j}} I_{i,j}(t) e^{-ft} dt + \\ & \sum_{i=1}^{m} \sum_{j=1}^{N-1} \pi_{i,j} e^{-fT_{i,j}} \beta_i \int_{T'_{i,j}}^{T_{i,j}} I_{i,j}(t) e^{-ft} dt + \sum_{i=1}^{m} \sum_{j=1}^{N-1} \pi'_{i,j} e^{-fT_{i,j}} (1-\beta_i) \int_{T'_{i,j}}^{T_{i,j}} I_{i,j}(t) e^{-ft} dt + \\ & \sum_{i=1}^{m} \sum_{j=1}^{N-1} \sum_{k=1}^{K} Q_{i,j} P_{i,k} U_{i,j,k} e^{-fT_{i,j}} \end{aligned} \tag{8}$$

The second objective can be formulated as the minimization of the total required storage space (Z_2). Since in a given period j, the order quantity of item i plus its inventory in the previous period require storage space, a total of $Q_{i,j} + X_{i,j}$ units, each requiring S_i unit of storage need space. Therefore, the second objective function becomes

$$\operatorname{Min} Z_2 = \sum_{i=1}^{m} \sum_{j=1}^{N-1} (Q_{i,j} + X_{i,j}) S_i \tag{9}$$

3.1. Constraints

The inventory of a certain product i at the end of period j, i.e. $I_{i,j}$, must be either positive denoted by $X_{i,j+1}$ (the beginning inventory of period $j+1$) or zero. In other words,

$$I_{i,j} = \begin{cases} X_{i,j+1}; & I_{i,j} > 0 \\ 0; & I_{i,j} \le 0 \end{cases} \tag{10}$$

Moreover, the beginning inventory of product i in period $j+1$ is equal to its beginning inventory in the previous period j, plus the order quantity, minus demand. Or

$$I_{i,j+1} = I_{i,j} + Q_{i,j} - D_{i,j}\left(T'_{i,j} L_{i,j} + T_{i,j}(1 - L_{i,j}) - T_{i,j-1}\right) \tag{11}$$

In the case a shortage happens for a product in a specific period, then $L_{i,j}$ takes the value of 1, $I_{i,j}$ becomes negative in Eq. (11), and the shortage becomes:

$$b_{i,j+1} = D_{i,j}\left(T'_{i,j} - T_{i,j-1}\right) - (I_{i,j} + Q_{i,j}) \tag{12}$$

Eq. (12) shows the shortage quantity of each item in a period completely depends on waiting time of its demand.

Since the order quantity of product i in period j, $Q_{i,j}$, is delivered in $V_{i,j}$ boxes, each containing B_i products, the next constraint becomes

$$Q_{i,j} = B_i V_{i,j} \tag{13}$$

Due to real-world limitations on the transportation methods (e.g. truck space), the order quantity of all products in a period cannot be greater than a given fixed number M_1. In other words,

$$\sum_{i=1}^{m} Q_{i,j} \le M_1; \quad \forall j \tag{14}$$

Furthermore, the number of available boxes to deliver product i in period j is limited and we have

$$V_{i,j} \le M_2 \tag{15}$$

The purchasing price per unit of product i is P_i, the order quantity of product i in period j is $Q_{i,j}$ and the total budget is C. As a result the budget constraint is

$$\sum_{i=1}^{m} \sum_{j=1}^{N-1} \sum_{k=1}^{K} Q_{i,j} P_{i,k} U_{i,j,k} e^{-fT_{i,j}} \le C \tag{16}$$

Finally, since at most one order can be placed for a product in a period and that it can be purchased at one price break point, we have

$$\sum_{k=1}^{K} U_{i,j,k} = 1 \tag{17}$$

In short, the mathematical formulation of the bi-objective inventory control problem at hand becomes

$$\begin{aligned} \operatorname{Min} Z_1 = {} & \sum_{i=1}^{m} \sum_{j=1}^{N-1} A_{i,j} W_{i,j} e^{-fT_{i,j}} + \sum_{i=1}^{m} \sum_{j=1}^{N-1} h_i e^{-fT_{i,j-1}} \int_{T_{i,j-1}}^{T_{i,j}(1-L_{i,j})+T'_{i,j}L_{i,j}} I_{i,j}(t) e^{-ft} dt + \\ & \sum_{i=1}^{m} \sum_{j=1}^{N-1} \pi_{i,j} e^{-fT_{i,j}} \beta_i \int_{T'_{i,j}}^{T_{i,j}} I_{i,j}(t) e^{-ft} dt + \sum_{i=1}^{m} \sum_{j=1}^{N-1} \pi'_{i,j} e^{-ft} (1-\beta_i) \int_{T_{i,j}}^{T_{i,j}} I_{i,j}(t) e^{-ft} dt + \\ & \sum_{i=1}^{m} \sum_{j=1}^{N-1} \sum_{k=1}^{K} Q_{i,j} P_{i,k} U_{i,j,k} e^{-fT_{i,j}} \\ \operatorname{Min} Z_2 = {} & \sum_{i=1}^{m} \sum_{j=1}^{N-1} (Q_{i,j} + X_{i,j}) S_i \end{aligned} \tag{18}$$

s.t.:

$$I_{i,j} = \begin{cases} X_{i,j+1}; & I_{i,j} \geq 0 \\ 0; & I_{i,j} < 0 \end{cases}$$

$$I_{i,j+1} = I_{i,j} + Q_{i,j} - D_{i,j}\left(T'_{i,j}L_{i,j} + T_{i,j}(1 - L_{i,j}) - T_{i,j-1}\right)$$

$$Q_{i,j} = B_i V_{i,j}$$

$$\sum_{i=1}^{m} Q_{i,j} \leq M_1; \quad \forall j$$

$$V_{i,j} \leq M_2$$

$$\sum_{i=1}^{m}\sum_{j=1}^{N-1}\sum_{k=1}^{K} Q_{i,j}P_{i,k}U_{i,j,k}e^{-fT_{i,j}} \leq C$$

$$\sum_{k=1}^{K} U_{i,j,k} = 1$$

$$i = 1, 2, \ldots, m, j = 0, 1, \ldots, N, k = 1, 2, \ldots, K$$

In the next section, two parameter-tuned multi-objective evolutionary algorithms are proposed to obtain Pareto-optimal set for the above constrained bi-objective inventory model.

4. Solution algorithms

Although different exact methods such as Lagrangian relaxation [17] and branch and bound [44] have been developed in the literature to solve less complicated inventory control models, due to multi-objectivity and complexity, they cannot be employed to find optimal solutions of the model at hand. As a result, in this work, a non-dominated sorting genetic algorithm (NSGA-II), a non-dominated ranked genetic algorithm (NRGA), and a multi-objective particle swarm optimization (MOPSO) algorithm are utilized in the following subsections, are employed to find Pareto solutions.

4.1. The proposed NSGA-II

Among various multi-objective optimization methods, the elitist non-dominated sorting genetic algorithm (NSGA), due to its simplicity, effectiveness, and minimum user interaction, is one of the most popular methods. NSGA-II, first introduced by Deb et al. [25], is an improved NSGA and it has been applied in many engineering design optimization successfully. Using Pareto dominance solutions, it is a computationally efficient algorithm implementing the idea of a selection method based on classes of dominance of all the solutions. It has a better sorting algorithm compared to NSGA, incorporates elitism, and requires no sharing parameter to be chosen a priori. The original NSGA-II algorithm consists of five operators: initialization, fast non-dominated sorting, crossover, mutation and the elitist crowded-comparison operator. The proposed NSGA-II of this research also involves these operators where they are described in the following subsections.

4.1.1. Initialization

In this research, a chromosome represents a box containing different number of products ordered in different periods. In other words, a chromosome is represented by a $m \times (N-1)$ matrix

$$Q = \begin{pmatrix} Q_{1,1} & Q_{1,2} & \cdots & Q_{1,N-1} \\ Q_{2,1} & Q_{2,2} & \cdots & Q_{2,N-1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ Q_{m,1} & Q_{m,2} & \cdots & Q_{m,N-1} \end{pmatrix}$$

Fig. 2. A chromosome representation.

$\left(Q = [Q_{i,j}; i = 1, 2, \ldots, m, j = 1, 2, \ldots, N-1]\right)$ shown in Fig. 2. Moreover, since the upper bound on the number of boxes in each period is M_2, in the initialization step of the proposed NSGA-II algorithm, a uniform integer random number m_2 in $[0, M_2]$ is generated for each chromosome. In order to accelerate the run process of the algorithm and reach a near optimal solution in a shorter time, infeasible solutions (those that violate the constraints) are penalized by a specific quantity determined by a penalty function shown in Eq. (19). This function was shown to be the best among several penalty functions used in a pilot study. In addition, several studies have utilized this function to penalize their infeasible constraints (Mousavi et al. [45,46]). For an infeasible solution when a constraint exceeds its right hand side value (R), both the objective values are fined to an amount found as the difference of the left-hand-side and the right-hand-side of that particular constraint as shown in Eq. (19).

$$\text{Penalty} = (Y(x) - R)^{\alpha} \tag{19}$$

Where $Y(x) \leq R$ is a constraint of the problem, $\alpha = 10$, and x is the decision variable. This penalty is added to the values of both objective functions.

4.1.2. Fast non-dominated sorting

In this step, the R populations that were generated in the previous step are compared and are sorted. To do this, all chromosomes in the first non-dominated front are first found. Since both objective functions in model (17) are of a minimization type, the chromosomes are chosen using the concept of domination, in which a solution x_i is said to dominate solution x_j, if $\forall o \in \{1, 2\}$ we have $Z_o(x_i) \leq Z_o(x_j)$ and $\exists\, o \in \{1, 2\}$ such that $Z_o(x_i) < Z_o(x_j)$. In this case, we say x_i is the non-dominated solution within the solution set $\left\{x_i, x_j\right\}$. Otherwise, it is not. Then, in order to find the chromosomes in the next non-dominated front, the solutions of the previous fronts are disregarded temporarily. This procedure is repeated until all solutions are set into fronts. It should be mentioned that the computational complexity of NSGA-II is $O(MR^2)$ where M and R are the number of objectives and the population size, respectively [25].

