Research Note

Modeling and optimization of the multi-objective single-buyer multi-vendor integrated inventory problem with multiple quantity discounts

A. Kamali\textsuperscript{a}, S.M.T. Fatemi Ghami\textsuperscript{a,}\textsuperscript{*} and F. Jolai\textsuperscript{b}

\textsuperscript{a}. Department of Industrial Engineering, Amirkabir University of Technology, 424 Hafez Avenue, Tehran, 1591634\textsuperscript{431}, Iran.
\textsuperscript{b}. Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran.

Received 8 June 2015; received in revised form 4 December 2015; accepted 25 April 2016

KEYWORDS
Integrated inventory; Multi-vendor; Multiple discounts; Multi-objective.

Abstract. This paper deals with a multi-objective integrated inventory model to coordinate a two-stage supply chain including a single buyer and multiple vendors. The earlier work on the problem is limited to considering only one type of discount. This paper extends the problem under the multiple quantity discount environment. We try to minimize the system cost, number of defective items, number of late delivered items and maximize the total purchasing value. Numerical examples are presented to provide some insights into the proposed model and different discount schemes. Results obtained from sensitivity analysis show that changes in unit prices have a relatively large effect on the objective function; as the upper bounds of discount intervals are reduced, the value of objective function decreases. It is also seen that the order quantity from the suppliers increases as the number of suppliers offering all unit quantity discounts increases. In addition, we use a solution approach that is not used by previous studies on this problem, and the obtained results show that the DE algorithm, proposed in this study, outperforms the PSO proposed by Kamali et al. [1] in both solution quality and computational time.

\textcopyright 2017 Sharif University of Technology. All rights reserved.

1. Introduction

With the growing focus on supply chain management, companies are pushed towards a close collaboration with their suppliers and customers. Researchers have studied different features of the supply chain coordination in recent years. Coordination models are presented to improve the overall performance of the supply chain by considering the benefits of all the members of the supply chain. In the case of one buyer and multiple vendors, the goal is to determine one or more vendors and allocate the order quantities to them, such that the efficiency of the entire system, consisting of buyer and vendors, is improved.

This paper considers the issue of coordination between one buyer and multiple potential suppliers in the supplier selection process. Quantity discount policies are used as common incentives from vendors to motivate the buyer to participate in the coordination. In this paper, quantity discounts are used to encourage the coordination. In this way, while the vendors benefit from the integrated model, quantity discounts offered by the vendors can encourage the buyer to participate in the coordination. As an example of this type of supply chain, we can mention the production patterns in steel, chemical, petrochemical, and electronic
industries where the upstream stage, with multiple production lines in parallel, produces a single product and feeds the outputs to the downstream stage [2]. For example, in the steel industry, several lines produce slabs in parallel and feed them to a rolling mill. In these situations, the problem of determining the right suppliers and allocating the required quantity to the selected suppliers arises. Another area of the problem is that our model may be applied to the automotive industry where buyers and vendors engage in strategic partnerships with the intention to create competitive advantages. In this case, using an integrated inventory model may help to coordinate material flows in the system [3].

Regarding the related recent studies to this work, Tabrizi el al. [4] considered a robust discount scheme that can be incorporated into this study as a future research. Giri and Chakraborty [5] considered a multi-supplier multi-buyer problem with stochastic demand and only one objective function maximizing the expected profit of the system. Kundu and Jain [6] developed a smart decision support system providing solutions for power difference issues amongst supply chain partners for a single-product N-supplier system. Aedini and Ventura [7] considered capacity and quantity restrictions on the problem. A supplier selection model in a make-to-order environment considering quantity discount was studied by Guan et al. [8]. They employed a multi-objective approach. In another work, Kermani et al. [9] considered a supplier selection problem with price, quality, and delivery performance criteria, but in a single-echelon supply chain, each manufacturer only tries to optimize his/her position while deteriorating other buyers’ situation. Chen and Sarkar [10] studied an integrated inventory lot-sizing and vehicle routing problem focusing on supplier/manufacturer cooperation in a JIT system with multi-supplier and single-assembly. Kamali et al. [1] developed a multi-objective single-buyer multi-vendor supply chain coordination model which considered the all-unit quantity discount policy for the vendors. They proposed a Particle Swarm Optimization (PSO) algorithm and a Scatter Search (SS) algorithm to simultaneously determine the number of suppliers to employ and the order quantity allocated to these suppliers.

Furthermore, in this paper, we propose a Differential Evolution (DE) algorithm to solve the current NP-hard problem. In this regard, studies of Shafictione and Sadeghi [11], Wang et al. [12], Shahrparvar et al. [13], and Lo [14] are instances of the recent studies employing single and multi-objective differential evolution algorithms for inventory problems in the supply chain. In order to enhance the performance of the proposed algorithms, we apply a local search procedure. Furthermore, we adopt a repair algorithm for constraint handling within the proposed algorithms. The performance of the proposed algorithm is evaluated by comparing its results with the PSO algorithm proposed by Kamali et al. [1]. The obtained results are also compared with the exact optimal solutions to small-sized problems.

The earlier work of Kamali et al. [1] is limited to considering only one type of discount. However, it is obvious that in practical situations, buyers face with multiple-sourcing in which suppliers may offer different price discount schemes [15]. Thus, in this paper, to capture a more practical situation, we extend the model to consider both types of quantity discount at the same time: all-unit discount and incremental discount. Here, we address the issue of coordination of a single-buyer multi-vendor supply chain in an integrated inventory model where each vendor can offer one of the two discount schemes, i.e. all-unit discount and incremental discount, on the unit price of the item. Considering both types of quantity discount and both quantitative and qualitative criteria in the model, it is made to be more practical as compared to that of other previous studies. In addition, we investigate the effect of the number of suppliers offering all-unit quantity discount on the values of the objective functions, providing some insights into the proposed model. Moreover, we use a solution approach to this problem that is not used by previous studies, and the obtained results show that the DE algorithm, proposed in this study, outperforms the PSO proposed by Kamali et al. [1] in both solution quality and computational time.

