Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Application of nature inspired algorithms for multi-objective inventory control scenarios

Ferdous Sarwar[a], Mushaer Ahmed[a*] and Mahjabin Rahman[a]

[a]*Department of Industrial and Production Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh*

CHRONICLE

Article history:
Received February 16 2020
Received in Revised Format
April 11 2020
Accepted September 4 2020
Available online
September, 4 2020

Keywords:
Multi Objective Optimization
Inventory Control
Metaheuristic Algorithm
Multi Objective Particle Swarm Optimization
Multi Objective Bat Algorithm
Taguchi Method

ABSTRACT

An inventory control system having multiple items in stock is developed in this paper to optimize total cost of inventory and space requirement. Inventory modeling for both the raw material storage and work in process (WIP) is designed considering independent demand rate of items and no volume discount. To make the model environmentally aware, the equivalent carbon emission cost is also incorporated as a cost function in the formulation. The purpose of this study is to minimize the cost of inventories and minimize the storage space needed. The inventory models are shown here as a multi-objective programming problem with a few nonlinear constraints which has been solved by proposing a meta-heuristic algorithm called multi-objective particle swarm optimization (MOPSO). A further meta-heuristic algorithm called multi-objective bat algorithm (MOBA) is used to determine the efficacy of the result obtained from MOPSO. Taguchi method is followed to tune necessary response variables and compare both algorithm's output. At the end, several test problems are generated to evaluate the performances of both algorithms in terms of six performance metrics and analyze them statistically and graphically.

1. Introduction

Advanced supply chain of any manufacturing organization has to deal with many conflicting objectives to earn profit and customer satisfaction at the same time. It is an ongoing process of integrating all the processes involved from manufacturing a product towards final consumption and after sales services. At the present age of mass commercialization, more emphasis has been given on effective supply chain management as a cost minimizing factor. In order to ensure that, strategic decisions throughout different stages of production like inventory, production, warehouse, distribution etc. is a must.

Inventory management is proved to be a major driving factor to establish a cost effective and responsive supply chain (Chopra & Meindl, 2001). A level of inventory that is more than necessary or less than required can affect a business enterprise equally. Raw materials are stocked in order to ensure maximum responsiveness towards the work in process. If a supplier encounters a vital inventory item stock-out, it could result in production stoppages. Again, items that are in the middle of production have equal responsibility towards the finished goods inventory. In a word, this classic scenario of managing stock holding is appropriately termed as inventory management. Traditionally inventory control system is structured on the basis of a single buyer- single supplier relationship with fixed demand at each period. That doesn't hold any more as demand is uncertain and only can be explained by a probability function (Kodama, 1995). Nowadays organizations tend to purchase different items from multiple suppliers. In that sense lead time, reliability, quality, transportation, shortage, ordering cost etc. have to be considered carefully. In an actual manufacturing plant, inventory control systems have many limitations within warehouse space, budget, supply, high demand, inflation etc. besides managing multiple goods (Pasandideh et al., 2013). Finding the

* Corresponding author
E-mail: ferdoussarwar@ipe.buet.ac.bd (F. Sarwar)

2021 Growing Science Ltd.
doi: 10.5267/j.ijiec.2020.9.001

optimum level of inventory considering some of these issues in any practical supply chain scenario has now gotten the attention of researchers.

From the operations research point of view, multi-item inventory control problems are now depicted as multi-objective optimization problem. The inventory level at any stages of production and its holding cost creates a dynamic situation which can be modeled as mathematical equations (Sutrisno & Wicaksono, 2015). To formulate an inventory system model, many different parameters like reordering point, economic ordering quantity (EOQ), economic production quantity (EPQ), backorder, shortage level etc. have to be considered simultaneously. These mathematical models are solved using various predictive and metaheuristic algorithms so that an optimum level of inventory can be achieved. Nature based optimization algorithms are applied in such cases and their effectiveness for various supply chain networks are measured. To take decision like the amount of material purchasing at a single period cause a difficult situation for management of any organizations. To predict these beforehand machine learning algorithms based on company's historical data can bring some interesting results.

With increasing awareness about environmental sustainability and governmental rules and regulations regarding environmental pollution, many companies are now interested in making the purchasing and production process green (Mirzapour Al-e-Hashem & Rekik, 2014). If suppliers are located far from manufacturing facility, fuel consumption and greenhouse gas (GHG) emission are two main factors to be considered. In recent literatures, incorporating environmental performance parameters in inventory modeling has been suggested strongly.

Many previous works have addressed the problem of multi-item inventory control modeling. The next section provides a brief summary of these researches. In section 3 and 4 the problem for raw material and WIP inventory are formulated. Proposed solution methodology is presented in section 5. Section 6 demonstrates the application of algorithms which is followed by result discussion in section 7 and a brief conclusion in section 8.

2. Literature Review

The simplest idea of inventory can be extended to goods of different forms present at different stages of operation which can be transformed to finished goods at a cost when demand arises (Arrow et al., 1951). The greatest challenge for any firm in supply chain is to maintain the optimal level of inventory. Classical inventory control theory assumes that most of the organization purchase or produce a single product. But this assumption is not suitable for real life problem solving as most of the producers today try to sell more than one product to capture more profit and customer attraction (Mousavi et al., 2014). Determining the ordering times and order amounts of goods in inventory are the two strategic decisions to either mitigate total costs or increase total profits (Tavana, 2016). In this work control model for raw material inventory and work in process inventory is developed.

The availability of raw material in right quality and amount to some extent determines the availability, quality and quantity of desirable output. For any manufacturing outfit determination of economic order quantity (EOQ) and optimum stock levels is important in raw material management (S Akindipe, 2014). Hayek and Salameh (2001) calculated the optimum quantity of output that minimizes the overall inventory expense per unit time despite imperfect quality, where shortages are allowed and reordered. Chiu & Chiu (2006) analyzed optimal restocking model for defective quality using EOQ calculated in a traditional, differential method approach. Before that Grubbström & Erdem (1999) developed a method to solve EOQ without using derivatives. Many researchers adopted this algebraic method to solve real life inventory model problem. Taleizadeh et al. (2008) expanded the EOQ model to include retention costs, fixed order costs, insurance costs, transport costs, and capital costs in a mutual replenishment strategy. Huang & Wu (2016) developed some cost functions based on some structural properties which can be solved using efficient algorithm for minimizing average inventory cost in order to solve a periodic inventory model problem considering backordering in B2B environment. Rossi et al.(2017) developed an EOQ model for a multi-item framework, using mixed integer linear programming to conduct a case study.

According to Conway et al. (1988), WIP has been defined as a material between manufacturing processes excluding raw material and finished goods inventory. Blinder & Maccini (1991) said that investment in work in process inventory is more than finished goods inventory. Taft (1918) first implemented the economic production quantity (EPQ) model to precisely determine the level of inventory to minimize the total cost of inventory output and meet demand. In this regard Goyal (1976) proposed an integrated model for a single supplier- single customer issue in this regard. His research is also applicable where a single supplier provides a single customer with a number of products. Most of the early work on WIP based on predetermined production rate. But the real-world production system always goes through changing rate of production. Darwish, (2008) presented the EPQ model with the relation between setup cost and run length of production. In this work two models are developed, the first of which is for the case when shortages are not permitted and the second one allows shortages. Chiu et al. (2015) used a differential approach to calculate the typical production cycle time for a multi-item Economic Production Quantity (EPQ) model which minimizes the production rate, storage, and delivery charges. Kim & Park (2016) developed theories and heuristics in order to conduct numerical studies to find the optimal solution for a make-to-order production system and also developed an optimal policy for inventory control management which is a continuous review system with lost sale cost.

To be effective in inventory controlling various inventory related criteria and their interrelation should be considered. Multi-item multi-objective inventory control problem have been investigated by different researchers over the years. According to Lenard & Roy (1995) multi-item model is often needed, particularly when there is a very large number of items. Rather than accomplishing a solitary objective, a few analysts endeavored to discover Pareto results for their multi-item inventory planning issues which as a rule comprise a number of clashing purposes. Roy & Maiti (1998) reformulated the issue of deteriorating items with storage-dependent demand in the form of a multi-objective inventory model within tight, inexact storage area and budget allocation. Taleizadeh (2009) developed a combination of Pareto, TOPSIS and genetic algorithm to optimize inventory problem with multiple constraints considering both continuous review and periodic review with fuzzy replenishments and fuzzy demand. Hybridization of different algorithms can also be used in solving multi-criteria inventory problem with stochastic replenishment and fuzzy demand. Storage capacity and lot size were not often included in past researches of inventory problem, although they are important issues that should not be overlooked. Kanyalkar & Adil (2005) and Mandal et al. (2006) designed inventory models taking into account limited storage capacity. If the size of a piece of item is fairly large and the storage capacity in a plant is restricted, the plant may store a limited number of items. Pasandideh et al. (2013) implemented MOPSO and NSGA both to investigate a dual-objective economic production quantity problem for faulty items. It was detailed in a nonlinear modeling setting where the point was to discover the request amounts of the item which would limit both the general inventory expense and the necessary storage space. Mousavi et al. (2014) developed a multi-item multi-period plan for inventory with variable demands which were known a priori under the limited budget scenario. The main aim was to minimize the overall inventory cost and warehouse space and formulated in combination of different weights as objective function. With this same objective another inventory optimization model was proposed by Taleizadeh et al. (2009) under inflation and discount. It was seen that for different performance metrics each algorithm shows a statistically different result.

Over the years, many metaheuristic algorithms have been developed by researchers to solve various multi-objective optimization problems. Among them particle swarm optimization (PSO) is a nature inspired algorithm which is introduced by Kennedy & Eberhart (1995) analyzing social behavior of flock of birds. Taleizadeh et al. (2010) solved a single customer-single supplier problem using a PSO method where demand is probabilistic, and the lead time is relied upon to contrast straightly as for lot size. Park & Kyung (2014) utilized PSO to propose a strategy to optimize net inventory expense and order fill rate by changing the original inventory condition. Again, as the complexity of controlling inventory is increasing day by day, need for multi-objective optimization arises. A successful conversion of PSO to MOPSO was proposed by Coello Coello & Lechuga (2002). They extend the general PSO approach to work on constrained multi-objective optimization problems. Pasandideh et al. (2013) built an EPQ multi-product model with defective products that are reworkable in which the entire number of orders per year was constrained. Multi-objective nonlinear programming context was used to formulate the problem, where the intention was to quantify the order amount so as to minimize overall inventory cost and the necessary storage space. Mousavi et al. (2014) used MOPSO to address a multi-period multi-item inventory planning problem under two distinct discount schemes with multiple conflicting objectives. The result of this work showed that with respect to fitness function values, MOPSO performs better than MOGA. To perfectly model inventory related optimization inventory cost, space and shortage are also important considerations. Tavana (2016) evaluated a bi-objective inventory optimization problem under inflation and discount. The goal of this work is to find Pareto optimal solution in different periods, whilst minimizing total inventory cost and total storage space. Huseyinov & Bayrakdar (2019) compared the effectiveness of NSGA -III and strength pareto evolutionary algorithm (SPEA II) for solving multi-objective inventory problem where the goal was to maximize profit and warehouse space utilization. In the same way, Niu et al. (2019) did an extensive in-depth analysis of comparing recently developed multi-objective optimization algorithms like competitive multi-objective particle swarm optimization (CMOPSO), bi-goal evolution (BiGE), cooperative multi-objective bacterial forging optimization (CMBFO) etc. for multi-objective SC problems.

