Scientia Iranica E (2018) 25(4), 2331–2346

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
http://scientiairanica.sharif.edu

Multi-machine economic production quantity of scrapped and reworked items considering shortages and allocation decisions

A.H. Nobil[a], A.H. Afshar Sedigh[b], and L.E. Cárdenas-Barrón[c,*]

a. *Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.*
b. *Department of Information Science, University of Otago, Dunedin, New Zealand.*
c. *School of Engineering and Sciences, Tecnológico de Monterrey, Ave. E. Garza Sada 2501 Sur, C.P. 64849, Monterrey, Nuevo León, México.*

Received 10 August 2016; received in revised form 7 February 2017; accepted 17 April 2017

KEYWORDS
EPQ;
Defective item;
MINLP;
Shortage;
Hybrid algorithm.

Abstract. This study considers a multi-product, multi-machine economic production quantity inventory problem in an imperfect production system that produces two types of defective items: items that require rework and scrapped items. The shortage is allowed and fully backordered. The scrapped items are disposed with a disposal cost, and the rework process is done at the end of the normal production period. Moreover, a potential set of available machines for utilization is considered, such that each has a specific production rate per item. Each machine has its own utilization cost, setup time, and production rate per item. The considered constraints are initial capital to utilize machines and production floor space. The proposed inventory model is a mixed integer non-linear programing mathematical model. The problem is solved using a bi-level approach; first, the set of machines to be utilized and the production allocation of items on each machine are obtained through a genetic algorithm. Then, using the convexity attribute of the second level problem, the optimum cycle length per machine is determined. The proposed hybrid genetic algorithm outperformed conventional genetic algorithm and a GAMS solver, considering solution quality and solving time. Finally, a sensitivity analysis is also given.

1. Introduction

The inventory model was first considered in early twentieth century by Harris [1], who introduced the Economic Order Quantity (EOQ) inventory model. Afterwards, industrialization and market competition were the motivations to optimize the inventory system considering production rate and demand. In this direction, Taft [2] improved the EOQ inventory model considering production rate; the result was the Economic Production Quantity (EPQ) inventory model. Pacheco-Velázquez and Cárdenas-Barrón [3] presented an example of an extension of EPQ inventory model by considering backorders and raw material inventory costs.

Inventory models that consider multiple manufacturing products on a machine may date back to the studies performed by Eilon [4] and Rogers [5]. Other pioneering studies of this problem are Bomberger [6], Madigan [7], Stankard and Gupta [8], Hodgson [9], and Baker [10]. A multi-product single-machine problem is considered in studies that are more recent; for example, Taleizadeh et al. [11] considered a service

*. *Corresponding author. Tel.: +52 81 83284235; Fax: +52 81 83284153*
E-mail addresses: amirhossein.nobil@yahoo.com (A.H. Nobil); amir.afshar@postgrad.otago.ac.nz (A.H. Afshar Sedigh); lecarden@tec.mx (L.E. Cárdenas-Barrón)

doi: 10.24200/sci.2017.4453

level constraint when there is a stochastic scrapped production rate and partial backordering. At the same time, Taleizadeh et al. [12] developed the former model while considering random defective items and repair failure.

Afterwards, Taleizadeh et al. [13] studied a multi-product single-machine problem with backorders and rework. Simultaneously, Taleizadeh et al. [14] considered demand uncertainty and a special discount situation. This study, followed by Taleizadeh et al. [15], developed an EPQ inventory model with an immediate rework process. Ramezanian and Saidi-Mehrabad [16] solved a Mixed Integer Nonlinear Programming (MINLP) model to optimize an unrelated parallel machine scheduling problem with multiple products considering imperfect products. Later on, Neidigh and Harrison [17] studied this problem with a non-linear production rate. At the same time, Taleizadeh et al. [18] considered repair failure in a system with random defective items. Subsequently, they [19] considered the rework of a system with budget and service level constraints. Moreover, Taleizadeh et al. [20] considered interruption in the process when the system copes with scrapping and rework processes. On the other hand, Wu and Sung [21] considered a multi-delivery policy in a production system with scrapped items. In recent studies, Pasandideh et al. [22] studied EPQ inventory model with rework and scrapping permission; they considered several classes of rework considering failure severity. Chiu et al. [23] proposed an algebraic method to solve a multi-product problem considering multi-shipment policy with rework. Shafiee-Gol et al. [24] considered pricing and production decisions with rework and discrete delivery.

On the other hand, due to the complexity of supply chain systems, some researchers have utilized heuristics or meta-heuristic algorithms. For instance, Vahdani et al. [25] employed Simulated Annealing (SA) to obtain optimal production size and schedule, considering that there exist one deteriorating item and several warehouses. Pasandideh et al. [26] used Non-dominated Sorting Genetic Algorithm (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO) algorithm to optimize the warehouse space and production cost in an imperfect production system with rework and limited orders. Recently, Forouzanfar et al. [27] employed Bee Colony Optimization (BCO) and Genetic Algorithm (GA) to make decisions about capacity and allocations in a closed-loop supply chain, considering transportation time and production costs. Mahmoodirad and Sanei [28] used three meta-heuristics, i.e., Differential Evolution (DE), Particle Swarm Optimization (PSO), and Gravitational Search Algorithm (GSA), for designing a multi-level solid supply chain with several products. In addition, it is important to mention that multi-product multi-machine problems have been considered in recent studies; for example, Neidigh and Harrison [29] extended their former study. Moreover, studies of Sarkar and Saren [30], Kang et al. [31], and Tayyab and Sarkar [32] are instances of some recent studies that considered the existence of defective items in different environments. Studies of Jaggi et al. [33] and Jaggi et al. [34] are some instances of recent extensions of inventory models related to non-instantaneous deteriorating items in two storage facilities. Finally, Nobil et al. [35] developed a multi-product, multi-machine problem by considering utilization and allocation decisions. They employed a hybrid genetic algorithm using convexity property of a multi-product single-machine problem.

This study presents a multi-product, multi-machine Economic Production Quantity (EPQ) inventory model for an imperfect production system considering allocation of products to machines. The imperfect production system produces two types of poor-quality items, which require rework and scrapping. The rework process is done after the termination of normal production period, and scrapped products are disposed. Moreover, in this study, the shortage is allowed and fully backordered. Further, a potential set of production machines is at hand in which each one can manufacture products at different rates. The utilization price of these machines and their performance are different with a distinct setup time for each product on each machine. The initial capital to buy machines and the production hall are limited. Therefore, in the studied problem, three fundamental questions exist that must be answered in order to minimize the total cost, including utilization, installation, production, rework, shortages, warehouse construction, and holding costs. These questions are as follows:

1. What machines should be utilized?
2. Which items should be allocated to each utilized machine?
3. What are the optimum amounts of production and shortage for each item?

A genetic algorithm is applied to solve this mixed integer linear programming (MINLP) problem. In this paper, first, the MINLP problem is transformed into a bi-level problem, where, in the first level, there are mixed integer linear programming and continuous nonlinear problems in the first and second levels, respectively.

2. Problem statement

This study extends two former studies of Nobil et al. [35] and Pasandideh et al. [26]. On the one hand, Nobil et al. [35] presented a multi-product, multi-machine economic production quantity problem with scrapped items. They studied a defective production system in which some of its products were disposed after normal production time. In their study, several potential machines for allocation of items were considered to be utilized, shortage was not permitted, and some constraints, such as budget and initial capital to buy the machines, were studied. They assumed that the items assigned to each machine have a common cycle time. The study aimed to answer the following questions with regard to minimizing the total cost including the cost of utilization, setup, production, maintenance, and scrapping: What machines should be purchased? What products should be allocated to each machine? How many of each item should be produced?

On the other hand, Pasandideh et al. [26] proposed a model for a single-machine, multi-product economic production quantity problem with defective items. In the studied problem, proportion of defective items requires rework, and the rest are scrapped items. The items requiring rework are classified into different categories based on the failure severity and associated rework rate. In their study, the shortage is permitted and fully backordered, and the items are manufactured on a single machine with limited capacity. The objective of the current study is to find the optimum amount of production and shortage of each item considering total costs minimization including setup, production, rework, scrapping, shortage holding, and warehouse construction costs.