The paper is organized as follows. In Section 2, the problem is defined, its assumptions are described, and its mathematical modeling is presented. The procedure to solve the problem is discussed in Section 3. In Section 4, a numerical example is presented to evaluate the results obtained from the proposed model. In Section 5, some experimental designs are presented. Finally, Section 6 is dedicated to the summary and conclusion of the paper.

2. Problem definition and formulation

Consider a system in which a single buyer is going to purchase the annual required quantity of a single item or product from the multiple vendors. Buyer’s demand is known with a predetermined annual rate (D); each vendor (indexed by i) is assumed to have a finite production rate (Pi). The vendors are also supposed to have different quality and delivery performances in terms of percentage of defective items and percentage of late delivery items, respectively. In this paper, two types of quantity discounts: all-unit discount and incremental discount, are considered, and each vendor can adopt one of the two quantity discount policies. In this problem, it is assumed that, in each period, after all ith vendor’s order quantities are consumed, (i+1)th vendor’s order quantity can be entered. Figure 1
depicts the inventory levels of the buyer and two vendors for the considered delivery structure. The aim of the model is to determine one or more vendors and allocate the order quantities to them. In the following, four objectives are considered in the modeling of the problem: 1) minimizing the total cost of the supply chain including vendors’ annual cost and buyer’s annual cost; 2) minimizing the total number of defective items; 3) minimizing the total number of late delivery items; and 4) maximizing the total purchasing value.

**Parameters**

- \( i \) Index of vendors
- \( k \) Index of discount intervals
- \( n \) Total number of vendors
- \( K_i \) Index of the last discount interval offered by the \( i \)th vendor
- \( u_i \) Number of vendors offering all-unit quantity discount
- \( u_{ik} \) The upper bound of the \( k \)th discount interval offered by the \( i \)th vendor
- \( u_{ik}^* \) Slightly smaller than \( u_{ik} \)
- \( c_{ik} \) Unit price of the product for the \( k \)th discount interval offered by the \( i \)th vendor
- \( d_i \) Percentage of defective items for the \( i \)th vendor
- \( l_i \) Percentage of late delivery items for the \( i \)th vendor
- \( w_i \) Total performance (weight) of the \( i \)th vendor in qualitative criteria
- \( D \) Annual demand rate of the buyer
- \( A_i \) Fixed/Order cost for the \( i \)th vendor

**Decision variables**

- \( q_{ik} \) Amount of the products ordered to the \( i \)th vendor at the \( k \)th discount interval
- \( y_{ik} \) Binary variable; if the order quantity from the \( i \)th vendor falls at the \( k \)th interval, then \( y_{ik} = 1 \); otherwise, \( y_{ik} = 0 \)

**Dependent variables**

- \( S_i \) The \( i \)th vendor’s setup cost
- \( P_i \) Annual production rate of the \( i \)th vendor
- \( z_i \) Unit production variable cost for the \( i \)th vendor
- \( h_b \) The buyer’s holding cost per unit per unit time
- \( h_i \) The \( i \)th vendor’s holding cost per unit per unit time

\[ Q_i = K_i = \sum_{k=1}^{K_i} q_{ik} \]

\[ Q = \sum_{i=1}^{n} Q_i \]

\[ C_P \] Annual purchasing cost of the buyer

\[ C_P(i) \] Annual production cost of the \( i \)th vendor

\[ C_A \] Annual ordering cost of the buyer

\[ C_S \] Annual setup cost of the \( i \)th vendor

\[ C_H \] Annual holding cost of the buyer

\[ TC(b) \] Total annual cost of the buyer

\[ TC(v) \] Total annual cost of the vendors

\[ TC \] Total cost of the supply chain
2.1. Cost function
The total annual cost of the supply chain, which includes the annual cost of the buyer and the annual cost of the vendors, is calculated as follows.

2.1.1. The buyer’s total cost
The total annual cost of the buyer is:

\[ TC(b) = C_{P_b} + C_{A_b} + C_{H_b}. \]  

(1)

\( C_{P_b} \) depends on the unit price of item, which is offered in the following two discount schemes: all-unit discount and incremental discount. Supposing that \( n_1 \) vendor(s) offer all-unit discount, the number of vendor(s) who offers incremental discount is \( n - n_1 \).

Under all-unit quantity discount policy, the unit cost corresponding to the size of the order is applied to the all-units of order. Hence, \( C_{P_b} \) under all-unit discount policy can be modeled as:

\[ D \frac{n_1}{Q} \sum_{k=1}^{K_i} c_{ik} q_{ik}. \]  

(2)

where \( \frac{D}{Q} \) represents the number of periods in the time horizon, \( c_{i1} > c_{i2} > \ldots > c_{iK_i} \), and at most one of the variables \( q_{ik} \) for \( k = 1, 2, \ldots, K_i \), can be positive and the rest are equal to zero.

