Author(s): Mousavi Abdehgah, Mohsen; Pardalos, Panos M.; Niaki, Seyed Taghi Akhavan; Fügenschuh, Armin; Fathi, Mahdi

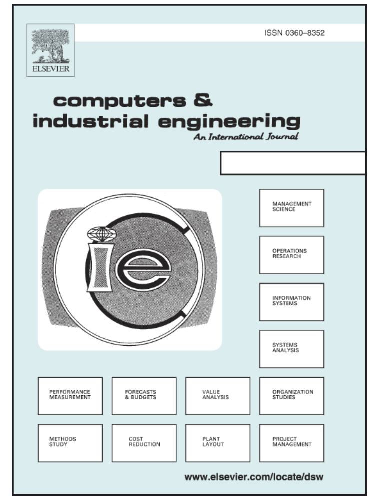
Title: Solving a continuous periodic review inventory-location allocation problem in vendor-buyer supply chain under uncertainty

Year: 2019

Version: Accepted version (Final draft)

Please cite the original version:

Mousavi Abdehgah, M., Pardalos, P. M., Niaki, S. T. A., Fügenschuh, A., & Fathi, M. (2019). Solving a continuous periodic review inventory-location allocation problem in vendor-buyer supply chain under uncertainty. Computers and Industrial Engineering, 128, 541-552. https://doi.org/10.1016/j.cie.2018.12.071



Solving a Continuous Periodic Review Inventory-Location Allocation Problem in Vendor-Buyer Supply Chain under Uncertainty

Seyed Mohsen Mousavi, Panos M. Pardalos, Seyed Taghi Akhavan Niaki, Armin Fügenschuh, Mahdi Fathi

PII: S0360-8352(18)30674-0
DOI: https://doi.org/10.1016/j.cie.2018.12.071
Reference: CAIE 5624

To appear in: *Computers & Industrial Engineering*

Received Date: 5 July 2018
Revised Date: 11 November 2018
Accepted Date: 29 December 2018

Solving a Continuous Periodic Review Inventory-Location Allocation Problem in Vendor-Buyer Supply Chain under Uncertainty

Seyed Mohsen Mousavi[*a]
[a]University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland, Email: smousavi@jyu.fi

Panos M. Pardalos[b]
[b]Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA, Email: pardalos@ufl.edu

Seyed Taghi Akhavan Niaki[c]
[c]Department of Industrial Engineering, Sharif University of Technology, P.O. Box 11155-9414 Azadi Ave, Tehran 1458889694 Iran, Email: niaki@sharif.edu

Armin Fügenschuh[d]
[d]Brandenburg University of Technology Cottbus-Senftenberg, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany, Email: fuegenschuh@b-tu.de

Mahdi Fathi[b]
[b]Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA, Email: mfathi.ie@gmail.com

[*] Corresponding author at: University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland, e-mail: smousavi@jyu.fi

Solving a Continuous Periodic Review Inventory-Location Allocation Problem in Vendor-Buyer Supply Chain under Uncertainty

Abstract

In this work, a mixed-integer binary non-linear two-echelon inventory problem is formulated for a vendor-buyer supply chain network in which lead times are constant and the demands of buyers follow a normal distribution. In this formulation, the problem is a combination of an (r, Q) and periodic review policies based on which an order of size Q is placed by a buyer in each fixed period once his/her on hand inventory reaches the reorder point r in that period. The constraints are the vendors' warehouse spaces, production restrictions, and total budget. The aim is to find the optimal order quantities of the buyers placed for each vendor in each period alongside the optimal placement of the vendors among the buyers such that the total supply chain cost is minimized. Due to the complexity of the problem, a Modified Genetic Algorithm (MGA) and a Particle Swarm Optimization (PSO) are used to find optimal and near-optimum solutions. In order to assess the quality of the solutions obtained by the algorithms, a mixed integer nonlinear program (MINLP) of the problem is coded in *GAMS*. A design of experiment approach named Taguchi is utilized to adjust the parameters of the algorithms. Finally, a wide range of numerical illustrations is generated and solved to evaluate the performances of the algorithms. The results show that the MGA outperforms the PSO in terms of the fitness function in most of the problems and also is faster than the PSO in terms of CPU time in all the numerical examples.

Keywords: *Inventory-location allocation problem; Mixed-integer binary non-linear programming; Two-echelon supply chain; Stochastic demands; Genetic Algorithm*

1. Introduction

In today's competitive markets companies have to update their logistic systems regularly to capture bigger market share by solving the existing difficulties involved in producing the items, the uncertainties in predicting the demands, the constraints in supplying the items and loading a wide range of items with varying volumes. To reach this aim, the companies need to use preferably the best strategy to integrate their logistic networks as well as their inventory systems, transportation, warehouses and vendors to minimize the total cost of operations. This research studies a real-world situation of a two-echelon inventory-supply chain problem in which some current limitations in the industry are considered.

Multi-products inventory control problems in finite time-periods have been addressed well by many researchers in recent years. Yang et al. (2017) proposed a mixed-integer linear program for a multi-item inventory problem in finite horizon under non-stationary demand, arbitrary review period, and restricted available inventory budget. Alikar et al. (2017a) modeled a multiple items multiple period inventory control problem for a series-parallel redundancy allocation problem (RAP) in which the total inventory cost was calculated with respect to the time value of money and inflation rates. The total budget for buying the items, the total storage spaces and the truck capacity for transferring the items were limited. Their research was conducted in a deterministic environment with a fixed demand where the lead times were not considered. Alikar et al. (2017b) developed a mixed-integer binary nonlinear model for a multi-product inventory control problem with a finite time-period in a series-parallel RAP problem, in which the products were bought under an all unit discount strategy. In their model, the storage space, the total available budget, the capacities of the vehicles and the system's total weight were constrained. The lead time was assumed to be negligible in their work and also the demands were deterministic. Shankar et al. (2018) presented a mixed-integer nonlinear model for a multiple-product multi-echelon finite horizon inventory-supply chain problem in which some vital factors of the automobile supply chain strategy were integrated. They assumed that no lead times were required. Considering time and cost restrictions, a multi-item multiple periods inventory control problem was improved for a routing model by Peres et al. (2017) where transhipment movements were handled by identical trucks with a unique capacity. They used an exact method and a meta-heuristic algorithm to solve the problem on a case study from a company in the Brazilian retail industry where the demand was assumed fixed and there was no lead time. Liao et al. (2017) proposed a multi-item inventory model in a finite horizon and fuzzy environment with the aim of maximizing the total profits of the retailers. In their work, the lead times were assumed negligible. Mousavi et al. (2013) used a genetic algorithm to optimize an inventory control problem with multiple products in finite time-period where the costs were computed with respect to the time value of money and inflation rates. In their work, discount policies, i.e., an all-unit discount and an incremental quantity discount were applied. The constraints of the problem were the limitation in storage space, supplying order quantity and the total budget at hand. They did not investigate the supply chain members in their work where the demands were deterministic and the lead time was assumed zero. A mixed-integer linear model was developed by Correia & Melo (2017) for a multi-period inventory location-allocation problem, in which customer segments had different sensitivity to delivery lead times. They used a general-purpose solver to optimize the formulated mixed-integer linear program. The demands in their work were considered deterministic.

In this study, an inventory control problem is formulated for a buyer-vendor supply chain where vendors store their produced items in their own warehouses in order to meet the demands. Supply chain inventory control problem with multiple products and multiple time periods is a popular topic studied by many scholars in different industries. Cárdenas-Barrón et al. (2015) presented a multiple items multi-period inventory lot-sizing problem for a supply chain, in which the best suppliers were to be chosen. To find a near-optimal solution, they solved their problem using an approximation method. No lead times were considered and the demands were deterministic. Sepehri (2011) studied a multiple products multiple time periods inventory model for a supply chain problem where a simulation approach was utilized to solve the problem. The retailers' demands were assumed fixed and there were no lead times for delivery of the products. An inventory control problem with a wide range of items and periods was proposed by Mirzapour Al-e-hashem & Rekik (2014) for a routing problem where items were delivered by capacitated trucks from the suppliers to a plant. Since the model was a mixed-integer linear programming, a standard solver (*IBM ILOG CPLEX*) was used to find the optimal solution of the problem. They modeled the problem with deterministic demands, which can be far from the real world applications. Mousavi et al. (2015) dealt with a multiple products finite horizon inventory-location allocation problem for a retailer-distributer supply chain problem where the distance between retailers and distributors were assumed to be Euclidean and Square Euclidean functions. Two discount strategies as well as all-unit discount and incremental quantity discount were considered and the orders were received in special packets. In their work, a fruit fly optimization algorithm was improved to optimize the proposed problem. Lead times were not considered in their work and the demands were supposed to be deterministic. Moreover, the quality of the solutions found by their applied algorithm was not justified with the one obtained by an exact solution method. A multi-product seasonal (multiple periods) inventory location-allocation problem was formulated by Mousavi et al. (2017b) in a two-echelon buyer-vendor supply chain in which the shortages were not allowed and all-unit discount policy was used to purchase the items. A modified particle-swarm optimization (PSO) algorithm along with a genetic algorithm was utilized to solve the problem. They While the lead times were assumed negligible in their work, they did not assess the performance of their solution algorithms with the one of an exact method. Paksoy & Chang (2010) considered a multi-stage inventory model in a finite horizon for a supply chain problem with multiple popup warehouses and developed a mixed-integer binary linear program. Three multiple goals were investigated where the revised multi-choice goal programming approach was utilized to solve the problem at hand on a real industrial case study. The customer demands were fixed and no lead times were considered in their research. Jonrinaldi & Zhang (2013) formulated an integrated production multi-item multi-period inventory control problem for a supply chain where several decision making processes and solving methods were used in the proposed mixed integer nonlinear model. Their model assumed constant demand rates and zero lead times.