This study considers a multi-product, multi machine economic production quantity problem for a defective production system with respect to the allocation of items to each machine. The defective items may be scrapped or require rework. The objective is to minimize the total cost including utilization, installation, production, rework, scrapping, shortage, holding, and warehouse construction costs. The constraints of this problem are initial capital, floor space for machines, and production machines capacity. Machine i produces item j with production rate of P_{ij}, and the defective items are θ_{ij} percent of the produced items. The ratio of rework required by the produced items is that α_{ij} and μ_{ij} items are scrapped. In other words, $\theta_{ij} = \alpha_{ij} + \mu_{ij}$. Moreover, the items that require rework immediately are repaired after ordinary production cycle with a higher production rate (λ_{ij} times faster). Other assumptions about the proposed inventory model are made as follows:

1. The shortage is permitted and fully backordered;
2. Proportion of defective items need rework and the rest are scrapped;
3. The rework ratio is a coefficient greater than one of ordinary production ratio;
4. Rework costs are different from ordinary production costs;
5. Rework process starts immediately when ordinary production stops and no scrapped items are produced during rework period;
6. The scrapped items should be disposed; therefore, the system faces disposal cost for each scrapped item;
7. The disposal of scrapped items occurs when the ordinary process stops;
8. There are different types of production machines to manufacture items;
9. Decision-maker faces maximum budget and production floor constraints on purchasing production machines;
10. The items allocated to each machine have the same production cycle; in other words, $T_{i1} = T_{i2} = \cdots = T_{in} = T_i$;
11. All parameters of this problem are known;
12. The warehouse space for item j includes storeroom plus passageways. The passageways for item j are a coefficient less than one of its storeroom.

3. Formulated problem

The following parameters, decision variables, and notations are employed in this paper for machines i; $i = 1, 2, \cdots, m$, and items j; $j = 1, 2, \cdots, n$:

m	Number of machines
n	Number of products
D_j	Demand rate of the jth product (units/unit time)
P_{ij}	Production rate of the jth product on machine i (units/unit time)
S_{ij}	Setup time of the ith machine to produce the jth product (unit time)
α_{ij}	Proportion of manufactured reworked items of the jth product on machine i (%)
μ_{ij}	Proportion of manufactured scrapped items of the jth product on machine i (%)
θ_{ij}	Proportion of manufactured imperfect quality products j on machine i (%) $\theta_{ij} = \alpha_{ij} + \mu_{ij}$
δ_{ij}	Binary parameter, $\delta_{ij} = 1$ if $(1-\theta_{ij})P_{ij} - D_j > 0$; otherwise, $\delta_{ij} = 0$
λ_{ij}	Ratio of the rework rate of the jth product to the jth item production rate on machine i ($\lambda_{ij} \geq 1$)

I_j	Maximum on-hand inventory of the jth product based on which the regular production process stops (units)
H_j	Maximum on-hand inventory of the jth product based on which the rework process stops (units)
Δ_j	Space required per unit of the jth product for holding (ft^2/product)
v_j	Ratio of the aisle space to the maximum level of on-hand inventory of the jth product
K_i	Required space of the ith machine (ft^2/machine)
R	Maximum available space for the production floor space (ft^2)
F	Maximum available budget ($)
f_i	Fixed cost of the utilization of the ith machine ($/machine)
A_{ij}	Setup cost of the ith machine to produce the jth product ($/setup)
c_{ij}	Unit production cost of the jth product on machine i ($/unit)
r_{ij}	Unit rework cost of the jth product on machine i ($/unit)
d_j	Disposal cost of scrapped product j per unit ($/unit)
h_j	Unit holding cost of the jth product per unit time ($/unit/unit time)
π_j	Unit backorder cost of the jth product per unit time ($/unit/unit time)
w_j	Unit warehouse construction cost of the jth product per unit space ($/unit)
TC	Total cost ($)
N_i	Number of cycles per unit time for the ith machine; dependent variables
Q_j	Production lot size of the jth product in a cycle (units); dependent variables
B_j	Total shortage quantity of the jth product in a cycle (units); decision variables
T_i	Cycle length of the ith machine (unit time); decision variables
y_i	$y_i = 1$ if machine i is utilized; otherwise, $y_i = 0$; decision variables
x_{ij}	$x_{ij} = 1$ if the jth product manufactured by machine i; otherwise, $x_{ij} = 0$; decision variables.

Figure 1 shows the inventory on hand and shortage of item j in each cycle, which is produced by machine i. During t_j^1 and t_j^5, machines produce the jth item; during t_j^2, t_j^3, and t_j^4 rework of item j, no production or rework is done. Based on Figure 1, these periods in each cycle of product j are calculated using Eqs. (1)-(5) as follows:

$$t_j^1 = \frac{I_j}{(1-\theta_{ij})P_{ij} - D_j} = \frac{Q_j}{P_{ij}} - \frac{B_j}{(1-\theta_{ij})P_{ij} - D_j}, \quad (1)$$

$$t_j^2 = \frac{H_j - I_j}{\lambda_{ij} P_{ij} - D_j} = \frac{\alpha_{ij} Q_j}{\lambda_{ij} P_{ij}}, \quad (2)$$

$$t_j^3 = \frac{H_j}{D_j}, \quad (3)$$

$$t_j^4 = \frac{B_j}{D_j}, \quad (4)$$

$$t_j^5 = \frac{B_j}{(1-\theta_{ij})P_{ij} - D_j}. \quad (5)$$

Moreover, based on Figure 1, it is obvious that:

$$I_j = [(1-\theta_{ij})P_{ij} - D_j]\frac{Q_j}{P_{ij}} - B_j, \quad (6)$$

and:

$$H_j = I_j + \alpha_{ij}(\lambda_{ij} P_{ij} - D_j)\frac{Q_j}{\lambda_{ij} P_{ij}}. \quad (7)$$

Therefore, the cycle length is obtained as follows:

$$T_i = T_{ij} = t_j^1 + t_j^2 + t_j^3 + t_j^4 + t_j^5 = \frac{(1-\mu_{ij})Q_j}{D_j}. \quad (8)$$

Hence:

$$Q_j = \frac{D_j T_i}{(1-\mu_{ij})}. \quad (9)$$

Total cost of the production system is the sum of total utilization cost, total setup cost, total production cost, total rework cost, total disposal cost, total backorder cost, total warehouse construction cost, and total holding cost of all products. These costs are derived as follows:

- Utilization cost: Utilization cost of machine i is equal to f_i, and y_i is the variable that indicates the utilization of this machine. So, the total utilization cost can be calculated as follows:

$$\text{Utilization cost} = \sum_{i=1}^{m} f_i y_i. \quad (10)$$

- Setup cost: The setup cost of machine i to produce item j is equal to A_{ij}. Therefore, the total cost of setting up machines can be obtained through the following equation:

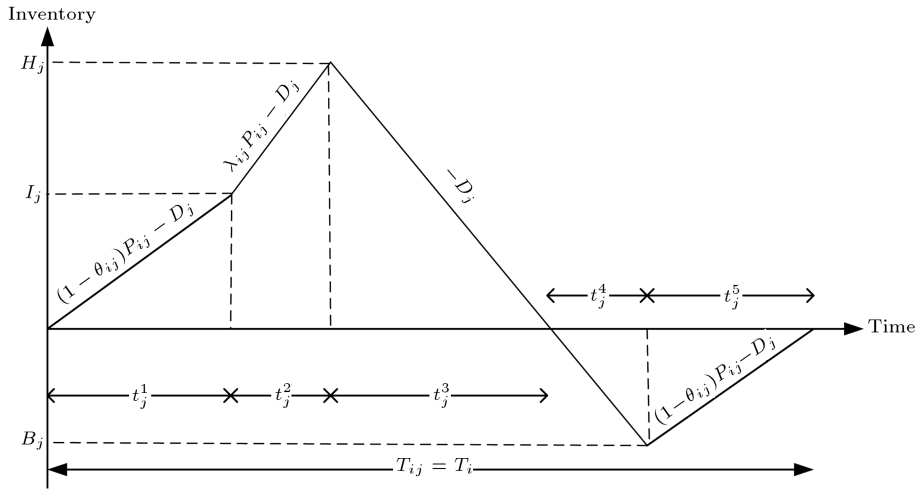

Figure 1. The inventory position of the jth item in a cycle that is produced by machine i.

$$\text{Setup cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}N_i x_{ij}. \tag{11}$$

Based on a joint production policy ($N_i = 1/T_i$):

$$\text{Setup cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{A_{ij}x_{ij}}{T_i}. \tag{12}$$

- Production cost: The amount of items that can be produced per cycle is Q_j; the production cost for each unit of item j on machine i is equal to c_{ij}. Therefore, the total production cost can be obtained as follows:

$$\text{Production cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}N_i x_{ij}Q_j$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{c_{ij}x_{ij}Q_j}{T_i}. \tag{13}$$

Inserting Q_j into Eq. (9) results in:

$$\text{Production cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}N_i x_{ij}Q_j$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{c_{ij}D_j x_{ij}}{(1-\mu_{ij})}. \tag{14}$$

- Rework cost: The items requiring rework per cycle are $\alpha_{ij}Q_j$, and rework cost for item j on machine i is r_{ij}. Thus, the rework cost can be calculated as follows:

$$\text{Rework cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}N_i x_{ij}\alpha_{ij}Q_j$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{r_{ij}x_{ij}\alpha_{ij}D_j x_{ij}}{(1-\mu_{ij})}. \tag{15}$$