This research has been inspired by Mousavi et al. (2014) and an extension of the work done in Sarwar et al. (2019). In this work, a multi-item multi-constraints inventory control model is proposed considering shortage and constant demand for different items. The objectives of this model are to find out the optimum purchase and production quantity to minimize the total cost of inventory and space requirement which is done by utilizing MOPSO algorithm. In order to evaluate the efficiency of the result obtained from MOPSO, another metaheuristic algorithm named multi-objective bat algorithm (MOBA) is used. It was developed by Xin-She Yang in 2011 (Yang, 2011). BA has many advantages like quick convergence, parameter controlling but quick transformation from exploration to exploitation lead this algorithm to stick after certain iterations. Not many works had been done using BA to control inventory related optimization problem. For numerical optimization, Wang et al. (2012) combined bat algorithm and harmony search to create some kind of hybrid algorithm. Sadeghi et al. (2014) proposed a hybrid bat algorithm (HBA) to solve a single manufacturer-single vendor multi-retailer (SM-SV-MR) supply chain related vendor-managed inventory (VMI) problem.

Determining the optimum combination of affecting parameters on solution performance is the main obstacle in the scenario of optimization problem. Taguchi (1990) first developed a family of fractional factorial experimentation (FFE) matrices to use in various situation. Moosavi et al. (2014) used an outer orthogonal array and generated a repetition data transformation to another value as a signal-to-noise ratio (S/N) to produce robust optimum results. They plotted S/N proportion for each degree of components of MOPSO and MOGA algorithm to solve a multi-item multi periodic inventory control problem. In recent literatures incorporating environmental sustainability into inventory control has been recommended and several models have been suggested (Zanoni et al. , 2014; Tiwari et al., 2018; Rau et al., 2018). A likely consequence among these models is that the efficiency of an inventory management strategy is vulnerable when accounting for greenhouse gas (GHG) emissions (e.g. CO2). The present work prioritizes environmental pollution as an integral part of inventory management and carbon emission cost is considered equivalently in total cost function. With increasing importance of artificial intelligence (AI), machine learning algorithms are becoming popular day by day in supply chain related forecasting issues. In this paper, shortage forecasting and scrap production rates are quantified using a machine learning algorithm called support vector regression (SVR). Traditional method of forecasting based on regression has a limitation of large data set. But forecasting future demand is a complex problem of having nonlinear relationship between factors affecting the prediction (Wu, 2010). The SVR method based on support vector machine (SVM) theory can overcome this problem. It uses kernel functions and support vector to predict future outcome of a time series with limited observations. Researches on application of SVR in demand forecasting clearly represents that it is way more effective in complex supply chain problems. Guanghui Wang (2012) evaluated that SVR shows comparatively lower value of relative mean square error and accurate forecasting capability in time series demand prediction. The works of Guo et al. (2014) showed that inventory strategy based on SVR prediction has more probability to optimize the inventory management and minimize total cost.

Table 1
Literature summary of the related works

References	Multi item	Multi objective	Demand type	Determining EOQ & EPQ	Green Supply Chain	Forecasting method	Objectives	Solving Methodology	DOE
Pasandideh et al. (2013)	√	√	Deterministic	–	–	–	Minimizing total inventory cost and storage space	NSGA-II and MOPSO	–
Sutrisno & Wicaksono (2015)	√	–	Constant	–	–	–	Minimizing total cost	Model Predictive Control	–
Mirzapour Al-e-Hashem & Rekik (2014)	√	–	Deterministic & time varying	–	√	–	Minimizing total cost	CPLEX	–
Mousavi et al. (2014)	√	√	Deterministic & variable	–	–	–	Minimizing total inventory cost and storage space	PSO & GA	Taguchi L_9
Tavana (2016)	√	√	Constant	–	–	–	Minimizing total inventory cost and storage space	NSGA-II, NRGA and MOPSO	Taguchi
Hayek & Salameh (2001)	–	–	Constant	–	–	–	Total inventory cost minimization	–	–
Taleizadeh et al. (2008)	√	–	Constant	EOQ	–	–	Minimizing total cost of joint replenishment problem	Harmony Search	–
Huang & Wu (2016)	√	–	Stationary, deterministic, and periodic	EOQ	–	–	Minimizing operational cost	Numerical Method	–
Rossi et al. (2017)	√	–	Constant	EOQ	–	–	Minimizing inventory level	Discrete event simulation	—-
Darwish (2008)	–	–	Constant and Deterministic	EPQ	–	–	Total cost minimization	Numerical Example	–
Chiu et al. (2015)	√	–	Constant	EPQ	–	–	Total production-inventory-delivery cost	Algebraic Method	–
Roy & Maiti (1998)	√	√	Stock dependent & probabilistic	–	–	–	Profit maximization & wastage minimization	FNLP and FAGP	–
Taleizadeh et al. (2013)	√	√	Constant	–	–	–	Maximizing Profit	Pareto, TOPSIS and GA	–

Table 1
Literature summary of the related works (Continued)

References	Multi item	Multi objective	Demand type	Determining EOQ & EPQ	Green Supply Chain	Forecasting method	Objectives	Solving Methodology	DOE
Kanyalkar & Adil (2005)	√	–	Constant	–	–	–	Minimizing sum of transportation and inventory carrying costs	Gnu Linear Programming Kit (GNLP)	–
Mandal et al. (2006)	√	–	Constant	EOQ	–	–	Minimizing Total average cost	Modified Geometric Programming (MGP) & Non-linear Programming (NLP)	–
Ata Allah Taleizadeh et al. (2010)	√	–	Stochastic	–	–	–	Total inventory cost minimization	PSO	–
Guanghui Wang (2012)	–	–	Deterministic	–	√	–	Minimizing total SC cost	Numerical method	–
Park & Kyung (2014)	√	–	Stochastic	–	–	–	Minimizing the total inventory cost and the order fill rate	PSO	–
Sarwar et al. (2019)	√	√	Constant	EOQ	√	–	Total inventory cost minimization and layout management	MOPSO	Taguchi L_9
Sadeghi et al. (2014)	–	√	Deterministic	–	–	–	Minimize total VMI cost & maximize the system reliability	Hybrid bat algorithm (HBA)	Taguchi
Huseyinov & Bayrakdar (2019)	–	√	Stochastic	–	–	–	Maximizing the profit and the warehouse occupancy level	NSGA-II and Strength Pareto Evolutionary Algorithm (SPEA2)	–
Tiwari et al. (2018)	–	–	Constant	–	√	–	Minimizing total inventory and carbon emission costs	Partial derivative approach	–
Rau et al. (2018)	–	√	Constant	–	√	–	Total cost and emission	Discrete PSO	–
Niu et al. (2019)	√	√	Deterministic	–	–	–	Minimum total cost and maximum customer satisfactory level	Cooperative Multi-objective Bacterial Foraging Optimization	–
This Research	√	√	Constant and deterministic	√	√	√	Minimizing total inventory cost and maximum utilization of warehouse space	MOPSO and MOBA	Taguchi L_9

Table 1 shows a summarized version of some of the literature that are reviewed for preparing this paper. As far as the authors are concerned, only a few studies addressed the inventory modeling issue in a multi-objective scenario using MOPSO and MOBA both. The present study not only tries to solve the proposed model but also represent a comparative evaluation report to find out the best among the two solution algorithms.

In a nutshell, the highlights of this study can be stated as follows:
Modelling of a multi-item multi-objective inventory control model in order to minimize total inventory cost and space management. Considering raw material and work in process (WIP) inventory while designing the model. Considering equivalent carbon emission cost by giving priority on the environmental impact while managing inventory. Determining EOQ and EPQ by differentiating the total inventory cost function. Optimizing the objective function by using nature inspired metaheuristic algorithm named multi-objective particle swarm optimization (MOPSO). Forecasting shortage and scrap production percentage by using support vector regression (SVR) analysis.

3. Raw Material Inventory Problem

For a periodic inventory control model having constant and deterministic demand of items, raw materials are supplied from a single supplier using third party logistics (3PL) service to the buyer. Holding, obtaining, carbon emanation, and transportation costs are the expenses related with the stock control process. Multiple items are considered with both the warehouse space and capacity constraints. In addition, lead-time is assumed to be zero, and the variables for decision are integer digits. The assumptions of this study are inspired from previous researches by Mousavi et al. (2014) and Roozbeh Nia et al. (2015). The purpose is to determine the level of inventory items and the necessary warehouse space, thereby minimizing the total cost of inventory.

3.1 Assumptions

- Independent demand rate of items
- Demand rate is constant in each period.
- Same cartons/pallets are used for different items. Therefore, a multiple of a fixed-size batch has to be order quantities.
- All truck has same capacity.
- No volume discounts.
- Holding, Ordering and Shortage costs are considered.

3.2 Notation

The parameters below are the decision variables used for items *i = 1, 2, ……………, n.*

n: number of items to be purchased
Q_i: order quantity of the ith item (decision variable)
D_i: annual demand of the ith item
S_i: ordering cost per ordering an item
H_i: unit inventory holding cost for item *i*
I_i: shortage level of the ith item
E: Green House Gas (GHG) emission level
C_t: fixed emission tax cost
P_i : Number of pallets for an order of item *i*
C_p: truck capacity
A_i: required storage space per unit of the ith item
F: total available space in warehouse
T_i: shipping cost per unit of demand
U_e: upper bound on total GHG emission
L_i: annual per-unit cost of shortages of the ith item
B_i: purchasing cost per unit of item
M: total budget
W: weightage related to total cost

3.3 Model Formulation

In order to calculate the total inventory cost, different costs are calculated in following manner:

Ordering Cost
The ordering costs are essentially incurred every time when an *order is placed.*

$$\text{Total Ordering Cost, OC} = \sum_{i=1}^{n} \frac{D_i}{Q_i} S_i \tag{1}$$

Holding Cost

The holding costs are incurred for stored inventories as raw material.

$$\text{Total Holding Cost, HC} = \sum_{i=1}^{n} \frac{H_i}{2Q_i}(Q_i - I_i)^2 \tag{2}$$

Shortage Cost

The shortage costs are occurred when company is out of stock for raw materials or finished goods.

$$\text{Total Shoratge Cost, SC} = \sum_{i=1}^{n} \frac{L_i}{2Q_i} {I_i}^2 \tag{3}$$

Carbon Emission Cost

Equivalent carbon emission cost from the vehicle used to transport the raw material.

$$\text{Total Carbon Emission Cost, TE} = \sum_{i=1}^{n} ED_iC_t \tag{4}$$

Transportation Cost

The transportation cost is obtained from per unit shipping cost and total demand.

$$\text{Total Transportation Cost, TC} = \sum_{i=1}^{n} D_iT_i \tag{5}$$

Objective Function

The total inventory cost is summed up using above costs which is the 1st objective of this problem.

$$\boldsymbol{Z_1} = \sum_{i=1}^{n} \frac{D_i}{Q_i} S_i + \frac{H_i}{2Q_i}(Q_i - I_i)^2 + \frac{L_i}{2Q_i} {I_i}^2 + ED_iC_t + D_iT_i \tag{6}$$

2nd objective of this problem is to minimize the requirement of warehouse space. That is:

$$Z_2 = \sum_{i=1}^{n} (Q_i - I_i)\, A_i \tag{7}$$

The Constraints

There are four non-equality constraints and some non-negativity constraints.

