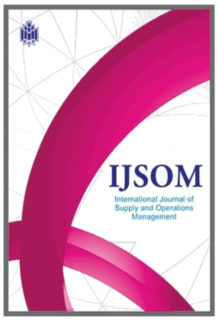
International Journal of Supply and Operations Management

IJSOM

May 2020, Volume 7, Issue 2, pp. 164-177
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com



Sustainable Supplier Selection and Order Allocation Applying Metaheuristic Algorithms

Amir Arabsheybani [a], Mohammad Mahdi Paydar [a,*], and Abdul Sattar Safaei [a]

[a] *Industrial Engineering Department, Faculty of Engineering, Babol Noshirvani University of Technology, Babol, Iran.*

Abstract
Supplier selection, order allocation and production planning are important and challenging decisions in supply chain management. There are many studies on mentioned topics separately. In this paper, a multi-objective mathematical model is proposed to optimize a sustainable supplier selection problem with order allocation and production planning simultaneously. This study considers a multi-supplier, multi-product, multi-item and multi-period supply chain. The designed mathematical model seeks to maximize total profit and minimize unsatisfied demand and total risk along with enforcing sustainability criteria in selecting suppliers. Supplier selection is a virtual process in every manufacturing company. On the other hand, this research considers all the important aspects of this problem. Therefore, the proposed framework can be implemented in many different companies like electronic, food, chemical industry. The proposed model is solved utilizing two metaheuristic algorithms including NSGA II and MOPSO. Moreover, algorithms are tuned utilizing Taguchi analysis. Furthermore, ten sample problems are generated and results are compared to identify the best algorithm for the proposed model.

Keywords: Integrated production-inventory model; Joint Economic lot size; Dual-channel supply chain; E-commerce.

1. Introduction

Supplier selection is one of the complex, sensitive and multifaceted decisions with the most profound effect on efficiency of the supply chain (Sarvestani et al., 2019; Valipour Parkouhi et al., 2019). Reducing production and inventory costs, quality improvement, flexibility and customer satisfaction are some advantages of a good supplier (Arabsheybani et al., 2018). Globalization in various industries has changed the procedures of sourcing to a global process. Hence, political, legal and cultural situation has huge influence on sourcing problem. Recently, supplier selection has been paid special attention and extremely emphasized in both academia and industrial (Parkouhi and Ghadikolaei, 2017). Moreover, intergovernmental panel on climate change (IPCC) figured out that industrial companies are the major reason for environmental pollution, global warming, and resource depletion. Moreover, in its plan for sustainable development, United Nations Environment Programme (UNNEP, 2016) introduced resource efficiency and sustainable production and consumption as a goal for 2030. For this goal, they identified three areas: making an enabling environment, accepting sustainable production and consumption activities across the whole supply chain; the last one defined as consumption pattern in the whole supply chain from supplier to retailer (Gupta and Barua, 2017). Hence, companies have recognized the necessity to reform their business, to attain sustainability. To obtain this commutative advantage, companies must use a new process for manufacturing like improving manufacturing procedure and production design and find a new method for the disposal of waste into the environment without leaving damage (Belin et al., 2009). For selecting sustainable suppliers, (Foroozesh and Tavakoli-Moghadam, 2017) consider a triple-bottom-line approach, including profit, people and planet, business operations, environmental effects and the social responsibilities of the suppliers. They proposed a new hybrid intelligent model, namely COA-LS-SVM.

Corresponding author email address: paydar@nit.ac.ir

(Tirkolaee et al., 2020) applied Fuzzy Analytic Network Process (FANP) to ranking criteria. Also, fuzzy Decision-Making Trial and Evaluation Laboratory (DEMATEL) was applied to evaluate the relationship among the criteria. Finally, they used TOPSIS to prioritize the suppliers. Then, the obtained weights were used as the parameters of the mathematical model. To show the applicability of the approach, a case study of the lamp was solved. (Negash et al., 2020) proposed an approach to obtain product quality using the process yield index. In their study, a nonlinear profile has been employed to characterize the product quality in a sustainable supplier selection problem. To obtain power and the required number of profiles a Monte Carlo simulation has been applied. (Manerba and Perboli, 2019) developed a mathematical model for multi-supplier multi-product procurement problem in the Automotive industry including supplier selection and order allocation decisions, further complicated by the presence of business activation costs, demand uncertainty and total quantity discounts. Although there is ample research in this field, to the best knowledge of authors there is no research with a comprehensive framework to consider suitability and quantity discount simultaneously. Moreover, in this article, metaheuristic algorithm has been implemented to solve supplier selection, order allocation and production planning problem in a reasonable CPU time. In most of the previous studies, supplier selection, order allocation and production planning are treated as distinct problems. Majority of previous studies treated supplier selection and order allocation as multiple objectives. The multi-objective problem was solved using the method for optimizing a single objective which was obtained by reforming multi-objective into single objective via scalarization methods like weighted sum (Türk et al., 2017). However, performances of such methods depend entirely on weights which select by the manager. On the other hand, it is incapability to find multiple trade-off solutions simultaneously after a run (Kim and De Weck, 2006). Accordingly, in this study, supplier selection is performed along with order allocation and production planning, simultaneously. Usually, production planning decisions are made every week or month, so using methods to solve the developed mathematical model in a reasonable time is essential. Additionally, most of the real-world supplier selection problems are very complex and unsolvable via normal computers due to a large number of indices. So, it may lead to the inefficiency of exact solution method (Babaveisi et al., 2017; Fahimnia et al., 2013). Therefore, in this study, two multi-objective metaheuristic algorithms are utilized: NSGA II and MOPSO. Metaheuristic algorithms like as any other stochastic local search method require the tuning of parameters. Tuning could have an important impact on the algorithm's performance. Determining the best parameters for algorithms is a challenging decision in applying efficient metaheuristic algorithms (Deb, 2007). The parameters of NSGA II and MOPSO are calibrated using Taguchi method (Ding et al., 2018). The paper is composed of the following sections. Problem description and designing a mathematical model are presented in section 2. The mathematical model is solved with metaheuristic methods in section 3. Performance analysis of the current study and parameter tuning are discussed with ten instances in section 4. Finally, conclusion and future research are given in section 5.