This article considers an inventory-supply chain problem under uncertainty while the demands of the buyers and the purchasing items from the vendors are stochastic. Rafie-Majd et al. (2018) formulated a three-echelon multi-item multi-period inventory-location problem for a routing supply chain problem where the demands of the customers were considered stochastic. Their approach takes into account the vehicle timetables, fuel consumption, product wastage, and setup cost. Qiu et al. (2017) developed a model for a multi-period inventory control problem structured in a dynamic program with demand uncertainty where a robust optimization method was used to solve the problem. No lead times were investigated in their work. Mousavi et al. (2014) studied an inventory control problem with multiple products in a finite time-period where the total available budget was limited and shortage costs were allowed for all products in combination with backorders and lost sales. They formulated the problem in a fuzzy environment in which the discount rates and the storage space for storing the items were considered as fuzzy numbers. The supply chain members were not brought to the model and the lead times were assumed negligible. Janakiraman et al. (2013) analyzed an inventory control problem in multiple periods for a newsvendor in which the lead times were stochastic and a dilation ordering of lead times implied an ordering of optimal costs. De & Sana (2014) considered a multi-period production-inventory problem with multiple producers in a plant with a multiple shop/delivery system and different machines where the cost function was considered to be fuzzy numbers. Aharon et al. (2009) modeled a multi-period multiple echelons supply chain problem with stochastic uncertainty where a robust optimization method called Affinely Adjustable Robust Counterpart was used to solve the problem. Nasiri et al. (2014) formulated a hierarchical model for designing a production-distribution inventory in a location-allocation problem with multiple-level capacitated warehouses. In order to obtain near-optimal solutions, both Lagrangian relaxation and a genetic algorithm were applied. In order to find better solutions in a shorter time, they employed the Taguchi approach to tune the parameters of their proposed algorithms. In this approach, the number of experiments needed to find the best values of the algorithms' parameters is reduced considerably. There are a number of works published recently in the literature that used the Taguchi approach for tuning the parameters in inventory and supply chain fields. Interested readers are referred to Mousavi et al. (2015), Mousavi et al. (2017b), Mousavi et al. (2013), and Mousavi et al. (2014) for more details.

The novelties involved in this paper are as follows. First, this work formulates a novel multi-item multi-period inventory-location allocation problem for a two-echelon buyer-vendor supply chain problem. The second novelty is that the problem is formulated under uncertainty while the demands of the buyers are considered stochastic. Moreover, the lead times are assumed constant while it was considered negligible in the related previous works. Furthermore, a modified version of the genetic algorithm, named MGA, and a PSO are applied to obtain near-optimal quantities of the items ordered by the buyers from the vendors in addition to finding near-optimal locations of each vendor placed among the buyers.

The rest of the paper is organized as follows. In the next section, the problem description is given. Indices, notations, and assumptions of the proposed problem come in Section 3. The problem formulations, including the objective function and also the constraints of the model, are presented in Section 4. In Section 5, a modified version of genetic algorithms (MGA) is developed to solve the problem. Section 6 describes the parameter calibration approach and Section 6 shows computational results to evaluate the MGA, in which 20 different numerical examples with different sizes are first generated, and then the Taguchi approach is utilized to tune the algorithm parameters on the generated examples. Finally, the conclusion of the work is described in Section 8.

2. Problem description

In this work, a two-echelon multi-item multi-period inventory control problem is formulated in a buyer-vendor supply chain network, in which the vendors manufacture different products and then store them in their own warehouses to meet the future demands of the buyers. Moreover, the vendors sell their products under an all-unit discount policy, where each vendor can propose different policy with different price break-points. In fact, when a buyer orders a particular item from a vendor, the vendor will charge the buyer based on the quantity of the item requested for which the price break-point provided by the vendor applies. The warehouse spaces, the total budget of the buyers and the total production capacity of the vendors are limited. Furthermore, the vendors deliver their products in special boxes each with a pre-determined number of products. In the model, the demands of the buyers are assumed to be stochastic and all follow a normal distribution where shortages are not allowed. Moreover, lead times of the products are assumed to be constant and there is a limitation on the service levels of the products in each period. The aim is to find out the reorder point in addition to the order quantity of each item so that the total supply chain cost is minimized. The proposed inventory-supply chain model is shown to be a mixed-integer binary non-linear programming type where two meta-heuristic algorithms, i.e., MGA and PSO, are used to solve the problem. In order to find suitable parameters of the algorithm, a design of experiment approach, i.e. the Taguchi method is used to adjust the MGA and PSO parameters.

Figure 1 shows the supply chain system under investigation. In the next section, the indices, notations, and assumptions of the problem will be presented.

Insert Figure 1 here

3. Indices, notations, and assumptions of the problem

All the notations and indices applied in this work are listed as follows.

3.1. Indices and notations

$i = 1, 2,..., I$ is the index of the buyers

$j = 1, 2,..., J$ is the index of the products

$k = 1, 2,..., K$ is the index of the vendors

$t = 0, 1,..., N$ is the index of the time periods

D_{ijkt} : Expected demand quantity of buyer i for product j produced by vendor k in period t

$f_{ijkt}\left(\mathrm{D}_{ijkt}\right)$: Probability density functions of D_{ijkt} (a normal distribution with mean $\mu_{D_{ijkt}}$ and standard deviation $\delta_{D_{ijkt}}$)

T_{ijkt} : The time at which the j^{th} product ordered by buyer i from vendor k is received

F_k : The production capacity of vendor k

h_{ijkt} : Inventory holding cost per unit of j^{th} product in the warehouse owned by vendor k sold to buyer i in period t

A_{ijkt} : Ordering cost (transportation cost) per unit of j^{th} product from vendor k to buyer i in period t

c_{ijktp} : Purchasing cost per unit of j^{th} product paid by buyer i to vendor k at p^{th} price break point in period t

s_{ijkt} : The required warehouse space for vendor k to store a unit of j^{th} product sold to buyer i in period t

S_i : The available capacity of i^{th} buyer's warehouse

TB: The total available budget

w_{ijkt}: A binary variable that is set to 1 if buyer i orders product j from vendor k in period t, and set to 0 otherwise

Q_{ijkt}: Ordering quantity of j^{th} product purchased by buyer i from vendor k in period t (decision variable)

V_{ijkt}: The number of special boxes of j^{th} product proposed by vendor k to buyer i in period t (decision variable)

n_j: The number of j^{th} product contained in each box

X_{ijkt}: The initial (remained) positive inventory of j^{th} product purchased by buyer i from vendor k in period t (decision variable)

I_{ijkt}: Inventory position j^{th} product for buyer i purchased from vendor k in period t

SS_{ijkt}: Safety stock of j^{th} product for buyer i purchased from vendor k in period t

r_{ijkt}: Reorder point of j^{th} product for buyer i purchased from vendor k in period t

L_{ijkt}: Lead time of j^{th} product for buyer i purchased from vendor k in period t

u_{ijktp}: p^{th} price break-point proposed by vendor k to buyer i for purchasing j^{th} product in period t

λ_{ijktp}: A binary variable that is set to 1 if buyer i purchases product j from vendor k at price break point p in period t, and set to 0 otherwise

$a_i = (a_{i1}, a_{2i})$: The coordinates of the location of buyer i

$y_k = (y_{1k}, y_{2k})$: The potential region of vendor k (decision variable)

M : Maximum inventory level

TC_h: Expected total holding cost

TC_P: Expected total purchasing cost

TC_O: Expected total ordering (transportation) cost

TC: Expected total supply chain cost

3.2. Assumptions

- The buyers' demand rates of all products are stochastic and follow a normal distribution.

- The initial positive inventory level of the items sold out by each vendor to each buyer is zero (i.e., $X_{ijk1} = 0$)
- All orders are placed on a given finite horizon that includes *N fixed* time periods of equal length.
- The orders must be received at the beginning of the next period; thus two scenarios may happen within a period, either the lead time is positive or zero. In other words, if the inventory level reaches below the reorder point, an order is placed and will be received at the beginning of the next period. Even if the inventory level does not reach the reorder point during a period, the order will be received immediately at the beginning of the next period.
- The total storage space, the total production capacity to produce items by each vendor and the total available budget to buy the items are restricted.
- No order is made in the last period.
- The orders arrive in special boxes of a pre-specified number of products.
- The orders should be received at time T, so the lead time would be between the time an order is placed and T.

4. The problem formulation

In this section, we propose a mixed-integer binary non-linear model for the inventory supply chain problem at hand. Figure 2 shows some scenarios of the inventory model.

Insert Figure 2 here

The objective function and the constraints of the model are formulated as follows.

4.1. The objective function

First, let us consider a problem in which shortages are not allowed and the stochastic demands follow a normal distribution. In this problem, the total cost of the proposed supply chain is calculated as:

$$TC = TC_O + TC_h + TC_P . \qquad (1)$$

For $\{T_{ijk1}, T_{ijk2}, ..., T_{ijkN}\}; T_{ijkt+1} > T_{ijkt}$ $(\mathit{for}\ t = 1,...,N)$ the total ordering (transportation) cost, the holding cost, and purchasing cost will be obtained as follows.