- Disposal cost: The amount of disposed items in each cycle equals $\mu_{ij}Q_j$, and disposal cost per unit is d_j. Thus, the disposal cost can be calculated as follows:

$$\text{Disposal cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} d_j N_i x_{ij}\mu_{ij}Q_j$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n} \Big(\frac{d_j\mu_{ij}D_j x_{ij}}{(1-\mu_{ij})}. \tag{16}$$

- Backorder cost: In this study, the shortage is fully backordered; moreover, with respect to Figure 1, the backorder cost for all items can be calculated as follows:

$$\text{Backorder cost} = \sum_{j=1}^{n} \pi_j N_i x_{ij}B_j\left(\frac{t_j^4+t_j^5}{2}\right)$$

$$= \sum_{j=1}^{n} \frac{\pi_j x_{ij}B_j}{T_i}\left(\frac{t_j^4+t_j^5}{2}\right). \tag{17}$$

Substituting t_j^4 and t_j^5 into Eqs. (4) and (5), respectively, results in:

$$\text{Backorder cost} = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(\frac{\pi_j(1-\theta_{ij})P_{ij}}{2(1-\theta_{ij})P_{ij}D_j-D_j^2}\right)$$

$$(x_{ij})\left(\frac{B_j^2}{T_i}\right). \tag{18}$$

- Warehouse construction cost: The decisions about warehouse area are made based on each item's space and aisles requirement. Therefore, the warehouse construction cost for all items can be obtained through the following relation:

$$\text{Construction cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} w_j\Delta_j(1+v_j)H_j x_{ij}. \tag{19}$$

Substituting H_j into Eq. (7) results in:

$$\text{Construction cost} = \sum_{i=1}^{m}\sum_{j=1}^{n}\Bigg[w_j\Delta_j(1+v_j)\left(\frac{((1-\theta_{ij})P_{ij}-D_j)D_j}{1-\mu_{ij})P_{ij}} + \frac{\alpha_{ij}(\lambda_{ij}P_{ij}-D_j)D_j}{(1-\mu_{ij})\lambda_{ij}P_{ij}}\right)T_i - w_j\Delta_j(1+v_j)B_j\Bigg]x_{ij}. \quad (20)$$

- Holding cost: Based on Figure 1, the total holding costs are as follows:

$$\text{Holding cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} N_i h_j\left(\frac{I_j}{2}(t_j^1) + \frac{H_j+I_j}{2}(t_j^2) + \frac{H_j}{2}(t_j^3)\right)x_{ij}. \quad (21)$$

Substituting t_j^1, t_j^2, and t_j^3 into Eqs. (1), (2), and (3), respectively, results in:

$$\text{Holding cost} = \sum_{i=1}^{m}\sum_{j=1}^{n} N_i h_j\left(\frac{I_j}{2}\left(\frac{Q_j}{P_{ij}} - \frac{B_j}{(1-\theta_{ij})P_{ij}-D_j}\right) + \frac{H_j+I_j}{2}\left(\frac{\alpha_{ij}Q_j}{\lambda_{ij}P_{ij}}\right) + \frac{H_j}{2}\left(\frac{H_j}{D_j}\right)\right)x_{ij}. \quad (22)$$

Substituting I_j and H_j into Eqs. (6) and (7), respectively, results in:

Holding cost =

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\Bigg[\frac{h_j((1-\theta_{ij})P_{ij}-D_j)D_j^2}{2P_{ij}^2(1-\mu_{ij})^2}(T_i) + \frac{h_j(\lambda_{ij}P_{ij}-D_j)(\alpha_{ij}D_j)^2}{2\lambda_{ij}^2P_{ij}^2(1-\mu_{ij})^2}(T_i) + \frac{h_j\alpha_{ij}((1-\theta_{ij})P_{ij}-D_j)D_j^2}{\lambda_{ij}P_{ij}^2(1-\mu_{ij})^2}(T_i) + \frac{h_j((1-\theta_{ij})P_{ij}-D_j)^2D_j}{2P_{ij}^2(1-\mu_{ij})^2}(T_i) + \frac{h_jD_j(\alpha_{ij}(\lambda_{ij}P_{ij}-D_j))^2}{2\lambda_{ij}^2P_{ij}^2(1-\mu_{ij})^2}(T_i) + \frac{h_j\alpha_{ij}((1-\theta_{ij})P_{ij}-D_j)(\lambda_{ij}P_{ij}-D_j)D_j}{\lambda_{ij}P_{ij}^2(1-\mu_{ij})^2}(T_i) + \frac{h_j(1-\theta_{ij})P_{ij}}{2(1-\theta_{ij})P_{ij}D_j-D_j^2}\left(\frac{B_j^2}{T_i}\right) - \frac{h_j(1-\theta_{ij})P_{ij}}{P_{ij}(1-\mu_{ij})}(B_j) - \frac{h_j\alpha_{ij}((\lambda_{ij}P_{ij}-D_j)+D_{ij})}{\lambda_{ij}P_{ij}(1-\mu_{ij})}(B_j)\Bigg]x_{ij}. \quad (23)$$

Therefore, based on Eqs. (10), (12), (14), (15), (16), (18), (20), and (23), the total cost is calculated by:

$$TC = \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m}\sum_{j=1}^{n}\Bigg[A_{ij}\left(\frac{1}{T_i}\right) + Z_{ij}^1(T_i) + Z_{ij}^2\left(\frac{B_j^2}{T_i}\right) + Z_{ij}^3 - Z_{ij}^4(B_j)\Bigg]x_{ij}, \quad (24)$$

where:

$$Z_{ij}^1 = w_j\Delta_j(1+v_j)\left(\frac{\lambda_{ij}((1-\theta_{ij})P_{ij}-D_j)D_j+\alpha_{ij}(\lambda_{ij}P_{ij}-D_j)D_j)}{\lambda_{ij}(1-\mu_{ij})P_{ij}}\right) + \frac{h_j(1-\theta_{ij})P_{ij}D_j}{2P_{ij}(1-\mu_{ij})^2} + \frac{h_jD_j\lambda_{ij}P_{ij}(\alpha_{ij})^2(\lambda_{ij}P_{ij}-D_j)}{2\lambda_{ij}^2P_{ij}^2(1-\mu_{ij})^2} + \frac{h_j\alpha_{ij}D_j((1-\theta_{ij})P_{ij}-D_j)}{P_{ij}(1-\mu_{ij})^2} \geq 0, \quad (25)$$

$$Z_{ij}^2 = \frac{(\pi_j+h_j)(1-\theta_{ij})P_{ij}}{2(1-\theta_{ij})P_{ij}D_j-D_j^2} \geq 0, \quad (26)$$

$$Z_{ij}^3 = \frac{(c_{ij}+r_{ij}\alpha_{ij}+d_j\mu_{ij})D_j}{(1-\mu_{ij})} \geq 0, \quad (27)$$

$$Z_{ij}^4 = w_j\Delta_j(1+v_j) + \frac{h_j[\lambda_{ij}(1-\theta_{ij})P_{ij}+\alpha_{ij}((\lambda_{ij}P_{ij}-D_j)+D_{ij})]}{\lambda_{ij}P_{ij}(1-\mu_{ij})} \geq 0. \quad (28)$$

The problem's constraints include allocation, setup, the maximum budget, maximum available floor space, and production capacity. These constraints are expressed as follows:

- Item allocation constraint: This constraint limits the allocation of items, such that each item type will not be produced by more than one machine:

$$\sum_{j=1}^{n} \delta_{ij} x_{ij} = 1 \qquad i = 1, 2, \cdots, m, \tag{29}$$

where δ_{ij} is a binary coefficient that shows the availability of machine i for producing the jth item.

$$\begin{cases} \delta_{ij} = 1 & (1-\theta_{ij})P_{ij} - D_j > 0 \\ \delta_{ij} = 0 & \text{otherwise} \end{cases}$$

- Machine utilization constraint: This constraint limits item production by a utilized machine:

$$x_{ij} \leq y_i \qquad i = 1, 2, \cdots, m; \quad j = 1, 2, \cdots, n. \tag{30}$$

- Budget constraint: Eq. (31) limits machines' utilization cost, such that the total cost would not to be greater than the maximum available budget.

$$\sum_{i=1}^{m} f_i y_i \leq F. \tag{31}$$

- Production floor space constraint: Eq. (32) limits a decision-maker to utilize machines, such that the total required space would not be greater than maximum available space.

$$\sum_{i=1}^{m} K_i y_i \leq R. \tag{32}$$

- Capacity of the single machine constraint: Eq. (33) shows the sum of the production, rework, and setup times for all items manufactured by the ith machine which must be smaller than or equal to the common cycle length of the ith machine:

$$\sum_{j=1}^{n} (t_j^1 + t_j^2 + S_{ij}) x_{ij} \leq T_i \quad i = 1, 2, \cdots, m. \tag{33}$$