Under incremental quantity discount policy, a unit price is applied only to the number of units above the breakpoint. \( C_{P_b} \) under incremental discount policy can be written as:

\[ D \frac{1}{Q} \sum_{i=n_1+1}^{n} \sum_{k=1}^{K_i} \left( c_{ik} (q_{ik} - y_{ik} u_{i,k-1}) + y_{ik} \sum_{j=1}^{k-1} c_{ij} (u_{ij} - u_{i,j-1}) \right), \]  

(3)

where \( u_{i,0} \) is a positive small value for all vendor, \( c_{i1} > c_{i2} > \ldots > c_{iK_i} \), and at most one of the variables \( q_{ik} \) for \( k = 1, 2, \ldots, K_i \), can be positive and the rest are equal to zero. Thus, \( C_{P_b} \) is obtained as:

\[ C_{P_b} = D \frac{1}{Q} \sum_{i=1}^{n_1} \sum_{k=1}^{K_i} c_{ik} q_{ik} + D \frac{1}{Q} \sum_{i=n_1+1}^{n} \sum_{k=1}^{K_i} \left( c_{ik} (q_{ik} - y_{ik} u_{i,k-1}) + y_{ik} \sum_{j=1}^{k-1} c_{ij} (u_{ij} - u_{i,j-1}) \right) \]

\[ = D \frac{1}{Q} \left[ \sum_{i=1}^{n_1} \sum_{k=1}^{K_i} c_{ik} q_{ik} + \sum_{i=n_1+1}^{n} \sum_{k=1}^{K_i} \left( y_{ik} \sum_{j=1}^{k-1} c_{ij} (u_{ij} - u_{i,j-1}) - c_{ik} u_{i,k-1} \right) \right]. \]  

(4)

The annual ordering cost of buyer \( (C_{A_b}) \) is:

\[ C_{A_b} = \frac{D}{Q} \sum_{i=1}^{n} \sum_{k=1}^{K_i} A_i y_{ik}. \]  

(5)

Referring to Figure 1, the average on hand inventory of the buyer for the \( i \)th vendor per unit time is:

\[ I_{bi} = \frac{Q_{bi}^2}{2Q} = \frac{(\sum_{k=1}^{K_i} q_{ik})^2}{2Q}. \]  

(6)

Thus, the annual holding cost of the buyer is obtained as:

\[ C_{H_b} = \frac{h_b}{2Q} \left( \sum_{i=1}^{n} \sum_{k=1}^{K_i} q_{ik} \right)^2. \]  

(7)

Hence, the total annual cost of the buyer is obtained as:

\[ TC(b) = C_{P_b} + C_{A_b} + C_{H_b} = \frac{D}{Q} \sum_{i=1}^{n_1} \sum_{k=1}^{K_i} (c_{ik} q_{ik} + A_i y_{ik}) \]

\[ + \frac{D}{Q} \sum_{i=n_1+1}^{n} \sum_{k=1}^{K_i} q_{ik} \left( \sum_{j=1}^{k-1} c_{ij} (u_{ij} - u_{i,j-1}) - c_{ik} u_{i,k-1} \right) + \frac{h_b}{2Q} \left( \sum_{i=1}^{n} \sum_{k=1}^{K_i} q_{ik} \right)^2. \]  

(8)

2.1.2. The vendors’ total cost
The annual production cost and the annual setup cost of the \( i \)th vendor are obtained using Eqs. (9) and (10), respectively:

\[ C_{P_i} = \frac{D}{Q} \sum_{k=1}^{K_i} z_{qik}. \]  

(9)

\[ C_{S_i} = \frac{D}{Q} \sum_{k=1}^{K_i} S_{ik} y_{ik}. \]  

(10)

According to Figure 1, the average on hand inventory of the \( i \)th vendor per unit time is:

\[ I_i = \frac{D Q_{P_i}^2}{2Q P_i} = \frac{D}{2Q} \left( \sum_{k=1}^{K_i} q_{ik} \right)^2. \]  

(11)

Thus, the annual holding cost of the \( i \)th vendor is obtained as:

\[ C_{H_i} = h_i I_i = \frac{D}{2Q} \times \frac{h_i}{P_i} \left( \sum_{k=1}^{K_i} q_{ik} \right)^2. \]  

(12)
Thus, the total annual cost of all vendors becomes:

\[ TC(v) = \sum_{i=1}^{n} (C_{P_i} + C_{S_i} + C_{H_i}) \]

\[ = \frac{D}{Q} \sum_{i=1}^{n} \sum_{k=1}^{K_i} \left( z_i + c_{ik}q_{ik} + (A_i + S_i)y_{ik} \right) \]

\[ + \frac{D}{2Q} \sum_{i=1}^{n} \sum_{k=1}^{K_i} \left( \frac{h_k}{P_i} \left( \sum_{k=1}^{K_i} q_{ik} \right)^2 \right) \ldots (13) \]

From Eqs. (8) and (13), the total cost of the supply chain, as the first objective function, is calculated as:

\[ Z_1 = TC = TC(b) + TC(v) \]

\[ = \frac{D}{Q} \sum_{i=1}^{n} \sum_{k=1}^{K_i} \left( z_i + c_{ik}q_{ik} + (A_i + S_i)y_{ik} \right) \]

\[ + \frac{D}{Q} \sum_{i=1}^{n} \sum_{k=1}^{K_i} \left( y_{ik} \left( \sum_{j=1}^{K_i-1} c_{ij} (u_{ij} - u_{ij-1}) \right) \right) \]

\[ - c_{ik}u_{i,k-1} \right) + \frac{D}{2Q} \sum_{i=1}^{n} \sum_{k=1}^{K_i} \left( \frac{h_k}{D} + \frac{h_i}{P_i} \right) \left( \sum_{k=1}^{K_i} q_{ik} \right)^2 \ldots (14) \]