4.1.3. Crowding distance

After sorting the populations, a measure called the crowding distance is defined to evaluate solution fronts of populations in terms of the relative density of individual solutions [25]. To do this, consider Z and f_k; $k = 1, 2, \ldots, M$ the number of non-dominated solutions in a particular front (F) and the objective functions, respectively. Besides, let d_i and d_j be the value of crowding distance on the solution i and j. Then, the crowding distance is obtained using the following steps:

1. Set $d_i = 0$ for $i = 1, 2, \ldots, Z$
2. Sort all objective functions f_k; $k = 1, 2, \ldots, M$ in ascending order
3. The crowding distance for end solutions in each front (d_1 and d_Z) are $d_1 = d_Z = \infty$

4. The crowding distance for d_j; $j = 2, 3, \ldots, (Z-1)$ are $d_j = d_j + (f_{k_{j+1}} - f_{k_{j-1}})$

4.1.4. Selection strategy

In order to select individuals of the next generation, the crowded tournament selection operator "$\succ$" is applied [25]. To do this, the following steps are required to be carried out

1. Choose n individuals in the population, randomly
2. Obtain non-dominated rank of each individual
3. Calculate the crowding distance of the solutions having equal non-dominated rank
4. At the end, the solutions with least rank are the selected ones. Moreover, if more than one individual share the least rank, the individual with highest crowding distance must be selected.

In other words, the NSGA-II comparison criterion can be written as

If $r_x < r_y$ or ($r_x = r_y$ and $d_x > d_y$) then $x \succ y$where r_x and r_y are the ranks and d_x and d_y are the crowding distances.

4.1.5. Crossover operator

Let P_C be the crossover probability and r_1 be a uniform random number between zero and one. For each of the R populations, if r_1 is less than P_C, select two parent chromosomes O_1 and O_2 randomly. Then, the crossover operator of the proposed NSGA-II algorithm is performed based on the following equations.

$$O'_1 = \lambda O_1 + (1-\lambda)O_2 \tag{20}$$

$$O'_2 = (1-\lambda)O_1 + \lambda O_2 \tag{21}$$

where O'_1 and O'_2 are offspring and λ is a uniform random number between zero and one.

4.1.6. Mutation operator

In order to reach better solution of the problem, several different forms of mutation operator provided in the literature are tested. It has been found that the uniform mutation operator was more suitable than the others. In this operations, for each chromosome for which $r_1 > P_C$, select two integer numbers y_1 and y_2 in intervals [1,m] and [1,N], first. Then, generate an integer number randomly between 0 and M_2 for row y_1 and column y_2 of the chromosome. Fig. 3 shows how to do mutation operator of NSGA-II algorithm for a problem with $m = 6$ and $N = 5$ since $y_1 = 3$ and $y_2 = 2$.

4.1.7. Concatenating the population

In this step, the parents and offspring population are combined to ensure the elitism. Since the combined population size is naturally greater than the original population size N, once more, non-domination sorting is performed [25]. In fact, chromosomes with higher ranks are selected and added to the populations until the population size becomes N. The last front is also consisted of the population based on the crowding distance.

The algorithm stops when a predetermined number of iterations is reached.

4.2. The proposed NRGA

NRGA is a new multi-objective genetic algorithm to find feasible Pareto front solutions. NRGA is similar to NSGA-II with the difference that in the selection operation the roulette wheel strategy is employed [33]. Similar to NSGA-II, here Eq. (19) is used to fine infeasible solutions violating the constraints. In NRGA, a fitness value representing rank is assigned to each individual of the population. In this regard, two ranked based roulette wheel selection features including: (I) select the fronts and (II) choose solutions from the fronts, are used. The selection probability of fronts, P_f, and the selection probability of solutions, P_{fs}, are obtained using Eqs. (21) and (22).

$$P_f = \frac{2 \times \mathrm{Rank}_f}{NF \times (NF+1)}; \quad f = 1, \ldots, NF \tag{22}$$

$$P_{fs} = \frac{2 \times \mathrm{Rank}_{fs}}{NS_f \times (NS_f+1)}; \quad f = 1, \ldots, NF, s = 1, \ldots, NS \tag{23}$$

where NF and NS_f are the number of fronts and the number of solutions in front f, respectively. Eq. (22) ensures that a front with highest rank has the highest probability to be selected. Similarly, based on Eq. (23), solutions with more crowding distance are assigned higher selection probability. The roulette wheel selection is iterated until a desired number of solutions are selected. At the end, the algorithm stops when a predetermined number of iterations is reached.

4.3. The proposed MOPSO

The MOPSO algorithm have been commonly used in the literature to some a wide range of optimization problems ([47–50]) including inventory control problems ([51,52]). The first, second, and third steps of the algorithm used in this study are initialization, fast non-dominated sorting, and crowding distance, similar to the NSGA-II algorithm. The fourth step is called updating and velocity of the particles (chromosomes) obtained using Eqs. (24) and (25) as follows:

$$v^i_{k+1,d} = w.v^i_{k,d} + C_1.r_1.(pbest^i_{k,d} - x^i_{k,d}) + C_2.r_2.(gbest^i_{k,d} - x^i_{k,d}) \tag{24}$$

$$x^i_{k+1,d} = x^i_{k,d} + v^i_{k+1,d} \tag{25}$$

where r_1 and r_2 are uniform random numbers between 0 and 1, coefficients C_1 and C_2 are the given acceleration constants towards $p\vec{b}est$ and $g\vec{b}est$, 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 [53,54]. Moreover, the linear distribution of the inertia weight is expressed as follows [53]

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{NOG}\text{iteration}, \tag{26}$$

where NOG is the maximum number of iterations and iteration is the current number of iteration. Eq. (26) presents how the inertia weight is updated, considering $w_{\max}$ and $w_{\min}$ are the initial and the final weights, respectively. We use parameters $w_{\max} = 0.9$ and $w_{\min} = 0.4$ in this study which have been investigated by Naka et al. [54] and Shi and Eberhart [53]. Note that the ranks and the crowding distances of the new chromosomes (offspring) are first determined. The parents and the offspring are then combined similar to the NSGA-II method. The algorithm stops when a predetermined number of iterations is reached.

5. Experiments and discussion

In order to demonstrate the application of the proposed methodology and to evaluate and compare the performances of the two MOE algorithms, the parameters of the algorithms are first tuned by the Taguchi method using two numerical examples.

5.1. Parameter tuning

As mentioned previously, NSGA-II and NRGA are employed to find Pareto fronts of the bi-objective optimization model (18), in which the population size (PS), the number of generations (NOG),

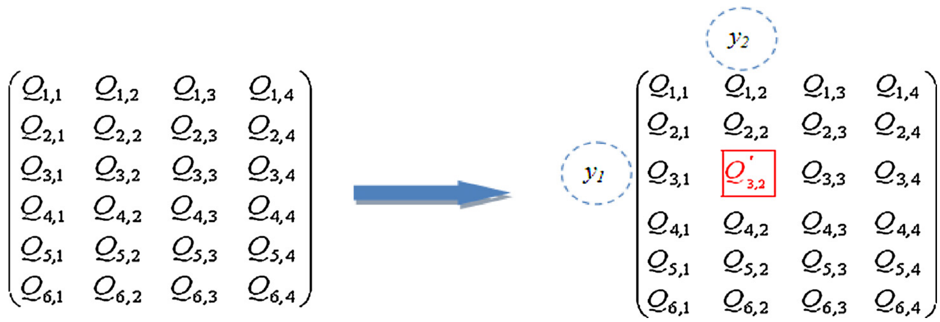

Fig. 3. The mutation operator.

the crossover probability (*CP*), and the mutation probability (*MP*) are the input parameters of both algorithms. Each algorithm runs 10 times and the performances are compared in terms of five indices; (1) an error ratio (*ER*), (2) the required CPU time (seconds), (3) the number of optimal Pareto solutions (*NPS*), (4) the mean ideal distance (MID) introduced in [55] to gauge the convergence rate of the Pareto fronts towards a specific point (0, 0), and (5) the diversification metric to measure the distribution of the solution set [55]. Moreover, *ER* is obtained in Eq. (27), in which e_u is zero if solution u belongs to Pareto front and one otherwise and n is the population size [56].

$$ER = \frac{\sum_{u=1}^{n} e_u}{n} \tag{27}$$

Since the meta-heuristic algorithms are severely sensitive to their parameters, a Taguchi procedure has been utilized to calibrate the parameters of the algorithms. Taguchi method is a fractional factorial experiment introduced by Taguchi as an efficient alternative for full factorial experiments [57]. Taguchi procedure uses orthogonal arrays for setting family of experiences to investigate a group of factors. In this procedure, factors are categorized into two groups consisted of controllable or signals factors and noise factors. Now, based on the concept of the robustness, the method seeks to minimize the effect of noise and to determine the optimal level of signal factors. To do so, the signal to noise ratio (S/N), which calculates the amount of variation involved in the response, is used. While there are different ways to obtain variation in Taguchi procedure, in this research, since the goal is to minimize S/N, the smaller-the-better type of response has been utilized, where S/N is given as

$$S/N = -10 \times \log\left(\frac{S\left(Y^2\right)}{n}\right) \tag{28}$$

In Eq. (28), Y and n are the response value and the number of orthogonal arrays, respectively, and $S(Y^2)$ is the summation of the responses Y^2. To consider the two main goals of Pareto-based algorithms (convergence and diversity) simultaneously, a new metric is introduced to contain the error ratio (*ER*) and the number of Pareto solutions (*NPS*). This ratio is defined as

$$C.V.R. = \frac{ER}{NPS} \tag{29}$$

Since this metric mimic the classical coefficient of variation, it is called coefficient of variation ratio (*C.V.R.*) and acts as the response variable of the Taguchi method.