Substituting t_j^1 and t_j^2 into Eq. (1) and (2), respectively, results in:

$$T_i \geq \left\{ \frac{\sum_{j=1}^{n} S_{ij} x_{ij} - \sum_{j=1}^{n} \frac{B_j x_{ij}}{(1-\theta_{ij})P_{ij} - D_j}}{1 - \sum_{j=1}^{n} \left(1 + \frac{\alpha_{ij}}{\lambda_{ij}}\right) \frac{D_j x_{ij}}{(1-\mu_{ij})P_{ij}}} = T_i^{\min} \right\}$$

$$i = 1, 2, \cdots, m. \tag{34}$$

Therefore, based on the objective function in Eq. (24) and the constraints in Eqs. (29) to (32), and (34), the proposed MINLP is formulated as follows:

$$\min \quad TC = \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} \left[A_{ij} \left(\frac{1}{T_i} \right) + Z_{ij}^1(T_i) + Z_{ij}^2 \left(\frac{B_j^2}{T_i} \right) + Z_{ij}^3 - Z_{ij}^4(B_j) \right] x_{ij},$$

$$\text{s.t.} \quad \sum_{j=1}^{n} \delta_{ij} x_{ij} = 1 \qquad i = 1, 2, \cdots, m,$$

$$x_{ij} \leq y_i \qquad i = 1, 2, \cdots, m; \qquad j = 1, 2, \cdots, n,$$

$$\sum_{i=1}^{m} f_i y_i \leq F, \qquad \sum_{i=1}^{m} K_i y_i \leq R,$$

$$T_i \geq T_i^{\min} \qquad i = 1, 2, \cdots, m,$$

$$T_i > 0 \qquad i = 1, 2, \cdots, m,$$

$$B_j \geq 0 \qquad j = 1, 2, \cdots, n,$$

$$y_i \in \{0, 1\} \qquad i = 1, 2, \cdots, m,$$

$$x_{ij} \in \{0, 1\} \qquad i = 1, 2, \cdots, m; \qquad j = 1, 2, \cdots, n, \tag{35}$$

where y_i is a binary variable that shows machine utilization, i.e., $y_i = 1$ if machine i is utilized and $y_i = 0$ otherwise. In addition, x_{ij} is a binary variable that shows item allocation to machines, i.e., $x_{ij} = 1$ if item j allocated to machine i is utilized, and $x_{ij} = 0$ otherwise. Moreover, B_j is a continuous decision variable that represents the backorder quantity of item j. Finally, T_i is a continuous decision variable that represents the cycle time of machine i.

4. Hybrid solution procedure

In this study, a heuristic method is employed to solve the proposed mixed integer non-linear programming model. It uses a genetic algorithm and the convexity attribute of a single machine problem to find a near-optimal solution. In this method, for the first step, Problem (35) is converted into a bi-level problem as follows:

$$\min \quad TC = \sum_{i=1}^{m} f_i y_i + \varphi(x_{ij}, T_i, B_j),$$

$$\text{s.t.} \quad \sum_{j=1}^{n} \delta_{ij} x_{ij} = 1 \qquad i = 1, 2, \cdots, m,$$

$$x_{ij} \leq y_i \qquad i = 1, 2, \cdots, m; \qquad j = 1, 2, \cdots, n,$$

$$\sum_{i=1}^{m} f_i y_i \leq F, \qquad \sum_{i=1}^{m} K_i y_i \leq R,$$

$$y_i \in \{0,1\} \qquad i=1,2,\cdots,m,$$

$$x_{ij} \in \{0,1\} \qquad i=1,2,\cdots,m; \qquad j=1,2,\cdots,n, \tag{36}$$

and:

$$\min \quad \varphi(x_{ij},T_i,B_j) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left[A_{ij}\left(\frac{1}{T_i}\right) + Z_{ij}^1(T_i) + Z_{ij}^2\left(\frac{B_j^2}{T_i}\right) + Z_{ij}^3 - Z_{ij}^4(B_j)\right]x_{ij},$$

$$\text{s.t.} \quad T_i \geq \frac{\sum_{j=1}^{n} S_{ij}x_{ij} - \sum_{j=1}^{n}\frac{B_j x_{ij}}{(1-\theta_{ij})P_{ij}-D_j}}{1-\sum_{j=1}^{n}\left(1+\frac{\alpha_{ij}}{\lambda_{ij}}\right)\frac{D_j x_{ij}}{(1-\mu_{ij})P_{ij}}}$$

$$i=1,2,\cdots,m,$$

$$T_i > 0 \qquad i=1,2,\cdots,m,$$

$$B_j \geq 0 \qquad j=1,2,\cdots,n. \tag{37}$$

The first level of this problem is a mixed integer nonlinear model, which can be solved using a genetic algorithm. In this level, the decision variables are x_{ij} and y_i; therefore, the number of variables equals $m(n+1)$, and the number of constraints is $m(n+1)+2$. After obtaining x_{ij} and y_i values, the second level problem is solved considering x_{ij} as an input parameter. The problem to be solved in the second level (Eq. (37)) is a non-linear continuous problem that can be optimized using derivatives. The decision variables in the second level are T_i and B_j; thus, the number of variables and constraints is $m+n$ and m, respectively.

Using the bi-level procedure, utilization and allocation are obtained randomly (using GA rules) in Eq. (36). Then, knowing that Eq. (37) is a convex NLP (see Appendix A), the optimum cycle length per machine and shortage value per item are calculated by derivatives method. Using this method, the necessity of searching for the optimal solution through continuous variables can be obviated. A general scheme of obtaining a new solutions' structure (chromosomes) is represented in Figure 2. In addition, the following steps for derivatives method are employed:

1. If $\sum_{j=1}^{n} S_{ij}x_{ij} - \sum_{j=1}^{n}\frac{B_j x_{ij}}{(1-\theta_{ij})P_{ij}-D_j}$ and $1-\sum_{j=1}^{n}(1+\frac{\alpha_{ij}}{\lambda_{ij}})\frac{D_j x_{ij}}{(1-\mu_{ij})P_{ij}}$ are simultaneously either positive or negative, then go to step 2. Otherwise, the solution is infeasible, and go to step 7;
2. If $(\sum_{j=1}^{n} Z_{ij}^1 x_{ij} - \sum_{j=1}^{n}\frac{(Z_{ij}^4)^2}{4Z_{ij}^2}x_{ij})$ is positive, then go to Step 3. Otherwise, the solution is infeasible, and go to Step 7;

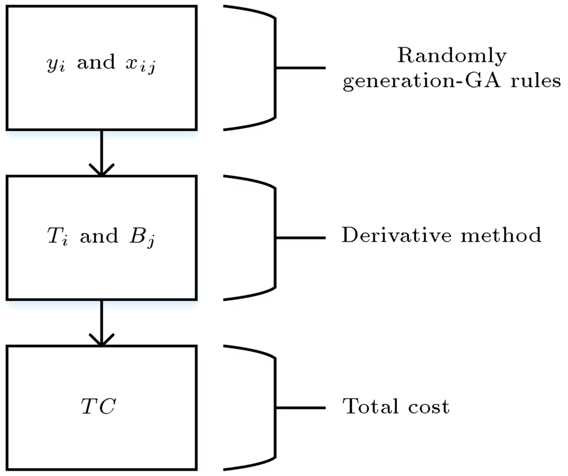

Figure 2. The structure of each solution (chromosome).

3. The following T_i and B_j are calculated based on x_{ij} value (a detailed calculation is represented in Appendix B):

$$T_i = \sqrt{\frac{\sum_{j=1}^{n} A_{ij}x_{ij}}{\left(\sum_{j=1}^{n} Z_{ij}^1 x_{ij} - \sum_{j=1}^{n}\frac{(Z_{ij}^4)^2}{4Z_{ij}^2}x_{ij}\right)}} \qquad i=1,2,\cdots,m, \tag{38}$$

$$B_j = \sum_{i=1}^{m}\frac{Z_{ij}^4 x_{ij}}{2Z_{ij}^2}T_i \qquad j=1,2,\cdots,n. \tag{39}$$

4. The lower bound of $T_i^{\min}$ is obtained as follows:

$$T_i^{\min} = \frac{\sum_{j=1}^{n} S_{ij}x_{ij} - \sum_{j=1}^{n}\frac{B_j x_{ij}}{(1-\theta_{ij})P_{ij}-D_j}}{1-\sum_{j=1}^{n}\left(1+\frac{\alpha_{ij}}{\lambda_{ij}}\right)\frac{D_j x_{ij}}{(1-\mu_{ij})P_{ij}}} \qquad i=1,2,\cdots,m. \tag{40}$$

5. If $T_i \geq T_i^{\min}$, then go to $T_i^* = T_i$. Otherwise, $T_i^* = T_i^{\min}$ and go to step 6;
6. Based on the value of T_i^*, obtain B_j^* using Eq. (39) and go to step 7.
7. Terminate the procedure.