Carbon emission has an upper limit above which tax cost becomes higher:

$$\sum_{i=1}^{n} E \leq U_e \tag{8}$$

Since the total available budget is M and purchasing cost per unit is C_b, budget constraint is given below:

$$\sum_{i=1}^{n} B_iQ_i \leq M \tag{9}$$

Order capacity have some limitations:

$$\frac{D_i}{Q_i} \leq C_p \tag{10}$$

Warehouse have some space constraints:

$$\sum_{i=1}^{n} A_iQ_i \leq F \tag{11}$$

Non-negativity constraints are:

$$Q_{i,} P_{i,} I_{i,} > 0 \tag{12}$$

Where, *1, 2, ……………, n;* where n is the number of items.

Final mathematical model of the total inventory control is to

$$\text{Minimize, } TOF = WZ_1 + (1 - W)Z_2 \tag{13}$$

where,

$$Z_1 = \sum_{i=1}^{n} \frac{D_i}{Q_i} S_i + \frac{H_i}{2Q_i}(Q_i - I_i)^2 + \frac{L_i}{2Q_i} {I_i}^2 + ED_iC_t + D_iT_i$$

$$Z_2 = \sum_{i=1}^{n} (Q_i - I_i)\, A_i$$

Subject to,

$$\sum_{i=1}^{n} E \leq U_e$$

$$\sum_{i=1}^{n} B_i Q_i \leq M$$

$$Q_{i,} P_{i,} I_{i,} > 0$$

where, *i = 1, 2, ……………, n;* where n is the number of items.

4. WIP Inventory Problem

One of the most important tasks in inventory control is to decide the optimal level of production quantity of several items. In this model the work in process (WIP) begins with raw materials which are converted into finished goods consisting imperfect, re-workable, and scrap items. Setup time and cost required for setup is considered along with machining time and inspection cost in order to develop the model. The assumptions and some models are inspired from previous research by Pasandideh et al. (2013). However, the goal is to identify the optimum level of produced items and required storage space, such that the total WIP inventory control system is minimized.

4.1 Assumptions

- Constant and deterministic demand
- Limited number of orders
- No shortage is allowed
- Each item has an inspection cost
- Production capacity is greater than demand
- Parameters are independent to production quantity
- Holding, setup, procurement and inspection costs are considered.

4.2 Notation

The parameters below are the decision variables used for items *i = 1, 2, ……………, n.*
n: number of items to be produced
q_i: ordering quantity of the *i*th item (decision variable)
d_i: annual demand of the *i*th item
M_i: raw material cost per unit of the *i*th item
h_i: unit inventory holding cost for produced item *i*
i_i: inspection cost of produced item per unit of the *i*th item
m_i: machining time per unit of the *i*th item
c_i: constant setup cost
p_i : scrap production percentage of item *i*
s_i: setup time per unit of item *i*
r_i: production rate per unit of the *i*th item
L: total number of orders
t_i: procurement cost per unit of the *i*th item
f_i: space occupied by each unit of items produced
a: total available space for keeping WIP inventory
b: total available budget
w: weight related to total cost ($0 \leq w \leq 1$)

4.3 Model Formulation

In order to calculate the total inventory cost, different costs are calculated in following manner:

Holding Cost
The holding costs are incurred for stored inventories which are still being processed.

$$\text{Total Holding Cost, HC} = \sum_{i=1}^{n} \frac{h_i d_i}{2(1-p_i)} [s_i + m_i q_i]\, [2M_i + \frac{r_i s_i}{q_i} + r_i M_i] \tag{13}$$

Setup Cost

Cost incurred for each setup of the produced item.

$$\text{Total Setup Cost, SC} = \sum_{i=1}^{n} \frac{c_i d_i}{q_i(1-p_i)} \tag{14}$$

Procurement Cost

Concerned with purchasing all of the goods and services that are important to the whole organization.

$$\text{Total Procurement Cost, PC} = \sum_{i=1}^{n} \frac{m_i d_i}{q_i(1-p_i)} \tag{15}$$

Inspection Cost

Cost incurred to inspect each item after being produced.

$$\text{Total Inspectation Cost, IC} = \sum_{i=1}^{n} \frac{i_i d_i}{q_i(1-p_i)} \tag{16}$$

Objective Function

The total inventory cost is summed up using above costs which is the 1st objective of this problem.

$$\boldsymbol{Z_1} = \sum_{i=1}^{n} \frac{h_i d_i}{2(1-p_i)} [s_i + m_i q_i]\, [2M_i + \frac{r_i s_i}{q_i} + r_i M_i] + \frac{c_i d_i}{q_i(1-p_i)} + \frac{m_i d_i}{q_i(1-p_i)} + \frac{i_i d_i}{q_i(1-p_i)} \tag{17}$$

2nd objective of this problem is to minimize the space needed to keep the WIP inventories. That is:

$$\boldsymbol{Z_2} = \sum_{i=1}^{n} q_i(1-p_i)\, f_i \tag{18}$$

The Constraints

There are three non-equality constraints and some non-negativity constraints.

Order capacity has some limitations:

$$\sum_{i=1}^{n} \frac{d_i}{q_i(1-p_i)} \leq L \tag{19}$$

Since the total available budget is b and procurement cost per unit is t_i, budget constraint is given below:

$$\sum_{i=1}^{n} t_i q_i \leq b \tag{20}$$

Space constraints for the WIP inventories:

$$\sum_{i=1}^{n} [1 - \frac{d_i}{(1-p_i)}]\, f_i q_i \leq a \tag{21}$$

Non-negativity constraints are:

$$q_{i,} p_{i,} i_{i,} c_{i,} > 0 \tag{22}$$

Where, *1, 2, ……………,n;* where n is the number of items.

Final mathematical model of the total inventory control is to

$$\text{Minimize, } TOF = wZ_1 + (1-w)Z_2 \tag{23}$$

where,

$$\boldsymbol{Z_1} = \sum_{i=1}^{n} \frac{h_i d_i}{2(1-p_i)} [s_i + m_i q_i]\, [2M_i + \frac{r_i s_i}{q_i} + r_i M_i] + \frac{c_i d_i}{q_i(1-p_i)} + \frac{m_i d_i}{q_i(1-p_i)} + \frac{i_i d_i}{q_i(1-p_i)}$$

$$\boldsymbol{Z_2} = \sum_{i=1}^{n} q_i(1-p_i)\, f_i$$

Subject to,

$$\sum_{i=1}^{n} E \leq U_e$$

$$\sum_{i=1}^{n} B_i Q_i \leq M$$

$$Q_{i,} P_{i,} I_{i,} > 0$$

where, *i = 1, 2, ……………, n;* where n is the number of items.

5. Solution Methodology

The proposed mathematical models for both inventory control scenarios are non-linear and multi-objective in nature. In order to reach in an analytical solution for this type of problem is highly unlikely (Gen & Cheng, 1997). Exact mathematical solution strategy like Lagrangian relaxation (Mirzazadeh et al., 2011), branch and bound (As'Ad & Demirli, 2011) and optimization solvers like CIPLEX and LINGO developed in various literature can only deal with less complicated and linear inventory optimization problems (Diabat, 2014). Real-life inventory optimization problem incorporates various conflicting objectives and even nonlinear constraints and that's why meta-heuristic algorithms are gaining popularity day by day because of their ability to deal with this kind of situations (Fattahi et al., 2015). In order to incorporate multiple constraints, continuous, integer and discrete variables and replicate real-life problem, metaheuristic methods are more efficient than conventional methods (Jones, Mirrazavi, & Tamiz, 2002). As a result, two nature inspired meta-heuristic algorithms; multi-objective particle swarm optimization (MOPSO) and multi-objective bat algorithm (MOBA) are employed in the following subsections to find out a Pareto optimal solution.

5.1 Multi-objective Particle Swarm Optimization (MOPSO)

Particle Swarm Optimization (PSO) technique is motivated by social conduct of creatures, similar to fish and bird. PSO is first introduced by James Eberhart and Russell C. Kennedy in the year of 1995 (Kennedy & Eberhart, 1995). While working on a model to explain birds and fish's social behavior, they found that their model is capable of solving problems with optimization. And that's how they came up with an algorithm based on their model called Particle Swarm Optimization. In this study a modified version of PSO algorithm name multi-objective particle swarm optimization (MOPSO) is used because of its diverse application field and the fast convergence rate and less computational effort.

Particle swarm optimization contains a population of candidates called "swarm". Swarm have a candidate of solution called "particle". Every particle in the swarm is a candidate solution for the optimization problem. PSO follows a search mechanism of these particles in the swarm. The particles try to find the best solution of the optimization problem by communicating with each other. They keep finding the best one till every particle in the swarm have found that best solution of the optimization problem. So, there are two principles of PSO algorithm for the cooperation of the particles. They are-

- *Communication:* informing the best solution of a particular particle to the other particles in the swarm.
- *Learning:* when particles move towards one another they actually learn about the location better solution. It is the concept of 'better' which is the main problem that has to be solved.

5.1.1 Swarm Intelligence

In swarm intelligence an intelligent behavior is created by means of some agents, like- birds, ants, and fish. The level of intelligence is unreachable for any member of the swarm. But with the cooperation among the members of the swarm, this level of intelligence can be reached. This is the concept of paradigm of swarm intelligence. Thus, PSO is a swarm intelligence method and it uses some unintelligent agents called particles to reach an upper level of intelligence which is completely unreachable for the particles alone.

5.1.2 Mathematical model of motion

The particles have a position in the search space of the optimization problem. For particle i, position vector is x_i, which is a member of search space X where, $x_i(t) \in X$. Here *t* is the time index to distinguish between discrete time steps and it shows the iteration number of the algorithm. Every particle in the swarm have velocity denoted by $v_i(t)$. It is also a vector and belongs to the same space.

In the space particles interact with and learn from each other by obeying a simple rule to find the best solution the optimization problem. Every particle finds their personal best denoted by $p_i(t)$ called the local best solution. There is a common best experience among the members of the swarm denoted by *g(t)* called the global best solution.

Initial position of the particle i is $x_i(t)$ and velocity is $v_i(t)$. Particles move toward the personal best and then to the global best and gain an updated position denoted by $x_i(t+1)$ and the addition of these beginning and end vector has a velocity of $v_i(t+1)$. So, the equation for updating the position is-

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{24}$$

where,

$$v_i(t+1) = wv_i(t) + C_1\big(p_i(t) - x_i(t)\big) + C_2\big(g(t) - x_i(t)\big) \tag{25}$$

A simplified approach is used to standardize the PSO equation and that is

$$v_i(t+1) = wv_i(t) + C_1 r_1 \left(x_{pbesti} - x_i(t)\right) + C_2 r_2 \left(x_{gbest} - x_i(t)\right) \tag{26}$$

where,

$w =$ inertia coefficient
$C_1, C_2 =$ accleration coefficients
$r_1, r_2 \in (0,1)$

In order to solve multi objective optimization problem, PSO needs some modifications:

1. Finding not a "universal best" solution, but a collection of Pareto Front solutions.
2. A non-dominated solution library is to be maintained, where all non-dominated solutions found at each iteration are deposited

In order to compare the efficiency of this algorithm another metaheuristic algorithm multi-objective bat algorithm (MOBA) is applied. Like MOPSO it is also a modified version of previously developed bat algorithm (BA) to deal with multiple conflicting objectives.