2. Problem description

The developed supplier selection procedure is presented in this section. The proposed model considers one production plant. This plant buys items from suppliers, and produces final products and sends them to markets. Hence, a comprehensive mathematical model to consider multi-product, multi-item, multi-supplier in a multi-period planning horizon is required. The developed model consists of three objective functions: the first objective maximizes total profit, the second objective minimizes imbalance between demand and lost sale and the last objective minimizes the total risk imposed to supply chain sustainability by purchasing items from suppliers. In the first objective, two common quantity discounts formulated completely in the model and suppliers are able to select pricing policy (all-unit or incremental quantity discount) freely. Figure 1 illustrates the structure of supplier selection problem. In this Figure, parameters which have an influence on each element of the supply chain are shown separately.

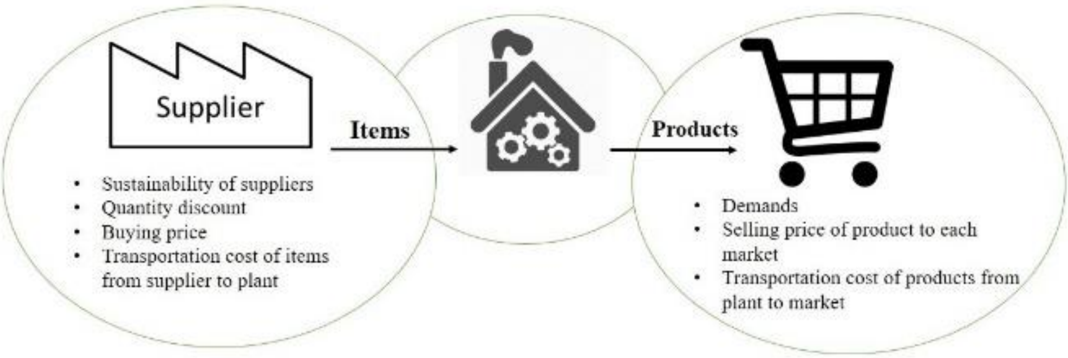


Figure 1. Structure of the proposed mathematical model

2.1. Indices, parameters and decision variables of the model

Indices:

s	Suppliers ($s=1,2,3,\ldots,S$)
i	Item ($i=1,2,\ldots.,I$)
p	Product($p=1,2,\ldots,P$)
m	Market ($m=1,2,\ldots,M$)
k_s	Discount range of supplier s ($k_s=1,2,\ldots.,K_S$)
t	Time period ($t=1,2,3,\ldots,T$)

Parameters:

se_{pt}	Price per unit of product p in period t
c_{pt}	Cost of producing a unit of product p in period t
d_{pmt}	Demand of market m for product p in period t
Cap_t	Plant's production capacity in period t
$supp_{si}$	Capacity of supplier s to produce item i
ti_p	Processing duration of product p
λ_{ip}	BOM of product p
LS_{pm}	Unmet demand cost for product p in market m
fx_{st}	Fixed ordering cost of supplier s in period t
TRI_{si}	Shipping price per unit of item i from supplier s
TRP_{pm}	Shipping price per unit of product p to market m
Nd_{si}	1 if supplier s does not offer any discounts for item i, 0 otherwise
Ad_{si}	1 if supplier s offers all-unit discounts for item i, 0 otherwise;
Id_{si}	1 if supplier s offers incremental discounts for item i, 0 otherwise
pr_{si}	Purchase price per unit of item i from supplier s
$Lowb_{sik_st}$	Lower bound of the discount range k_s of supplier s for item i in period t
Upb_{sik_st}	Upper bound of the discount range k_s of supplier s for item i in period t
AB_{sik_s}	Buying price per unit of supplier s for item i in k_s range
IB_{sik_s}	Buying price per unit of supplier s for item i in k_s range
$risk_{si}$	Risk of supplier s for item i
RX_i	Maximum acceptable risk of item i
M	Big Positive number

Decision Variables:

W_{pt}	Quantity of product p during period t
X_{sit}	Order quantity of item i from supplier s in period t
T_{pmt}	Quantity of product p transport to market m in period t
L_{pmt}	Unmet demand of product p in market m in period t
VF_{st}	1 if an order is placed with supplier s in period t, 0 otherwise
AP_{sik_st}	1 if k_s range selected for supplier s and item i in period t, 0 otherwise (all-unit discount)
IP_{sik_st}	1 if k_s range selected for supplier s and item i in period t, 0 otherwise (incremental discount)

Objective functions:

Total profit

The first objective maximizes total profit as follows:

$Max\ OB_1=$

$$\sum_{p=1}^{P}\sum_{m=1}^{M}\sum_{t=1}^{T} se_{pt}T_{pmt} \tag{1}$$

$$-\sum_{p=1}^{P}\sum_{t=1}^{T} c_{pt}W_{pt} \tag{2}$$

$$-\sum_{s=1}^{S}\sum_{t=1}^{T} fx_{st}VF_{st} \tag{3}$$

$$-\sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{t=1}^{T} TRI_{si}X_{sit}-\sum_{p=1}^{P}\sum_{m=1}^{M}\sum_{t=1}^{T} TRP_{pm}T_{pmt} \tag{4}$$

$$-\sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{t=1}^{T} X_{sit}\, pr_{si}\, Nd_{si} \tag{5}$$

$$-\sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{k_s=1}^{K_S}\sum_{t=1}^{T} AB_{sik_s} X_{sit} AP_{sik_s t}\, Ad_{si} \tag{6}$$

$$-\sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{k_s=1}^{K_s}\sum_{t=1}^{T}\begin{pmatrix} \left(X_{sit} IP_{sik_s t} - Upb_{si\,(k_s-1)t} IP_{si\hat{k}_s t}\right) IB_{sik_s} Id_{si} \\ + \sum_{\hat{k}_s=0}^{K_s-1}\left(Upb_{si\hat{k}_s t} - Lowb_{si\hat{k}_s t}\right) IB_{si\hat{k}_s} Id_{si} IP_{si\hat{k}_s t} \end{pmatrix} \tag{7}$$

The first term calculates the total selling price. Equation (2) shows the production cost. Equation (3) imposes fixed ordering cost to the model. Term (4) calculates total transportation cost from suppliers to the markets. The last three summations of objective function represent buying items. Equation (5) is used when a supplier does not offer any discount. Equation (6) is applied when a supplier offers all-unit discount, and equation (7) is used to calculate total purchase price with an incremental discount.

Lost sale balance

The second objective function considers the lost sale. It makes a balance between lack of product and the amount of demand. In other words , a market with more demand confronts more paucity of product.

$$Min\ OB_2 = \sum_{p=1}^{P}\sum_{t=1}^{T} \underset{m}{Max}\ \frac{d_{pmt} - T_{pmt}}{d_{pmt}} \tag{8}$$

Total risk

This objective seeks to achieve a minimum total risk of suppliers which imposes to sustainability by purchasing items.

$$MinOB_3 = \sum_{s=1}^{S}\sum_{i=1}^{I}\sum_{t=1}^{T} risk_{si} X_{sit} \tag{9}$$

Constraints:

The following constraints are embedded in the model to consider different limitations.

Discounting constraints

Equation (10) expresses that an order with all-unit discounts cannot be allocated if a supplier does not offer it. Furthermore, this constraint guarantees that only one of the discount ranges can be selected.

$$\sum_{k_s=1}^{K_S} AP_{sik_s t} = Ad_{si} \qquad \forall s, i, t \tag{10}$$

Constraints (11), (12) express that order quantity for all-unit discounts must be in the range.

$$\sum_{t=1}^{T} X_{sit} \le Upb_{sik_s t} + M\,(1 - AP_{sik_s t} \times Ad_{si}) \qquad \forall s, i, k_s, t \tag{11}$$

$$\sum_{t=1}^{T} X_{sit} > Lowb_{sik_s t} - M\,(1 - AP_{sik_s t} \times Ad_{si}) \qquad \forall s, i, k_s, t \tag{12}$$

Similar to the all-unit discount, three constraints for the incremental discount are presented as follows:

$$\sum_{k_s=1}^{K_S} IP_{sik_s t} = Id_{si} \qquad \forall s, i, t \tag{13}$$

$$\sum_{t=1}^{T} X_{sit} \le Upb_{sik_s t} + M\,(1 - IP_{sik_s t} \times Id_{si}) \qquad \forall s, i, k_s, t \tag{14}$$

$$\sum_{t=1}^{T} X_{sit} > Lowb_{sik_s t} - M\,(1 - IP_{sik_s t} \times Id_{si}) \qquad \forall s, i, k_s, t \tag{15}$$

Demand constraint

This constraint stipulates that the demand of markets for each product in a period must be equal to satisfying demand and unmet demand.

$$d_{pmt} = T_{pmt} + L_{pmt} \qquad \forall m, p, t \tag{16}$$

Fix ordering cost constraint

This constraint guarantees that fix ordering cost is calculated in the first objective function.

$$\sum_{i=1}^{I} X_{sit} \le M \times VF_{st} \qquad \forall s, t \tag{17}$$

Capacity constraints

Capacity constraints guarantee that the total production cannot exceed the maximum time in each period (18). Constraint (19) is the maximum capacity of suppliers.

$$\sum_{p=1}^{P} ti_p \times W_{pt} \le cap_t \qquad \forall t \qquad (18)$$

$$X_{sit} \le supp_{si} \qquad \forall s,i,t \qquad (19)$$

Inventory constraints

The following constraints are designed to make a balance between produced items, the quantity of purchased items and sent items.

$$W_{pt} = \sum_{m=1}^{M} T_{pmt} \qquad \forall p,t \qquad (20)$$

$$\sum_{s=1}^{S} X_{sit} = \sum_{i=1}^{I}\sum_{p=1}^{P} \lambda_{ip} W_{pt} \qquad \forall i,t \qquad (21)$$

Maximum risk constraint

Sustainability risk imposed on the system for each item must be less or equal to the maximum risk determined by experts.

$$\sum_{s=1}^{S} risk_{si} X_{sit} \le RX_i \qquad \forall i,t \qquad (22)$$

Variable domain

$$\begin{aligned} &W_{pt}, X_{sit}, T_{pmt}, L_{pmt} \ge 0 and \in Integer \\ &VF_{st}, AP_{sik_s t}, IP_{sik_s t} \in \{0,1\} \end{aligned} \qquad (23)$$

3. Solution methodology

There are different solution methods for solving mathematical models. For simple and small problems, routine methods can be adopted. However, for large-scale problems, it may be inefficient. Hence, using evolutionary techniques for optimization is useful. Since the model is involved with three conflicting objective functions in the current study, two multi-objective metaheuristic algorithms are utilized. Non-dominated sorting genetic algorithm (NSGA II) and multi-objective particle swarm optimization (MOPSO) are used for this purpose. The mentioned algorithms are population-based and request for random population generation. Therefore, the procedure for population generating will be initially described. The populations will be generated according to the chromosomes in both algorithms. Chromosomes employ priority-based method for generating population. The priority-based method is developed by Gen & Cheng in 2000 (Gen & Cheng, 2000) and has two main steps: chromosome preparing or chromosome encoding and chromosome decoding.