Regarding Figure 2, y_i and x_{ij} are obtained randomly using GA, and then optimum values of T_i and B_j are calculated based on x_{ij} by the derivative method. Then, the inventory system's total cost is determined based on these values. To solve the MINLP problem, a Hybrid Genetic Algorithm (HGA) is employed that combines GA and the derivatives method. The solution procedure of the proposed inventory model using the hybrid genetic algorithm is as follows:

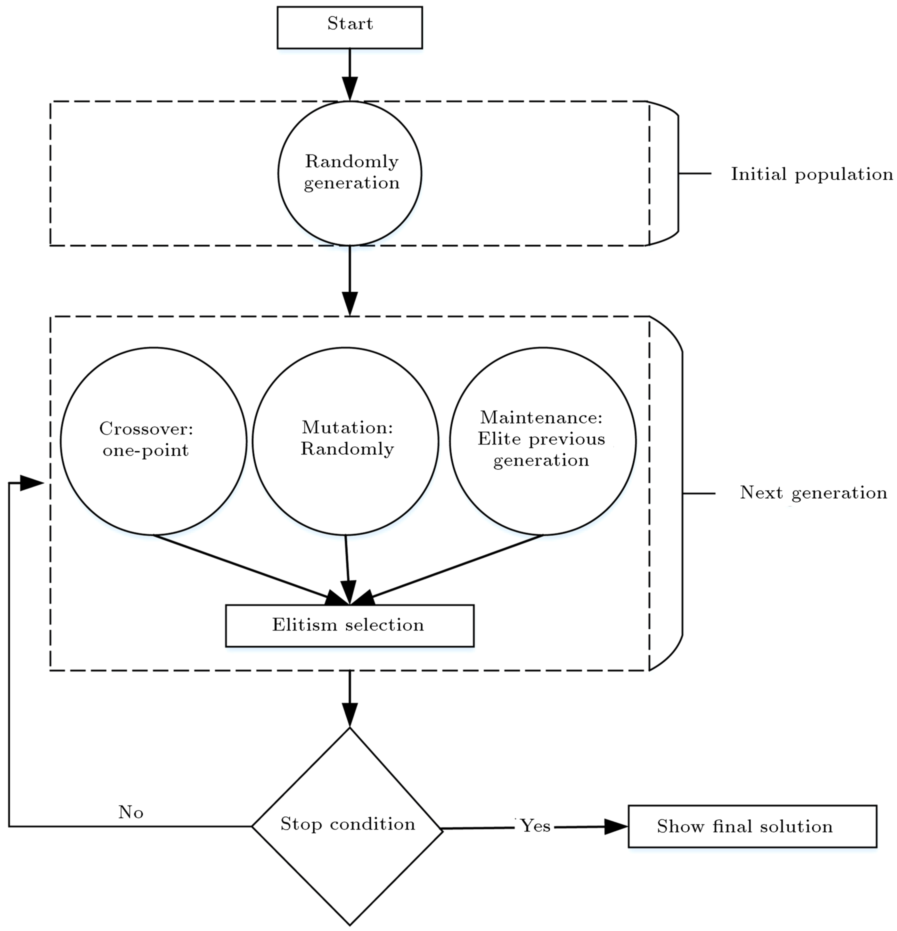

Figure 3. The flowchart of the proposed Hybrid Genetic Algorithm (HGA).

1. In the genetic algorithm, the chromosomes are produced randomly in the first step of the algorithm, called the initial generation production;
2. After producing the initial generation, the crossover and mutation operators are used to reproduce new chromosomes. Further, a percentage of better chromosomes of the last generation is saved;
3. Elite selection transfers the best chromosomes to the next generation per iteration;
4. This procedure repeats until solution convergence is met.

A flowchart and a general procedure of the proposed hybrid genetic algorithm are proposed in Figures 3 and 4, respectively.

4.1. Initial definitions

- Chromosome: Solutions of model are called chromosomes;
- Generation population: Number of chromosomes in an iteration ($npop$);
- Crossover probability (Pc): Crossover chance per chromosome. Therefore, $Pc \times npop$ is the number of crossovers per generation;
- Mutation probability (Pm): Mutation chance per chromosome. Therefore, $Pm \times npop$ is the number of mutations per generation;
- Maintenance probability (Pr): The probability of maintenance per generation. Therefore, $Pr \times npop$ is the number of maintained chromosomes per generation.

4.2. Chromosome scheme

Each chromosome is composed of a number of genes. These problem solutions, i.e., chromosomes, consist of two types of genes: x_{ij} and y_i. A schematic overview of an arbitrary chromosome is represented in Figure 5.

4.3. Initial population

$npop$ chromosomes are generated randomly and stored in POP to form an initial generation. To do so, random numbers from $\{0, 1\}$ are generated considering Constraints (31) and (32) to create genes related to y_i. Afterwards, random numbers from $\{0, 1\}$ are generated considering Constraints (29) and (30) to create genes

Preliminary
Initialize POP(K); POP(K): population of chromosome of the Kth generation.
Evaluate all chromosomes in POP(K)
Chromosomes of the POP(1) are generated randomly (initial population).
While (termination condition is not true) **do**;
//Select probabilistically of y_i and x_{ij}
Crossover: crossover the pairs of chromosomes in POP(K) to yield P1(K) by employing a crossover probability, then go to **second level**
Mutation: mutation of the genes of POP(K) to yield P2(K) by employing a mutation probability, then go to **second level**
Maintenance: the elite chromosomes of POP(K) to yield P3(K) by employing a maintenance probability, then go to **second level**
Each chromosome (y_i, x_{ij}) is fed to the second level to compute $\varphi(x_{ij}, T_i, B_j)$
Start (second level)
For each chromosome
Running the derivatives method
Compute $\varphi(x_{ij}, T_i, B_j)$
End
Objective function is calculated for each chromosome $TC(y_i, x_{ij}, T_i, B_j)$
Evaluate pop1(K); pop2(K); pop3(K) and pop4(K)
End while
Output the resulting of the best chromosome
Termination

Figure 4. The proposed hybrid genetic algorithm's general procedure.

$j \downarrow i \rightarrow$	1	2	$\cdots$	m
1	x_{11}	x_{21}	$\cdots$	x_{m1}
2	x_{21}	x_{22}	$\cdots$	x_{m2}
$\vdots$	$\vdots$	$\vdots$		$\vdots$
n	x_{1n}	x_{2n}	$\cdots$	x_{mn}
	y_1	y_2	$\cdots$	y_m

Figure 5. The example of chromosome.

related to x_{ij}. Moreover, some rules to improve feasibility and optimality are employed as follows:

1. If $y_i = 1$, then at least an item should be produced on machine i;
2. An item should be set up only on one machine;
3. Machines investment ($\sum_{\forall i} f_i y_i$) should not exceed the total budget;
4. Whole floor space that machines occupy, i.e., ($\sum_{\forall i} K_i y_i$), should be less than or equal to available floor space.

4.4. Crossover operator

One-point crossover is applied to generate two offspring (new chromosomes) from two parents (randomly chosen chromosomes). To do so, a random number from $\{1, m-1\}$ is generated and parents are cut from there for both y_i and x_{ij}. This method is able to generate $P2 = Pc \times npop$ new chromosomes, such that all $P2$ members are feasible.

4.5. Mutation operator

A randomly chosen offspring is mutated by reversing a random gene of y_i based on Eq. (41). Then, considering Constraints (29) and (30), $P3 = Pm \times npop$ genes are generated randomly from $\{0, 1\}$ to form x_{ij}, such that all $P3$ members are feasible:

$$y_i^{\text{offspring}} = 1 - y_i^{\text{parent}}. \tag{41}$$

4.6. Maintenance operator

$P4 = Ph \times npop$ chromosomes from a former generation are maintained.

4.7. Fitness function

T_i and B_j are computed for a chromosome based on y_i and x_{ij} employing the derivatives method (the second level). Moreover, chromosomes' fitness function can be calculated as follows:

$$\min TC = \sum_{i=1}^{m} f_i y_i + \varphi(x_{ij}, T_i, B_j),$$

$$\varphi(x_{ij}, T_i, B_j) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left[A_{ij} \left(\frac{1}{T_i} \right) + Z_{ij}^1(T_i) + Z_{ij}^2 \left(\frac{B_j^2}{T_i} \right) + Z_{ij}^3 - Z_{ij}^4(B_j) \right] x_{ij}. \tag{42}$$

It is possible to have infeasible solutions during calculation of T_i and B_j using the derivatives method, i.e., the first step. As a result, for infeasible solutions (chromosomes), the following penalty function to the fitness function is added:

$$\text{Penalty} = -inf, \tag{43}$$

where inf is a sufficiently large number.