To implement the Taguchi procedure, first, the levels of the factors are provided in Tables 1 and 2 for the three algorithms. In each of the algorithms, three levels are considered for each factor. Then, by selecting the L9 design and using the Minitab Software, the orthogonal arrays along with their responses are presented in Tables 3 and 4. We found the L9 design of orthogonal arrays (OA) to be the best to tune the parameters of the algorithms using several tests. This has been a common approach in several studies such as [31,45,46,58].

Table 1
NSGA-II and NRGA parameter ranges and levels.

Algorithm parameters	Parameters range	Low (1)	Medium (2)	High (3)
PS	50–80	50	60	80
CP	0.7–0.90	0.9	0.8	0.7
MP	0. 1–0.3	0. 1	0.2	0.3
NOG	100–300	100	200	300

Table 2
MOPSO parameter ranges and levels.

Algorithm Parameters	Parameters range	Low (1)	Medium (2)	High (3)
C_1	1.5–2.5	1.5	2	2.5
C_2	1.5–2.5	1.5	2	2.5
NOP	50–100	50	80	100
NOG	100–500	100	300	500

Table 3
Calibration process of NSGA-II and NRGA.

Run order	**Algorithm parameters**				**C.V.R.**	
	PS	***NOG***	***CP***	***MP***	**NSGA-II**	**NRGA**
1	1	1	1	1	0.0911	0.0336
2	1	2	2	2	0.1050	0.0221
3	1	3	3	3	0.1800	0.0120
4	2	1	2	3	0.1800	0.0280
5	2	2	3	1	0.1833	0.0141
6	2	3	1	2	0.1262	0.0106
7	3	1	3	2	0.1542	0.0217
8	3	2	1	3	0.1304	0.0101
9	3	3	2	1	0.2375	0.0059

Table 4
Calibration process of MOPSO.

Run order	Algorithm parameters			**C.V.R.**	
	C_1	C_2	***NOP***	***NOG***	**MOPSO**
1	1	1	1	1	0.2140
2	1	2	2	2	0.0809
3	1	3	3	3	0.0101
4	2	1	2	3	0.1532
5	2	2	3	1	0.1201
6	2	3	1	2	0.0902
7	3	1	3	2	0.1802
8	3	2	1	3	0.0347
9	3	3	2	1	0.1532

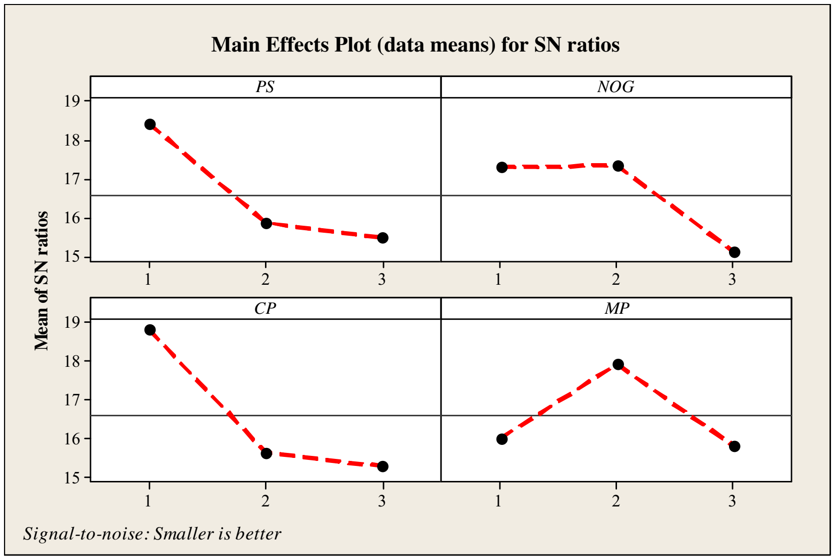

Fig. 4. Taguchi S/N ratio plot for NSGA-II.

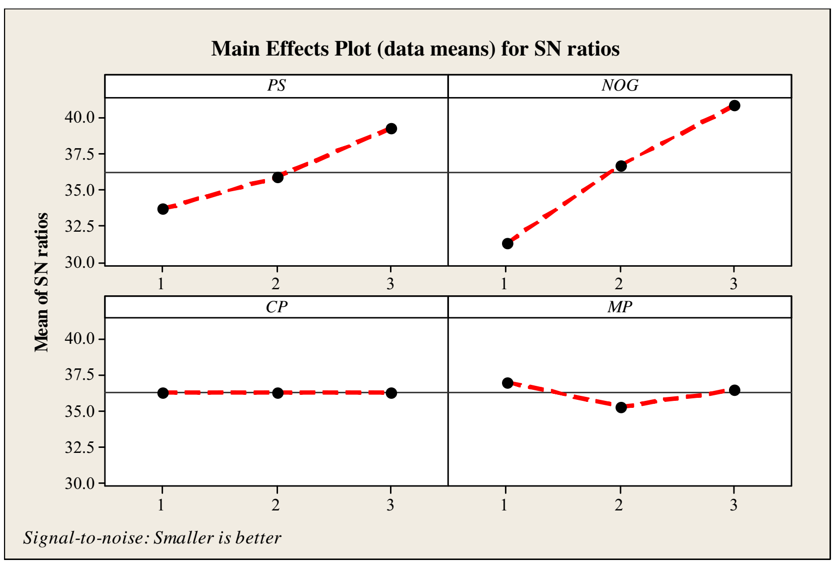

Fig. 5. Taguchi S/N ratio plot for NRGA.

Table 5
Optimal values of the parameters.

Algorithms	Parameters	Optimal value
NSGA-II	*PS*	50
	CP	0.8
	MP	0.1
	NOG	200
NRGA	*PS*	80
	CP	0.7
	MP	0.1
	NOG	100
MOPSO	C_1	1.5
	C_2	2.5
	NOP	100
	NOG	500

The effect plots of S/N ratio are given in Figs. 4–6 for NSGA-II, NRGA, and MOPSO, respectively. Using these plots, the optimal values of the parameters for each algorithm are obtained in Table 5. We used Figs. 4–6 to select the optimal values of the parameters according to the parameter levels of Low, Medium, and High, where the highest response is selected. The ranges of values defined for each level of Low, Medium, and High are selected based on a pilot study with several test problems. In other words, we set a wide range of values for the parameters and then run the problem to find the best ranges for each parameter. Interested readers can refer to [31,45,46,58] for more details on the selection of these optimal values.

5.2. Analysis of results

The computer coding of both algorithms are developed using MATLAB software and the experiments are performed on a computer with 2.50 GHz of core 2 CPU and 3.00 GB of RAM.

Numerical Example 1: Consider an inventory control problem with 6 products and 4 periods where the input data is given in Table 6. The other parameters of the problem are: $T_j=2$; ($j=1, 2, \ldots, 5$); $M_2=3200$; $f=0.08$; $M_1=100$; $C=24000$. Moreover, the price-break points for products 1 and 2 are defined as follows:

$$P_i = \begin{cases} 12 & 0 \le Q_{i,j} < 20 \\ 11 & 20 \le Q_{i,j} < 50 \\ 9 & 50 \le Q_{i,j} \end{cases}$$

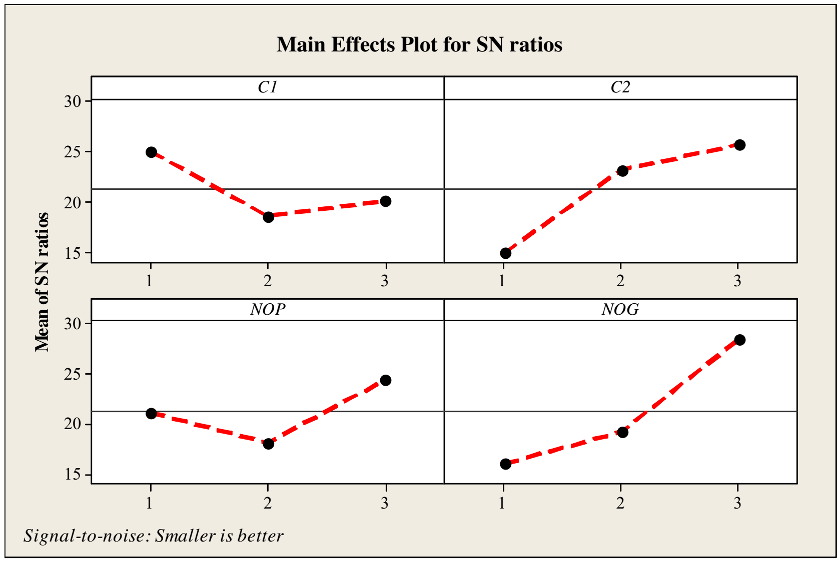

Fig. 6. Taguchi S/N ratio plot for MOPSO.

Table 6
Input data for the first numerical example.