The total transportation cost is given by Eq. (2).

$$TC_O = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{N}(Q_{ijkt}A_{ijkt}d(y_k, a_i)), \quad (2)$$

where $d(y_k, a_i)$ is the distance function between the location of vendor *k* and buyer *i*, considering to be the Euclidean function defined as follows:

$$d(y_k, a_i) = \sqrt{(y_{k1} - a_{1i})^2 + (y_{k2} - a_{2i})^2}$$

From Fig 2, the total holding cost will be given by:

$$TC_h = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{N}\int_{T_{ijkt}}^{T_{ijkt+1}} h_{ijkt} I_{ijkt} dt \quad (3)$$

The demands of the buyers should be covered by the positive level of inventory during lead time L_{ijkt} with a given probability $1-\alpha$ called the inventory service level specified by the decision makers (DMs) where this service level can be formulated as

$$Pr(D_{ijkt} \leq r_{ijkt}) = 1\text{-}\alpha \quad (4)$$

and we have:

$$r_{ijkt} = \bar{D}_{ijkt} + SS_{ijkt} \quad (5)$$

According to (Miranda & Garrido, 2004), the following formula is the result:

$$r_{ijkt} = E(D_{ijkt}).E(L_{ijkt}) + Z_{1-\alpha}.(\sqrt{(E(D_{ijkt}))^2 \delta^2_{L_{ijkt}} + E(L_{ijkt})\delta_{D_{ijkt}}}) \quad (6)$$

Equation (6) is simplified as the following formula when the L_{ijkt} is supposed to be a constant:

$$r_{ijkt} = D_{ijkt}.L_{ijkt} + Z_{1-\alpha}.\sqrt{\delta_{D_{ijkt}}}\sqrt{L_{ijkt}} \quad (7)$$

In Eq. (7), $Z_{1-\alpha}$ is the upper $(1-\alpha)$ percentile point of the standard normal distribution

Figure 2 shows the reorder point situations in the proposed model. Using the Weber problem (Drezner & Hamacher, 2001), the average holding cost rate in the interval period $[T_{ijkt}, T_{ijkt+1}]$ based on the equation above is computed as:

$$TC_h = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{N-1}\left\{h_{ijkt}\left(\frac{Q_{ijkt} + X_{ijkt}}{2}\right) + h_{ijkt}.Z_{1-\alpha}.\sqrt{L_{ijkt}}\sqrt{\delta_{D_{ijkt}}}\right\} \quad (8)$$

Eq. (8) includes the average cost borne due to storing the order quantity $(Q_{ijkt} + X_{ijkt})$ as the first part which is the inventory level of item *j* applied to cover the buyer demand received during two successive orders. The safety stock is the second average cost included in (8) which is stored in the storage owned by each vendor.

In this work, the vendors sell their products under some discount policies, i.e., all-unit discount and incremental quantity discount. The following equation is the price-break points proposed by the vendors for an all-unit discount policy:

$$\begin{cases} c_{ijkt1} & u_{ijkt1} \le Q_{ijkt} < u_{ijkt2} \\ c_{ijkt2} & u_{ijkt2} \le Q_{ijkt} < u_{ijkt3} \\ & \vdots \\ c_{ijktP} & u_{ijktP} \le Q_{ijkt} \end{cases}$$

Then, the total cost for purchasing the items from the vendor under all-unit discount strategy is calculated as:

$$TC_p = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{N-1}\sum_{p=1}^{P} Q_{ijkt} c_{ijktp} \lambda_{ijktp} \tag{9}$$

4.2. The constraints

The initial positive inventory of each buyer in each period remained from the previous period is formulated as follows:

$$X_{ijkt+1} = X_{ijkt} + \mathrm{Q}_{ijkt} - \mathrm{D}_{ijkt}(T_{ijkt+1} - T_{ijkt}) \tag{10}$$

Each vendor’s warehouse has a limited capacity that is shown by the following equation:

$$\sum_{j=1}^{J}\sum_{k=1}^{N}(Q_{ijkt} + x_{ijkt})s_{ijkt} \le S_i \tag{11}$$

The products are provided by each vendor in special boxes V_{ijkt} with the number of item n_j where its relevant constraint comes as follows:

$$Q_{ijkt} = n_j V_{ijkt} \tag{12}$$

When the production capacity of each plant owned by each vendor is restricted, the related constraint would be formulated as follows:

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{N} Q_{ijkt} \le F_k \tag{13}$$

The total available budget to buy the products from the vendors is limited which is given by the following formula:

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{N-1}\sum_{p=1}^{P} Q_{ijkt} c_{ijktp} \lambda_{ijktp} \le TB \tag{14}$$

While the order quantity Q_{ijkt} plus the remaining inventory cannot exceed the maximum inventory *M*, the relevant constraint is shown as:

$$Q_{ijkt} + X_{ijkt} \le M \tag{15}$$

Finally, the following constraint describes that a product can be only bought by each buyer at a price break point in each time.

$$\sum_{p=1}^{P} \lambda_{ijktp} = \begin{cases} 1 & if \ \mathrm{Q}_{ijkt} > 0 \\ 0 & otherwise \end{cases} \tag{16}$$

Therefore, the supply chain model for the first model is obtained as follows:

$$MinTC = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{N}(Q_{ijkt}A_{ijkt}d(y_k, a_i)) + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{N-1}\{h_{ijkt}.(\frac{Q_{ijkt}+X_{ijkt}}{2}) + h_{ijkt}.Z_{1-\alpha}.\sqrt{L_{ijkt}}\sqrt{\delta_{D_{ijkt}}}\} + \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{N-1}\sum_{p=1}^{P} Q_{ijkt}c_{ijktp}\lambda_{ijktp}$$

Subject to:

$$X_{ijkt+1} = X_{ijkt} + \mathrm{Q}_{ijkt} - \mathrm{D}_{ijkt}(T_{ijkt+1} - T_{ijkt})$$

$$\sum_{j=1}^{J}\sum_{k=1}^{K}(Q_{ijkt} + x_{ijkt})s_{ijkt} \le S_i$$

$$Q_{ijkt} = n_j V_{ijkt}$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{N} Q_{ijkt} \le F_k$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{t=1}^{N-1}\sum_{p=1}^{P} Q_{ijkt}c_{ijktp}\lambda_{ijktp} \le TB$$

$$Q_{ijkt} + X_{ijkt} \le M$$

$$r_{ijkt} = D_{ijkt}.L_{ijkt} + Z_{1-\alpha}.\sqrt{\delta_{D_{ijkt}}}\sqrt{L_{ijkt}}$$

$$\sum_{p=1}^{P} \lambda_{ijktp} = \begin{cases} 1 & if \ \mathrm{Q}_{ijkt} > 0 \\ 0 & otherwise \end{cases} \tag{17}$$

$$Q_{ijkt} \in Z, x_{ijkt} \ge 0; y_{ijkt}, \lambda_{ijktp} \in \{0,1\}; \ (for \ i = 1,2,...,I; \ j = 1,2,...,J; \ k = 1,2,...,K; \ t = 1,2,...,N)$$

5. Solving methodologies

The modified GA and PSO are the solution algorithms used in this paper to solve the problem modeled in (17).

5.1. The MGA

In this research, due to the complexity of the problem a modified version of the genetic algorithm called MGA is used to find out near-optimal order quantities of the products bought from each vendor by each buyer. The MGA steps are described as follows:

- *Initialization of the parameters and representation of the solutions*: The parameters of the MGA are the number of chromosomes (solutions) in the population (*Pop*), the probability of crossover (P_c), the probability of mutation (P_m), and the number of generation (G*en*). The decision variables proposed in this study are *Q* and *y*, where the rest of the decision variables will be obtained, automatically after having *Q* and *y*.
- *Evaluation of the solutions*: In this stage, all the chromosomes of the population are evaluated by the objective function *TC* proposed in Eq. (17). Figure 3 depicts the population of the generated chromosomes evaluated by the *TC* function.

Insert Figure 3 here

- *Selection operator*: After testing several approaches on the problem, a two-chromosome tournament approach is chosen to select two different chromosomes each time randomly and compare them in terms of *TC* after sorting TC of all population solutions in ascending order. The chromosome with the minimum *TC* will be selected to enter the reproduction pool.
- *Crossover operator*: In order to generate new solutions, a crossover operator is performed. First, a number between 0 and 1 is generated randomly for each solution of the population. Then, if the value is less than P_c, the related solution will be chosen for crossover operation. For two different chosen chromosomes R_1 and R_2, the crossover operator is performed using the following formulae:

$$R_1^* = R_1.\mu + R_2(1-\mu)$$
$$R_2^* = R_1.(1-\mu) + R_2\mu \quad , \qquad (18)$$

  where μ is a random number generated between 0 and 1 and R_1^* and R_2^* are the offspring. Note the value R and R^* include Q and y, where Q is an integer number and y is a number greater than or equal to zero.
- *Mutation operator:* In this paper, a one-point mutation operator is found to be the best approach to generate new solutions for the next generation. First, a random number is

generated between 0 and 1 for each chromosome. If that number is less than P_m, the related chromosome is chosen for the mutation operator. In the chosen chromosome, one gene of Q and two genes of y related to a location are selected randomly and then are changed in the range randomly.

- *Termination criteria:* The algorithm is ended up while the number of generation reaches a pre-specific value (*Gen)*.

5.2. The PSO

In order to validate the results obtained by the proposed MGA, a PSO algorithm is also used to solve the problem. The steps involved in PSO are summarized as follows (Mousavi, et al., 2017a):

- Initializing the parameters and representing the particles the same as shown in Figure 3.
- Initializing the position and velocity of each particle the same as the method performed in (Mousavi et al., 2017a).
- Selecting the process of particles using Pbest and Gbest of each generation.
- Generating new solutions for each particle by updating the positions and velocities.
- Reaching the maximum number of generation as a termination criterion.

6. Experimental design

Tuning parameters in an appropriate way can usually have an impressive effect on the performance of a meta-heuristic algorithm. Since the quality of the solution obtained by any meta-heuristic algorithm such as PSO and GA depends on the values of their chosen parameters, in this section, the Taguchi method is used to tune the parameters. In the work proposed by Eiben & Smit (2011) a conceptual framework for parameter setting in evolutionary algorithms is presented emphasising on two approaches to choose a parameter value: (1) parameter tuning approach, in which the parameter values are set during running the algorithm and (2) the parameter control approach, where the parameter values are changing while running the algorithm. In this work, the first approach is employed.