5.2 Multi Objective Bat Algorithm (MOBA)

Bat algorithm (BA) is a biologically oriented algorithm. Like MOPSO, it is also a nature-inspired typically based on swarm intelligence. Solving multi objective optimization problems using BA are more entangled than optimizing single objective problem. BA needs some modification to accommodate multiple objectives properly for that reason optimality fronts are approximated.

5.2.1 Bat motion

A speed v_i^t and a position x_i^t are two variables related with each bat at iteration *t* in a d-dimensional inquiry area (Yang & He, 2013). In midst of all this bat, there exists a current best location x_*. So, the equation stands:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{27}$$
$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*)f_i \tag{28}$$
$$x_i^t = x_i^{t-1} + v_i^t \tag{29}$$

where, $\beta \in [0, 1]$ is an arbitrary vector.

5.2.2 Loudness and pulse emission rate

As the loudness normally diminishes once a bat has discovered its prey, while the pace of heartbeat emanation expands, the loudness can be picked as any estimation of convenience, between A_{min} & A_{max}, assuming $A_{min} = 0$ (Yang & He, 2013). With these assumptions,

$$A_i^{t+1} = \alpha A_i^t \tag{30}$$
$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{31}$$

where, α and γ are constants.

5.2.3 Pareto optimality

A solution vector $u = (u_1, \dots\dots, u_n)^T \in \mathcal{F}$, is said to dominate another vector $v = (v_1, \dots\dots, v_n)^T$ if and only if $u_1 \leq v_1$. No component of *u* is larger than the corresponding component of *v*, and at least one component is smaller.

$$u \preceq v \leftrightarrow u \prec v \lor u = v \tag{32}$$

A multi-objective Pareto front, PF can be defined as the set of non-dominated solutions so that

$$PF = \{s \in \mathcal{S} \mid \nexists\, s' \in \mathcal{S} : s' \prec s\} \tag{33}$$

or in term of the Pareto optimal set in the search space,

$$PF^{*} = \{x \in \mathcal{F} \mid \nexists\, x' \in \mathcal{F} : f(x') \prec f(x)\} \tag{34}$$

Where, $f = (f_1, \dots \dots, f_K)^T$

5.2.4 MOBA

For simplicity a weighted sum has been used to combine all objectives f_k into a single objective,

$$f = \sum_{k=1}^{K} w_k f_k \tag{35}$$

$$\sum_{k=1}^{K} w_k = 1 \tag{36}$$

As the weights are created self-assertively from a uniform distribution, the loads can be changed with adequate variety so the Pareto front can be appropriately approximated (Yang, 2012).

5.3 PseudoCodes

Pseudocode algorithm of MOPSO and MOBA are given below:

Pseudocode 1: Pseudo Code of MOPSO Algorithm (Mousavi et al., 2014)

```
for i = 1 to Pop
    initialize position (i)
    initialize velocity (i)
    if position (i) and velocity (i) be a feasible candidate solution
        penalty = 0
    else penalty = a positive number
    end if
end for
w = [0.4, 0.9]
do while Iter <= Gen
    for j = 1 to Pop
        Calculate new velocity of the particle
        Calculate new position of the particle
        pbest (iter) = min (pbest(i))
    end for
    gbest (iter) = min (gbest)
    w = w_max – ((w_max - w_min)/iter_max) × iter
    modifying the velocity and position of the particle
end while
```

Pseudocode 2: Pseudo Code of MOBA Algorithm (Yang, 2012)

```
Objective functions $f_1(x), \dots \dots, f_K(x),\ x = (x_1, \dots \dots, x_d)^T$
Initialize the bat population $x_i\ (i = 1,2, \dots \dots, n)\ and\ v_i$
for j = 1 to N (points on Pareto fronts)
  Generate K weights $w_k \geq 0$ so that $\sum_{k=1}^{K} w_k = 1$
  Form a single objective $f = \sum_{k=1}^{K} w_k f_k$
  while ( $t <$ Max number of iterations)
      Generate new solutions and update by $f_i = f_{min} + (f_{max} - f_{min})\beta$ to $x_i^t = x_i^{t-1} + v_i^t$
      if ($rand > r_i$)
        Random walk around a selected best solution
      end if
      Generate a new solution by flying randomly
      if ($rand < A_i$ & $f(x_i) < f(x_*)$)
        Accept the new solutions,
        and increase $r_i$ & reduce $A_i$
      end if
      Rank the bats and find the current best $x_*$
  end while
  Record $x_*$ as a non-dominated solution
end
```

6. Application of MOPSO and MOBA

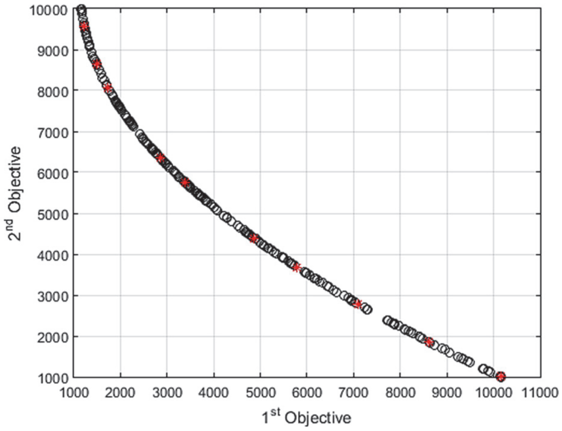
MATLAB platform is used to solve the non-liner programming model for both raw material inventory control and work-in-process (WIP) inventory control. The obtained Pareto front plotting developed by both MOPSO and MOBA algorithm are shown in Figs. 1-4 for two previously developed inventory models. The outcomes of this solution process are Pareto front of all local optimum solutions, optimum solution for both objectives and related parameters value, total elapsed time to reach solution.

Table 2
Parameter values of MOPSO Pareto front

Iteration No.	C_1	C_2	Pop	Rep	1st Objective	2nd Objective
200	2.5	1.5	300	10	10160	1000

Table 3
Parameter values of MOBA Pareto front

Iteration No.	n	A	r	1st Objective	2nd Objective
200	100	.25	.75	10829	115



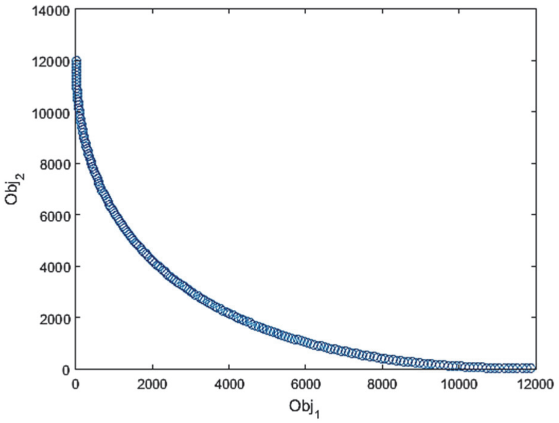


Fig. 1. MOPSO Pareto front plot

Fig. 2. MOBA Pareto front plot

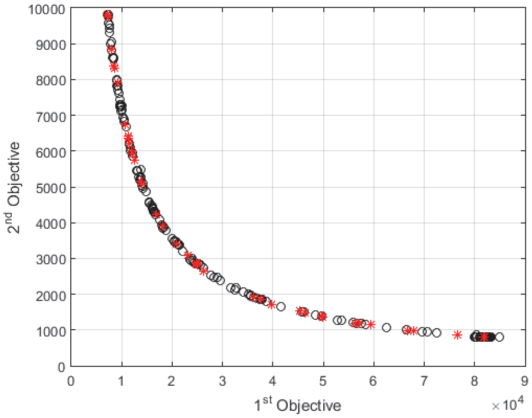
Table 4
Parameter values of MOPSO Pareto front

Iteration No.	C_1	C_2	Pop	Rep	1st Objective	2nd Objective
200	2	1.5	200	40	80193	800

Table 5
Parameter values of MOBA Pareto front

Iteration No.	n	A	r	1st Objective	2nd Objective
200	60	.5	.25	71598	310

Both the algorithms are implemented simultaneously for the two inventory models developed in section 3 and 4. The near optimal solutions for raw material inventory along with the parameters value after iterating 200 times are shown in Table 2 and 3. Similarly, obtained optimal value of two objectives and their parameters for WIP inventory are presented in Table 4 and 5. It is clear from the result generated by the proposed two algorithms are different from each other. In the next section a parameter tuning method Taguchi L_9 is implemented to find out the optimal level of the parameters.



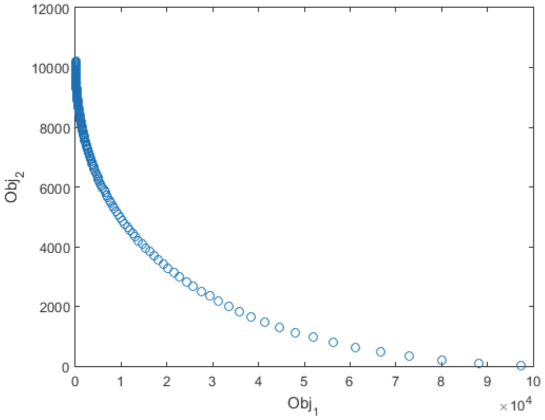


Fig. 3. MOPSO Pareto front

Fig. 4. MOBA Pareto front

Applying a differential approach on Eq.(6) and Eq. (17), equation of economic quantity to order and produce can be formulated as follows:

$$EOQ = \sqrt{\frac{2DS}{H} + \frac{LI^2}{H} + I^2} \tag{37}$$

$$EPQ = \sqrt{\frac{2(c + m + i) + hrs^2}{hMm(r + 2)}} \tag{38}$$

See Appendix A and B for detail formulation process.

7. Analysis and Discussion

7.1 Tuning parameters of algorithm using Taguchi Design

Minitab18 is used to apply Taguchi approach to finely tune the level of algorithms' parameters. Four factors are chosen for two of the algorithms and for each factor three level of value is selected to implement Taguchi L_9 design are shown in Table 6 and 7. As a result, nine different combinations of parameter value along with their objective values for both MOPSO and MOBA are obtained which are presented in Table 8 and 9. Fig. 5 and 6 demonstrates the mean S/N ration plot for each level of parameters of MOPSO and MOBA and finally from the plot the optimal level of parameters' value is chosen. Table 10 and 11 represents these findings and the optimal value for each algorithm using those optimal levels of parameters.