I	1	2	3	4	5	6
D_{i1}	250	420	170	200	400	350
D_{i2}	340	180	410	130	270	290
D_{i3}	320	210	110	205	415	190
B_i	8	7	9	8	9	6
H_i	9	7	8	8	10	7
O_{i1}	20	18	20	18	13	13
O_{i2}	16	21	16	21	17	17
O_{i3}	11	15	12	14	15	16
S_i	8	11	12	10	11	9
$\dot{\pi}_{i1}$	8	7	7	7	7	8
$\dot{\pi}_{i2}$	6	8	4	8	4	7
$\dot{\pi}_{i3}$	3	4	5	6	7	4
π_{i1}	9	5	9	5	9	9
π_{i2}	4	9	4	9	7	4
π_{i3}	6	5	4	7	8	6
β_i	0.4	0.5	0.5	0.6	0.3	0.4

For products 3, 4, and 5 the price-break points are

$$P_i = \begin{cases} 10 & 0 \le Q_{i,j} < 40 \\ 8 & 40 \le Q_{i,j} \end{cases}$$

Finally, the price-break points for product 6 are

$$P_6 = \begin{cases} 13 & 0 \le Q_{i,j} < 60 \\ 11 & 60 \le Q_{i,j} \end{cases}$$

Table 8
ANOVA for NPS of numerical example 1.

Source	DF	SS	MS	F	p-value
Factor	2	72.80	36.40	8.63	0.001
Error	27	113.90	4.22	–	–
Total	29	186.70	–	–	–

Table 9
ANOVA for ER of numerical example 1.

Source	DF	SS	MS	F	p-value
Factor	2	0.006047	0.003023	5.93	0.007
Error	27	0.013760	0.000510		
Total	29	0.019807			

In the presence of inflation, the objective is to find the optimal order quantities of the products in different cycles such that both the total inventory cost and the required warehouse space are minimized.

The five performance measures obtained by employing the three parameter-tuned algorithms on all 10 replications of numerical example 1 are given in Table 7. The last two rows in Table 7 show the average (*Ave.*) and the standard deviation (*St.dev.*) of the values of the metrics in these 10 replications. In addition, five one-way ANOVAs are used to statistically compare the performances of the three algorithms in terms of the five abovementioned criteria. Tables 8–12 show the one-way ANOVA of the performance indices

Table 7
Experimental results of 10 replications in numerical example 1.

Run No.	NRGA					NSGA-II					MOPSO				
	NPS	ER	CPU (Sec)	MID	DM	NPS	ER	CPU (Sec)	MID	DM	NPS	ER	CPU (Sec)	MID	DM
1	9	0.888	34.89	131.19	12.95	7	0.86	21.82	399.39	46.574	16	0.84	39.73	362.48	80.77
2	10	0.875	36.71	64.562	71.87	7	0.86	20.92	306.49	70.369	9	0.91	41.25	297.9	68.89
3	11	0.863	35.92	119.65	59.33	8	0.84	20.38	499.49	148.83	8	0.92	40.52	205.8	64.67
4	10	0.875	35.82	195.22	64.5	8	0.84	21.34	209.26	115.03	15	0.85	41.53	460.84	83.63
5	8	0.9	33.49	247.66	74.62	6	0.88	19.88	275.16	85.345	8	0.92	39.97	378.05	107.7
6	10	0.875	34.56	103.04	68.19	7	0.86	21.29	238.22	159.38	9	0.91	41.49	195.27	146.1
7	7	0.913	35.28	191.54	98.26	7	0.86	20.31	213.01	34.346	8	0.92	39.77	223.38	64.15
8	11	0.863	34.71	174.51	77.13	6	0.88	19.87	240.17	33.832	13	0.87	40.29	237.32	75.19
9	9	0.888	35.92	189.15	36.24	8	0.84	21.43	176.14	119.24	9	0.91	40.82	445.93	115.7
10	8	0.9	36.49	201.36	54.14	7	0.86	20.22	491.76	130.48	14	0.86	41.09	254.51	117.8
Ave.	9.3	0.884	35.379	161.79	61.72	7.1	0.858	20.746	304.91	94.342	10.9	0.891	40.646	306.15	92.46
St.dev.	1.33749	0.017	0.984	55.246	23.53	0.7379	0.015	0.700	118.04	46.846	3.212	0.032	0.6905746	99.161	27.76

Table 10
ANOVA for CPU (Sec) of numerical example 1.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	2126.253	1063.127	1646.67	0.000
Error	27	17.432	0.646		
Total	29	2143.685			

Table 11
ANOVA for MID of numerical example 1.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	137750	68875	7.70	0.002
Error	27	241371	8940		
Total	29	37912			

Table 12
ANOVA for DM of numerical example 1.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	6708	3354	2.86	0.075
Error	27	31668	1173		
Total	29	3837			

NPS, ER, CPU (Sec), MID, and DM at 95% confidence level along with the values of the corresponding *p*-values. Tables 8–11 show while there are significant differences between the three algorithms in terms of the means of NPS, ER, CPU (Sec), and MID, there are no significant differences among the proposed algorithms in term of the means of DM. Moreover, Fig. 7 exhibits the best Pareto solutions of the three algorithms in 10 runs of the algorithms for the numerical example 1.

Figs. 8–12 show the plots and the box plots of NPS, ER, CPU (Sec), MID, and DM in the three algorithms for numerical example 1.

Numerical Example 2: Consider an inventory control system with 8 products and 4 periods. The input data is shown in Table 13, where the other data are: $M_2 = 3500$; $f = 0.08$; $M_1 = 100$; $C = 25000$. The price-break points for all 8 products, $(i = 1, 2, \ldots, 8)$, are defined as follows:

$$P_i = \begin{cases} 16 & 0 \le Q_{i,j} < 25 \\ 13 & 25 \le Q_{i,j} < 60 \\ 11 & 60 \le Q_{i,j} \end{cases}$$

Table 14 shows the five performance measures obtained by employing the three parameter-tuned algorithms on all 10 replications of numerical example 2. The last two rows of Table 14 exhibits the average (*Ave.*) and the standard deviation (*St.dev.*) of the values of the metrics in 10 the replications.

Tables 15–19 show the one-way ANOVA for the performance metrics NPS, ER, CPU (Sec), MID, and DM at 95% confidence level along with their corresponding *p*-vales. The results show while there are significant differences among the three algorithms in terms of the means of ER, CPU (Sec) MID, and DM, there are no significant differences among the proposed algorithms in term of the means of NPS. In addition, Fig. 13 shows the best Pareto

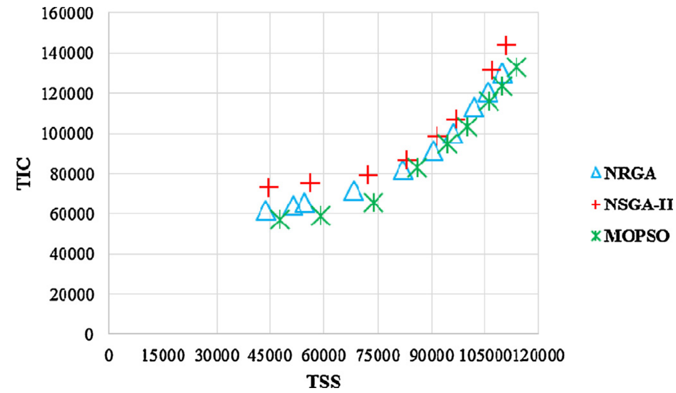

Fig. 7. The best Pareto solution in 10 runs of the algorithms in numerical example 1.

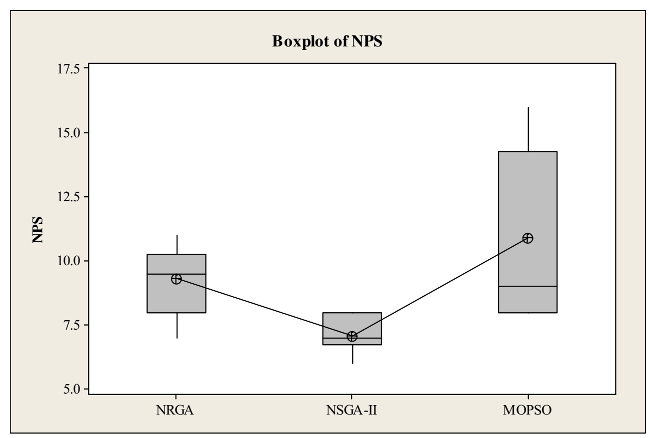

Fig. 8. The plot and the box-plot of the NPS metric in numerical example 1.

Table 13
Input data for the second numerical example.

I	1	2	3	4	5	6	7	8
D_{i1}	250	420	170	200	400	350	330	150
D_{i2}	340	180	410	130	270	290	210	240
D_{i3}	320	210	110	205	415	190	190	190
B_i	8	7	9	8	9	6	6	8
H_i	9	7	8	8	10	7	9	10
O_{i1}	20	18	20	18	13	13	21	14
O_{i2}	16	21	16	21	17	17	16	16
O_{i3}	11	15	12	14	15	16	18	17
S_i	8	11	12	10	11	9	12	9
π'_{i1}	8	7	7	7	7	8	6	4
π'_{i2}	6	8	4	8	4	7	7	9
π'_{i3}	3	4	5	6	7	4	4	6
π_{i1}	9	5	9	5	9	9	10	8
π_{i2}	4	9	4	9	7	4	4	5
π_{i3}	6	5	4	7	8	6	8	9
β_i	0.4	0.5	0.5	0.6	0.3	0.4	0.3	0.45

Table 14
Experimental results of 10 runs in numerical example 2.