In a meta-heuristic algorithm, the parameters are controllable factors, the problem being solved is the process input, and the fitness function is the process output. Hence, the best way would be to tune the algorithm's parameters using the experimental design methods as explained as follows instead of applying the values set by other researchers or using a trial and error procedure. In the Taguchi method (Roy, 1990), the factors (here the parameters) which effect on the efficiency (response) of a process are classified into two types: noise factors *N* which are uncontrollable, and those factors *S* such as the parameters of a meta-heuristic algorithm that are controllable. The Taguchi employs the orthogonal arrays to design the experiments, and then uses an approach to control *N* in order to decrease the variation or scatter around the target; in other words, the design that is impressed less by *N* is a robust design (Sadeghi et al., 2013). In order to analyze the values obtained by the Taguchi, the standard approach and the signal to noise ratio (S/N) approach are utilized. In the standard approach, an analysis of variance is used for experiments with only one iteration whereas the second approach is employed for experiments with more than one iteration. In the meta-heuristic algorithms proposed in this work, more than one replication is needed and thus the second approach has to be applied.

According to *S*/*N* analysis, a good condition is observed if the signal is more than the noise (i.e. $S > N$). In this paper, the aim is to reach a condition that optimizes *S*/*N*. Three categories of characters exist in the Taguchi method, "smaller is better" for which the objective function is of a minimization type, "nominal is the best" for which the objective function has modest variance around its target and "bigger is better", where the objective function is of a maximization type. The *S*/*N* analysis of these three categories is formulated respectively by (Roy, 1990):

$$(S/N)_s = -10\ \log\left(\frac{1}{n}\sum_{m=1}^{n} {a_m}^2\right) \tag{19}$$

$$(S/N)_N = -10\ \log\left(\frac{1}{n}\sum_{m=1}^{n} (a - a_m)^2\right) \tag{20}$$

$$(S/N)_B = -10\ \log\left(\frac{1}{n}\sum_{m=1}^{n} \frac{1}{{a_m}^2}\right), \tag{21}$$

where n is the number of iteration, a_m is the response in m^{th} iteration, and a is the average response. Using the design of experiment method, i.e. the Taguchi provides the following advantages: (i) reducing the number of iterations, (ii) finding the optimal values of the algorithm parameters and, (iii) reducing the runtime taken by the algorithms to find the best solutions. The implementation of the Taguchi method is explained in the numerical examples in the next section.

7. Computational results and discussions

Some numerical examples are solved in this section in order to demonstrate the application of the proposed methodology as well as to assess the performances of the solution algorithms.

7.1. Numerical examples

As a new type of problem has been addressed in this work, there is no benchmark available in the literature. As such, in this section, 40 numerical examples classified in 20 small-size and 20 large-size problems are generated and solved in order to evaluate the performance of the proposed solution methods. Then, the Taguchi method is used to obtain the near-optimal values of the MGA parameters on the 40 numerical examples, for which the L_9 array is used. Table 1 shows the input data used to generate the 40 numerical examples with different sizes where the demands of the buyers follow a normal distribution with mean 20 and standard deviation 10 and the other parameters follow a uniform distribution. From Table 1, the coordinates of both the buyers and the vendors are chosen randomly in a region [0, 100]. Tables 2 and 3 depict the 20 small-size numerical examples and their parameter values along with the best and worst results in terms of their fitness values and their required CPU times obtained by the MGA and PSO, respectively. In small-size numerical examples shown in Tables 2 and 3, the number of buyers is between 2 to 15 while this value is between 1 to 10 for the vendors, for the items, and for the time periods while these numbers in large-size numerical examples are 10 to 25 for the buyers, 10 to 20 for the items, 10 to 20 for the vendors and 2 to 3 for the time periods which are shown in Tables 5 and 6. The sixth to ninth columns of Tables 2, 3, 5 and 6 show the optimal levels of the MGA and PSO parameters tuned by the Taguchi method for each numerical problem, respectively. Since the problem is considered as mixed-integer binary nonlinear programming in order to evaluate the quality of the solutions obtained by the MGA and PSO, the problem is coded in *GAMS version 24.1.2* using *MINLP* function. The fitness values and CPU time of small-size numerical examples obtained by GAMS for all 20 small-size numerical examples are shown in Table 4.

In order to clarify how the Taguchi's method works, Problem number 6 (*Prob. No. 6*) of small-size numerical examples is described for the MGA parameters in detail as an example. Table 7 displays the parameters (factors) of the MGA and PSO and their levels which have been found the best values for the generated problems after running the problems many times with different values of the parameters. The Taguchi approach with an L_9 array of the MGA designed for *Prob. No. 6* of small-size numerical examples is shown in Table 8 where the TC value of each combination is brought in the last column. Figure 4 depicts the mean S/N ratio plot for different levels of the parameters for *Prob. No. 6* of small-size numerical examples for the MGA. According to Fig. 4, the best levels of the MGA parameters are $Pop = 200$, $P_C = 0.6$, $P_m = 0.2$ and $Gen = 1000$. In order to show the difference between the best results obtained by the MGA, PSO, and *GAMS* on small-size numerical examples problem, the pictorial representation of the results for TC and CPU time (*hours*) is demonstrated in Figs 5 and 6, respectively. The convergence path of the best results obtained by the MGA for *Prob. No. 6* of small-size numerical examples is shown in Fig 7. Moreover, the obtained optimal orders of the items made by the buyers from the vendors and the optimal locations of the vendors among the buyers resulted by the MGA and GAMS for *Prob. No.*

6 of small-size numerical examples are displayed in Tables 9 and 10, respectively. Figure 8 shows the graphical representation of the optimal locations of the vendors among the buyers for *Prob. No. 6* of small-size numerical examples. In addition, Tables 11 and 12 depict the one-way ANOVA to compare the MGA and PSO for both small-size and large-size numerical examples in terms of the best fitness values and CPU time respectively.

Insert Figures 4 to 8 here

Insert Tables 1 to 12 here

7.2. Discussions

In this section, the results obtained by the proposed methods are analyzed. Since there is no benchmark fit to the model in the literature, 40 different problems are randomly generated and classified into two categories, small-size and large-size, each with 20 numerical examples. This classification is based on the results achieved by the *GAMS version 24.1.2* software and is based on whether the best fitness value can be reached or not running the problem in 6 days continuously. From Tables 2, 3, and 4, the fitness values obtained by the three solution methods are the same for *Prob. No. 1*. However, while the optimal solution is found by all algorithms, the MGA reaches this value faster than the other methods in terms of CPU time (*sec*). In addition, the fitness value obtained by the PSO for *Prob. No. 2* is optimal and is equal to the one achieved by the *GAMS*. Nonetheless, while the solution found by the MGA is not optimal, this algorithm performs better than PSO and *GAMS* in terms of the CPU time. Meanwhile, in *Prob. Nos. 4* and *6*, the MGA reaches the optimal solution in comparison with *GAMS* while it is still the fastest solution method with the lowest CPU time. Moreover, the results in Table 4 show that *GAMS* is not able to solve *Prob. Nos. 14-15* and *17-20* and thus their optimal fitness values are left unknown. In other words, *GAMS* cannot solve the numerical examples of the problems with the number of buyers more than 8, the number of items more than 5, and the number of vendors more than 4 regardless of running the algorithm problem in 6 days conterminously. In fact, the CPU time taken by *GAMS* to solve the numerical examples increases exponentially with the size of the problems which states that the exact methods such as *GAMS* are not suitable for solving the numerical examples of the problem when the dimension of the problem increases.

Comparing MGA with PSO, the results in Tables 2 and 3 are in favor of MGA in terms of the fitness value, except in *Prob. No. 2* where the PSO found a better fitness value. In addition, both algorithms found identical fitness values for *Prob. Nos. 1* and *7*. Furthermore, MGA is the faster algorithm in all the numerical examples solved.

From Tables 5 and 6, the PSO outperforms the MGA in *Prob. Nos. 5, 10, 15, 17* and *19* in terms of fitness value while both algorithms have the same performance to solve *Prob. Nos. 2, 4, 7, 13* and *18.* Of course, the results of fitness values for the rest of the numerical examples are in favor of MGA. The MGA is still faster than PSO in all 20 numerical examples.

To compare the results obtained by both algorithms statistically, the analysis of variance (a one-way ANOVA) is used. Tables 6 and 7 show the one-way ANOVA derived to compare the MGA and the PSO in terms of the fitness value and CPU time for both small-size and large-size numerical examples. According to the p-values shown in these tables, there is no significant difference between the two algorithms in terms of the fitness value and CPU time.

8. Conclusion

In this work, a novel multi-item multi-period inventory-location allocation problem was formulated for a two-echelon buyer-vendor supply chain problem in which the demands of the buyers were considered to be stochastic following a normal distribution. The distance among the buyers and the vendors were assumed to be Euclidean while the available budget, the production capacity, and the storage space to store the items were limited. The objective was to find out the optimal order quantity demanded by the buyers from the vendors and the optimal locations of the vendors among the buyers so that total supply chain cost was as small as possible. While the model was shown to be a mixed-integer binary nonlinear program, the MGA and PSO were used to solve the proposed problem and to find a near-optimum solution. In order to evaluate the quality of the solutions obtained by the algorithms, some small-size numerical examples of the proposed problem were coded and solved by the *GAMS* software. The results showed that with increasing the dimension of the problem, the CPU time taken to solve the problem rose exponentially. The Taguchi's method was also applied to obtain the best parameters value of the algorithms on 40 generated problems of different sizes. The computational results of running both algorithms indicated that the MGA was the better algorithm in most of the numerical examples in terms of the minimum cost and the faster algorithm to solve all problems.

As for recommendations for future, the model can be extended for a routing problem. In addition, the model can be formulated under shortage, inflation and time value of money. Furthermore, some other meta-heuristic algorithms can be used to solve the problem.