Table 6
Parameter values and levels of MOPSO and MOBA (raw material inventory model)

Algorithms	Factors	Levels [1 2 3]	Factors	Levels [1 2 3]
MOPSO	**C_1**	[1.5 2 2.5]	**Pop**	[100 200 300]
	C_2	[1.5 2 2.5]	**Rep**	[10 30 50]
MOBA	**n**	[10 50 100]	**r**	[.25 .5 .75]
	A	[.25 .5 .75]	**Pop**	[100 150 200]

Table 7
Parameter values and levels of MOPSO and MOBA (WIP inventory model)

Algorithms	Factors	Levels [1 2 3]	Factors	Levels [1 2 3]
MOPSO	**C_1**	[1.5 2 2.5]	**Pop**	[100 150 200]
	C_2	[1.5 2 2.5]	**Rep**	[40 50 60]
MOBA	**n**	[20 40 60]	**r**	[.25 .5 .75]
	A	[.25 .5 .75]	**Pop**	[80 140 200]

Table 8
Taguchi L_9 formulation with objective function value (raw material inventory model)

Run No.	A	B	C	D	MOPSO	MOBA
1	1	1	1	1	10273	9773
2	1	2	2	2	10155	9976
3	1	3	3	3	10253	10567
4	2	1	2	3	10156	11829
5	2	2	3	1	10225	10773
6	2	3	1	2	10163	11075
7	3	1	3	2	10356	11824
8	3	2	1	3	10250	12083
9	3	3	2	1	10192	10098

Table 9
Taguchi L_9 formulation with objective function value (WIP inventory model)

Run No.	A	B	C	D	MOPSO	MOBA
1	1	1	1	1	80221	79687
2	1	2	2	2	79358	80994
3	1	3	3	3	79599	83981
4	2	1	2	3	81515	82345
5	2	2	3	1	80766	76843
6	2	3	1	2	80606	78942
7	3	1	3	2	81239	81110
8	3	2	1	3	79351	85621
9	3	3	2	1	80652	79890

Table 10
Optimal level of different parameters (raw material inventory model)

Algorithms	Factors	Optimal Levels	Factors	Optimal Levels	Optimal Values
	C_1	2.5	**Pop**	300	
MOPSO	**C_2**	1.5	**Rep**	10	10160
	n	100	**r**	.75	
MOBA	**A**	.25	**Pop**	200	10829

Table 11
Optimal level of different parameters (WIP inventory model)

Algorithms	Factors	Optimal Levels	Factors	Optimal Levels	Optimal Values
	C_1	2	**Pop**	200	
MOPSO	**C_2**	1.5	**Rep**	40	80193
	n	60	**r**	.25	
MOBA	**A**	.5	**Pop**	200	71598

7.2 Comparison of MOPSO and MOBA

To compare the performance of MOPSO and MOBA, at first, a graphical approach is taken. Minitab18 is used to develop the scatterplot of MOPSO and MOBA algorithm shown in Fig. 7 and Fig. 8. Aim of this study is to minimize the total inventory cost and space. After comparing the two Pareto fronts it is found that MOBA gives more appropriate result to fulfill this objective. But as it is known that these nature inspired algorithms always give near optimum solutions. So only a Pareto front cannot draw a conclusive solution rather some other criteria should be considered.

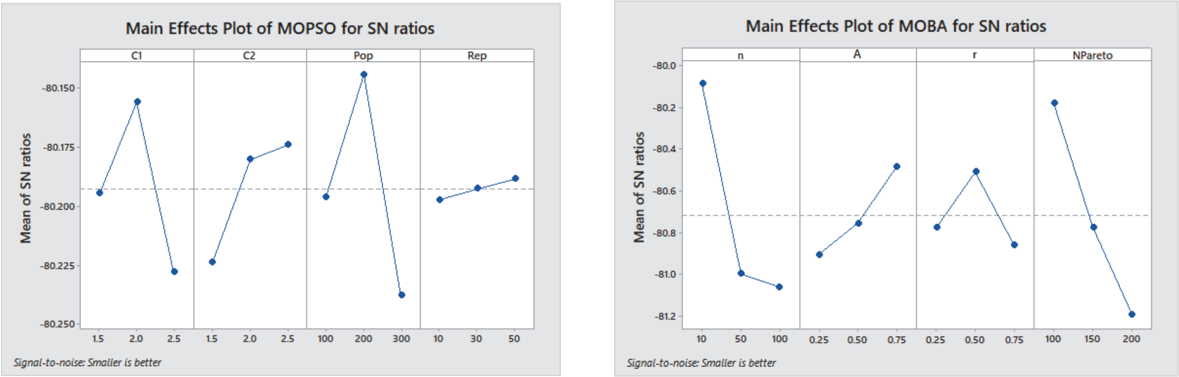


Fig. 5. Main effect plot of MOPSO and MOBA for S/N ratios (raw material inventory model)

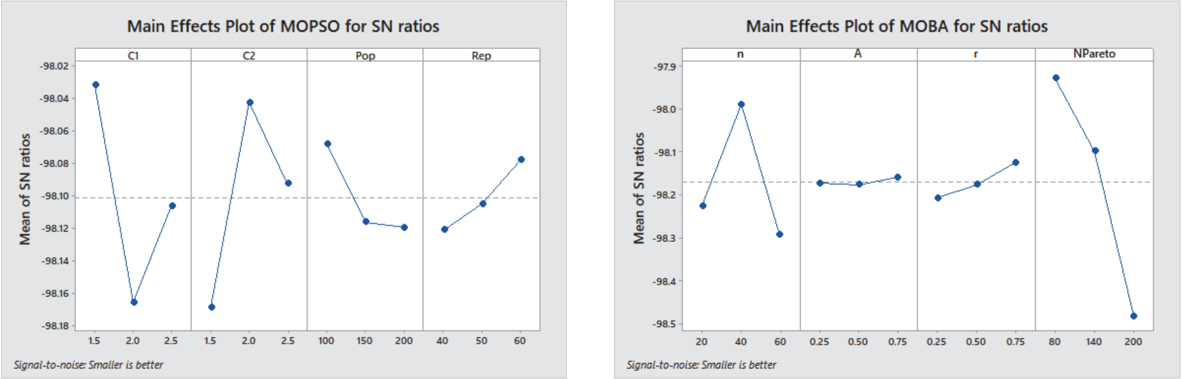


Fig. 6. Main effect plot of MOPSO and MOBA for S/N ratios (WIP inventory model)

The obtained Pareto front of both this algorithm shows significant difference in terms of sensitivity. For MOBA deviation in one objective function affects other in larger extent than MOPSO. Also, at the peak value of one objective than the other becomes almost zero. In this regard, MOPSO gives better solution as it never shows zero value for any of the objectives. The convergence rate of MOBA is quick at an early stage, but gradually slow down. On the other hand, MOPSO converges faster than any other metaheuristics algorithm. It requires less computational effort (Branke & Mostaghim, 2016). Again, mathematical analysis for linking parameters to convergence rates is constrained, as it is difficult to understand the meaning for which it converges. Besides the graphical comparison, the followings six standard metrics of the multi - objective algorithms is applied to come to a concluding decision:

- Spacing metric (SM): measures the expanse of spread among the solutions for which smaller value shows better results (Schott, 1995).
- Diversity metric (DM): measures the extent of the obtained Pareto fronts for which larger value means better results (Zitzler & Thiele, 1998).
- Mean ideal distance (MID): measures the convergence rate of Pareto fronts to a certain point (0, 0) (Zitzler & Thiele, 1998).
- Error ratio (ER): measures the percentage of solutions that are not included in Pareto optimal set (Van Veldhuizen, 1999).
- Number of solutions (NOS): measures the number of solutions in the Pareto optimal set for which the bigger value is the better.
- Time: measures the computational (CPU) time for the algorithm to reach optimum solutions.

An experiment of 10 test problems based on two proposed model with various number of items is implemented. Each problem goes through at least thirty iterations under different random scenarios and their average is taken as final responses. Table 12 and 13 summarize these responses of different metrics on 10 test problems. In order to statistically compare the performance of proposed algorithms, an analysis of variance (ANOVA) test is applied. Table 14 shows these test results along with *P*-values in terms of defined metrics. To validate the statistical comparison, the boxplots of six metrics on MOPSO and MOBA are represented in Fig. 9 and Fig. 10. In addition, graphical comparisons of these matrices are shown in Fig. 11 and Fig. 12, respectively. From the ANOVA test results in Table 14 along with the boxplots in Fig. 9 it is clear that in terms of MID, SM and ER MOBA shows better performance than MOPSO with statistically significant test results. But on the other hand MOPSO dominates MOBA in terms of DM. For the rest of two metrics, NOS and time both the algorithms work statistically same and show no mentionable difference. However, statistical results for WIP inventory model differs from the former one. Though MOPSO clearly shows a far better performance than MOBA in terms of CPU time, it is dominated by MOBA in terms of MID and ER. For SM, DM and NOS, no concluding decision can be made as their test result is not significantly different. All these conclusions are made at the 95% confidence level and graphical comparisons in Fig. 11 and Fig. 12 also confirms that. It should be mentioned that decisions regarding *P*-values are not always convincing because of its extreme assumption of the null hypothesis. Complexity of the proposed design of inventory model and assumptions in optimization problem can also play an important part in obtaining such test results.

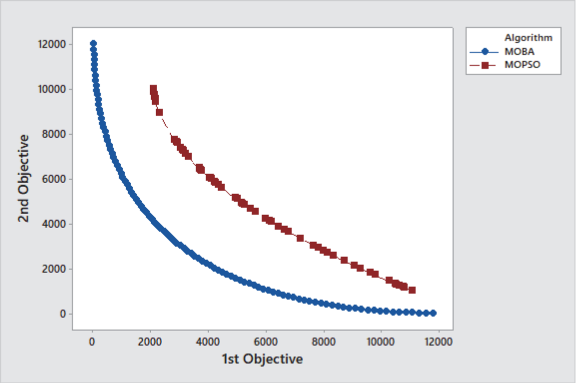


Fig. 7. Scatterplot of the performance of MOPSO & MOBA (raw material inventory model)

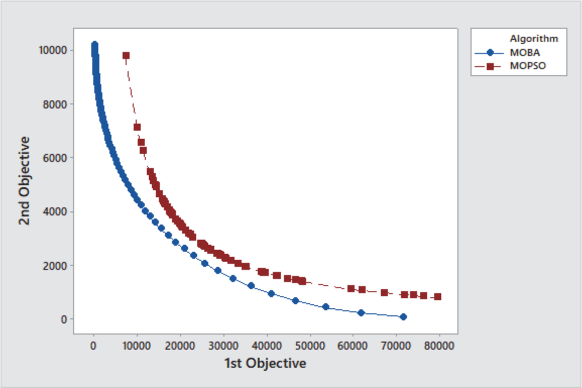


Fig. 8. Scatterplot of the performance of MOPSO & MOBA (WIP inventory model)