4.8. Selection operator

$P2$, $P3$, and $P4$ are merged per iteration to obtain $npop$ better solutions featuring smaller fitness functions and forming next generation. Finally, if there is no obvious enhancement with respect to the solutions for nIt iterations, then the algorithm is terminated.

Table 1. Comparison of algorithms.

	Size	Proposed HGA		Conventional GA		GAMS solver Couenne	
	$n \times m$	Total cost ($)	CPU time (second)	Total cost ($)	CPU time (second)	Total cost ($)	CPU time (second)
1	2×2	127565.7781971	45.6279	127565.7781971	87.6438	127565.7781971	0.0202
2	2×3	136978.5888335	49.8007	136982.7001745	92.2613	136980.9735568	0.0989
3	2×5	165920.4156557	56.4961	165997. 3489096	102.2991	—	—
4	3×6	284632.6823419	59.5632	288915.0912641	118.2832	—	—
5	3×10	289473.8272197	61.7463	292038.0537104	147.2121	—	—
6	4×10	494736.1758372	68.5492	529472.6951114	167.0090	—	—
7	5×12	542800.5102909	85.5382	581082.0475678	249.3129	—	—
8	6×15	905370.2278785	96.1931	951113.2076686	320.4512	—	—
9	6×20	1022337.5288482	107.7371	1126859.6852671	352.6391	—	—
10	7×25	1291611.2514883	156.0021	1380281.4415222	423.8735	—	—

Table 2. Input parameters of the proposed problem.

$P_{ij} \sim U(15000, 25000); \alpha_{ij} \sim U(0.01, 0.05); \mu_{ij} \sim U(0.001, 0.007); \lambda_{ij} \sim U(1, 4); A_{ij} \sim U(100, 300)$
$f_i \sim U(120000, 220000); D_j \sim U(1000, 3000); h_j \sim U(10, 20); w_j \sim U(2, 10); \pi_j \sim U(2, 10); \Delta_j \sim U(2, 5)$
$v_j \sim U(0.2, 0.8); c_{ij} \sim U(200, 300); r_{ij} \sim U(10, 30); d_j \sim U(20, 40); S_{ij} \sim U(0.03, 0.08);$
$F \sim U(300000, 400000); R \sim U(1000, 2000); K_i \sim U(400, 800)$

Note: U is uniform distribution.

5. Numerical examples

In this section, the numerical results of the proposed mixed integer linear programming problem and sensitivity analysis are discussed. The model is solved using three different approaches, i.e., hybrid genetic algorithm, conventional genetic algorithm, and a software, i.e., GAMS solver Couenne, for 10 sample problems of different sizes proposed in Table 1. In these 10 instances, the input parameters are randomly selected from Table 2. As seen from Table 1, for small-sized instances, three methods obtain almost equal solutions. However, increasing problem dimension affects the accuracy and running time, and solutions' quality decreases dramatically. Contrary to GAMS, the conventional GA and proposed hybrid GA can obtain an acceptable solution to large dimensions of this MINLP problem. By comparing the results of two GAs, it is obvious that the proposed hybrid GA outperforms conventional GA in terms of solution quality and solving time. Therefore, based on Table 1, it can be concluded that the proposed hybrid genetic algorithm has an appropriate performance for this MINLP.

There is a vast literature concerning sensitivity analysis of multi-product non-linear problems on the effect of production rate, demand rate, and cost parameters on inventory system total costs (see [18,19,22,35]). Therefore, this problem focuses on the parameters central to the production system which have an impact on total costs. These parameters are as follows: rework ratio, disposal ratio, and rework speed. To do so, a 2×5 problem is considered, and these three parameters change based on Table 3 to study their impact on the total cost and final solution. Based on Table 3, it is obvious that increasing rework speed decreases the total cost. By contrast, increasing rework and scrapping ratio leads to higher system costs.

6. Conclusion and future research directions

This paper proposes a Mixed Integer Non-Linear Programing (MINLP) mathematical model and its optimization procedure for a multi-product, multi-machine Economic Production Quantity (EPQ) inventory problem in an imperfect production environment. The considered system produced two types of defective items: items that need rework and scrapped items. The shortage was allowed such that demands for unavailable items were totally backordered. The scrapped items were disposed with a disposal cost, and rework process was done after finishing the normal production period. Moreover, the system was considered when there was a potential set of available machines with a specific production rate along with its utilization cost as well as setup time per item. This system was studied under some constraints such as initial capital for machines' utilization and production floor space. A bi-level method was used to solve this problem. First, a set of machines to be utilized and a production allocation of items to each machine were obtained

Table 3. Sensitivity analysis of three parameters: rework ratio, disposal ratio, and rework speed.

Parameter	% changes	The proposed HGA	
		Total cost ($)	**% changes in**
Initial problem	0	165920.41565577	0
	+100	166180.509967279	1.00156758473923
α_{ij}	–50	165788.089543842	0.999202472393738
	–100	165654.260980893	0.998395889536408
	+100	166016.570453697	1.00057952360803
μ_{ij}	–50	165872.699892375	0.999712417768444
	–100	165825.222680232	0.999426273281912
	+100	165858.860829722	0.999629009933439
λ_{ij}	+50	165879.354287614	0.999752523714495
	–50	166044.198464805	1.00074603724048

by the genetic algorithm. Then, using the convexity attribute of the second level problem, the final solution was calculated. Furthermore, a comparison made among the proposed hybrid genetic algorithm, a conventional genetic algorithm, and a GAMS solver was presented. The outcomes suggest that the proposed method outperformed other methods considering both solution quality and solving time. The proposed inventory model was applicable to real-world instances in which corporate's managers deal with procurement of facilities as well as their allocation to corporate operations. In the aforementioned environment, a manager should make a decision based on technological advantages of available facilities, i.e., production rate, failure rate, rework rate, and so forth. Moreover, another concern that a decision-maker should take into the account is the demand rate of each item, shortage costs, budget constraint, desired quality, and facilities production and procurement costs. In this study, a mathematical model was proposed to make a near-optimal decision about facilities procurement and product allocation considering system costs. To do so, a multi-item production system with defective products and shortage with a set of potential machines was optimized by utilizing single machine specifications and GA altogether.

It is significant to consider other objectives, such as maximizing the profit or minimizing the warehouse space, as some extensions of the proposed problem. Moreover, considering some environmental considerations in the form of manager's preference or *internalizing the externalities* in choosing production facilities is another possible future extension of this model. The solution procedure may be extended by composing combinatorial optimizations and convexity attribute of the problem. However, it is required to branch on both y and x variables, i.e., binary variables; then, a convex problem similar to the mentioned problem will emerge. It is worth mentioning that the proposed method can provide an estimation of each branch solution and its quality. On the other hand, this problem can be studied under different conditions: perishable items, several classes of rework, and aggregate rework. Further, all rework processes of items produced by machine i should be done in a separate cycle. Moreover, the following are some interesting future research directions: interruption in machines' manufacturing processes, maintenance policy consideration, stochastic production failure, discount on imperfect products and fuzzy/stochastic demands or capacity, or a combination of these.

References

1. Harris, F.W. "How many parts to make at once", *Factory, The Magazine of Management*, **10**(2), pp. 135-136, and 152 (1913).

2. Taft, E.W. "The most economical production lot", *Iron Age*, **101**(18), pp. 1410-1412 (1918).

3. Pacheco-Velázquez, E.A., and Cárdenas-Barrón, L.E. "An economic production quantity inventory model with backorders considering the raw material costs", *Scientia Iranica E*, **23**(2), pp. 736-746 (2016).

4. Eilon, S. "Scheduling for batch production", *Journal of Institute of Production Engineering*, **36**, pp. 549-570 and 582 (1957).

5. Rogers, J.A. "Computational approach to the economic lot scheduling problem", *Management Science*, **4**(3), pp. 264-291 (1958).