References

Aharon, B.-T., Boaz, G. & Shimrit, S. (2009). Robust multi-echelon multi-period inventory control. European Journal of Operational Research, 199(3), 922-935.
Alikar, N., Mousavi, S.M., Ghazilla, R.A.R., Tavana, M. & Olugu, E.U. (2017a). A bi-objective multi-period series-parallel inventory-redundancy allocation problem with time value of money and inflation considerations. Computers & Industrial Engineering, 104, 51-67.
Alikar, N., Mousavi, S.M., Raja Ghazilla, R.A., Tavana, M. & Olugu, E.U. (2017b). Application of the NSGA-II algorithm to a multi-period inventory-redundancy allocation problem in a series-parallel system. Reliability Engineering & System Safety, 160, 1-10.
Cárdenas-Barrón, L.E., González-Velarde, J.L. & Treviño-Garza, G. (2015). A new approach to solve the multi-product multi-period inventory lot sizing with supplier selection problem. Computers & Operations Research, 64(Supplement C), 225-232.
Correia, I. & Melo, T. (2017). A multi-period facility location problem with modular capacity adjustments and flexible demand fulfillment. Computers & Industrial Engineering, 110(Supplement C), 307-321.
De, S.K. & Sana, S.S. (2014). A multi-periods production–inventory model with capacity constraints for multi-manufacturers – A global optimality in intuitionistic fuzzy environment. Applied Mathematics and Computation, 242(Supplement C), 825-841.
Drezner, Z. & Hamacher, H.W. (2001). Facility location: applications and theory: Springer Science & Business Media, Berlin.
Eiben, A.E. & Smit, S.K. (2011). Parameter tuning for configuring and analyzing evolutionary algorithms. Swarm and Evolutionary Computation, 1(1), 19-31.
Janakiraman, G., Park, S.J., Seshadri, S. & Wu, Q. (2013). New results on the newsvendor model and the multi-period inventory model with backordering. Operations Research Letters, 41(4), 373-376.
Jonrinaldi & Zhang, D.Z. (2013). An integrated production and inventory model for a whole manufacturing supply chain involving reverse logistics with finite horizon period. Omega, 41(3), 598-620.
Liao, Z., Leung, S.Y.S., Du, W. & Guo, Z. (2017). A Me-based rough approximation approach for multi-period and multi-product fashion assortment planning problem with substitution. Expert Systems with Applications, 84, 127-142.
Miranda, P.A. & Garrido, R.A. (2004). Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. Transportation Research Part E: Logistics and Transportation Review, 40(3), 183-207.
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(Supplement C), 80-88.
Mousavi, S.M., Alikar, N., Niaki, S.T.A. & Bahreininejad, A. (2015). Optimizing a location allocation-inventory problem in a two-echelon supply chain network: A modified fruit fly optimization algorithm. Computers & Industrial Engineering, 87, 543-560.
Mousavi, S.M., Alikar, N., Tavana, M. & Di Caprio, D. (2017a). An improved particle swarm optimization model for solving homogeneous discounted series-parallel redundancy allocation problems. Journal of Intelligent Manufacturing. In press, DOI: https://doi.org/10.1007/s10845-017-1311-9.
Mousavi, S.M., Bahreininejad, A., Musa, S.N. & Yusof, F. (2017b). A modified particle swarm optimization for solving the integrated location and inventory control problems in a two-echelon supply chain network. Journal of Intelligent Manufacturing, 28(1), 191-206.
Mousavi, S.M., Hajipour, V., Niaki, S.T.A. & Alikar, N. (2013). Optimizing multi-item multi-period inventory control system with discounted cash flow and inflation: Two calibrated meta-heuristic algorithms. Applied Mathematical Modelling, 37(4), 2241-2256.

Mousavi, S.M., Sadeghi, J., Niaki, S.T.A., Alikar, N., Bahreininejad, A. & Metselaar, H.S.C. (2014). Two parameter-tuned meta-heuristics for a discounted inventory control problem in a fuzzy environment. Information Sciences, 276, 42-62.

Nasiri, G.R., Zolfaghari, R. & Davoudpour, H. (2014). An integrated supply chain production-distribution planning with stochastic demands. Computers & Industrial Engineering, 77(Supplement C), 35-45.

Paksoy, T. & Chang, C.-T. (2010). Revised multi-choice goal programming for multi-period, multi-stage inventory controlled supply chain model with popup stores in Guerrilla marketing. Applied Mathematical Modelling, 34(11), 3586-3598.

Peres, I.T., Repolho, H.M., Martinelli, R. & Monteiro, N.J. (2017). Optimization in inventory-routing problem with planned transshipment: A case study in the retail industry. International Journal of Production Economics, 193, 748-756.

Qiu, R., Sun, M. & Lim, Y.F. (2017). Optimizing (s, S) policies for multi-period inventory models with demand distribution uncertainty: Robust dynamic programing approaches. European Journal of Operational Research, 261(3), 880-892.

Rafie-Majd, Z., Pasandideh, S.H.R. & Naderi, B. (2018). Modelling and solving the integrated inventory-location-routing problem in a multi-period and multi-perishable product supply chain with uncertainty: Lagrangian relaxation algorithm. Computers & chemical engineering, 109(Supplement C), 9-22.

Roy, R. A primer on the Taguchi method, Society of Manufacturing Engineers, Michigan, 1990.

Sadeghi, J., Mousavi, S.M., Niaki, S.T.A. & Sadeghi, S. (2013). Optimizing a multi-vendor multi-retailer vendor managed inventory problem: Two tuned meta-heuristic algorithms. Knowledge-Based Systems, 50, 159-170.

Sepehri, M. (2011). Cost and inventory benefits of cooperation in multi-period and multi-product supply. Scientia Iranica, 18(3), 731-741.

Shankar, R., Bhattacharyya, S. & Choudhary, A. (2018). A decision model for a strategic closed-loop supply chain to reclaim End-of-Life Vehicles. International Journal of Production Economics, 195, 273-286.

Yang, L., Li, H., Campbell, J.F. & Sweeney, D.C. (2017). Integrated multi-period dynamic inventory classification and control. International Journal of Production Economics, 189(Supplement C), 86-96.

The Tables

Table 1. The input data for generating the numerical problems

Parameters	Distribution function
D	$N(20,10)$
F	$U(50000,1000000)$
h	$U(3,20)$
A	$U(5,20)$
c	$U(10,20)$
s	$U(1,10)$
S	$U(1000000,5000000)$
TB	$U(1000000,10000000)$
n	$U(2,6)$
u	$U(0,50)$
a	$U(0,100)$
y	$U(0,100)$
M	$U(0,150)$
μ	$U(20,50)$
σ	$U(10,15)$

Table 2. The general data for different small-size numerical examples along with the fitness function and CPU time of the MGA

Prob. No.	Number of Buyers	Number of Items	Number of Vendors	Number of Time periods	MGA							
					Pop	Pc	Pm	Gen	Fitness		CPU time (Sec)	
									Best	Worst	Best	Worst
1	2	2	1	2	50	0.6	0.2	500	**$1.919e^{4}$**	$2.102e^{4}$	2.75	2.95
2	2	2	2	2	50	0.6	0.2	500	$2.212e^{4}$	$2.745e^{4}$	1.58	1.83
3	3	2	2	2	50	0.7	0.1	500	$3.129e^{4}$	$3.986e^{4}$	1.13	1.45
4	4	3	2	2	50	0.6	0.2	500	**$2.090e^{5}$**	$2.432e^{5}$	7.21	10.49
5	4	4	2	2	100	0.6	0.2	500	$3.033e^{5}$	$3.477e^{5}$	17.61	21.21
6	5	2	2	2	200	0.6	0.2	1000	**$8.793e^{4}$**	$1.139e^{5}$	12.74	13.18
7	5	4	3	3	100	0.6	0.2	500	$6.724e^{5}$	$7.771e^{5}$	22.31	23.11
8	5	5	3	3	200	0.6	0.2	500	$9.146e^{5}$	$1.222e^{6}$	28.56	29.42
9	5	5	4	5	200	0.6	0.2	500	$3.608e^{6}$	$3.714e^{6}$	52.99	55.77
10	8	2	2	2	100	0.6	0.2	500	$2.618e^{5}$	$3.110e^{6}$	23.12	24.905
11	8	3	3	3	100	0.6	0.2	500	$3.393e^{6}$	$2.441e^{6}$	26.39	28.58
12	8	4	4	4	100	0.6	0.2	500	$6.080e^{6}$	$6.218e^{6}$	36.42	38.41
13	8	5	4	4	200	0.6	0.2	500	$7.379e^{6}$	$7.650e^{6}$	83.25	86.42
14	8	5	5	5	200	0.6	0.2	1000	$8.690e^{6}$	$8.803e^{6}$	240.76	248.60
15	8	6	6	6	200	0.6	0.2	1000	$2.397e^{7}$	$2.409e^{7}$	303.81	310.02
16	10	2	2	2	200	0.6	0.2	500	$3.778e^{5}$	$4.175e^{5}$	54.54	56.71
17	10	4	4	4	200	0.6	0.2	500	$7.074e^{6}$	$7.275e^{6}$	69.08	72.14
18	10	8	5	5	200	0.7	0.2	1000	$1.814e^{7}$	$1.826e^{7}$	459.48	464.53
19	10	8	8	8	200	0.7	0.2	1000	$7.856e^{7}$	$7.901e^{7}$	992.94	998.23
20	15	10	10	10	200	0.8	0.1	1000	$2.162e^{8}$	$2.315e^{8}$	1085.16	1098.75

Table 3. The general data for different small-size numerical examples along with the fitness function and CPU time of the PSO