Table 12
Experimental results of six metrics in raw material inventory model

Test			MOPSO						MOBA			
	SM	DM	MID	ER	NOS	Time	SM	DM	MID	ER	NOS	Time
1	9097.17	32178.46	184.03	0.97	8.00	2.08	983.3	12002.6	74.21	0.7	9	5.67
2	6320.58	37394.74	168.59	0.97	10.00	2.29	1625.15	11999.22	74.65	0.63	6	4.7
3	3449.67	30651.89	165.53	0.73	6.00	3.38	1471.25	12422.19	74.29	0.8	7	1.88
4	3208.92	18416.96	191.47	0.77	8.00	4.60	810.73	11425	74.76	0.57	10	2.45
5	4111.71	33799.57	190.01	0.77	8.00	4.60	1617.19	13280.38	74.58	0.7	8	2.45
6	2904.03	17748.90	171.96	0.73	10.00	2.89	1338.59	12751.61	74.06	0.67	8	2.35
7	12244.74	68020.89	213.99	0.80	8.00	1.35	965.23	13560.53	73.62	0.53	12	2.86
8	4146.76	14915.85	172.09	0.83	7.00	3.33	1284.06	13165.04	73.94	0.6	9	1.91
9	3941.99	31196.05	179.74	0.67	9.00	3.08	1039.62	13409.18	73.69	0.6	11	2.32
10	3216.34	8466.32	171.55	0.87	5.00	3.88	1620.87	11995.95	74.63	0.67	6	1.4

Table 13
Experimental results of six metrics in WIP inventory model

Test Problem No.	MOPSO						MOBA					
	SM	DM	MID	ER	NOS	Time	SM	DM	MID	ER	NOS	Time
1	3329.48	76910.09	167.80	0.97	32.00	3.42	5602.74	62643.17	106.58	0.96	34	46.3
2	2728.99	53221.45	224.25	0.98	34.00	4.63	5742.29	60761.67	106.46	0.95	32	40.54
3	2867.062	90386.05	181.65	0.97	36	5.60	5463.135	64385.13	106.67	0.94	36	55.48
4	17178.08	86365.75	184.18	0.97	34	4.46	5532.62	63529.4	98.91	0.96	35	46.07
5	5082.65	73701.97	158.32	0.96	40	4.23	5879.74	58723.29	106.34	0.94	30	42.48
6	21589.2	19069.07	173.15	0.98	29	3.22	6076.4	55331.43	102.33	0.96	28	38.53
7	3382.12	72374.02	169.63	0.98	29	5.04	5190.42	67506.69	106.85	0.95	40	68.77
8	4310.18	73010.99	166.72	0.98	24	4.49	5947.26	57640.09	106.27	0.93	29	41.7
9	66245.65	591500.74	246.38	0.98	28	5.36	5811.04	59762.72	106.4	0.93	31	45.31
10	3196.68	73683.99	167.09	0.97	28	3.94	5059.48	68914.49	106.93	0.94	42	52.45

Table 14
ANOVA comparison results with *P*-values

Model	Metric's name	*P*-value	Test Results
Raw material inventory model	SM	0.001	Null hypothesis is rejected
	DM	0.005	Null hypothesis is rejected
	MID	0	Null hypothesis is rejected
	ER	0.001	Null hypothesis is rejected
	NOS	0.400	Fail to reject null hypothesis
	Time	0.525	Fail to reject null hypothesis
WIP inventory model	SM	0.257	Fail to reject null hypothesis
	DM	0.277	Fail to reject null hypothesis
	MID	0	Null hypothesis is rejected
	ER	0	Null hypothesis is rejected
	NOS	0.285	Fail to reject null hypothesis
	Time	0	Null hypothesis is rejected

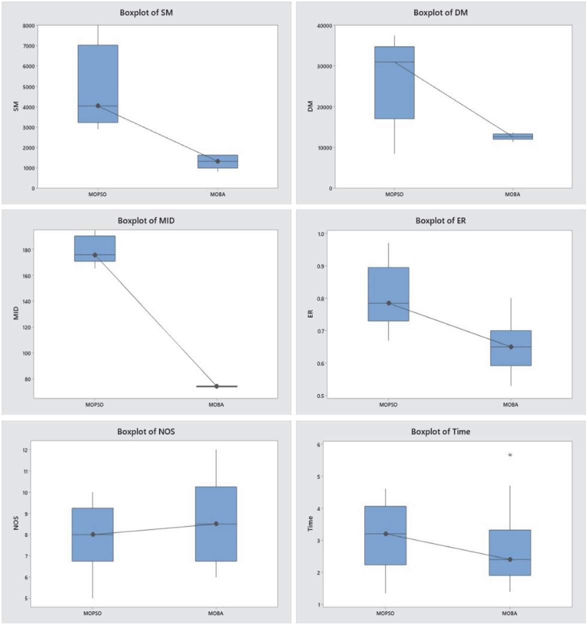


Fig. 9. Boxplots of metrics vs. algorithms (raw material inventory model)

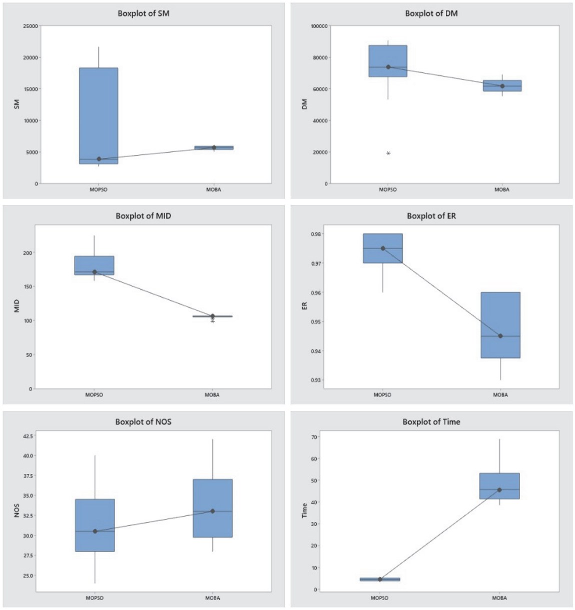
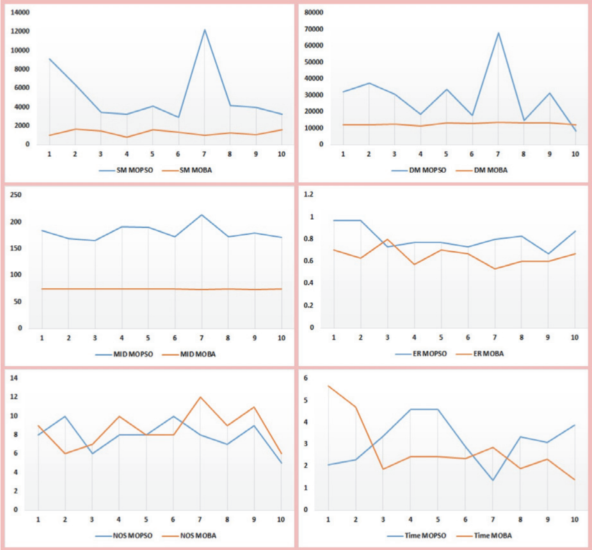


Fig. 10. Boxplots of metrics vs. algorithms (WIP inventory model)



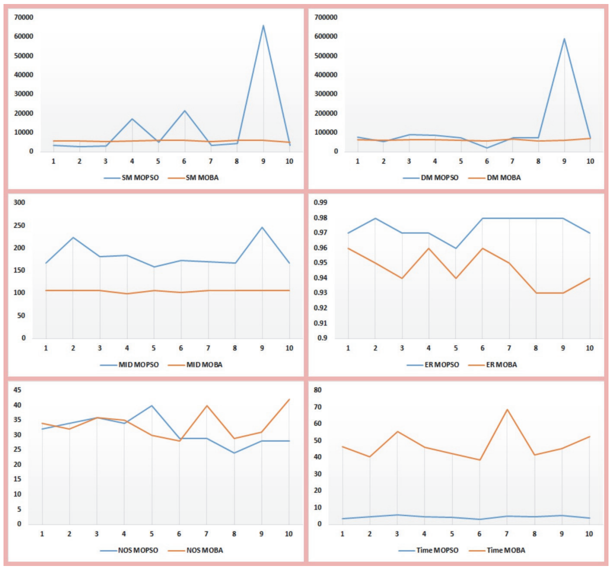
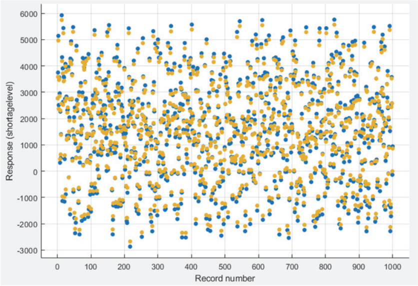


Fig. 11. Graphical comparisons of algorithms (raw material inventory model)

Fig. 12. Graphical comparisons of algorithms (WIP inventory model)



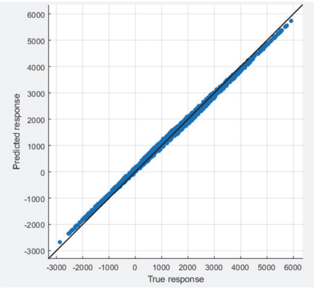
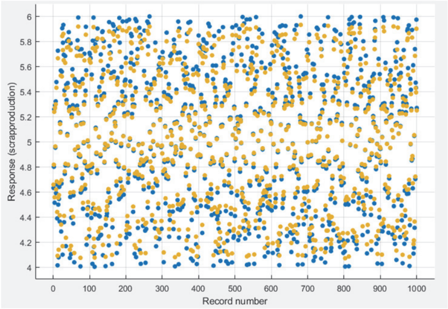


Fig. 13. Shortage prediction plot (yellow dot- record number and blue dot – response)

Fig. 14. Predicted response vs true response for shortage prediction



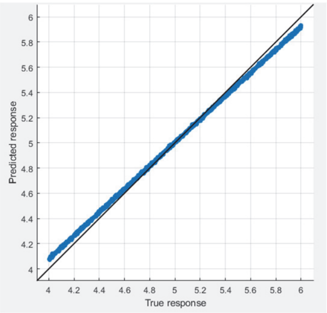


Fig. 15. Scrap prediction plot (yellow dot- record number and blue dot – response)

Fig. 16. Predicted response vs true response for scrap

8. Shortage and scrap prediction using SVR

MATLAB Regression Learner App is used to import a large number of close to real data (Appendix C) to predict shortage quantity for raw material inventory and scrap production rate for WIP inventory. The minimization problem is shaped in a quadratic programming formation and resolved by quadratic SVM technique (Scholkopf et al., 1997). In this study, response variables are shortage level and scrap production rate. 5-fold cross-validation is applied to increase model performance on new data and choose best model. Obtained response result and plotting of actives prediction are presented in Fig. 13 for shortage prediction and Fig. 15 for predicting scrap production. The model is trained into all six SVM regression model type and among them quadratic model gives the lowest root mean square error (RMSE) value. All these forecasting parameter values to evaluate the efficiency of prediction are summarized in Table 15. Again, efficiency of prediction is deduced from graphical representation and it is quite good for both model as it is seen from the predicted response vs true response plotting depicted by Fig. 14 and Fig. 16. Interpretability is hard for this type of model but flexibility is comparatively better than others. Run time for training is quite long as it requires huge memory to store.