2.2. Defective items function

In order to improve the product quality and effectiveness of the supply chain, this objective function is applied which minimizes the total number of defective items in the supply chain as follows:

\[ Z_2 = \frac{D}{Q} \sum_{i=1}^{n} \sum_{k=1}^{K_i} d_{ik}q_{ik} \ldots (15) \]

2.3. Late delivery function

This objective function minimizes the total number of late delivery items in the supply chain and can be stated as follows:

\[ Z_3 = \frac{D}{Q} \sum_{i=1}^{n} \sum_{k=1}^{K_i} l_{ik}q_{ik} \ldots (16) \]

2.4. Total purchasing value function

The Analytical Hierarchy Process (AHP), introduced by Saaty [16], is an effective approach to deal with the multi-criteria problems. By using AHP, the model is enabled to make a trade-off between several tangible and intangible factors with different priorities [17]. The main steps of the algorithm are [17]:

1. Define the criteria for supplier selection;
2. Calculate the weights of the criteria;
3. Rate the alternative suppliers;
4. Compute the overall score of each supplier;
5. Build the linear model and maximize the objective function.

Let \( w_i \) be the total performance of the \( i \)th vendor in qualitative criteria calculated by AHP. Hence, the total purchasing value, which has to be maximized, is obtained as:

\[ Z_4 = \frac{D}{Q} \sum_{i=1}^{n} \sum_{k=1}^{K_i} w_i q_{ik} \ldots (17) \]

2.5. The constraints

The total annual production quantity of all vendors \( \left( D / Q \sum_{i=1}^{n} \sum_{k=1}^{K_i} q_{ik} \right) \) is equal to the annual demand rate of buyer \( (D) \). Thus, we will have:

\[ \sum_{i=1}^{n} \sum_{k=1}^{K_i} q_{ik} = Q \ldots (18) \]

The total quantity ordered to the \( i \)th vendor \( \left( D / Q \sum_{k=1}^{K_i} q_{ik} \right) \) is equal or less than the production rate of this vendor \( (P_i) \). Hence:

\[ \sum_{k=1}^{K_i} q_{ik} \leq \frac{P_i}{D} Q, \quad \forall i; \quad i = 1, 2, \ldots, n. \ldots (19) \]

The discount intervals constraints can be modeled as follows:

\[ q_{ik} \leq u_{ik} y_{kk} q_{kk}, \quad \forall i; \quad i = 1, 2, \ldots, n; \quad \forall k; \quad k = 1, 2, \ldots, K_i \ldots (20) \]

\[ q_{ik} \geq u_{ik,k-1} y_{kk} q_{kk}, \quad \forall i; \quad i = 1, 2, \ldots, n; \quad \forall k; \quad k = 1, 2, \ldots, K_i \ldots (21) \]

\[ \sum_{k=1}^{K_i} y_{kk} \leq 1, \quad \forall i; \quad i = 1, 2, \ldots, n \ldots (22) \]

where \( u_{ik} \) is a positive small value for all vendors.

Finally, to satisfy the buyer’s demand and due to the fact that phrase \( Q \) is the denominator of the fractions in the objective functions, we set \( Q \) greater than a small positive value \( (i.e., \ Q > \varepsilon) \).

2.6. The multi-objective model

The problem can be formulated as a multi-objective mixed integer nonlinear programming model as follows:
Minimize \([Z_1, Z_2, Z_3, - Z_4]\), subject to:

\[
\sum_{i=1}^{n} \sum_{k=1}^{K_i} q_{ik} = Q,
\]

\[
\sum_{k=1}^{K_i} q_{ik} \leq \frac{P_i}{D_i} Q; \quad \forall i; k = 1, 2, \ldots, n,
\]

\[
q_{ik} \leq u_{ik} y_{ik}; \quad \forall i; k = 1, 2, \ldots, K_i,
\]

\[
q_{ik} \geq u_{i,k-1} y_{ik}; \quad \forall i; k = 1, 2, \ldots, K_i,
\]

\[
\sum_{k=1}^{K_i} y_{ik} \leq 1; \quad \forall i; k = 1, 2, \ldots, n,
\]

\[
Q \geq \epsilon.
\]

\[
q_{ik} \geq 0; \quad \forall i; k = 1, 2, \ldots, n,
\]

\[
y_{ik} \in \{0, 1\}; \quad \forall i; k = 1, 2, \ldots, K_i.
\]

One common approach in dealing with multi-objective problems is to minimize the summation of relative deviations of all objectives from their individual optimal values [18]. Suppose that the decision-maker assigns relative weight \(W_i (\forall i = 1, 2, 3, 4)\) to the \(i\)th objective. The weighted objective function as the summation of normalized deviation of each objective from its optimal value is established as:

\[
\text{Minimize } Z = W_1 \times \left( \frac{Z_1 - Z_1^*}{Z_1^*} \right) + W_2 \times \left( \frac{Z_2 - Z_2^*}{Z_2^*} \right)
\]

\[
+ W_3 \times \left( \frac{Z_3 - Z_3^*}{Z_3^*} \right)
\]

\[
+ W_4 \times \left( \frac{Z_4 - Z_4^*}{Z_4^*} \right).
\]

In this formulation, \(Z_i^* (\forall i = 1, 2, 3, 4)\) is the best value of the \(i\)th objective function obtained by optimizing the objective function under the constraints of the problem ignoring other objective functions.