3.1. Chromosome encoding

One of the most important parts of metaheuristic algorithms is showing the feasible solution in a form of string characters. This string of characters is called chromosome and each member of the chromosome is a Gen. A chromosome can be encoded by using three different methods of encoding based on edge, based on vertex and based on edge-vertex. In this article, encoding was performed based on edge-vertex. Encoding based on edge-vertex is a general form of source and depot problem. Consider $|S|$ and $|W|$ as the number of source and depot, respectively. The length of the chromosome will be $|W|+|S|$ and number of genes will be random numbers from 1 to $|W|+|S|$.

According to the priority-based method, we need to design a chromosome for each stage of the supply chain. As the current model considers two stages of the supply chain, it is necessary to design two types of chromosomes. The first stage is between markets and plant and the second stage is among plants and suppliers. In both stages, we should consider one element as a source and the other one as depot, and then transportation problem should be solved to obtain the flow of items and products. The representation for the chromosome of each stage is shown in Figure2.

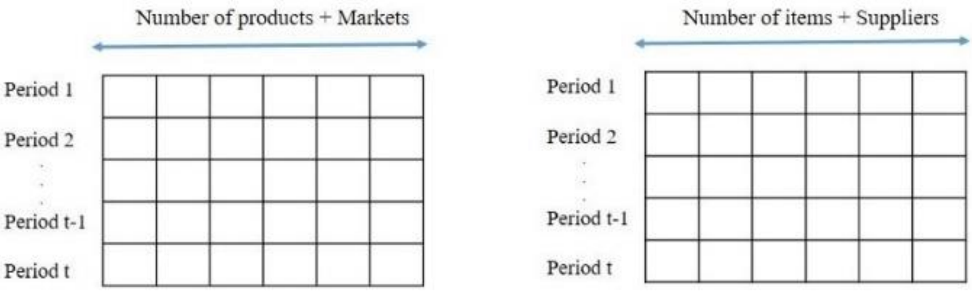


Figure 2. Chromosome représentation of Stage 1 and 2

3.2. Chromosome decoding

Decoding algorithm of the priority-based method should be carried out for all stages of the supply chain. As mentioned before, the ransom numbers and the gene number are the same. These random numbers are used in decoding. Gen with the higher number will be select first, then according to the place of the gen, it will be a source (or depot). By referring to transportation costs matrix, depot (source) with minimum transportation cost will be selected. After selecting source and depot with minimum transportation cost, material's flow must be allocated which is equal to the minimum capacity of source and depot. Then, capacity of source and depot should be updated by subtracting material's flows from an initial capacity. Furthermore, gen with the high number should be changed to zero because allocation for that gen is already performed. This process continues to reach a stop condition. It should be noticed that stages in priority-based method start from a final node of the supply chain (Demands) and continue to suppliers. A simple form of source and depot problem is shown in Figure 3. Furthermore, pseudo-code for decoding stage 1 is available in Figure 4. Similar to the first stage, decoding applies in the second stage.

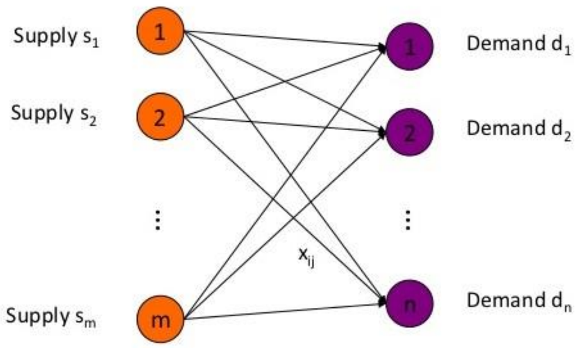


Figure 3. Source and depot problem

Procedure of priority based decoding for the first stage

Inputs:	*S: sets of supplier, I: sets of items, P: sets of product, M: sets of market, T: sets of period* *TRP_{pm}: Shipping price per unit of product* p *to market* m *D_{pmt}: Demand of market m for product p in period t* *Cap_t: Production capacity of period t* *Ti_P: processing time of product p* *$landa_{ip}$: BOM of product p* *$Ch(t\times(p+m))$: chromosome, $\forall p \in P, m \in M$*
Outputs:	*T_{pmt}: Quantity of product* p *transport to market* m *in period* t *W_{pt}: Production quantity of product p for period t* *ITEM_NEED: items required for the second stage*
Procedure:	*For t=1 to T* *While all(ch(:))>0 or sum(sum(D_{pmt}(:,:,t)))>0* *Maximum arguman {ch(u) $u \in p+m$} $\longrightarrow$ va* *If va<p* *V^*=va, a producet is selected* *M^*=minimum arguman {TRP_{pm^*} | ch(t,m) $\neq 0$, $p \in P$}, selecting a market with minumum transporation cost* *If D_{pmt}* $Ti_P < Cap_t$* *$T_{pmt} = D_{pmt}$* *Else D_{pmt}* $Ti_P > Cap_t$*