Table 15
Values of forecasting parameters

Prediction	Parameters	Values
Shortage	RMSE	0.03704
	R-squared	1
	MSE	0.001372
	MAD	0.0282
Scrap	RMSE	0.044192
	R-squared	0.99
	MSE	0.0019529
	MAD	0.038063

9. Conclusion and Recommendations for Future Research

A multi-item inventory control problem was investigated in this paper with the aim of minimizing both the total cost of inventory and the total storage space required. Considering independent demand rates of items, no volume discount and shortages were allowed. The problem was first formulated as a raw material inventory control model and after that as a WIP inventory control model. The aim was to identify the optimal level of order and the optimal quantity of production, so that objective function is minimized and constraints retain. Both these models were implemented into multi-objective particle swarm optimization and multi-objective bat algorithm to find a Pareto front solution. Taguchi L_9 design was applied to calibrate the parameters of the algorithm and the combination that best suited to the objective was chosen. An experiment, including several test problems was conducted to compare the performance of two algorithms in terms of six standard metrics. Two outcomes that worth mentioning are:

- MOBA works better than MOPSO in terms of SM, MID and ER in the proposed raw material inventory control scenario;
- MOPSO is the best algorithm in terms of time in the proposed WIP inventory control scenario;

A number of closer to reality data were generated at the end for both control models to predict shortage quantity and scrap production rate using support vector regression method.

Some suggestions for further research in future are to develop model to cover finished goods inventory can be developed, consider volume discount, lead time uncertainty, defective items, inflation and time value of money and employ recently developed meta-heuristic nature inspired algorithms to solve the problem.

References

Arrow, K. J., Harris, T., & Marschak, J. (1951). Optimal inventory policy. *Econometrica: Journal of the Econometric Society*,*19*(3), 250–272.

As'Ad, R., & Demirli, K. (2011). A bilinear programming model and a modified branch-and-bound algorithm for production planning in steel rolling mills with substitutable demand. *International Journal of Production Research*, *49*(12), 3731–3749. https://doi.org/10.1080/00207541003690116

Blinder, A. S., & Maccini, L. J. (1991). Taking stock: a critical assessment of recent research on inventories. *The Journal of Economic Perspectives*, *5*(1), 73–96. https://doi.org/10.1016/j.jmoneco.2004.08.005

Bonney, M., & Jaber, M. Y. (2011). Environmentally responsible inventory models: Non-classical models for a non-classical era. *International Journal of Production Economics*, *133*(1), 43–53.

Branke, J., & Mostaghim, S. (2016). Comprehensive Comparision of MOPSO methods: Study of Convergence and Diversity-Survey of State of The Art.

Chiu, S. W., & Chiu, Y. S. P. (2006). Mathematical modeling for production system with backlogging and failure in repair. *Journal of Scientific & Industrial Research*, *65*, 499-506.

Chiu, Y. S. P., Wu, M. F., Chiu, S. W., & Chang, H. H. (2015). A simplified approach to the multi-item economic production quantity model with scrap, rework, and multi-delivery. *Journal of Applied Research and Technology*, *13*(4), 472–476. https://doi.org/10.1016/j.jart.2015.09.004

Chopra, S., & Meindl, P. (2001). Strategy, Planning, and Operation. *Supply Chain Management*.

Coello Coello, C. A., & Lechuga, M. S. (2002). MOPSO: A proposal for multiple objective particle swarm optimization. *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, *2*, 1051–1056. https://doi.org/10.1109/CEC.2002.1004388

Conway, R., Maxwell, W., McClain, J. O., & Thomas, L. J. (1988). The Role of Work-in-Process Inventory in Serial Production Lines. *Operations Research*, *36*(2), 229–241. https://doi.org/10.1287/opre.36.2.229

Darwish, M. A. (2008). EPQ models with varying setup cost. *International Journal of Production Economics*, *113*(1), 297–306. https://doi.org/10.1016/j.ijpe.2007.07.010

Diabat, A. (2014). Hybrid algorithm for a vendor managed inventory system in a two-echelon supply chain. *European Journal of Operational Research*, *238*(1), 114–121. https://doi.org/10.1016/j.ejor.2014.02.061

Fattahi, P., Hajipour, V., & Nobari, A. (2015). A bi-objective continuous review inventory control model: Pareto-based meta-heuristic algorithms. *Applied Soft Computing Journal*, *32*, 211–223. https://doi.org/10.1016/j.asoc.2015.02.044

Goyal, S. K. (1976). An integrated inventory model for a single supplier- single customer problem. *International Journal of Production Research*, *15*(1), 107–111. https://doi.org/http://dx.doi.org/10.1080/00207547708943107

Grubbström, R. W., & Erdem, A. (1999). EOQ with backlogging derived without derivatives. *International Journal of Production Economics*, *59*(1), 529–530. https://doi.org/10.1016/S0925-5273(98)00015-2

Guo, X., Liu, C., Xu, W., Yuan, H., & Wang, M. (2014). A prediction-based inventory optimization using data mining models. *Proceedings - 2014 7th International Joint Conference on Computational Sciences and Optimization, CSO 2014*, 611–615. https://doi.org/10.1109/CSO.2014.118

Hayek, P. A., & Salameh, M. K. (2001). Production lot sizing with the reworking of imperfect quality items produced. *Production Planning & Control*, *12*(6), 584–590.

Huang, B., & Wu, A. (2016). *EOQ model with batch demand and planned backorders*. *Applied Mathematical Modelling* (Vol. 40). Elsevier Inc. https://doi.org/10.1016/j.apm.2016.01.004

Huseyinov, I., & Bayrakdar, A. (2019). Performance Evaluation of NSGA-III and SPEA2 in Solving a Multi-Objective Single-Period Multi-Item Inventory Problem. *UBMK 2019 - Proceedings, 4th International Conference on Computer Science and Engineering*, (4), 531–535. https://doi.org/10.1109/UBMK.2019.8907139

Jones, D. F., Mirrazavi, S. K., & Tamiz, M. (2002). Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, *137*(1), 1–9.

Kanyalkar, A. P., & Adil, G. K. (2005). An integrated aggregate and detailed planning in a multi-site production environment using linear programming. *International Journal of Production Research*, *43*(20), 4431–4454. https://doi.org/10.1080/00207540500142332

Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization Proceedings., IEEE International Conference. *Proceedings of ICNN'95 - International Conference on Neural Networks*, *11*(1), 111–117.

Kim, E., & Park, T. (2016). Admission and inventory control of a single-component make-to-order production system with replenishment setup cost and lead time. *European Journal of Operational Research*, *255*(1), 91–102. https://doi.org/10.1016/j.ejor.2016.04.021

Kodama, M. (1995). Some Probabilistic Multiperiod Inventory Problems. *IFAC Proceedings Volumes*, *28*(7), 217–222. https://doi.org/10.1016/S1474-6670(17)47112-6

Lenard, J. D., & Roy, B. (1995). Multi-item inventory control: A multicriteria view. *European Journal of Operational Research*, *87*(3), 685–692. https://doi.org/10.1016/0377-2217(95)00239-1

Gen, M. and Cheng, R. (1997) *Genetic Algorithms & Engineering Design.* John Wiley & Sons, Inc., Hoboken.

Mandal, N. K., Roy, T. K., & Maiti, M. (2006). Inventory model of deteriorated items with a constraint: A geometric programming approach. *European Journal of Operational Research*, *173*(1), 199–210.

Mirzapour Al-e-Hashem, S. M. J., & Rekik, Y. (2014). Multi-product multi-period Inventory Routing Problem with a transshipment option: A green approach. *International Journal of Production Economics*, *157*(1), 80–88. https://doi.org/10.1016/j.ijpe.2013.09.005

Mirzazadeh, A., Fatemi Ghomi, S. M. T., & Seyed Esfahani, M. M. (2011). A multiple items inventory model under uncertain external inflationary conditions. *Trends in Applied Sciences Research*, *6*(5), 472–480.

Moosavi, V., Talebi, A., & Shirmohammadi, B. (2014). Producing a landslide inventory map using pixel-based and object-oriented approaches optimized by Taguchi method. *Geomorphology*, *204*, 646–656. https://doi.org/10.1016/j.geomorph.2013.09.012

Mousavi, S. M., Niaki, S. T. A., Bahreininejad, A., & Musa, S. N. (2014). Multi-item multiperiodic inventory control problem with variable demand and discounts: A particle swarm optimization algorithm. *Scientific World Journal*, *2014*. https://doi.org/10.1155/2014/136047

Niu, B., Tan, L., Liu, J., Liu, J., Yi, W., & Wang, H. (2019). Cooperative bacterial foraging optimization method for multi-objective multi-echelon supply chain optimization problem. *Swarm and Evolutionary Computation*, *49*(May 2018), 87–101. https://doi.org/10.1016/j.swevo.2019.05.003

Park, K., & Kyung, G. (2014). Optimization of total inventory cost and order fill rate in a supply chain using PSO. *International Journal of Advanced Manufacturing Technology*, *70*(9–12), 1533–1541. https://doi.org/10.1007/s00170-013-5399-6

Pasandideh, S. H. R., Niaki, S. T. A., & Sharafzadeh, S. (2013). 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), 764–770. https://doi.org/10.1016/j.jmsy.2013.08.001

Rau, H., Daniel, S., & Agus, G. (2018). Optimization of the multi-objective green cyclical inventory routing problem using discrete multi-swarm PSO method. *Transportation Research Part E*, *120*(September), 51–75. https://doi.org/10.1016/j.tre.2018.10.006

Roozbeh Nia, A., Hemmati Far, M., & Niaki, S. T. A. (2015). A hybrid genetic and imperialist competitive algorithm for green vendor managed inventory of multi-item multi-constraint EOQ model under shortage. *Applied Soft Computing Journal*, *30*, 353–364. https://doi.org/10.1016/j.asoc.2015.02.004

Rossi, T., Pozzi, R., & Testa, M. (2017). EOQ-based inventory management in single-machine multi-item systems. *Omega (United Kingdom)*, *71*, 106–113. https://doi.org/10.1016/j.omega.2016.10.002

Roy, T. K., & Maiti, M. (1998). Multi-objective inventory models of deteriorating items with some constraints in a fuzzy environment. *Computers & Operations Research*, *25*(12), 1085–1095.

S Akindipe, O. (2014). The Role of Raw Material Management in Production Operations. *International Journal of Managing Value and Supply Chains*, *5*(3), 37–44. https://doi.org/10.5121/ijmvsc.2014.5303

Sadeghi, J., Mousavi, S. M., Niaki, S. T. A., & Sadeghi, S. (2014). Optimizing a bi-objective inventory model of a three-echelon supply chain using a tuned hybrid bat algorithm. *Transportation Research Part E: Logistics and Transportation Review*, *70*(1), 274–292. https://doi.org/10.1016/j.tre.2014.07.007

Sarwar, F., Rahman, M., & Ahmed, M. (2019). A Multi-Item Inventory Control Model using Multi Objective Particle Swarm Optimization (MOPSO), 1311–1321.

Scholkopf, B., Sung, K.-K., Burges, C. J. C., Girosi, F., Niyogi, P., Poggio, T., & Vapnik, V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, *45*(11), 2758–2765.

Schott, J. (1995). Fault Tolerant Design Using Single and Multi-Criteria Genetic Algorithms. *Master's Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology*.