3. The proposed DE algorithm

Burke et al. [19] stated that supplier selection problems under all-unit or incremental quantity discount policies are categorized among the NP-hard problems. There are various evolutionary algorithms to solve the complex problems. Differential Evolution (DE) is from the class of population-based evolutionary algorithms, developed by Storn and Price [20], which is a reliable and versatile method and easy to use. Also, DE has been shown to be efficient for global optimization over continuous real spaces [20]. Recently, DE has been used by Qu et al. [21] for solving a stochastic location-inventory problem with two replenishment policies, joint replenishment, and independent replenishment; results show the effectiveness of the algorithm. Wang et al. [22] developed a simple and improved differential evolution algorithm to solve a procurement approach using the joint replenishment and channel coordination policy in a two-echelon supply chain considering the coordination cost. Also, Cui et al. [23] adopted a differential evolution algorithm to solve a RFID-based investment evaluation model for the joint replenishment and delivery problem under stochastic demand. Hence, we utilize this algorithm to solve the current problem. Furthermore, Kamali et al. [1] showed the good performance of the Particle Swarm Optimization (PSO) in finding the near optimal solutions in a supplier selection and coordination problem. Therefore, this paper adopts the same representation and extends it for the current problem.

In DE algorithm, for each individual in the population, an offspring is generated using the weighted difference of parent solutions. Then, the offspring is replaced with the parent if it is fitter. Otherwise, the parent retains its place in the population in the next generation. These steps are repeated until the stopping criterion is satisfied. (See Storn and Price [20] for more details). In the following, the overall structure of the proposed algorithm is presented.

3.1. Overall structure of the proposed DE algorithm

In the considered problem, solution vectors have the following format [1]:

\[
X = \begin{bmatrix}
Q_1 & Q_2 & \cdots & Q_n \\
q_{11} & q_{12} & \cdots & q_{1K_1} \\
q_{21} & q_{22} & \cdots & q_{2K_2} \\
\vdots & \vdots & \ddots & \vdots \\
q_{n1} & q_{n2} & \cdots & q_{nK_n}
\end{bmatrix}.
\]

We use an approach to generate the initial solutions and a local search algorithm to enhance the performance of the proposed algorithm similar to those implemented by Kamali et al. [1]; readers can refer to this article for more details. Furthermore, during the evolutionary algorithms, solutions may violate the constraints related to the vendors’ annual production rate (Constraint (19)), resulting in infeasible solutions. Hence, we employ a procedure to repair generated infeasible solutions [1].

Algorithm 1 describes the overall structure of the proposed DE algorithm. As it can be seen in this algorithm, after initializing target vector \(X_i\), it is improved by the local search procedure and if the result
for $i = 1$ to population size $m$ do
  initialize the target vector $X_i$
  improve $X_i$ using the local search algorithm;
  if $X_i$ is a feasible solution, then repair it using the repair algorithm;
end for
while maximum iteration or minimum acceptable error is not attained do
  for $i = 1$ to population size $m$ do
    choose integer numbers $r_0$, $r_1$ and $r_2$ randomly in range $[1, n]$ such that $r_0 \neq r_1 \neq r_2 \neq i$;
    $j_{rand} \leftarrow$ an integer random number in range $[1, n]$;
    for $j = 1$ to number of suppliers $n$ do
      if $\frac{r_j}{r_0} \leq C$ or $j = j_{rand}$ then
        $u_{ij} = s_{r_0,j} + F(s_{r_1,j} - s_{r_2,j});$ /* check for out-of-bounds */
      else $u_{ij} = s_{ij};$
    end for
    $U_i$ using the local search algorithm;
    if $U_i$ is an infeasible solution, then repair it using the repair algorithm;
  end for
  // select the next generation //
  for $i = 1$ to population size $m$ do
    if fitness ($U_i$) is better than fitness ($X_i$), then $X_i \leftarrow U_i$;
  end for
end while

Algorithm 1. The proposed DE algorithm.

is an infeasible solution, repair algorithm is applied to transform it to a feasible solution. Furthermore, in the consecutive iterations, the local search procedure is applied to each trial vector $U_i$, and if the result is infeasible, the repair algorithm replaces the infeasible solution with its feasible version.

4. Numerical example

A supply chain system consisting of three suppliers and a single buyer is considered. The buyer’s demand is 100000 units per year, and his (her) inventory holding cost per unit per unit time is equal to 3.24. Discount intervals offered by the suppliers are shown in Table 1, and Table 2 gives some other information of the suppliers.

Table 3 shows the obtained results considering $W_1 = 0.3$, $W_2 = 0.4$, $W_3 = 0.2$, and $W_4 = 0.1$ in different levels of $n_1$. As we can see, the order quantity from the suppliers increases as $n_1$ increases. Table 3 also indicates that in the final solutions, $Z_2$ and $Z_3$ are in their optimal values, and the values of $Z_1$ and $Z_4$ are worse than their optimal values.

We conduct a sensitivity analysis on the main parameters of the model. Table 4 shows percentage of changes in the associated objective function in levels of -25%, -10%, +10%, and +25% changes in the value of each parameter. We can see from the table that changes in unit prices ($c_i$) have relatively large effect.