T_{pmt}= *Round toward negative infinity*(Cap_t/ Ti_P)
End
$D_{pmt}= D_{pmt} - T_{pmt}$
$Cap_t= Cap_t - D_{pmt} * Ti_P$
Else
V^*=*va, a market is selected*
P^*=*minimum arguman* {TRP_{p^*m} | $ch(t,v) \neq 0$, $m \in M$}, *selecting a product with minumum transporation cost*
If $D_{pmt} * ti_P < Cap_t$
$T_{pmt}= D_{pmt}$
Else $D_{pmt} * Ti_P > Cap_t$
T_{pmt}= *Round toward negative infinity*(Cap_t/ Ti_P)
End
$D_{pmt}= D_{pmt} - T_{pmt}$
$Cap_t= Cap_t - D_{pmt} * Ti_P$
End
For p and t
$W_{pt}=sum(T_{pmt}(p,:,t))$
End
For t, i, p
$Item\_need\_pit=landa_{ip} * W_{pt}$
End
For i,t
ITEM_NEED=sum(Item_need_pit(t,:,i))
End

Figure 4. Pseudocode for the first stage decoding

3.3. NSGA II

NSGA II is a multi-objective form of the genetic algorithm (GA). Mechanism of the NSGA II is based on the GA. According to NSGA II, populations are ranked using non-dominated sorting procedure. The result of this ranking is a set of solutions where none is better than the others. This non-dominated solution is called Pareto front. This algorithm generates a new population by using operators. The operators of NSGA II algorithm are crossover and mutation. This algorithm is explained completely in (Deb et al., 2002).

Crossover and mutation operators

There are different types of simple crossovers operators like one or two-point crossover. However, using these traditional and simple code operators in priority-based method leads to generating infeasible solutions. Hence, in the current study order crossover (OX) was implemented. Mechanism of OX will be explained according to Figure 5. First of all, two random cut on the length and weight of both parents should be performed. Chromosome A must be prepared by putting the yellow part in a zero chromosome. Then, chromosome B should be obtained by changing to zero arrays of the second parent which is equal to yellow arrays of the first parent. Finally, to generate the first child, for each row, the nonzero number in chromosome B should be put on zero number of chromosome A, respectively. The second child generation is almost the same as the first child, but different is that the yellow part of the chromosome should be selected from the second parent. For mutation operations, two positions change randomly for all the rows (periods). For mutation operation, as illustrated in Figure 6 two positions change randomly for all the rows (periods).

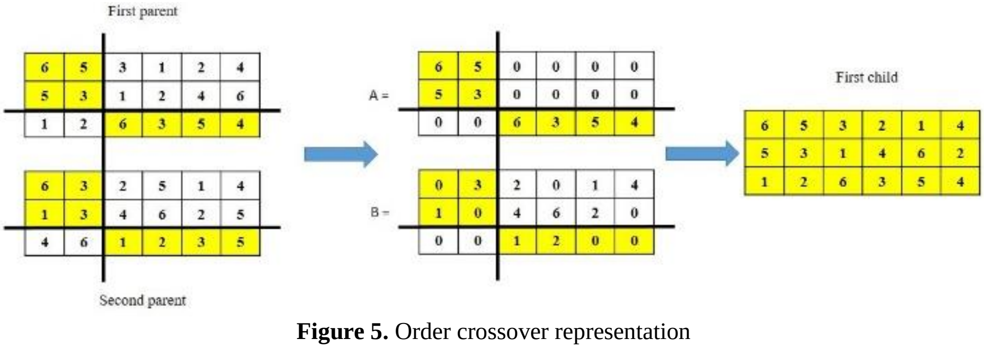


Figure 5. Order crossover representation

Period 1	6	5	3	1	2	4
Period 2	5	3	1	2	4	6
Period 3	1	2	6	3	5	4

Figure 6. Mutation representation

3.4. MOPSO

Particle swarm optimization (PSO) is introduced by (Eberhart & Kennedy, 1995) originally taken from fish and bird movements. This algorithm searches solution space by particles updated in each iteration. MOPSO is a multi-objective form of PSO. According to this algorithm, each particle has two attributes: position and velocity. After initializing a swarm of particle randomly they will be modified based on the best local (Pbest) and best global experience (Gbest). Pbest is the best experience of each particle during all iterations and Gbest is the best position among all of the particles. According to the amount of Pbest and Gbest, the velocity is calculated for each particle. We will explain updating the mechanism of velocity in the next section.

3.4.1 Velocity update

Routine velocity formula cannot apply for velocity updating on priority-based method. In equation (24), c1 and c2 are the local and global learning coefficients.

$$\begin{aligned} v_{ij}^{t} &= v_{ij}^{t-1} + c_1 r_1 \left(p_{ij}^{t-1} - x_{ij}^{t-1}\right) + c_2 r_2 \left(G_j^{t-1} - x_{ij}^{t-1}\right) \\ x_{ij}^{t} &= x_{ij}^{t-1} + v_{ij}^{t} \end{aligned} \tag{24}$$

In the current research, a specific procedure for velocity updating is used to keep priority value. According to this procedure, the theory of vector similarity should be defined. The theory of vector says that the solution must be considered as a vector. The vector resemblance-degree between the current and personal best position (pi) and the current and global best position (g). Hence, the current vectors (xi) must be changed to obtain velocity (Babaveisi et al., 2017). A vector has two indicators: direction and magnitude. The magnitude of a vector is calculated by equation (25).

$$\| x_i \| = \sqrt{\sum_{j=1}^{k} x_{ij}^{2}} = \sqrt{\sum_{j=1}^{k} j^{2}}, i = 1,2,...,k \tag{25}$$