Sutrisno, & Wicaksono, P. A. (2015). Optimal Strategy for Multi-product Inventory System with Supplier Selection by Using Model Predictive Control. *Procedia Manufacturing*, *4*(Iess), 208–215. https://doi.org/10.1016/j.promfg.2015.11.033

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

Taguchi, G. (1990). Introduction to quality engineering, Tokyo. *Asian Productivity Organization*.

Taleizadeh, A A, Moghadasi, H., Niaki, S. T. A., & Eftekhari, A. (2008). An economic order quantity under joint replenishment policy to supply expensive imported raw materials with payment in advance. *Journal of Applied Sciences*. https://doi.org/10.3923/jas.2008.4263.4273

Taleizadeh, Ata Allah, Niaki, S. T. A., & Aryanezhad, M. B. (2009). A hybrid method of Pareto, TOPSIS and genetic algorithm to optimize multi-product multi-constraint inventory control systems with random fuzzy replenishments. *Mathematical and Computer Modelling*, *49*(5–6), 1044–1057. https://doi.org/10.1016/j.mcm.2008.10.013

Taleizadeh, Ata Allah, Niaki, S. T. A., Aryanezhad, M. B., & Shafii, N. (2013). A hybrid method of fuzzy simulation and genetic algorithm to optimize constrained inventory control systems with stochastic replenishments and fuzzy demand. *Information Sciences*, *220*, 425–441. https://doi.org/10.1016/j.ins.2012.07.027

Taleizadeh, Ata Allah, Niaki, S. T. A., Shafii, N., Meibodi, R. G., & Jabbarzadeh, A. (2010). A particle swarm optimization approach for constraint joint single buyer-single vendor inventory problem with changeable lead time and (r,Q) policy in supply chain. *International Journal of Advanced Manufacturing Technology*, *51*(9–12), 1209–1223. https://doi.org/10.1007/s00170-010-2689-0

Tavana, M. (2016). A bi-objective inventory optimization model under inflation and discount using tuned Pareto-based algorithms : NSGA-II , NRGA , and MOPSO, *43*, 57–72.

Tiwari, S., Daryanto, Y., & Wee, H. M. (2018). Sustainable inventory management with deteriorating and imperfect quality items considering carbon emission. *Journal of Cleaner Production*, *192*, 281–292. https://doi.org/10.1016/j.jclepro.2018.04.261

Van Veldhuizen, D. A. (1999). Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. *Ph. D. thesis, Air Force Institute of Technology Wright-Pattersonafb OH School of Engineering*.

Wang, Gaige, Guo, L., Duan, H., Liu, L., & Wang, H. (2012). A bat algorithm with mutation for UCAV path planning. *The Scientific World Journal*, *2012*. https://doi.org/10.1100/2012/418946

Wang, Guanghui. (2012). Demand forecasting of supply chain based on Support Vector Regression method. *Procedia Engineering*, *29*, 280–284. https://doi.org/10.1016/j.proeng.2011.12.707

Wu, Q. (2010). Product demand forecasts using wavelet kernel support vector machine and particle swarm optimization in manufacture system. *Journal of Computational and Applied Mathematics*, *233*(10), 2481–2491. https://doi.org/10.1016/j.cam.2009.10.030

Yang, X. S., (2011), Bat Algorithm for Multiobjective Optimization, *International Journal of Bio-Inspired Computation, 3*(5), 267-274.

Yang, X. S., & He, X. (2013). Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation*, *5*(3), 141. https://doi.org/10.1504/IJBIC.2013.055093

Zitzler, E., & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms - A comparative case study. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *1498 LNCS*(September), 292–301. https://doi.org/10.1007/bfb0056872

Appendix A. Formulation of EOQ

From total cost function of raw material inventory (Eq. 6),

$$TC = \frac{D_i}{Q_i}S_i + \frac{H_i}{2Q_i}(Q_i - I_i)^2 + \frac{L_i}{2Q_i}{I_i}^2 + ED_iC_t + D_iT_i$$

Differentiating by Q,

$$\frac{d(TC)}{dQ} = DS\left(-\frac{1}{Q^2}\right) + \frac{H}{2}\left(-\frac{1}{Q^2}\right)(Q-I)^2 + \frac{H}{2Q} \times 2(Q-I) + \frac{LI^2}{2}\left(-\frac{1}{Q^2}\right) + 0 + 0$$

$$or, 0 = 2DS + LI^2 + H(Q-I)^2 - 2Q(Q-I)H$$

$$or, H(Q-I)^2 + 2HQ(Q-I) = -(2DS + LI^2)$$

$$or, (Q-I)(Q-I-2Q) = -\frac{(2DS + LI^2)}{H}$$

$$or, Q^2 - I^2 = \frac{(2DS + LI^2)}{H}$$

$$or, Q^2 = \frac{(2DS + LI^2)}{H} + I^2$$

$$or, Q = \sqrt{\frac{2DS}{H} + \frac{LI^2}{H} + I^2}$$

$$EOQ = \sqrt{\frac{2DS}{H} + \frac{LI^2}{H} + I^2} \tag{A.1}$$

Appendix B. Formulation of EPQ

From total cost function of WIP inventory (Eq. 17),

$$TC = \frac{h_id_i}{2(1-p_i)}[s_i + m_iq_i]\left[2M_i + \frac{r_is_i}{q_i} + r_iM_i\right] + \frac{c_id_i}{q_i(1-p_i)} + \frac{m_id_i}{q_i(1-p_i)} + \frac{i_id_i}{q_i(1-p_i)}$$

Differentiating by q,

$$\frac{d(TC)}{dq} = \frac{hd}{2(1-p)}2(s+mq)\left(-\frac{rs}{q^2}\right) + \frac{hd}{2(1-p)}m\left[2M + \frac{rs}{q} + rM\right] + \frac{cd}{(1-p)}\left(-\frac{1}{q^2}\right) + \frac{md}{(1-p)}\left(-\frac{1}{q^2}\right) + \frac{id}{(1-p)}\left(-\frac{1}{q^2}\right)$$

$$or, 0 = \frac{hd}{2(1-p)}\left[2Mm + \frac{rsm}{q} + rMm - \frac{rs^2}{q^2} - \frac{rsm}{q}\right] - \frac{cd + md + id}{(1-p)q^2}$$

$$or, \frac{(c+m+i)d}{(1-p)q^2} = \frac{hd}{2(1-p)}\left[2Mm + rMm - \frac{rs^2}{q^2}\right]$$

$$or, \frac{(c+m+i)}{q^2} = \frac{h}{2}Mm(r+2) - \frac{h}{2}\frac{rs^2}{q^2}$$

$$or, \frac{1}{q^2}\left(c + m + i + \frac{hrs^2}{2}\right) = \frac{h}{2}Mm(r+2)$$

$$or, q^2 = \frac{\left(c + m + i + \frac{hrs^2}{2}\right)}{\frac{h}{2}Mm(r+2)}$$

$$or, q = \sqrt{\frac{2(c+m+i) + hrs^2}{hMm(r+2)}}$$

$$EPQ = \sqrt{\frac{2(c+m+i) + hrs^2}{hMm(r+2)}} \tag{B.1}$$

Appendix C

Table C1
Data for shortage prediction

Order Quantity	Demand	Ordering Cost	Holding Cost per Unit	Shortage Level	GHG Level	Emission Cost	Pallet Count	Truck capacity	Storage Space	Available Storage	Shipping cost	Upper Bound on GHG	Unit cost of Shortage	Purchase Cost	Total Budget
5870	7805	308	50	1935	0.462	120.00	117	6	4.23	6	0.32	0.36	0.2	3	30000
7947	5504	382	75	-2443	0.149	120.00	159	8	5.72	6	0.32	0.36	0.2	3	30000
4209	5962	353	58	1753	0.763	120.00	84	4	3.03	6	0.32	0.36	0.2	3	30000
4616	7701	461	67	3085	0.357	120.00	92	5	3.32	6	0.32	0.36	0.2	3	30000
7495	7795	567	60	300	0.714	120.00	150	7	5.40	6	0.32	0.36	0.2	3	30000
4959	5811	434	69	852	0.621	120.00	99	5	3.57	6	0.32	0.36	0.2	3	30000
6905	8323	594	52	1418	0.236	120.00	138	7	4.97	6	0.32	0.36	0.2	3	30000
5425	7361	380	66	1936	0.345	120.00	109	5	3.91	6	0.32	0.36	0.2	3	30000
7447	6048	414	61	-1399	0.806	120.00	149	7	5.36	6	0.32	0.36	0.2	3	30000
5435	7405	538	58	1970	0.560	120.00	109	5	3.91	6	0.32	0.36	0.2	3	30000
5689	7671	547	53	1982	0.608	120.00	114	6	4.10	6	0.32	0.36	0.2	3	30000
7307	5510	474	65	-1797	0.551	120.00	146	7	5.26	6	0.32	0.36	0.2	3	30000
6415	6423	577	54	8	0.360	120.00	128	6	4.62	6	0.32	0.36	0.2	3	30000
5672	9535	534	50	3863	0.301	120.00	113	6	4.08	6	0.32	0.36	0.2	3	30000
6542	8040	315	54	1498	0.968	120.00	131	7	4.71	6	0.32	0.36	0.2	3	30000

Table C2
Data for scrap rate prediction

Order Quantity	Demand	R/M Cost per Unit	Holding Cost per Unit	Inspection Cost per Unit	Machining Time per Unit	Constant Setup Cost	Scrap Production %	Setup Time per Unit	Production Rate per Unit	Total Order	Procurement Cost per Unit	Space Occupied	Space Available	Total Budget
4513	9159	82	74	124	1308	662	5.18	886	1.902	8980	2200	3.41	4.5	50000
4992	7787	93	90	141	1622	613	4.60	783	1.975	9784	2492	3.43	4.5	50000
7806	5329	91	72	130	1497	624	5.33	877	2.193	9362	3902	3.41	4.5	50000
8028	8439	95	77	142	1315	649	4.41	621	2.225	8167	2873	3.44	4.5	50000
7749	5670	73	71	120	1549	602	4.96	655	1.007	8927	4270	3.42	4.5	50000
8771	7275	96	73	140	1556	656	5.38	791	1.262	8641	2919	3.41	4.5	50000
4656	6514	81	79	150	1663	648	4.11	733	1.453	8636	4478	3.45	4.5	50000
5544	7139	77	89	145	1341	692	4.44	624	1.545	9157	3431	3.44	4.5	50000
8425	7337	80	71	120	1333	692	4.44	759	1.104	8752	2150	3.44	4.5	50000
5976	6958	93	74	137	1669	643	4.49	871	1.453	8566	2319	3.44	4.5	50000
7396	7138	98	73	140	1312	624	4.92	644	1.950	8485	2365	3.42	4.5	50000
4327	5007	74	82	147	1211	672	5.00	653	2.547	9694	4330	3.42	4.5	50000
4916	9354	80	88	143	1282	623	4.94	859	1.734	9771	3018	3.42	4.5	50000
8498	8041	86	71	120	1247	656	5.15	865	1.261	8494	3689	3.41	4.5	50000
8670	9521	95	84	126	1321	679	5.16	698	1.738	8656	2932	3.41	4.5	50000