<table>
<thead>
<tr>
<th>Supplier</th>
<th>Intervals</th>
<th>Unit prices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier 1</td>
<td>0 &lt; $Q$ &lt; 1500</td>
<td>6.1</td>
</tr>
<tr>
<td>Supplier 2</td>
<td>0 &lt; $Q$ &lt; 2000</td>
<td>5</td>
</tr>
<tr>
<td>Supplier 3</td>
<td>0 &lt; $Q$ &lt; 3000</td>
<td>6</td>
</tr>
<tr>
<td>Supplier 1</td>
<td>1500 &lt; $Q$ &lt; 3000</td>
<td>6</td>
</tr>
<tr>
<td>Supplier 2</td>
<td>2000 &lt; $Q$ &lt; 4000</td>
<td>4.9</td>
</tr>
<tr>
<td>Supplier 3</td>
<td>3000 &lt; $Q$ &lt; 4500</td>
<td>4.8</td>
</tr>
<tr>
<td>Supplier 1</td>
<td>3000 &lt; $Q$ &lt; 4500</td>
<td>5.9</td>
</tr>
<tr>
<td>Supplier 2</td>
<td>4000 &lt; $Q$ &lt; 6000</td>
<td>4.8</td>
</tr>
<tr>
<td>Supplier 3</td>
<td>5000 &lt; $Q$ &lt; 6000</td>
<td>5.8</td>
</tr>
<tr>
<td>Supplier 1</td>
<td>4500 &lt; $Q$ &lt; 6000</td>
<td>5.8</td>
</tr>
<tr>
<td>Supplier 2</td>
<td>6000 &lt; $Q$ &lt; 7500</td>
<td>4.7</td>
</tr>
<tr>
<td>Supplier 3</td>
<td>6000 &lt; $Q$ &lt; 7500</td>
<td>5.7</td>
</tr>
<tr>
<td>Supplier 1</td>
<td>6000 &lt; $Q$ &lt; 7500</td>
<td>5.7</td>
</tr>
<tr>
<td>Supplier 2</td>
<td>7500 &lt; $Q$ &lt; 8000</td>
<td>5.6</td>
</tr>
<tr>
<td>Supplier 3</td>
<td>12000 &lt; $Q$ &lt; 15000</td>
<td>5.5</td>
</tr>
</tbody>
</table>

Table 1. Discount intervals of the suppliers.

Table 3. Solutions of the example in different levels of $n_1$.

<table>
<thead>
<tr>
<th>$n_1$</th>
<th>$Z$</th>
<th>$Z_1$</th>
<th>$Z_2$</th>
<th>$Z_3$</th>
<th>$Z_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.038</td>
<td>1012483</td>
<td>28650</td>
<td>25800</td>
<td>915</td>
</tr>
<tr>
<td>1</td>
<td>0.0412</td>
<td>1012483</td>
<td>28650</td>
<td>25800</td>
<td>915</td>
</tr>
<tr>
<td>2</td>
<td>0.0712</td>
<td>993473.3</td>
<td>28650</td>
<td>25800</td>
<td>915</td>
</tr>
<tr>
<td>3</td>
<td>0.0336</td>
<td>978223.1</td>
<td>28650</td>
<td>25800</td>
<td>915</td>
</tr>
</tbody>
</table>

Table 2. Information of the suppliers.

<table>
<thead>
<tr>
<th>Supplier</th>
<th>$z$</th>
<th>$S$</th>
<th>$P$</th>
<th>$A$</th>
<th>$h$</th>
<th>$d$</th>
<th>$l$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.03</td>
<td>33</td>
<td>40000</td>
<td>24</td>
<td>2.19</td>
<td>0.09</td>
<td>0.95</td>
<td>0.46</td>
</tr>
<tr>
<td>2</td>
<td>3.64</td>
<td>35</td>
<td>35000</td>
<td>34</td>
<td>2.36</td>
<td>0.01</td>
<td>0.15</td>
<td>0.31</td>
</tr>
<tr>
<td>3</td>
<td>4.45</td>
<td>50</td>
<td>75000</td>
<td>34</td>
<td>2.85</td>
<td>0.05</td>
<td>0.36</td>
<td>0.23</td>
</tr>
</tbody>
</table>
on the objective function, while changes in the bounds of the discount intervals ($u_i$) have no serious effect. It can also be seen that the objective function is highly sensitive to the values of demand rate of the buyer, production rate of the suppliers, variable costs, defective rates, and late delivery rates. It can also be seen that as the upper bounds of discount intervals are reduced, the value of objective function decreases; the reason is that with a reduction of $u_i$, the buyer can order lower quantities with the same unit price that leads to a reduction in holding cost of buyer, and subsequently the total cost of supply chain decreases.

5. Experimental design

The aim of this section is to evaluate the performance of the proposed DE algorithm. The algorithms are implemented in Visual C#.Net 2008 and are executed on a computer with Intel Core2 Duo 2.66 GHz PC at 4 GB RAM. In this paper, to determine the best value of each objective function for large-scaled problems ($Z_i^*$), all problems are solved by each algorithm using 10 different seeds by considering only one objective individually, and the best obtained value in all runs is used as $Z_i^*$ ($i = 1, 2, 3, 4$) for each objective function. Then, they are replaced in Eq. (24). Furthermore, the Relative Percentage Deviation (RPD) is used as a common performance measure to compare methods and parameters given by the following equation:

$$\text{RPD}(\%) = \frac{\text{Alg}_{sol} - \text{Min}_{sol}}{\text{Min}_{sol}} \times 100,$$

where $\text{Alg}_{sol}$ is the value of $Z$ obtained by an algorithm for a given problem and $\text{Min}_{sol}$ is the best solution known to a given problem. Obviously, lower values of RPD are preferred. The maximum number of iterations is also used as the stopping criterion.

5.1. Parameter tuning

It is clear that performance of a hybrid algorithm is significantly affected by choosing the level of its parameters. This paper applies a full factorial Design Of Experiments (DOE) [24] to analyze the different parameters required by the algorithm. The best levels of parameters are obtained as: Population size = 150, $Cr = 0.4$, and $F = 0.4$.

5.2. Experimental results

This subsection presents some computational experiments to show the performance of the model proposed in Section 3 and verify the efficiency and effectiveness of the proposed algorithm. In order to conduct the experiments, some sample problems with different number of suppliers are generated. In order to evaluate the performance of the proposed DE algorithm, we compare its results with the particle swarm optimization algorithm proposed in Kamali et al. [1]. Each problem is solved by each algorithm using five different seeds for each case, and the average value is considered. Table 5 shows the results of experiments in terms of average RPD on different weights grouped by $n$ and $n_1$.