Prob. No.	Number of Buyers	Number of Items	Number of Vendors	Number of Time periods	PSO							
					C_1	C_2	Pop	Gen	Fitness		CPU time (Sec)	
									Best	Worst	Best	Worst
1	2	2	1	2	1.5	2	70	700	**$1.919e^4$**	$2.131e^4$	2.83	3.19
2	2	2	2	2	2	1.5	100	700	**$2.205e^4$**	$2.890e^4$	1.63	1.89
3	3	2	2	2	1.5	2.5	100	700	$3.163e^4$	$4.222e^4$	1.32	1.46
4	4	3	2	2	2	1.5	200	1000	$2.136e^5$	$2.583e^5$	7.51	11.60
5	4	4	2	2	2	1.5	200	1000	$3.209e^5$	$3.524e^5$	18.08	22.43
6	5	2	2	2	2	2.5	200	1200	$8.901e^4$	$1.540e^5$	12.95	13.88
7	5	4	3	3	2	1.5	100	700	$6.724e^5$	$7.997e^5$	24.78	25.91
8	5	5	3	3	2.5	1.5	200	1000	$9.381e^5$	$1.420e^6$	30.24	32.62
9	5	5	4	5	2	1.5	200	700	$3.611e^6$	$3.783e^6$	55.47	57.32
10	8	2	2	2	2	2.5	100	700	$2.685e^5$	$3.222e^6$	23.45	27.20
11	8	3	3	3	2	1.5	200	1000	$3.396e^6$	$2.521e^6$	27.74	31.43
12	8	4	4	4	1.5	1.5	100	1200	$6.200e^6$	$6.821e^6$	37.66	42.23
13	8	5	4	4	2	2	100	700	$7.411e^6$	$8.005e^6$	86.98	91.32
14	8	5	5	5	2	1.5	200	1000	$8.719e^6$	$8.851e^6$	248.28	256.72
15	8	6	6	6	1.5	1.5	200	1200	$2.405e^7$	$2.430e^7$	311.46	319.56
16	10	2	2	2	2	2.5	200	700	$3.796e^5$	$4.272e^5$	57.33	59.21
17	10	4	4	4	2	2	200	1000	$7.144e^6$	$7.327e^6$	72.31	76.84
18	10	8	5	5	2.5	2	100	1200	$1.823e^7$	$1.872e^7$	468.92	476.21
19	10	8	8	8	2	1.5	200	1200	$7.898e^7$	$8.051e^7$	1005.38	1034.28
20	15	10	10	10	2	2.5	200	1000	$2.173e^8$	$2.386e^8$	1104.20	1118.39

Table 4. The general data for different small-size numerical examples and the fitness function and CPU time obtained by GAMS

Prob. No.	Number of Buyers	Number of Items	Number of Vendors	Number of Time periods	GAMS	
					Fitness	CPU time (Sec)
1	2	2	1	2	**$1.919e^4$**	26.31
2	2	2	2	2	**$2.205e^4$**	43.72
3	3	2	2	2	$3.125e^4$	59.26
4	4	3	2	2	**$2.090e^5$**	213.45
5	4	4	2	2	$3.029e^5$	418.75
6	5	2	2	2	**$8.793e^4$**	421.96
7	5	4	3	3	$6.716e^5$	1022.75
8	5	5	3	3	$9.138e^5$	7653.44
9	5	5	4	5	$3.590e^6$	24536.52
10	8	2	2	2	$2.604e^5$	18782.35
11	8	3	3	3	$3.382e^6$	108369.39
12	8	4	4	4	$6.066e^6$	232136.31
13	8	5	4	4	$7.359e^6$	475183.49
14	8	5	5	5	-	-
15	8	6	6	6	-	-
16	10	2	2	2	$3.758e^5$	432540.23
17	10	4	4	4	-	-
18	10	8	5	5	-	-
19	10	8	8	8	-	-
20	15	10	10	10	-	-

Table 5. The general data for different large-size numerical examples along with the fitness function and CPU time of the MGA

Prob. No.	Number of Buyers	Number of Items	Number of Vendors	Number of Time periods	MGA							
					Pop	Pc	Pm	Gen	Fitness		CPU time (Sec)	
									Best	Worst	Best	Worst
1	10	10	10	2	100	0.7	0.2	500	$3.723e^{6}$	$4.130e^{6}$	292.11	354.30
2	10	15	10	2	100	0.7	0.1	500	$4.826e^{6}$	$5.103e^{6}$	558.97	631.21
3	10	15	13	2	100	0.6	0.2	1000	$5.165e^{6}$	$5.596e^{6}$	610.20	676.22
4	10	10	15	2	100	0.7	0.1	500	$5.004e^{6}$	$5.496e^{6}$	588.53	690.13
5	15	10	10	2	100	0.7	0.1	500	$6.171e^{6}$	$6.587e^{6}$	692.98	741.27
6	15	15	10	2	100	0.6	0.2	1200	$7.381e^{6}$	$7.823e^{6}$	802.03	859.08
7	15	10	15	3	200	0.6	0.2	1000	$7.237e^{6}$	$7.892e^{6}$	851.26	932.38
8	15	15	12	3	100	0.7	0.2	1200	$8.401e^{6}$	$8.923e^{6}$	1000.39	1201.36
9	15	15	14	3	200	0.7	0.1	1000	$8.900e^{6}$	$9.310e^{6}$	1031.93	1201.58
10	15	15	15	2	100	0.7	0.1	1000	$9.803e^{6}$	$1.060e^{7}$	1307.38	1443.61
11	10	15	15	3	100	0.7	0.2	1200	$9.119e^{6}$	$9.831e^{6}$	1281.32	1399.02
12	17	15	10	3	100	0.7	0.2	1200	$1.989e^{7}$	$2.197e^{7}$	1532.24	1675.28
13	20	10	10	3	200	0.6	0.2	1000	$1.123e^{7}$	$1.238e^{7}$	1885.25	2100.63
14	20	10	15	2	200	0.6	0.2	1000	$3.230e^{7}$	$3.402e^{7}$	2799.00	3320.45
15	20	15	10	2	100	0.7	0.1	1200	$5.128e^{7}$	$5.652e^{7}$	2895.32	3579.65
16	20	15	15	2	200	0.7	0.2	1200	$8.220e^{7}$	$8.959e^{7}$	4010.25	4950.38
17	20	20	10	2	200	0.7	0.2	1200	$7.456e^{7}$	$7.838e^{7}$	4302.46	5181.33
18	20	20	15	2	200	0.8	0.1	1200	$8.233e^{7}$	$8.881e^{7}$	4831.92	5920.64
19	20	20	20	2	200	0.8	0.2	1200	$9.010e^{7}$	$9.263e^{7}$	5098.52	6672.20
20	25	20	15	2	200	0.7	0.2	1200	$2.230e^{8}$	$2.432e^{8}$	7002.28	9543.37

Table 6. The general data for different large-size numerical examples along with the fitness function and CPU time of the PSO

Prob. No.	Number of Buyers	Number of Items	Number of Vendors	Number of Time periods	PSO							
					C_1	C_2	Pop	Gen	Fitness		CPU time (Sec)	
									Best	Worst	Best	Worst
1	10	10	10	2	2	2	100	1000	$3.811e^6$	$4.231e^6$	312.23	398.23
2	10	15	10	2	1.5	2	70	1200	$4.826e^6$	$5.132e^6$	591.21	603.28
3	10	15	13	2	1.5	1.5	100	1000	$5.273e^6$	$5.641e^6$	645.32	691.65
4	10	10	15	2	2	2	100	1000	$5.004e^6$	$5.412e^6$	627.50	711.35
5	15	10	10	2	2.5	1.5	100	1000	$6.160e^6$	$6.616e^6$	718.22	769.18
6	15	15	10	2	1.5	2	200	1000	$7.409e^6$	$7.900e^6$	822.33	871.53
7	15	10	15	3	2	2	100	1200	$7.237e^6$	$7.856e^6$	903.12	1012.01
8	15	15	12	3	1.5	1.5	200	1000	$8.383e^6$	$8.955e^6$	1032.22	1152.86
9	15	15	14	3	1.5	1.5	200	1000	$8.919e^6$	$9.341e^6$	1142.40	1225.15
10	15	15	15	2	2	1.5	200	700	$9.720e^6$	$1.022e^7$	1343.21	1401.21
11	10	15	15	3	2	1.5	100	1200	$9.122e^6$	$9.815e^6$	1327.11	1410.22
12	17	15	10	3	2	2	100	1000	$2.021e^7$	$2.371e^7$	1622.24	1731.06
13	20	10	10	3	1.5	1.5	200	1200	$1.123e^7$	$1.220e^7$	1956.20	2190.30
14	20	10	15	2	2	2	200	1000	$3.235e^7$	$3.418e^7$	2986.18	3572.71
15	20	15	10	2	1.5	2	100	1200	$5.125e^7$	$5.654e^7$	3012.51	3809.60
16	20	15	15	2	1.5	1.5	200	1000	$8.238e^7$	$8.962e^7$	4200.20	4969.33
17	20	20	10	2	2	2	200	1000	$7.453e^7$	$7.831e^7$	4272.22	5109.33
18	20	20	15	2	1.5	2.5	200	1200	$8.233e^7$	$8.870e^7$	4901.90	5996.38
19	20	20	20	2	2	2	200	1200	$9.003e^7$	$9.251e^7$	5325.02	7030.29
20	25	20	15	2	1.5	2.5	200	1200	$2.235e^8$	$2.400e^8$	7112.20	9918.37

Table 7. The parameters levels of the algorithms

Algorithm	Parameter	Low (1)	Medium (2)	High (3)
MGA	Pop	50	100	200
	Pc	0.5	0.6	0.7
	Pm	0.1	0.15	0.2
	Gen	200	500	1000
PSO	C1	1.5	2	2.5
	C2	1.5	2	2.5
	Pop	70	100	200
	Gen	700	1000	1200

Table 8. The Taguchi array of the MGA for *Prob. No. 6* of small-size numerical examples

Array	Pop	Pc	Pm	Gen	TC
1	1	1	1	1	883241
2	1	2	2	2	881250
3	1	3	3	3	880568
4	2	1	2	3	879985
5	2	2	3	1	881456
6	2	3	1	2	881425
7	3	1	3	2	879985
8	3	2	1	3	880365
9	3	3	2	1	881135

Table 9. The optimal orders made by the buyers from the vendors obtained by the MGA and GAMS for *Prob. No. 6* of small-size numerical examples

	Two types of item produced by Vendor 1 in the first period		Two types of item produced by Vendor 2 in the first period	
Buyer	1	2	1	2
1	100	29	102	93
2	34	74	86	75
3	30	54	100	76
4	65	130	29	94
5	55	25	89	105

Table 10. The optimal location of the vendors among the buyers obtained by the MGA and GAMS for *Prob. No. 6* of small-size numerical examples