Run No.	NRGA					NSGA-II					MOPSO				
	NPS	ER	CPU (Sec)	MID	DM	NPS	ER	CPU (Sec)	MID	DM	NPS	ER	CPU (Sec)	MID	DM
1	13	0.8375	34.89	154.263	60.779	14	0.72	21.82	447.7485	144.1789	10	0.9	39.73	481.563	144.4657
2	8	0.9	36.71	100.972	72.669	12	0.76	20.92	490.4855	37.81176	12	0.88	41.25	417.112	126.0461
3	9	0.8875	35.92	55.8665	59.755	9	0.82	20.38	157.9268	93.37863	17	0.83	40.52	358.572	134.2151
4	11	0.8625	35.82	176.781	39.416	8	0.84	21.34	370.5643	53.09099	13	0.87	41.53	160.662	141.5532
5	12	0.85	33.49	156.549	46.058	9	0.82	19.88	112.6219	119.4878	11	0.89	39.97	290.768	145.2846
6	10	0.875	34.56	159.577	34.933	13	0.74	21.29	415.8193	77.7906	12	0.88	41.49	278.095	118.2833
7	9	0.8875	35.28	209.876	63.526	11	0.78	20.31	443.5279	36.31316	15	0.85	39.77	285.284	53.72055
8	12	0.85	34.71	98.9977	21.975	10	0.8	19.87	462.5986	121.1048	9	0.91	40.29	274.952	148.5214
9	9	0.8875	35.92	56.3748	20.486	10	0.8	21.43	499.4592	42.86496	11	0.89	40.82	402.019	92.0069
10	11	0.8625	36.49	244.159	41.465	9	0.82	20.22	205.3058	98.61423	13	0.87	41.09	254.287	73.64249
Ave.	10.4	0.87	35.379	141.342	46.106	10.5	0.79	20.746	360.605	82.463	12.3	0.877	40.646	320.331	117.773
St.dev.	1.646	0.020	0.984	62.541	17.765	1.957	0.039	0.700	145.625	38.914	2.359	0.023	0.690	93.867	33.389

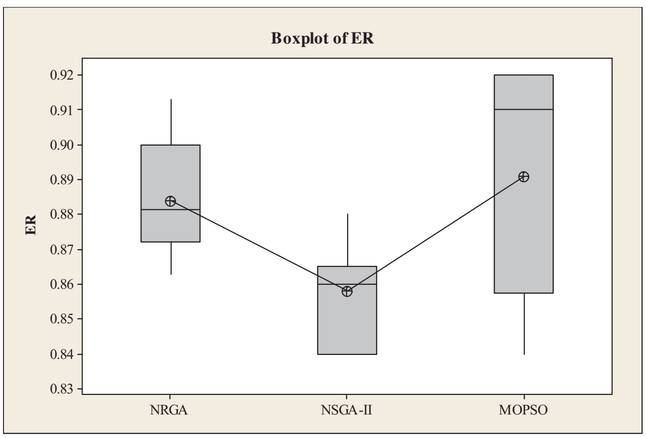

Fig. 9. The plot and the box-plot of the ER metric in numerical example 1.

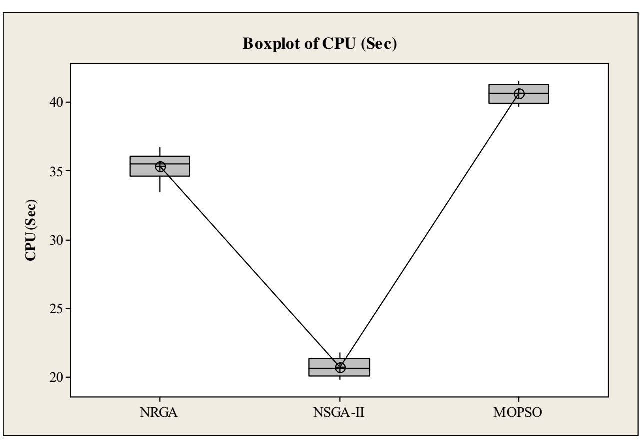

Fig. 10. The plot and the box-plot of the CPU metric in numerical example 1.

solutions for the numerical example 2 and the three algorithms in 10 runs.

Figs. 14–18 show the plots and the box plots of NPS, ER, CPU (Sec), MID, and DM of the three algorithms for the numerical example 2.

Numerical Example 3: Consider an inventory control problem with 10 products and 5 periods where the input data is given in Table 20. The other parameters of the problem are: $T_j = 2\,; (j = 1,\ 2,\ \ldots,\ 5)$; $M_2 = 10000$; $f = 0.08$; $M_1 = 100$; $C = 40000$.

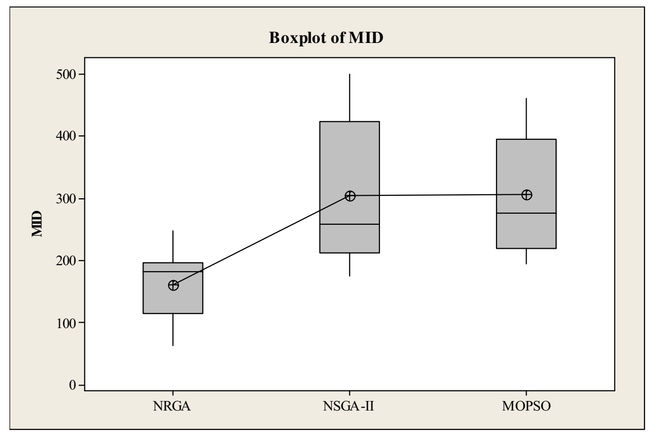

Fig. 11. The plot and the box-plot of the MID metric in numerical example 1.

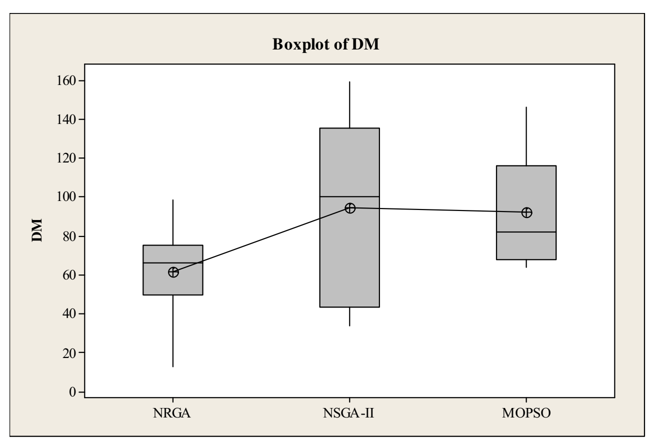

Fig. 12. The plot and the box-plot of the DM metric in numerical example 1.

Table 15
ANOVA for NPS of numerical example 2.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	22.87	11.43	2.83	0.076
Error	27	109.00	4.04		
Total	29	131.87			

Table 16
ANOVA for ER of numerical example 2.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	0.046727	0.023363	27.88	0.000
Error	27	0.022623	0.000838		
Total	29	0.069349			

Table 17
ANOVA for CPU (Sec) of numerical example 2.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	2126.253	1063.127	1646.67	0.000
Error	27	17.432	0.646		
Total	29	2143.685			

Table 18
ANOVA for MID of numerical example 2.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	272454	136227	12.05	0.000
Error	27	305364	11310		
Total	29	577818			

Table 19
ANOVA for DM of numerical example 2.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	25683	12842	13.08	0.000
Error	27	26503	982		
Total	29	52186			

Furthermore, the price-break points for different products are given as follows.

$$\text{For } (i=1,2,3),\ P_i = \begin{cases} 14 & 0 \le Q_{i,j} < 35 \\ 13 & 35 \le Q_{i,j} < 50 \\ 10 & 50 \le Q_{i,j} \end{cases}$$

$$\text{For } (i=4,5,6,7),\ P_i = \begin{cases} 18 & 0 \le Q_{i,j} < 40 \\ 17 & 40 \le Q_{i,j} < 70 \\ 13 & 70 \le Q_{i,j} \end{cases}$$

$$\text{For } (i=8,9,10),\ P_i = \begin{cases} 21 & 0 \le Q_{i,j} < 30 \\ 19 & 30 \le Q_{i,j} \end{cases}.$$

Using the optimal values of the parameters, the five performance measures obtained by employing the three

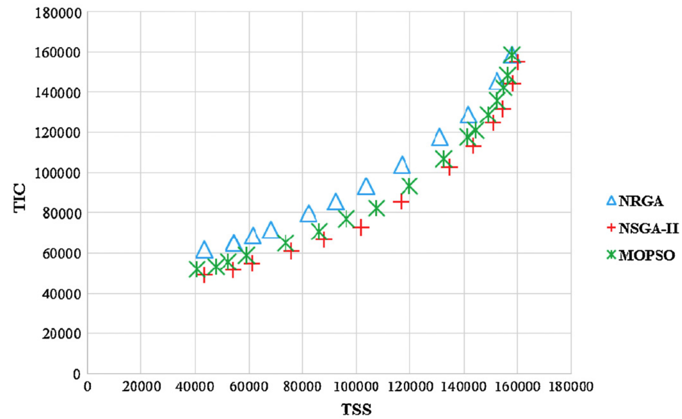

Fig. 13. The best Pareto solution in 10 runs of the algorithms in numerical example 2.

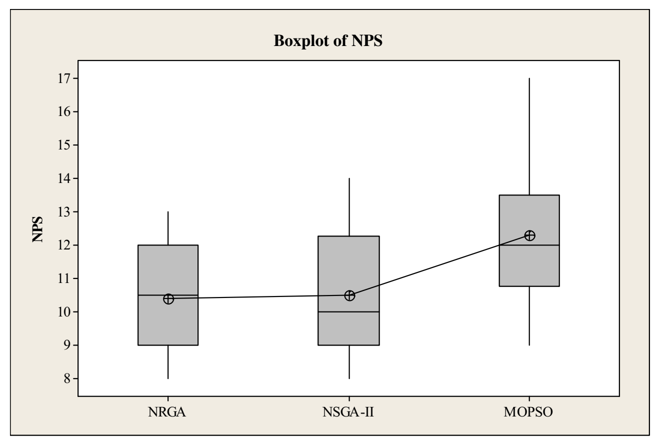

Fig. 14. The plot and the box-plot of the NPS metric in numerical example 2.

parameter-tuned algorithms on all 10 replications of numerical example 3 are given in Table 21. The last two rows of Table 21 shows the average (*Ave.*) and the standard deviation (*St.dev.*) of the values of the metrics for the 10 replications.