As can be seen from Table 5, the obtained RPD values for small-sized problems are zero or very small compared to the optimal solutions. Average computational time of the LINGO for problems with 4 and 6 suppliers is 95 and 2254 seconds, while the DE's is 0.6 and 1.5, respectively. The average RPD values of DE and PSO algorithms are 0.538% and 0.776%, respectively. Furthermore, in order to compare the results more precisely, we carry out an ANOVA on the results where the type of algorithm is the single controlled factor. Figure 2 depicts the means plot and LSD intervals for two algorithms at 95% confidence level. As can be seen, there is a significant difference between two algorithms and DE proposed in this paper, providing statistically better results than PSO.

Figure 3 shows the average RPDs of two algorithms for different levels of the number of suppliers. As shown in this figure, for all the problem sizes, the
Table 5. Average relative percentage deviation (RPD%) for two algorithms.

<table>
<thead>
<tr>
<th>Number of suppliers (n)</th>
<th>Algorithm</th>
<th>W₁ = 0.4, W₂ = 0.3, W₃ = 0.2, W₄ = 0.1</th>
<th>Algorithm</th>
<th>W₁ = 0.3, W₂ = 0.2, W₃ = 0.1, W₄ = 0.4</th>
<th>Algorithm</th>
<th>W₁ = 0.1, W₂ = 0.4, W₃ = 0.2, W₄ = 0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DE</td>
<td>PSO</td>
<td>DE</td>
<td>PSO</td>
<td>DE</td>
<td>PSO</td>
</tr>
<tr>
<td>4</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>0.006</td>
<td>0.001</td>
<td>0.001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.003</td>
</tr>
<tr>
<td>3</td>
<td>0.001</td>
<td>0.006</td>
<td>0.001</td>
<td>0.000</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>0</td>
<td>0.002</td>
<td>0.006</td>
<td>0.001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td>0.016</td>
<td>0.040</td>
<td>0.319</td>
<td>0.342</td>
<td>0.106</td>
<td>0.096</td>
</tr>
<tr>
<td>4</td>
<td>0.011</td>
<td>0.021</td>
<td>0.003</td>
<td>0.013</td>
<td>0.089</td>
<td>0.061</td>
</tr>
<tr>
<td>0</td>
<td>0.032</td>
<td>0.033</td>
<td>0.149</td>
<td>0.075</td>
<td>0.107</td>
<td>0.070</td>
</tr>
<tr>
<td>10</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.007</td>
<td>0.017</td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
<td>0.001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.009</td>
<td>0.029</td>
</tr>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
</tr>
<tr>
<td>12</td>
<td>0.909</td>
<td>0.860</td>
<td>0.321</td>
<td>0.470</td>
<td>0.936</td>
<td>1.922</td>
</tr>
<tr>
<td>6</td>
<td>0.562</td>
<td>0.388</td>
<td>0.312</td>
<td>0.361</td>
<td>0.626</td>
<td>1.971</td>
</tr>
<tr>
<td>0</td>
<td>0.362</td>
<td>0.493</td>
<td>0.218</td>
<td>0.232</td>
<td>1.957</td>
<td>3.295</td>
</tr>
<tr>
<td>16</td>
<td>0.527</td>
<td>1.182</td>
<td>0.424</td>
<td>0.476</td>
<td>0.186</td>
<td>0.879</td>
</tr>
<tr>
<td>8</td>
<td>0.688</td>
<td>1.371</td>
<td>0.542</td>
<td>0.488</td>
<td>1.003</td>
<td>1.135</td>
</tr>
<tr>
<td>0</td>
<td>0.412</td>
<td>0.293</td>
<td>0.271</td>
<td>0.391</td>
<td>1.573</td>
<td>1.614</td>
</tr>
<tr>
<td>20</td>
<td>1.423</td>
<td>1.733</td>
<td>0.096</td>
<td>0.175</td>
<td>1.306</td>
<td>2.781</td>
</tr>
<tr>
<td>10</td>
<td>0.819</td>
<td>1.413</td>
<td>0.041</td>
<td>0.110</td>
<td>0.908</td>
<td>2.222</td>
</tr>
<tr>
<td>0</td>
<td>0.978</td>
<td>2.025</td>
<td>0.120</td>
<td>0.215</td>
<td>1.094</td>
<td>1.974</td>
</tr>
<tr>
<td>25</td>
<td>0.431</td>
<td>1.270</td>
<td>1.034</td>
<td>0.589</td>
<td>1.228</td>
<td>1.141</td>
</tr>
<tr>
<td>15</td>
<td>1.400</td>
<td>1.589</td>
<td>0.585</td>
<td>1.173</td>
<td>1.417</td>
<td>1.784</td>
</tr>
<tr>
<td>0</td>
<td>0.982</td>
<td>1.551</td>
<td>1.577</td>
<td>1.223</td>
<td>1.470</td>
<td>1.902</td>
</tr>
<tr>
<td>30</td>
<td>0.536</td>
<td>0.907</td>
<td>0.510</td>
<td>1.517</td>
<td>1.257</td>
<td>2.887</td>
</tr>
<tr>
<td>15</td>
<td>0.760</td>
<td>1.524</td>
<td>0.645</td>
<td>2.000</td>
<td>1.746</td>
<td>2.159</td>
</tr>
<tr>
<td>0</td>
<td>0.851</td>
<td>1.124</td>
<td>0.592</td>
<td>1.347</td>
<td>3.119</td>
<td>2.617</td>
</tr>
<tr>
<td>40</td>
<td>1.448</td>
<td>0.974</td>
<td>0.167</td>
<td>0.817</td>
<td>1.324</td>
<td>1.184</td>
</tr>
<tr>
<td>20</td>
<td>0.652</td>
<td>1.931</td>
<td>0.880</td>
<td>1.538</td>
<td>1.056</td>
<td>1.815</td>
</tr>
<tr>
<td>0</td>
<td>1.606</td>
<td>1.636</td>
<td>0.678</td>
<td>1.306</td>
<td>2.195</td>
<td>3.033</td>
</tr>
<tr>
<td>Average</td>
<td>0.516</td>
<td>0.746</td>
<td>0.343</td>
<td>0.495</td>
<td>0.823</td>
<td>1.196</td>
</tr>
</tbody>
</table>