Optimal location of Vendor 1		Optimal location of Vendor 2	
y_{11}	y_{12}	y_{21}	y_{22}
8	12	13	26

Table 11. The one-way ANOVA to compare MGA and PSO for small-size numerical examples in terms of the best fitness values and CPU time

	Source	DF	Adj SS	Adj MS	F-Value	P-Value
Fitness value	Factor	1	1.00255E+11	1.00255E+11	0.00	0.995
	Error	38	9.44628E+16	2.48586E+15		
	Total	39	9.44629E+16			
CPU time	Factor	1	147	147	0.00	0.970
	Error	38	3911720	102940		
	Total	39	3911867			

Table 12. The one-way ANOVA to compare MGA and PSO for large-size numerical examples in terms of the best fitness values and CPU time

	Source	DF	Adj SS	Adj MS	F-Value	P-Value
Fitness value	Factor	1	27772900000	27772900000	0.00	0.998
	Error	38	1.08420E+17	2.85316E+15		
	Total	39	1.08420E+17			
CPU time	Factor	1	54701	54701	0.01	0.904
	Error	38	141322528	3719014		
	Total	39	141377228			

The Figures

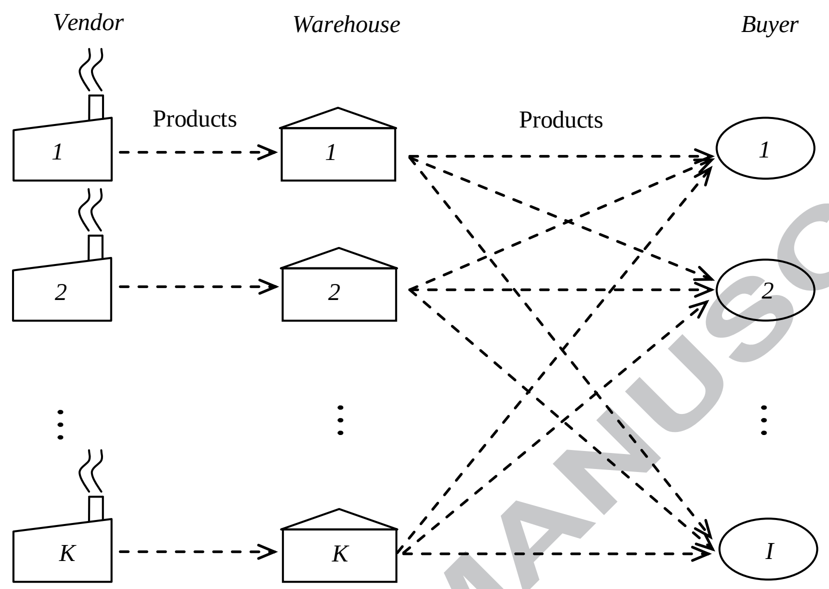


Fig 1. The supply chain system

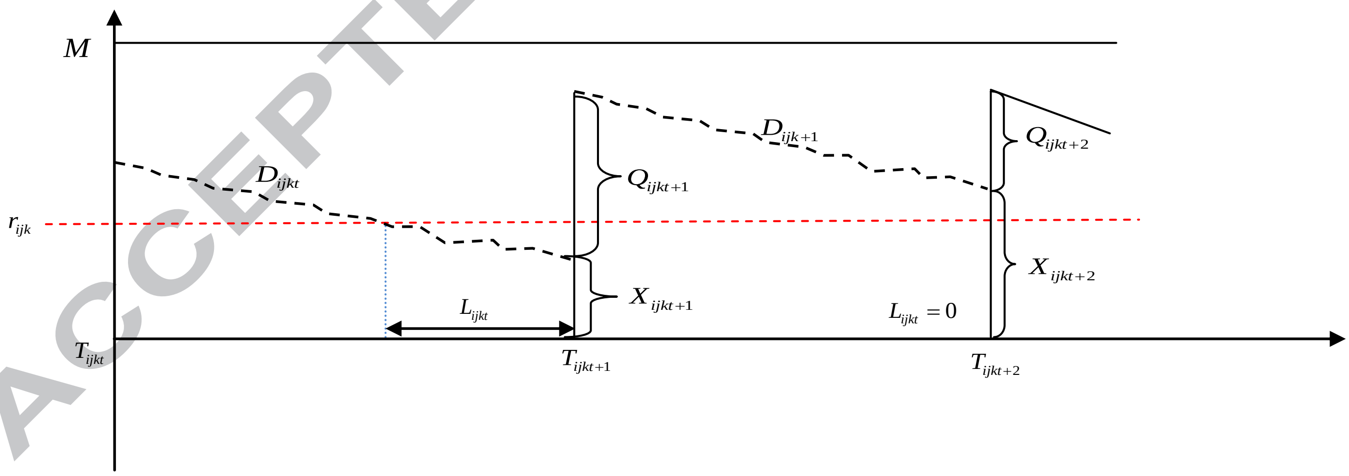


Fig 2. Two different scenarios of net stock vs. time for the inventory model

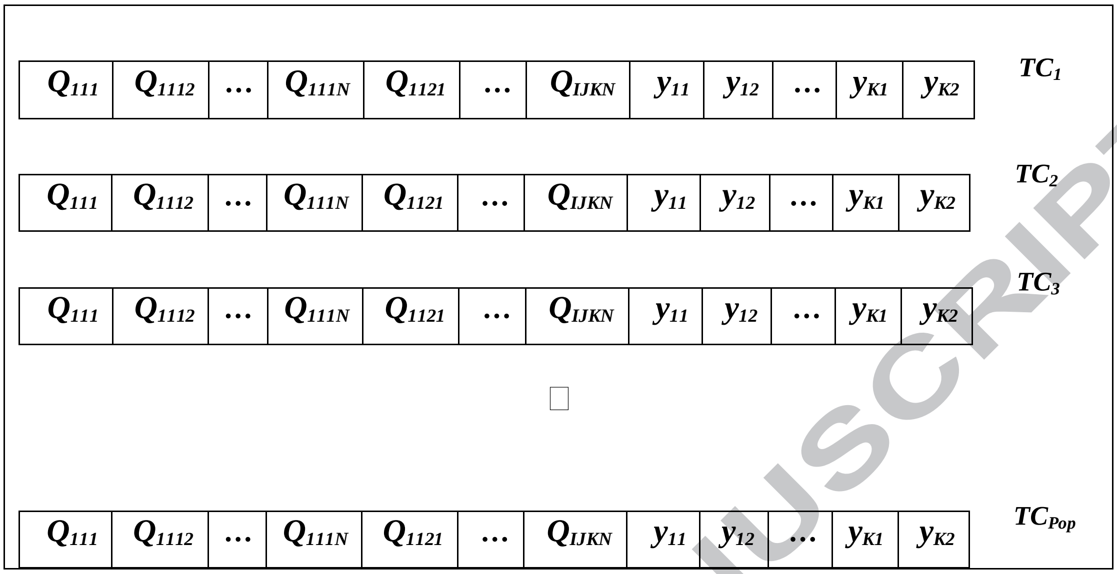


Fig 3. The representation of a chromosome

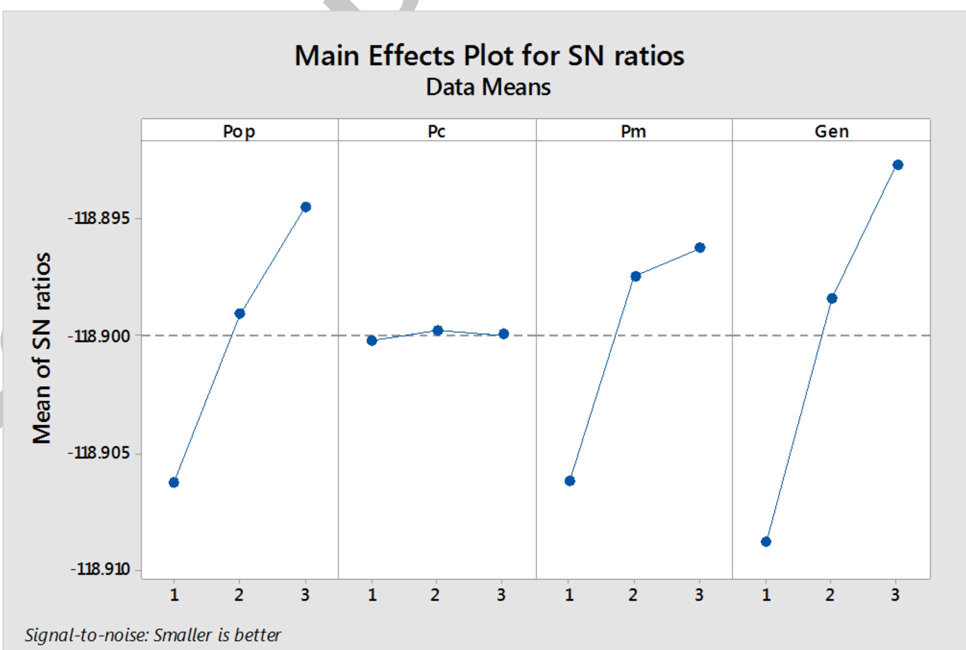


Fig 4. The mean S/N ratio plot for different levels of the parameters for *Prob. No. 6* of small-size numerical examples for the MGA