The vector resemblance degree is calculated using equation (26) (cosine of xi and Pbest or Gbest).

$$\begin{aligned} m(p_i, x_i) &= \frac{\sum_{j=1}^{k} x_{ij} * p_{ij}}{\sqrt{\sum_{j=1}^{k} x_{ij}^{2} * \sum_{j=1}^{k} p_{ij}^{2}}} \\ m(g, x_i) &= \frac{\sum_{j=1}^{k} x_{ij} * g_j}{\sqrt{\sum_{j=1}^{k} x_{ij}^{2} * \sum_{j=1}^{k} g_j^{2}}} \end{aligned} \tag{26}$$

Assume *r* as a random number within [0,1] and *w* as an inertia weight. Equation (27) is used to determine the requested changes in dimensions.

$$s_i^{i+1} = (r.w)s_i^t + (1 - r.w)\left(\frac{1 - m(p_i, x_i^t)}{2} + \frac{1 - m(g, x_i^t)}{2}\right) \tag{27}$$

Finally, new velocity can be obtained by the following equation:

$$v_i^{t+1} = w * v_i^t + c_1 r(p_i - x_i^t) + c_2 r(g - x_i^t) \tag{28}$$

3.4.2 Position update

In MOPSO algorithm, positions are updated considering velocity. Indeed, the solutions are created with a permutation of integer numbers. Since positions are integer numbers, velocity should be an integer value. Updating the process for permutations has two steps as follows:

First step: Rounding number of dimensions and velocity.

$$v_{ik}^{t+1} = \text{int}(v_{ik}^{t+1})$$
$$s_{ik}^{t+1} = \text{int}(s_{ik}^{t+1}) \quad (29)$$

Second step: Obtaining the new positions of a vector using equation (30).
This formula generates value within [1,n]. Thus, in the permutation, when a value changes from x1 to x2, another change must be applied to keep permutation exclusive. This change is switching x2 by x1.

$$x_{ik}^{t+1} = \begin{cases} \dfrac{x_{ik}^{t} + v_{ik}^{t+1}}{n}, & v_{ik}^{t+1} \geq 0 \\ \dfrac{x_{ik}^{t} + v_{ik}^{t+1}}{n} + n, & v_{ik}^{t+1} < 0 \end{cases} \quad (30)$$

For a better understanding of this process, graphical representation of position updating is illustrated in Figure 7.

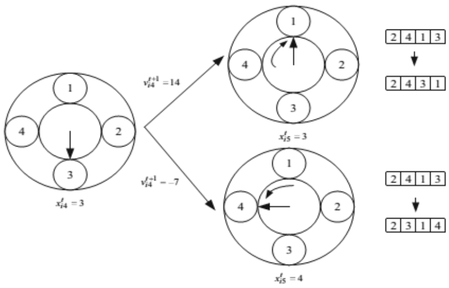


Figure 7. Procedure for position update (Peng and Wei, 2008)

3.5. Metaheuristic performance measurement and parameter tuning

In a multi-objective problem, instead of one single solution, there are sets of non-dominated solutions. Therefore, comparing objective functions directly is not possible. Due to this issue, performance measurement methods for evaluating metaheuristic algorithms should be used. There are various performance measurements explained completely in (Babaveisi et al., 2017).
Performance of an algorithm greatly depends on its parameters. Therefore, if these parameters are compatible with the size of the problem, the algorithm can report better solutions. Moreover, parameter tuning can reduce the number of experiments and enhanced solution time (Sarrafha et al., 2015).
Each metaheuristic algorithm uses different parameters to achieve solutions. For instance, NSGA II uses mutation and crossover rate, number of populations and iterations. There are different approaches for parameters tuning such as full factorial experiment, trial and error, design of experiments (DOE), response level and neural network. In the current study, Taguchi method is utilized. Preparing a set of experiments which make meaningful change in variables is the purpose of the Taguchi. The response variable in this case is obtained by the integration of MID, SNS and CPU time. For simplicity, orthogonal arrays are designed that provides number of necessary experiments by considering the number of levels and factors.
For analyzing Taguchi results, two methods are recommended: using analysis of variance (ANOVA), and signal to noise ratio (S/N). The higher S/N shows a lower variance around a specific amount (Roy, 2010).

4. Experiments and Results

In this section, a validation has been prepared to ensure that encoding and decoding are correct. Furthermore, ten instances are generated to select the best algorithm (NSGA II or MOPSO) for the current model. Finally, Taguchi method and sensitivity analysis are presented.

4.1. Metaheuristic validation

Since the multi-objective result is a Pareto front and it is not comparable so we generate four different instances with different scales according to Table 1 to validate and compare CPU time. The first objective function of the model solved singularity with GA, PSO and LINGO (to obtain the global optimum solution). The results are illustrated in Table 2. As shown, the metaheuristic algorithms have a gap between 4 to 12 percent that is reasonable for a supply chain problem.

Table 1. Dimension of the instances

Index Instance	size	S	I	P	M	T
1	Small problem	2	4	2	1	1
2		4	5	3	2	2
3		6	5	3	3	3
4		8	6	4	4	4
5	Medium problem	12	7	5	6	5
6		15	10	7	8	6
7		17	12	8	9	7
8	Large problem	18	13	9	10	8
9		19	14	10	11	9
10		21	20	12	13	10

Table 2. Metaheuristic performance compared with the exact method

Instance number	Total profit by exact method (LINGO)	Total profit by PSO	Total profit by GA	PSO deviation from exact	GA deviation from exact
1	100074100	95356201	93004523	4%	7%
2	1066376000	1001376000	996325890	6%	6%
3	2396712000	2300745250	2100769100	4%	12%
4	6825920000	629831840	6272250980	7%	8%

In order to make sure the reliability of the result of the large instance , ANOVA test can be applied (Najjartabar et al., 2016). For briefly, ANOVA results for this case has been omitted from the paper.