DE provides better results with lower RPD values. Furthermore, Figure 4 shows the computational time of each algorithm for different levels of the number of suppliers. As can be seen in this figure, the DE algorithm is faster than PSO for all the problem sizes. Also, Figure 5 shows the value of objective function against the number of iterations for 150 iterations for a problem with 30 suppliers for both algorithms. We can see from this figure that the DE algorithm rapidly converges to the near optimal solution after 5~10 iterations, while the PSO algorithm finds a worse solution even after 75~85 iterations.

With attention to the obtained results, we can conclude that the DE is clearly better in performing algorithm in this study. It finds the lower RPDs for all the problem sizes and has a lower computational time.
quantity discount, i.e. all-unit discount and incremental
discount, were considered simultaneously in the
problem. A problem with three suppliers is solved
in different levels of the number of suppliers offering
incremental quantity discount, and the obtained results
were carefully analyzed. The results show that the
order quantity from the suppliers increases as the num-
ber of suppliers offering all-unit quantity discount \( n_1 \)
increases. Also, an increase in \( n_2 \) leads to an improve-
ment in the values of \( Z_2 \) and \( Z_3 \) and a deterioration in
the value of \( Z_4 \). It is also concluded that in all cases,
the summation of deviations of all objective functions
from their optimal values \( (Z) \) increases by imposing
threshold levels on an objective function. In addition,
the impact of changes in the parameters of model
on the obtained results was identified by conducting a
sensitivity analysis. Results show that changes in
unit prices have relatively large effect on the objective
function, while changes in the bounds of the discount
intervals have no serious effect. Moreover, the objective
function is highly sensitive to the values of demand rate
of the buyer, production rate of the suppliers, variable
costs, defective rates, and late delivery rates. It is also
seen that as the upper bounds of discount intervals are
reduced, the value of objective function decreases; the
reason is that with a reduction of the upper bounds of
discount intervals, buyer can order lower quantities
with the same unit price that leads to a reduction in
holding cost of buyer, and subsequently the total cost
of supply chain decreases. Due to the NP-hard nature
of the problem, a differential evolution algorithm was
developed to solve the problem. The performance
of the proposed DE algorithm was compared with
the PSO algorithm. Results showed that there is a
significant difference between the two algorithms in
terms of solution quality and DE proposed in this paper
which provided statistically better results than PSO.

Acknowledgements
The authors would like to thank anonymous referees for
their valuable and helpful comments. The suggestions
have strongly improved the presentation and clarifica-
tion of earlier version of the paper.

References
1. Kamali, A., Fatemi Ghomi, S.M.T. and Johi, F. “A
multi-objective quantity discount and joint optimiza-
tion model for coordination of a single-buyer multi-
vendors supply chain”, Computers and Mathematics
2. Park, S.S., Kim, T. and Hong, Y. “Production alloca-
tion and shipment policies in a multiple-manufacturer-
single-retailer supply chain”, International Journal of


Biographies

Amir Kamaei received his BS degree in Industrial Engineering from Isfahan University of Technology, Isfahan in 2009, and the MS degree in Industrial Engineering from Amirkabir University of Technology, Tehran, in 2011. He is currently a major engineer in steel industry at Neyshabour, Iran. His research interests are quality assurance, management systems, EFQM model, and production planning and control.

Seyyed Mohammad Taghi Fatemih Ghomi was born in Ghom, in Iran, in 1952. He received his BS degree in Industrial Engineering from Sharif University of Technology, Tehran, Iran, in 1973, and a PhD degree
in the same subject from the University of Bradford, England in 1980. He worked as planning and control expert in construction and cement industries, and the Organization of National Industries of Iran from 1980-1983, where he founded the Department of Industrial Training in 1981. He is currently a Professor in the Department of Industrial Engineering at Amir Kabir University of Technology, Tehran, Iran, where he was recognized as one of the best researchers of years 2004 and 2006, and best Professor at the university in 2014. He was also recognized by the Ministry of Science and Technology as one of the best Professors of Iran for 2010. He is author and co-author of more than 350 technical papers and six books in the area of Industrial Engineering and has supervised 133 MSc and 19 PhD theses. His research and teaching interests are in stochastic activity networks, production planning, scheduling, queuing theory, statistical quality control, and time series analysis and forecasting.

Fariborz Jolai is currently a Professor of Industrial Engineering at College of Engineering, University of Tehran, Tehran, Iran. He obtained his PhD degree in Industrial engineering from INPG, Grenoble, France in 1998. He completed his BS and MSc in Industrial Engineering at Amir Kabir University of Technology, Tehran, Iran. His current research interests are scheduling and production planning, supply chain modeling, and optimization problems under uncertainty conditions.
Reproduced with permission of copyright owner. Further reproduction prohibited without permission.