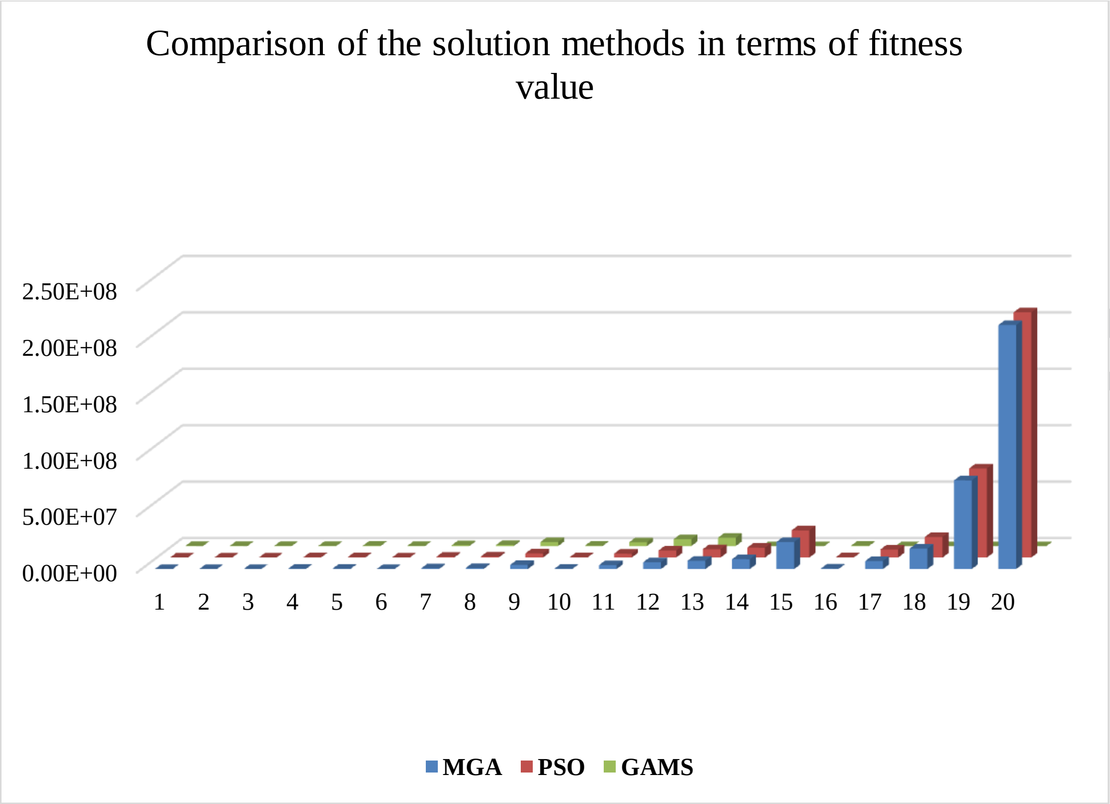


Fig 5. The gap between the best and the worst results of TC obtained by the MGA, PSO and GAMS for small-size numerical examples

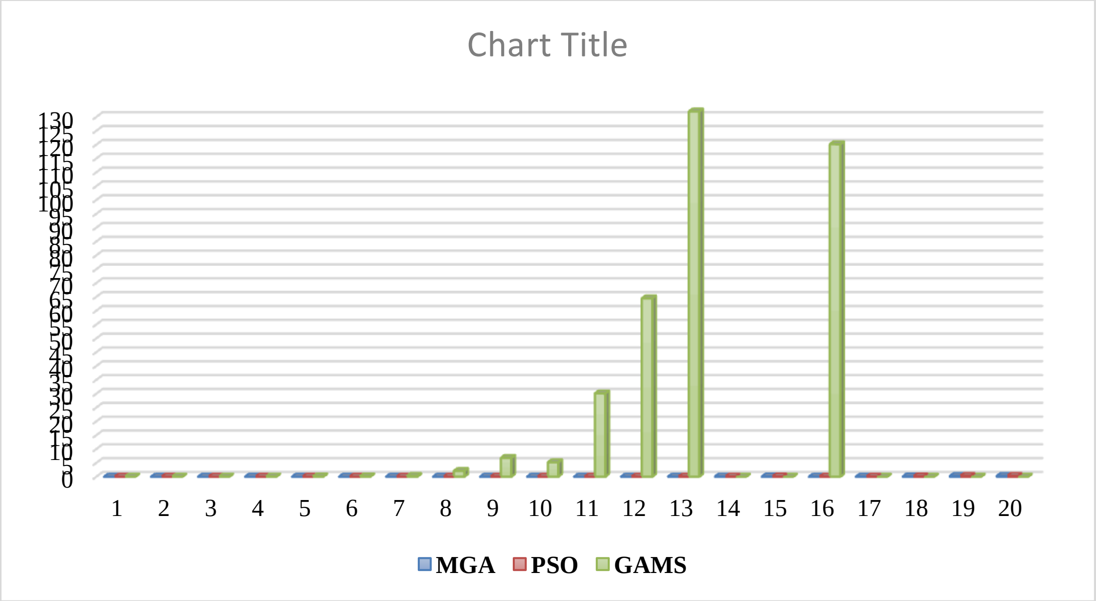


Fig 6. The gap between the best and the worst results of CPU time (*hours*) obtained by the MGA, PSO, and GAMS for small-size numerical examples

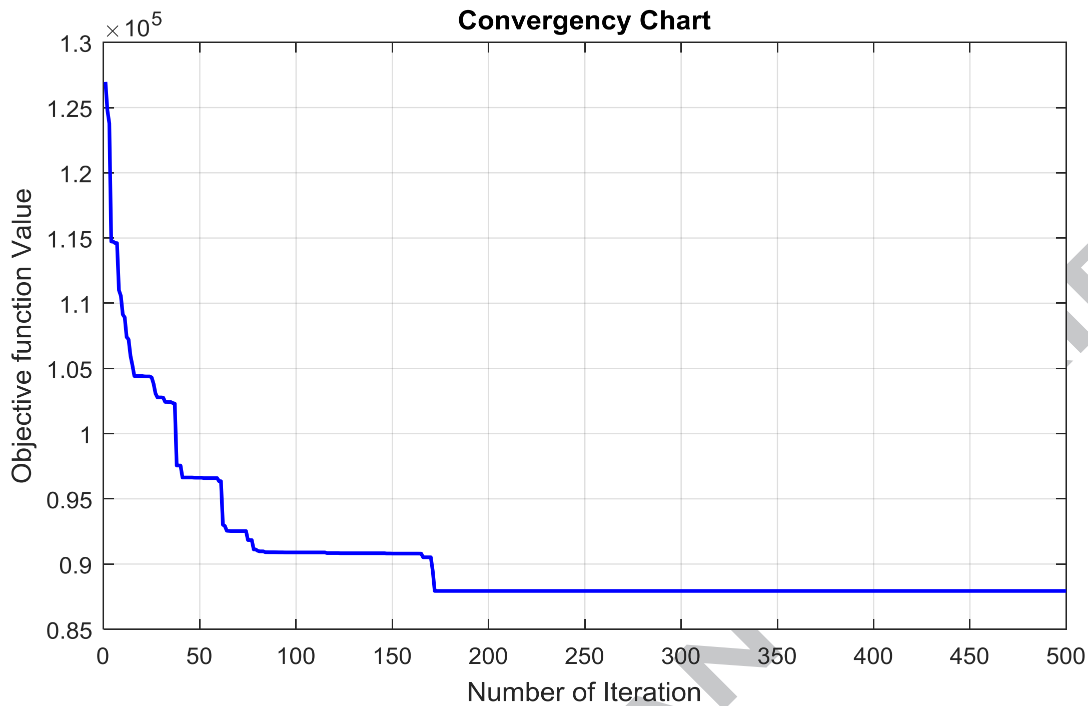


Fig 7. The convergence path of the best results obtained by the MGA for *Prob. No. 6* of small-size data

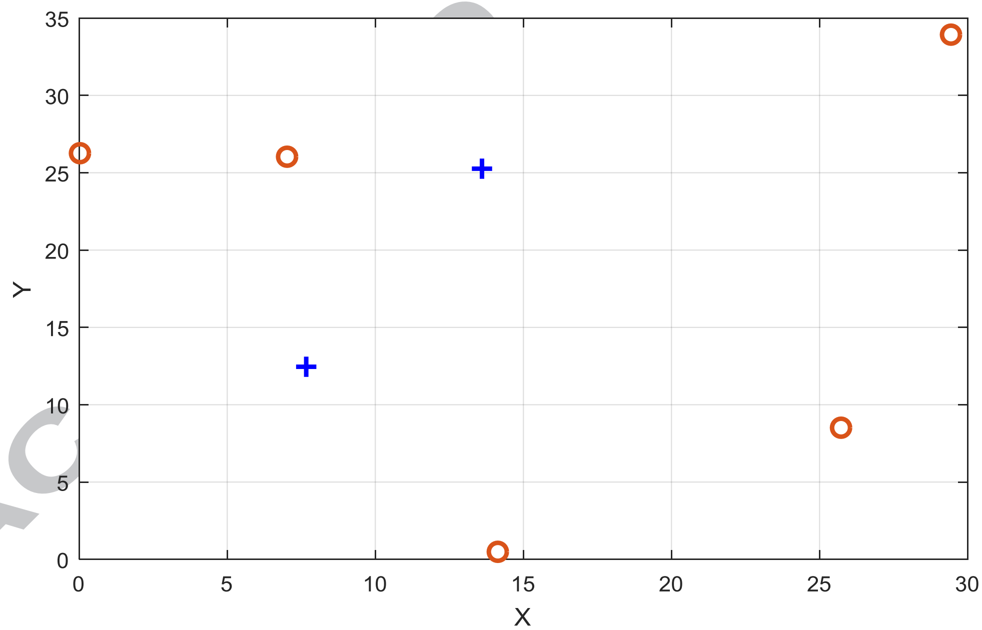


Fig 8. The optimal location of the vendors (blue points) among the buyers (orange circles) obtained by the MGA and GAMS for *Prob. No. 6* of small-size data

Highlights

1. A mixed-integer binary non-linear two-echelon stochastic inventory problem is formulated where the demands of buyers are stochastic.
2. The problem is formulated to be a combination of an (r,Q) and periodic review policies
3. The aim is to find the optimal order quantities and the optimal placement of the vendors such that the costs are minimized.
4. A Genetic Algorithm and Particle Swarm Optimization are used.
5. A design of experiment approach is utilized to adjust the parameters of the algorithms.