Table 20
Input data for the third numerical example.

i	1	2	3	4	5	6	7	8	9	10
D_{i1}	220	350	214	130	300	250	330	150	230	250
D_{i2}	180	210	260	315	220	170	305	130	340	150
D_{i3}	320	210	110	205	415	190	190	190	220	320
D_{i4}	150	230	190	165	325	300	200	140	160	350
B_i	9	4	8	7	6	9	3	5	4	3
H_i	10	7	9	6	4	11	5	3	7	8
O_{i1}	16	17	19	22	13	21	14	16	15	15
O_{i2}	15	19	22	21	11	14	15	19	12	14
O_{i3}	16	13	14	12	16	13	13	18	23	20
O_{i4}	20	17	13	16	16	14	19	16	21	15
S_i	12	9	11	10	8	5	10	9	13	12
π'_{i1}	7	8	8	8	6	7	6	8	7	5
π'_{i2}	5	6	5	8	7	10	11	6	13	8
π'_{i3}	5	3	7	6	4	8	11	8	7	10
π'_{i4}	8	10	7	4	7	9	11	5	8	10
π'_{i1}	9	5	9	5	9	9	10	8	6	4
π'_{i2}	4	9	4	9	7	4	4	5	6	7
π'_{i3}	6	5	4	7	8	6	8	9	10	11
π'_{i4}	8	11	10	5	8	8	9	6	10	7
β_i	0.3	0.5	0.6	0.4	0.4	0.6	0.45	0.75	0.3	0.5

Table 21
Experimental results of 10 runs in numerical example 3.

Run No.	NRGA					NSGA-II					MOPSO				
	NPS	ER	CPU (Sec)	MID	DM	NPS	ER	CPU (Sec)	MID	DM	NPS	ER	CPU (Sec)	MID	DM
1	8	0.9	34.89	76.6587	43.0799	11	0.78	21.82	326.936	170.403	14	0.86	39.73	263.32	74.2682
2	9	0.8875	36.71	70.5399	29.14	13	0.74	20.92	213.426	53.102	13	0.87	41.25	250.606	64.0386
3	12	0.85	35.92	193.575	50.11	10	0.8	20.38	234.741	32.7093	19	0.81	40.52	235.362	137.018
4	10	0.875	35.82	225.96	79.0469	12	0.76	21.34	468.651	97.3591	16	0.84	41.53	441.825	114.996
5	15	0.8125	33.49	52.7862	12.7621	11	0.78	19.88	490.677	62.673	13	0.87	39.97	144.967	100.73
6	12	0.85	34.56	52.5268	61.1231	8	0.84	21.29	475.205	66.2259	17	0.83	41.49	322	77.1019
7	12	0.85	35.28	118.697	97.6741	10	0.8	20.31	107.694	76.1852	15	0.85	39.77	136.945	51.0466
8	11	0.8625	34.71	173.379	32.9046	9	0.82	19.87	495.903	70.3995	14	0.86	40.29	496.642	53.9655
9	13	0.8375	35.92	187.67	98.4059	13	0.74	21.43	228.612	37.3401	19	0.81	40.82	362.952	82.1962
10	14	0.825	36.49	72.5523	20.656	11	0.78	20.22	181.502	63.2768	18	0.82	41.09	463.573	54.5603
Ave.	11.6	0.855	35.379	122.434	52.490	10.8	0.784	20.746	322.335	72.967	15.8	0.842	40.646	311.819	80.992
St.dev.	2.170	0.027	0.984	66.381	30.812	1.619	0.032	0.700	148.131	38.923	2.347	0.023	0.690	127.878	28.596

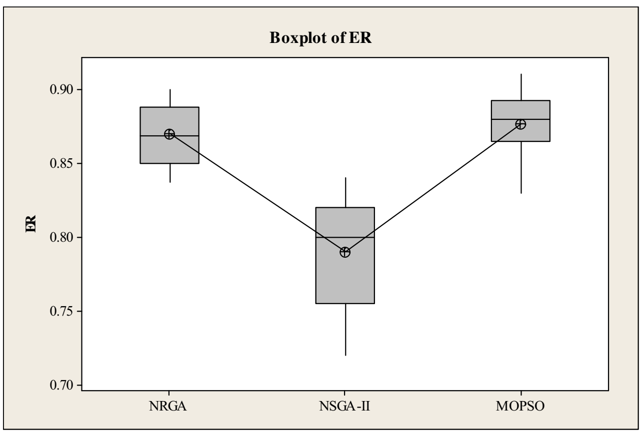

Fig. 15. The plot and the box-plot of the ER metric in numerical example 2.

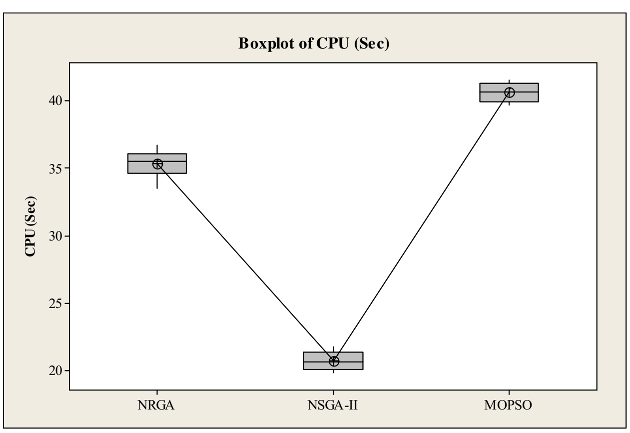

Fig. 16. The plot and the box-plot of the CPU metric in numerical example 2.

Table 22
ANOVA for NPS of numerical example 3.

Source	DF	SS	MS	*F*	*p*-VALUE
Factor	2	144.27	72.13	16.85	0.000
Error	27	115.60	4.28		
Total	29	259.87			

Table 23
ANOVA for ER of numerical example 3.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	0.028580	0.014290	18.35	0.000
Error	27	0.021025	0.000779		
Total	29	0.049605			

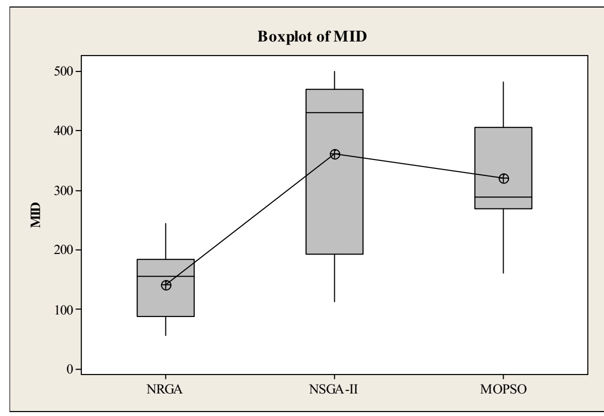

Fig. 17. The plot and the box-plot of the MID metric in numerical example 2.

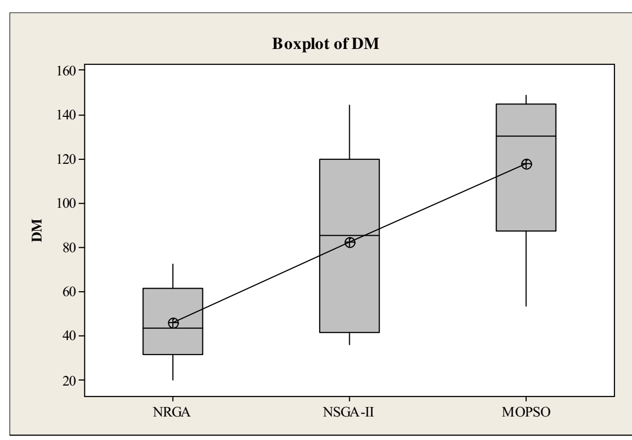

Fig. 18. The plot and the box-plot of the DM metric in numerical example 2.

Table 24
ANOVA for CPU (Sec) of numerical example 3.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	2126.253	1063.127	1646.67	0.000
Error	27	17.432	0.646		
Total	29	2143.685			

Table 25
ANOVA for MID of numerical example 3.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	253124	126562	8.89	0.001
Error	27	384318	14234		
Total	29	637442			

Table 26
ANOVA for DM of numerical example 3.

Source	DF	SS	MS	*F*	*p*-value
Factor	2	4320	2160	1.97	0.158
Error	27	29540	1094		
Total	29	33860			

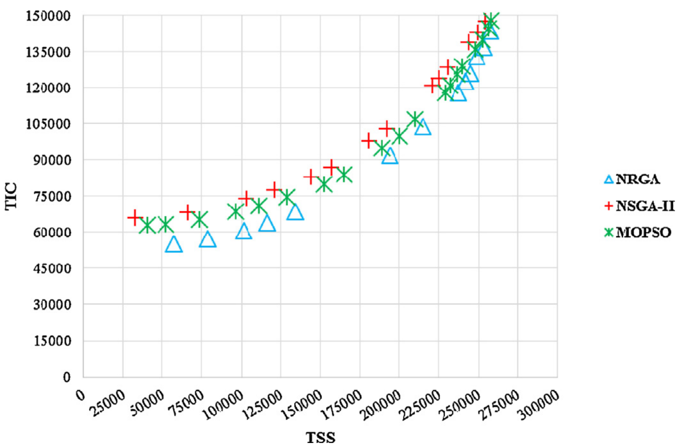

Fig. 19. The best Pareto solution in 10 runs of the algorithms in numerical example 3.