4.2.Performance of metaheuristic algorithms

In this section, 10 instances generated in Table 1 should be solved to identify the best algorithm for the current mathematical model. Algorithm's parameters and their levels are prepared in Table 3.

The NSGA II algorithm has four parameters and three levels. Thus, appropriate Taguchi orthogonal array is L9 and it is L27 for MOPSO. According to orthogonal arrays and the number of instance, 270 runs for MOPSO and 90 runs for NSGA II are performed by a desktop computer with Core i7 3.4GHz and 8GB Ram. Therefore, totally 360 runs are done. The results are normalized using Equation (31) and shown in Table 4.

Since the MID and SNS do not have the same importance, Equation (32) has been used. The weights are 0.6 and 0.4 for MID and SNS, respectively. The normalized weight score and CPU time are shown in Table 5. The last row of this table is the average amount of all instances. Table 6 is prepared based on these averages for comparing the performance of algorithms. It is clear that MOPSO has better performance on CPU time; however, in the weighted scores *w* (MID and SNS) NSGA II has better performance. To determine the best algorithm a multi-criteria decision-making method of TOPSIS is used (Triantaphyllou, 2000). TOPSIS prepare a rank of alternative by considering different criteria. To implement this method, a decision matrix should be prepared. Table 6 is a decision matrix for this problem.

$$NORMAL = \frac{|X - Y|}{Y} \tag{31}$$

$$Score = \sum_{i=1}^{I} W_i V_i \tag{32}$$

Table 3. Algorithm parameters and their level

	Parameters	Level 1	Level 2	Level 2
NSGA II	**Population**	150	250	500
	Rate of crossover	0.7	0.8	0.93
	Rate of mutation	0.2	0.3	0.4
	Iteration	100	300	500
MOPSO	**Local learning coefficient**	0.7	0.83	0.93
	Global learning coefficient	0.95	0.85	0.75
	Inertia weight	0.2	0.5	0.8
	Grid number	6	12	18
	Iteration	100	150	250
	Population	100	150	250
	Repository	90	140	190

Table 4. Normalized result of algorithms

	NSGA II			MOPSO		
	MID	**SNS**	**CPU**	**MID**	**SNS**	**CPU**
1	0	0	111	0.024	0	76
2	0	0	155	0	0.022	112
3	0.0066	0.101	326	0.045	0.060	251
4	0	0	699	0.141	0	140
5	0	0.578	455	0	0.020	222
6	0.1598	0	502	0.125	0	245
7	0.4063	0	751	0.084	0	134
8	0	0.105	463	0	0.114	173
9	0.1018	0	262	0.163	0	107
10	0.1865	0	501	0.062	0	126
Ave	**0.08585**	**0.026422**	**2755.6**	**0.086799**	**0.040029**	**524**

Table 5. Normalized weight score and time for algorithms

Instance	**NSGA II**		**MOPSO**	
	w	**CPU time**	***w***	**CPU time**
1	0	1113	0.146853	76
2	0.00000066	155	0.009139	112
3	0.04437	326	0.051705	251
4	0	699	0.084804	140
5	0.023147	455	0.08103	222
6	0.095884	5024	0.075079	245
7	0.0242176	7513	0.050932	134
8	0.042137	4632	0.045812	1738
9	0.061127	2627	0.097993	1070
10	0.111937	5012	0.037569	1256
Average	**0.06284**	**2755.6**	**0.068092**	**524.4**

Table 6. Two performance algorithms compared

	w	**CPU time**
NSGA II	0.062084	2755.6
MOPSO	0.068092	524.4

Determining the weight of each criterion (*w* and CPU time) is complicated and completely depends on the decision-maker. In this study, a sensitivity analysis has been performed on the weight of each criterion. The different weight that decision-maker may choose is applied to a decision matrix and the result is illustrated in Figure 9. As is shown, TOPSIS runs 8 times for different weights. Generally, MOPSO has better performance except the last run where the weigh of CPU time is almost 0 and *w* is 1. It is clear that selecting zero as CPU time's weight is not reasonable because one of the most important reasons for using metaheuristic algorithm is reducing the CPU time. Thus, it can be concluded that MOPSO has better performance for the current mathematical model. According to the analysis, MOPSO is selected as the best algorithm. Parameters tanning can enhance the results of the MOPSO. Instance 1 is selected as a representative of all instances. To interpret the result of Taguchi, mean of means and signal to noise ratio are used. The lower amount of mean of means and higher signal to noise ratio shows the best level of parameters. Figure 8 shows charts of Taguchi results for instance 1 obtained by Minitab 17 software.