6. Bomberger, E.E. "A dynamic programming approach to a lot size scheduling problem", *Management Science*, **12**(11), pp. 778-784 (1966).
7. Madigan, J.G. "Scheduling a multi-product single machine system for an infinite planning period", *Management Science*, **14**(11), pp. 713-719 (1968).
8. Stankard, M.F. and Gupta, S.K. "A note on Bomberger's approach to lot size scheduling: Heuristic proposed", *Management Science*, **15**(7), pp. 449-452 (1969).
9. Hodgson, T.J. "Addendum to Stankard and Gupta's note on lot size scheduling", *Management Science*, **16**(7), pp. 514-517 (1970).
10. Baker, K.R. "On Madigan's Approach to the deterministic multi-product production and inventory problem", *Management Science*, **16**(9), pp. 636-638 (1970).
11. Taleizadeh, A.A., Wee, H.M., and Sadjadi, S.J. "Multi-product production quantity model with repair failure and partial backordering", *Computers and Industrial Engineering*, **59**(1), pp. 45-54 (2010).
12. Taleizadeh, A.A., Niaki, S.T.A., and Najafi, A.A. "Multiproduct single-machine production system with stochastic scrapped production rate, partial backordering and service level constraint", *Journal of Computational and Applied Mathematics*, **233**(8), pp. 1834-1849 (2010).
13. Taleizadeh, A.A., Sadjadi, S.J., and Niaki, S.T.A. "Multiproduct EPQ model with single machine, backordering and immediate rework process", *European Journal of Industrial Engineering*, **5**(4), pp. 388-411(2011).
14. Taleizadeh, A.A,. Shavandi, H., and Haji, R. "Constrained single period problem under demand uncertainty", *Scientia Iranica E*, **18**(6), pp. 1553-1563 (2011).
15. Taleizadeh, A.A., Cárdenas-Barrón, L.E., Biabani, J., and Nikousokhan, R. "Multi products single machine EPQ model with immediate rework process", *International Journal of Industrial Engineering Computations*, **3**(2), pp. 93-102 (2012).
16. Ramezanian, R. and Saidi-Mehrabad, M. "Multi-product unrelated parallel machines scheduling problem with rework processes", *Scientia Iranica E*, **19**(6), pp. 1887-1893 (2012).
17. Neidigh, R.O. and Harrison, T.P. "Optimising lot sizing with nonlinear production rates in a multi-product single-machine environment", *International Journal of Production Research*, **51**(12), pp. 3561-3573 (2013).
18. Taleizadeh, A.A., Wee, H.M., and Jalali-Naini, S.G. "Economic production quantity model with repair failure and limited capacity", *Applied Mathematical Modelling*, **37**(5), pp. 2765-2774 (2013a).
19. Taleizadeh, A.A., Jalali-Naini, S.G., Wee, H.M., and Kuo, T.C. "An imperfect multi-product production system with rework", *Scientia Iranica E*, **20**(3), pp. 811-823 (2013b).
20. Taleizadeh, A.A., Cárdenas-Barrón, L.E., and Mohammadi, B.A. "deterministic multi product single machine EPQ model with backordering, scraped products, rework and interruption in manufacturing process", *International Journal of Production Economics*, **150**, pp. 9-27 (2014).
21. Wu, M.F. and Sung, P.C. "Optimization of a multi-product EPQ model with scrap and an improved multi-delivery policy", *Journal of Engineering Research*, **2**(4), pp. 1-16 (2014).
22. Pasandideh, S.H.R., Niaki, S.T.A., Nobil, A.H., and Cárdenas-Barrón, L.E. "A multiproduct single machine economic production quantity model for an imperfect production system under warehouse construction cost", *International Journal of Production Economics*, **169**, pp. 203-214 (2015).
23. Chiu, S.W., Sung, P.C., Tseng, C.T., and Chiu, Y.S.P. "Multi-product FPR model with rework and multi-shipment policy resolved by algebraic approach", *Journal of Scientific and Industrial Research*, **74**(10), pp. 555-559 (2015).
24. Shafiee-Gol, S., Nasiri, M.M., and Taleizadeh, A.A. "Pricing and production decisions in multi-product single machine manufacturing system with discrete delivery and rework", *OPSEARCH*, **53**(4), pp. 873-888 (2016).
25. Vahdania, M., Dolatib, A., and Bashiria, M. "Single-item lot-sizing and scheduling problem with deteriorating inventory and multiple warehouses", *Scientia Iranica E*, **20**(6), pp. 2177-2187 (2013).
26. Pasandideh, S.H.R., Niaki, S.T.A., and Sharafzadeh, S. "Optimizing a bi-objective multi-product EPQ model with defective items, rework and limited orders: NSGA-II and MOPSO algorithms", *Journal of Manufacturing Systems*, **32**(4), pp. 764-770 (2013).
27. Forouzanfar, F., Tavakkoli-Moghaddam, R., Bashiri, M., and Baboli, A. "A new bi-objective model for a closed-loop supply chain problem with inventory and transportation times", *Scientia Iranica E*, **23**(3), pp. 1441-1458 (2016).
28. Mahmoodirad, A. and Sanei, M. "Solving a multi-stage multi-product solid supply chain network design problem by meta-heuristics", *Scientia Iranica E*, **23**(3), pp. 1428-1440 (2016).
29. Neidigh, R.O. and Harrison, T.P. "Optimising lot sizing with nonlinear production rates in a multi-product multi-machine environment", *International Journal of Production Research*, **57**(4), pp. 939-959 (2017).
30. Sarkar, B. and Saren, S. "Product inspection policy for an imperfect production system with inspection errors and warranty cost", *European Journal of Operational Research*, **248**(1), pp. 263-271 (2016).
31. Kang, C.W., Ullah, M., Sarkar, B., Hussain, I., and Akhtar, R. "Impact of random defective rate on lot size focusing work-in-process inventory in manufacturing system", *International Jour-*

nal of Production Research, pp. 1-19 (2016). DOI http://dx.doi.org/10.1080/00207543.2016.1235295.

32. Tayyab, M. and Sarkar, B. "Optimal batch quantity in a cleaner multi-stage lean production system with random defective rate", *Journal of Cleaner Production*, **139**, pp. 922-934 (2016).

33. Jaggi, C.K., Tiwari, S., and Goel, S.K. "Credit financing in economic ordering policies for non-instantaneous deteriorating items with price dependent demand and two storage facilities", *Annals of Operations Research*, **248**(1) pp. 253-280 (2017).

34. Jaggi, C.K., Tiwari, S., and Goel, S.K. "Replenishment policy for non-instantaneous deteriorating items in a two storage facilities under inflationary conditions", *International Journal of Industrial Engineering Computations*, **7**(3), pp. 489-506 (2016).

35. Nobil, A.H., Sedigh, A.H.A., and Cárdenas-Barrón, L.E. "A multi-machine multi-product EPQ problem for an imperfect manufacturing system considering utilization and allocation decisions", *Expert Systems with Applications*, **56**, pp. 310-319 (2016).

Appendix A

Proof of the convexity of the objective function in Eq. (37)

Based on the objective function in Eq. (37):

$$\varphi = \sum_{i=1}^{m}\sum_{j=1}^{n}\left[A_{ij}\left(\frac{1}{T_i}\right) + Z_{ij}^1(T_i) + Z_{ij}^2\left(\frac{B_j^2}{T_i}\right) + Z_{ij}^3 - Z_{ij}^4(B_j)\right]x_{ij}.$$

So:

$$\frac{\partial\varphi}{\partial T_i} = \sum_{j=1}^{n} Z_{ij}^1 x_{ij} - \sum_{j=1}^{n}\frac{(A_{ij}+Z_{ij}^2B_j^2)}{(T_i)^2}x_{ij};$$

$$\frac{\partial^2\varphi}{\partial^2 T_i} = \sum_{j=1}^{n}\frac{2(A_{ij}+Z_{ij}^2B_j^2)x_{ij}}{(T_i)^3};$$

$$\frac{\partial^2\varphi}{\partial T_i\partial T_l} = 0,$$

$$\frac{\partial\varphi}{\partial B_j} = \sum_{i=1}^{m}\frac{2Z_{ij}^2B_jx_{ij}}{T_i} - \sum_{i=1}^{m} Z_{ij}^4x_{ij};$$

$$\frac{\partial^2\varphi}{\partial^2 B_j} = \sum_{i=1}^{m}\frac{2Z_{ij}^2x_{ij}}{T_i};$$

$$\frac{\partial^2\varphi}{\partial B_j\partial B_l} = 0; \qquad \frac{\partial^2\varphi}{\partial B_j\partial T_i} = -\frac{2B_jZ_{ij}^2x_{ij}}{(T_i)^2},$$

Hessian matrix is defined in Boxes A.I and A.II.

Since x_{ij} and A_{ij} are greater than or equal to zero; X^TAX is greater than zero. As a result, the Hessian matrix of objective function Eq. (37) is greater than or equal to zero and is a convex function.