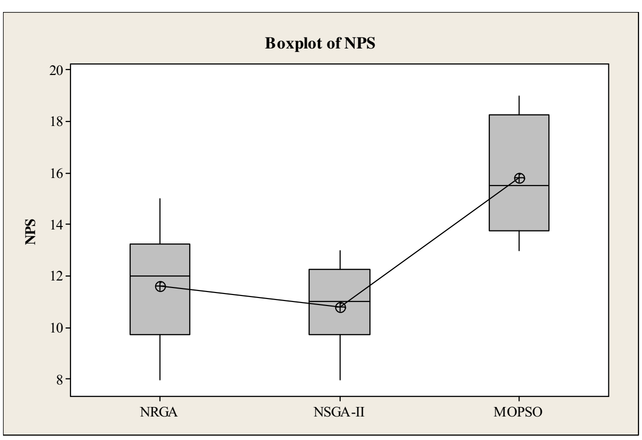

Fig. 20. The plot and the box-plot of the NPS metric in numerical example 3.

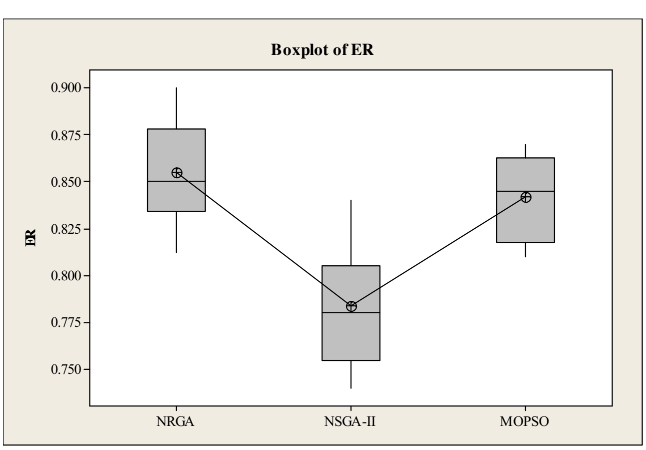

Fig. 21. The plot and the box-plot of the ER metric in numerical example 3.

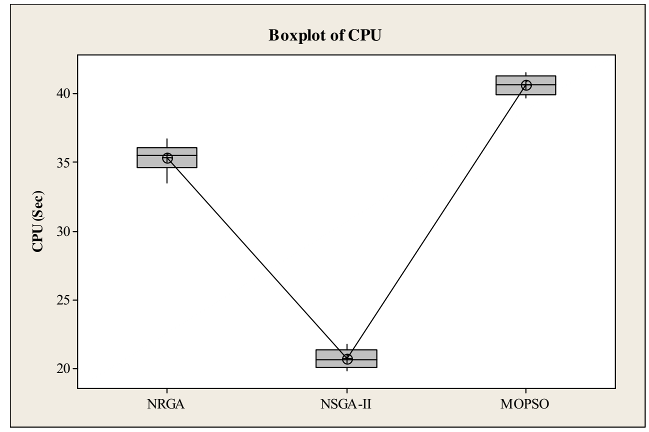

Fig. 22. The plot and the box-plot of the CPU metric in numerical example 3.

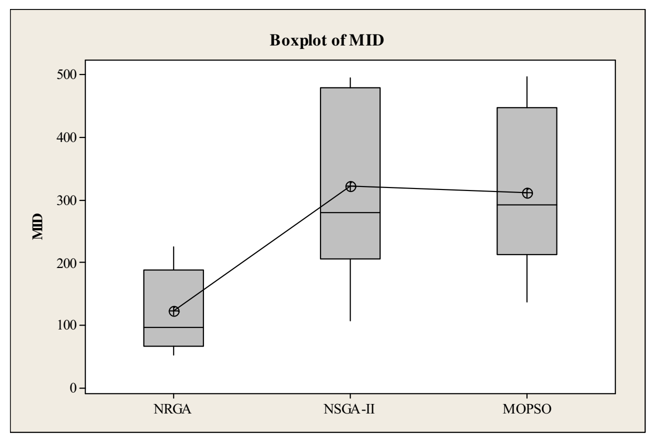

Fig. 23. The plot and the box-plot of the MID metric in numerical example 3.

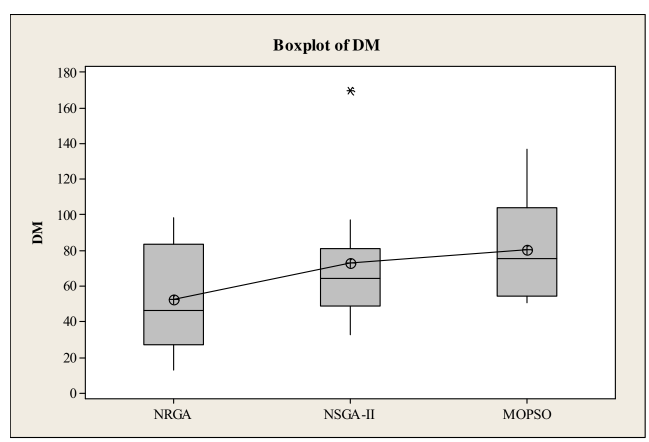

Fig. 24. The plot and the box-plot of the DM metric in numerical example 3.

In addition, Tables 22–26 show the one-way ANOVA of the performance indices NPS, ER, CPU (Sec), MID, and DM at 95% confidence level along with the values of the corresponding p-values. The results in these tables also indicate that while there are significant differences among the three algorithms in terms of the means of NPS, ER, CPU (Sec), and MID, there are no significant differences among the utilized algorithms in term of the means of DM. Moreover Fig. 19 shows the best Pareto solutions of the three algorithms in 10 runs in numerical example 3.

Figs. 20–24 show the plots and box plots of NPS, ER, CPU (Sec), MID, and DM of the three algorithms in numerical example 3.

6. Conclusion and future research directions

In this paper, a closer to reality multi-product inventory control problem was investigated with the goals of minimizing both the total inventory cost and the total required storage space. The number of replenishment cycles was limited, shortages were allowed, and the costs were obtained under inflation. The problem was first formulated into a bi-objective model and then three MOEA, namely NSGA-II, NRGA, and MOPSO, were employed to find Pareto front solutions of the model. Moreover, Taguchi method was implemented to tune the parameters of all algorithms, in which a novel metric that considers both the convergence and the diversity was introduced as the response variable. Three numerical examples were next given to demonstrate the application of the proposed methodology and to evaluate and compare the performances of the three algorithms in terms of NPS, ER, CPU time, MID, and DM criteria. The results showed that the three algorithms had different performances in terms of different metrics. In other words, there found statistically significant differences among the algorithms in terms almost all the performance metrics.

As recommendations for future research, demands or inflation rate can be considered stochastic or fuzzy to make the model's usage more realistic. Moreover, the selling prices of the products can be used and the problem can be modeled into the framework of a supply chain management. For example, one can extend a two-level multi-buyer multi-vendor supply chain problem in which each vendor has storage to store products and each buyer order each product from each vendor.

Acknowledgements

The authors would like to thank the anonymous reviewers, the associate editor, and the editor-in-chief for their constructive comments and suggestions.

References

[1] A.A. Taleizadeh, H. Moghadasi, S.T.A. Niaki, A. Eftekhari, An EOQ-joint replenishment policy to supply expensive imported raw materials with payment in advance, J. Appl. Sci. 8 (2008) 4263–4273.
[2] A.A. Taleizadeh, S.J. Sadjadi, S.T.A. Niaki, Multiproduct EPQ model with single machine, backordering, and immediate rework process, Eur. J. Ind. Eng. 5 (2011) 388–411.
[3] S.H.R. Pasandideh, S.T.A. Niaki, N.A. Roozbeh Niaki, An investigation of vendor managed inventory application in supply chain: the EOQ model with shortage, Int. J. Adv. Manuf. Technol. 49 (2010) 329–339.
[4] S.H.R. Pasandideh, S.T.A. Niaki, S.S. Mirhosseyni, A parameter-tuned genetic algorithm to solve multi-products EPQ model with defective items, rework, and constrained space, Int. J. Adv. Manuf. Technol. 49 (2010) 827–837.
[5] S.H.R. Pasandideh, S.T.A. Niaki, A genetic algorithm approach to optimize a multi-products EPQ model with discrete delivery orders and constrained space, Appl. Math. Comput. 195 (2008) 506–514.
[6] S.H.R. Pasandideh, S.T.A. Niaki, Optimizing the economic production quantity model with discrete delivery orders, J. Econom. Comput. Econom. Cyber. Stud. Res. 44 (2010) 49–62.
[7] A. Saha, A. Roy, S. Kar, M. Maiti, Inventory models for breakable items with stock dependent demand and imprecise constraints, Math. Comp. Model. 52 (2010) 1771–1782.
[8] S. Ahmed, U. Cakmak, A. Shapiro, Coherent risk measures in inventory problems, Eur. J. Operat. Res. 182 (2007) 226–238.
[9] M. Sepehri, Cost and inventory benefits of cooperation in multi-period and multi-product supply, Scientia Iranica 18 (2011) 731–741.
[10] D. Zhang, H. Xu, Y. Wu, Single and multi-period optimal inventory control models with risk-averse constraints, Eur. J. Op. Res. 199 (2009) 420–434.
[11] T.M. Chio, C.H. Chiu, P.L. Fu, Periodic review multiperiod inventory control under a mean–variance optimization objective, IEEE Trans. Syst. Man Cyber. Part A: Syst. Humans 41 (2011) 678–682.
[12] Y.W. Zhou, A multi-warehouse inventory model for items with time-varying demand and shortages, Comp. Op. Res. 30 (2003) 2115–2134.
[13] A.A. Taleizadeh, S.T.A. Niaki, M.B. Aryanezhad, A.F. Tafti, A genetic algorithm to optimize multiproduct multiconstraint inventory control systems with stochastic replenishment intervals and discount, Int. J. Adv. Manuf. Technol. 51 (2010) 311–323.
[14] J.A. Buzacott, Economic order quantities with inflation, Op. Res. Quart. 26 (1975) 553–558.
[15] J.K. Dey, S.K. Mondal, M. Maiti, Two storage inventory problems with dynamic demand and interval valued lead-time over finite time horizon under inflation and time-value of money, Eur. J. Op. Res. 185 (2008) 170–194.
[16] B. Sarkar, L. Moon, An EPQ model with inflation in an imperfect production system, Appl. Math. Comput. 217 (2011) 6159–6167.
[17] A. Mirzazadeh, S.M.T. Fatemi Ghomi, M.M. Seyed Esfahani, A multiple items inventory model under uncertain external inflationary conditions, Trends Appl. Sci. Res. 6 (2011) 472–480.
[18] N.K. Mahapatra, M. Maiti, Decision process for multiobjective, multi-item production- inventory system via interactive fuzzy satisfying technique, Comp. Math. Appl. 49 (2005) 805–821.
[19] K. Maiti, M. Maiti, A numerical approach to a multi-objective optimal inventory control for deteriorating multi-items under fuzzy inflation and discounting, Comp. Math. Appl. 55 (2008) 1794–1807.
[20] Y. Zhang, S. Song, H. Zhang, C. Wu, W. Yin, A hybrid genetic algorithm for two-stage multi-item inventory system with stochastic demand, Neural Comput. Appl. 21 (2011) 1087–1098.
[21] C.H. Tsou, Multi-objective inventory planning using MOPSO and TOPSIS, Expert Syst. Appl. 35 (2008) 136–142.
[22] C.H. Tsou, D.Y. Yang, J.H. Chen, Y.H. Lee, Estimating exchange curve for inventory management through evolutionary multi-objective optimization, Afr. J. Bus. Manage. 5 (2011) 4847–4852.
[23] J. Rezaei, M. Davoodi, Multi-objective models for lot-sizing with supplier selection, Int. J. Prod. Econom. 130 (2011) 77–86.
[24] A. Ojha, A.O. Dos, S. Mondal, M. Maiti, A stochastic discounted multi-objective solid transportation problem for breakable items using analytical hierarchy process, Appl. Math. Model. 34 (2010) 2256–2271.
[25] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2002) 182–197.
[26] S.H. Liao, C.L. Hsieh, A capacitated inventory-location model: Formulation, solution approach and preliminary computational results, Lecture Notes in Comp. Sci. (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) 5579 LNAI (2009) 323–332.
[27] R. Bhattacharya, S. Bandyopadhyay, Solving conflicting bi-objective facility location problem by NSGA II evolutionary algorithm, Int. J. Adv. Manuf. Technol. 51 (2010) 397–414.
[28] X. Yuan, L. Quanfeng, Bicriteria parallel machines scheduling problem with fuzzy due dates based on NSGA-II, Intel. Comput. Intel. Syst. 3 (2010) 520–524.
[29] Y.H. Kang, Z. Zhang, W. Huang, NSGA-II algorithms for multi-objective short-term hydrothermal scheduling, in: Power and Energy Engineering Conference 2009; APPEEC, Asia-Pacific, 2009, pp. 1–5.
[30] L. Amodeo, H. Chen, A. El Hadji, Supply chain inventory optimization with multiple objectives: an industrial case study, Adv. Comput. Intel. Transport Logistics Supply Chain Manage. 144 (2008) 211–230.
[31] S.M. Mousavi, S.T.A. Niaki, N. Alikar, A. Bahreininejad, Two tuned multi-objective meta-heuristic algorithms for solving a fuzzy multi-state redundancy allocation problem under discount strategies, Appl. Math. Model. 39 (2015) 6968–6989.
[32] J. Sadeghi, S.T.A. Niaki, Two parameter tuned multi-objective evolutionary algorithms for a bi-objective vendor managed inventory with trapezoidal fuzzy demand, Appl. Soft Comput. 30 (2015) 567–576.
[33] O. Al Jaddan, L. Rajamani, C.R. Rao, Nondominated ranked genetic algorithm for solving constrained multi-objective optimization problems, J. Theor. Appl. Info. Technol. 5 (2009) 640–651.
[34] H. Moradi, M. Zandieh, I. Mahdavi, Non-dominated ranked genetic algorithm for a multi-objective mixed-model assembly line sequencing problem, Int. J. Prod. Res. 49 (2011) 3479–3499.
[35] K. Rahmani, I. Mahdavi, H. Moradi, H. Khorshidian, M. Solimanpur, A nondominated ranked genetic algorithm for bi-objective single machine preemptive scheduling in just-in-time environment, Int. J. Adv. Manuf. Technol. 55 (2011) 1135–1147.
[36] V. Kayvanfar, M. Zandieh, I. Mahdavi, Economic lot scheduling problem with allowable shortage: a multi-objective approach, Ind. Eng. Eng. Manage. 2 (2011) 920–923.
[37] S.M. Mousavi, S.T.A. Niaki, E. Mehdizadeh, M.R. Tavarroth, The capacitated multi-facility location-allocation problem with probabilistic customer location and demand: two hybrid meta-heuristic algorithms, Int. J. Syst. Sci. 44 (2013) 897–1912.
[38] M. Bagher, M. Zandieh, H. Farsijani, Balancing of stochastic U-type assembly lines: an imperialist competitive algorithm, Int. J. Adv. Manuf. Technol. 54 (2011) 271–285.
[39] S. Forouharfard, M. Zandieh, An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems, Int. J. Adv. Manuf. Technol. 51 (2010) 1179–1193.
[40] A. Ayough, M. Zandieh, H. Farsijani, GA and ICA approaches to job rotation scheduling problem: considering employee's boredom, Int. J. Adv. Manuf. Technol. 60 (2012) 651–666.
[41] S.M. Mousavi, V. Hajipour, S.T.A. Niaki, N. Alikar, Optimizing multi-item multi-period inventory control system with discounted cash flow and inflation: two calibrated meta-heuristic algorithms, Appl. Math. Model. 37 (2012) 2241–2256.

[42] S.M. Mousavi, S.T.A. Niaki, Capacitated location allocation problem with stochastic location and fuzzy demand: a hybrid algorithm, Appl. Math. Model. 37 (2013) 5109–5119.
[43] S.T.A. Niaki, M.J. Ershadi, A parameter-tuned genetic algorithm for statistically constrained economic design of multivariate CUSUM control charts: a Taguchi loss approach, Int. J. Syst. Sci. 43 (2012) 2275–2287.
[44] R. As'ad, K. Demirli, A bilinear programming model and a modified branch-and-bound algorithm for production planning in steel rolling mills with substitutable demand, Int. J. Prod. Res. 49 (2011) 3731–3749.
[45] S.M. Mousavi, V. Hajipour, S.T.A. Niaki, N. Alikar, Optimizing multi-item multi-period inventory control system with discounted cash flow and inflation: two-calibrated meta-heuristic algorithms, Appl. Math. Model. 37 (2013) 2241–2256.
[46] S.M. Mousavi, V. Hajipour, S.T.A. Niaki, N. Alikar, A multi-product multi-period inventory control problem under inflation and discount: a parameter-tuned particle swarm optimization algorithm, Int. J. Adv. Manuf. Technol. 70 (2014) 1739–1756.
[47] M. Hu, J.D. Weir, T. Wu, An augmented multi-objective particle swarm optimizer for building cluster operation decisions, Appl. Soft Comput. 25 (347) (2014) 359.
[48] S.T. Torabi, N. Sahebjamnia, S.A. Mansouri, M.A. Bajestani, A particle swarm optimization for a fuzzy multi-objective unrelated parallel machines scheduling problem, Appl. Soft Comput. 13 (2013) 4750–4762.
[49] H. Ali, W. Shahzad, F.A. Khan, Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization, Appl. Soft Comput. 12 (2012) 1913–1928.
[50] H. Ali, F.A. Khan, Attributed multi-objective comprehensive learning particle swarm optimization for optimal security of networks, Appl. Soft Comput. 13 (2013) 3903–3921.
[51] P. Fattahi, V. Hajipour, A. Nobari, A bi-objective continuous review inventory control model: Pareto-based meta-heuristic algorithms, Appl. Soft Comput. 32 (2122) (2015) 223.
[52] S.H.R. Pasandideh, S.T.A. Niaki, S. Sharafzadeh, Optimizing a bi-objective multi-product EPQ model with defective items, rework and limited orders: NSGA-II and MOPSO algorithms, J. Manuf. Syst. 32 (764) (2013) 770.
[53] J. Kennedy, R. Eberhart, Y. Shi, Swarm intelligence, Morgan Kaufmann Publishers, San Francisco, 2001.
[54] S. Naka, T. Genji, T. Yura, Y. Fukuyam, Practical distribution state estimation using hybrid particle swarm optimization, in: Proceedings of the IEEE Power Engineering Society Winter Meeting, 2001.
[55] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms a comparative case study, in: in: Parallel problem solving from nature—PPSN V, Springer, 1998, pp. 292–301.
[56] D.A. Van Veldhuizen, Multiobjective evolutionary algorithms: Classifications analyzes, and new innovations. Ph.D. dissertation, Department of Electrical and Computer Engineering., Graduate School of Engineering, Air Force Institute Technology, Wright-Patterson AFB, OH, May, 1999.
[57] G.S. Peace, Taguchi methods, Addison-Wesley, 1993.
[58] S.M. Mousavi, S.T.A. Niaki, Capacitated location allocation problem with stochastic location and fuzzy demand: A hybrid algorithm, Appl. Math. Model. 37 (5109) (2013) 5119.