Appendix B

Finding the optimal value of the decision variables

The derivative of objective function in Eq. (37) should be calculated with respect to T_i, as follows:

$$\text{Hessian} = \begin{bmatrix}
\frac{\partial^2\varphi}{\partial^2T_1} & \frac{\partial^2\varphi}{\partial T_1\partial T_2} & \cdots & \frac{\partial^2\varphi}{\partial T_1\partial T_2} & \frac{\partial^2\varphi}{\partial T_1\partial B_1} & \frac{\partial^2\varphi}{\partial T_1\partial B_2} & \cdots & \frac{\partial^2\varphi}{\partial T_1\partial B_n}\\
\frac{\partial^2\varphi}{\partial T_2\partial T_1} & \frac{\partial^2\varphi}{\partial^2T_2} & \cdots & \frac{\partial^2\varphi}{\partial T_2\partial T_m} & \frac{\partial^2\varphi}{\partial T_2\partial B_1} & \frac{\partial^2\varphi}{\partial T_2\partial B_2} & \cdots & \frac{\partial^2\varphi}{\partial T_2\partial B_n}\\
 & & \vdots & & & & \vdots & \\
\frac{\partial^2\varphi}{\partial T_m\partial T_1} & \frac{\partial^2\varphi}{\partial T_m\partial T_2} & \cdots & \frac{\partial^2\varphi}{\partial^2T_m} & \frac{\partial^2\varphi}{\partial T_m\partial B_1} & \frac{\partial^2\varphi}{\partial T_m\partial B_2} & \cdots & \frac{\partial^2\varphi}{\partial T_m\partial B_n}\\
\frac{\partial^2\varphi}{\partial B_1\partial T_1} & \frac{\partial^2\varphi}{\partial B_1\partial T_2} & \cdots & \frac{\partial^2\varphi}{\partial B_1\partial T_m} & \frac{\partial^2\varphi}{\partial^2B_1} & \frac{\partial^2\varphi}{\partial B_1\partial B_2} & \cdots & \frac{\partial^2\varphi}{\partial B_1\partial B_n}\\
\frac{\partial^2\varphi}{\partial B_2\partial T_1} & \frac{\partial^2\varphi}{\partial B_2\partial T_2} & \cdots & \frac{\partial^2\varphi}{\partial B_2\partial T_m} & \frac{\partial^2\varphi}{\partial B_2\partial B_1} & \frac{\partial^2\varphi}{\partial^2B_2} & \cdots & \frac{\partial^2\varphi}{\partial B_2\partial B_n}\\
 & & \vdots & & & & \vdots & \\
\frac{\partial^2\varphi}{\partial B_n\partial T_1} & \frac{\partial^2\varphi}{\partial B_n\partial T_2} & \cdots & \frac{\partial^2\varphi}{\partial B_n\partial T_m} & \frac{\partial^2\varphi}{\partial B_n\partial B_1} & \frac{\partial^2\varphi}{\partial B_n\partial B_2} & \cdots & \frac{\partial^2\varphi}{\partial^2B_n}
\end{bmatrix}.$$

Box A.I

$$
\text{Hessian} = \left[\begin{matrix}
\sum_{j=1}^{n} \frac{2x_{1j}A_{1j}+Z_{1j}^2 B_j}{(T_1)^3} & 0 & \cdots & 0 \\
0 & \sum_{j=1}^{n} \frac{2x_{2j}(A_{2j}+Z_{2j}^2 B_j)}{(T_2)^3} & \cdots & 0 \\
 & & \vdots & \\
0 & 0 & \cdots & \sum_{j=1}^{n} \frac{2x_{mj}(A_{mj}+Z_{mj}^2 B_j)}{(T_m)^3} \\
\frac{-2x_{11}Z_{11}^2 B_1}{(T_1)^2} & \frac{-2x_{21}Z_{21}^2 B_1}{(T_2)^2} & \cdots & \frac{-2x_{m1}Z_{m1}^2 B_1}{(T_m)^2} \\
\frac{-2x_{12}Z_{12}^2 B_2}{(T_1)^2} & \frac{-2x_{22}Z_{22}^2 B_2}{(T_2)^2} & \cdots & \frac{-2x_{m2}Z_{m2}^2 B_2}{(T_m)^2} \\
 & & \vdots & \\
\frac{-2x_{1n}Z_{1n}^2 B_n}{(T_1)^2} & \frac{-2x_{2n}Z_{2n}^2 B_n}{(T_2)^2} & \cdots & \frac{-2x_{mn}Z_{mn}^2 B_n}{(T_m)^2}
\end{matrix}\right.
$$

$$
\left.\begin{matrix}
\frac{-2x_{11}Z_{11}^2 B_1}{(T_1)^2} & \frac{-2x_{12}Z_{12}^2 B_2}{(T_1)^2} & \cdots & \frac{-2x_{1n}Z_{1n}^2 B_n}{(T_1)^2} \\
\frac{-2x_{21}Z_{21}^2 B_1}{(T_2)^2} & \frac{-2x_{22}Z_{22}^2 B_2}{(T_2)^2} & \cdots & \frac{-2x_{2n}Z_{2n}^2 B_n}{(T_2)^2} \\
 & & \vdots & \\
\frac{-2x_{m1}Z_{m1}^2 B_1}{(T_m)^2} & \frac{-2x_{m2}Z_{m2}^2 B_2}{(T_m)^2} & \cdots & \frac{-2x_{mn}Z_{mn}^2 B_n}{(T_m)^2} \\
\sum_{i=1}^{m} \frac{-2x_{i1}Z_{i1}^2}{T_i} & 0 & \cdots & 0 \\
0 & \sum_{i=1}^{m} \frac{-2x_{i2}Z_{i2}^2}{T_i} & \cdots & 0 \\
 & & \vdots & \\
0 & 0 & \cdots & \sum_{i=1}^{m} \frac{-2x_{in}Z_{in}^2}{T_i}
\end{matrix}\right],
$$

$$
X^T A X = \begin{bmatrix} T_1 & T_2 & \cdots & T_m & B_1 & \cdots & B_n \end{bmatrix} (\text{Hessian}) \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_m \\ B_1 \\ \vdots \\ B_n \end{bmatrix} = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\frac{2A_{ij}}{T_i} \right) x_{ij}.
$$

Box A.II

$$\frac{\partial \varphi}{\partial T_i} = \sum_{j=1}^{n} Z^1_{ij} x_{ij} - \sum_{j=1}^{n} \frac{A_{ij} + Z^2_{ij} B^2_j}{(T_i)^2} x_{ij} = 0 \rightarrow T_i = \sqrt{\sum_{j=1}^{n} \left(\frac{A_{ij} + Z^2_{ij} B^2_j}{Z^1_{ij}} \right) x_{ij}}. \quad \text{(B.1)}$$

Moreover, by calculating derivative with respect to B_j, we have:

$$\frac{\partial \varphi}{\partial B_j} = \sum_{i=1}^{m} \frac{2 Z^2_{ij} B_j x_{ij}}{T_i} - \sum_{i=1}^{m} Z^4_{ij} x_{ij} = 0 \rightarrow B_j = \sum_{i=1}^{m} \left(\frac{Z^4_{ij}}{2 Z^2_{ij}} T_i \right) x_{ij}. \quad \text{(B.2)}$$

Substituting Eq. (B.1) into Eq. (B.2) leads to:

$$T_i = \sqrt{\frac{\sum_{j=1}^{n} A_{ij} x_{ij}}{\left(\sum_{j=1}^{n} Z^1_{ij} x_{ij} - \sum_{j=1}^{n} \frac{(Z^4_{ij})^2}{4 Z^2_{ij}} x_{ij} \right)}}; \quad i = 1, 2, \cdots, m, \quad \text{(B.3)}$$

and:

$$B_j = \sum_{i=1}^{m} \left(\frac{Z^4_{ij} x_{ij}}{2 Z^2_{ij}} \right) T_i; \quad j = 1, 2, \cdots, n. \quad \text{(B.4)}$$

Biographies

Amir Hossein Nobil is currently a PhD Candidate in Industrial Engineering at Qazvin Islamic Azad University, Qazvin, Iran. He received his BSc degree in Industrial Engineering and, then, his MSc degree in Industrial Engineering both from Qazvin Islamic Azad University, Qazvin, Iran. His research interests include supply chain management, inventory control, and production planning.

Amir Hosein Afshar Sedigh is currently a PhD Student in Information Science at University of Otago, Dunedin, New Zealand. He received a BSc degree from Qazvin Islamic Azad University, Qazvin, Iran in Industrial Engineering. Thereafter, he received his MSc degree in Industrial Engineering Sharif University of Technology, Tehran, Iran. His research interests include supply chain management, inventory control, and queuing theory.

Leopoldo Eduardo Cárdenas-Barrón is currently a Professor at School of Engineering and Sciences at Tecnológico de Monterrey, Campus Monterrey, México. He is also a faculty member at the Department of Industrial and Systems Engineering at Tecnológico de Monterrey. He was the Associate Director of the Industrial and Systems Engineering programme from 1999 to 2005. Moreover, he was also the Associate Director of the Department of Industrial and Systems Engineering from 2005 to 2009. His research areas include primarily related to inventory planning and control, logistics, and supply chain. He has published papers and technical notes in national and international Journal. He has co-authored one book in the field of simulation in Spanish. He is also an editorial board member in several international journals.