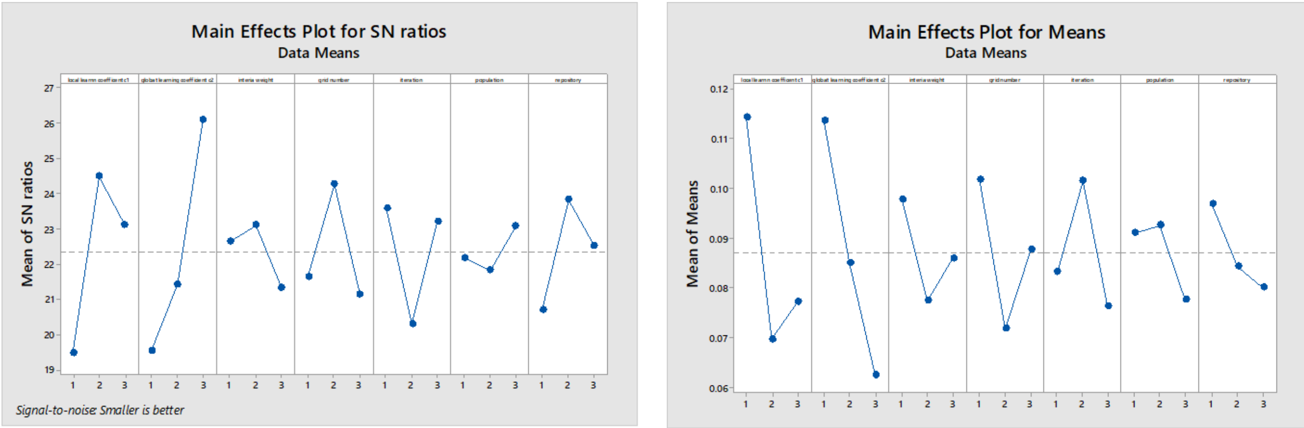
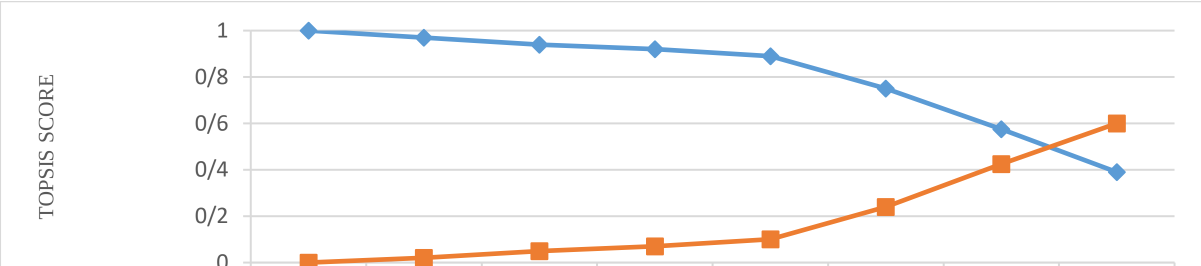


Figure 8. Taguchi plot for MOPSO



	w=0/1 time =0/9	w=0/2 time =0/8	w=0/4 time =0/6	w=0/5 time =0/5	w=0/6 time =0/4	w=0/8 time =0/2	w=0/9 time =0/1	w=0/95 time =0/05
MOPSO score	1	0/97	0/94	0/92	0/89	0/75	0/5754	0/39
NSGA II score	0	0/02	0/05	0/07	0/1	0/24	0/4245	0/6

Figure 9. Sensitivity analysis on the weight of metaheuristic decision matrix

5. Conclusion

In the current study, a supply chain management problem considering sustainable supplier selection, order allocation and production planning are addressed. The first objective maximizes total profit along with quantity discounts that suppliers can offer for each item. The second objective is designed to make a balance between unsatisfied demands and the total demand of each market. Due to the importance of sustainability in a supply chain, the third objective of the model considers the sustainability of suppliers. The developed model is solved utilizing two multi-objective metaheuristic algorithms including NSGA II and MOPSO. Hence, the current approach is capable of capturing the trade-off between total profits, unsatisfied demand and risk. Applied metaheuristic algorithms in addition to reducing CPU time, provides flexibility to select a solution from a set of trade-offs. Moreover, the experimental results show acceptable deviation from the exact and metaheuristic solutions. Furthermore, the priority-based method has been used in all algorithms to control the feasibility of results.

In this study, ten instances with different scale are randomly generated to compare the capability of algorithms. TOPSIS method is adopted for two alternatives (MOPSO and NSGA II) based on two criteria. TOPSIS results show that MOPSO has better performance for the current model. Therefore, Taguchi method is used to select the best operational parameters for MOPSO and enhance the performance of the algorithm. The main finding of the paper is as follows:

- Achieving a competitive advantage by selecting sustainable suppliers.
- Obtaining more profit by considering all-unit and incremental quantity discount
- Improving the applicability of supplier selection and production planning by reducing CPU time with metaheuristic algorithms.
- Solving large-scale problems with normal computers

In future work, we intend to consider the uncertainty of the parameters like demand and transportation cost, enhance parameter tuning and mathematical solution methodology. Furthermore, other objective functions could be added according to case studies. Although, MOPSO is performed reasonably well for three objective functions in this case, this might not work efficiently when the number of objectives increases. So using enhanced algorithms for multi-objective can be helpful. For these cases, (Deb and Jain, 2014) proposed NSGA III as the extinction of NSGA II with a major change in the operator of selection.

References

Arabsheybani, A., Paydar, M.M., and Safaei, A.S. (2018). An integrated fuzzy MOORA method and FMEA technique for sustainable supplier selection considering quantity discounts and supplier's risk. *Journal of Cleaner Production*, Vol. 190, pp. 577-591.

Babaveisi, V., Paydar, M.M., and Safaei, A.S. (2017). Optimizing a multi-product closed-loop supply chain using NSGA-II, MOSA, and MOPSO meta-heuristic algorithms. *Journal of Industrial Engineering International*, pp.1-22.

Belin, J., Horbach, J., and Oltra, V. (2009). Determinants and specificities of eco-innovations–An econometric analysis for France and Germany based on the Community Innovation Survey. In article présenté au DIME Workshop on «Environmental innovation, industrial dynamics and entrepreneurship», Utrecht, The Netherlands, pp. 10-12.

Deb, K. (2007). Evolutionary multi-objective optimization without additional parameters. Parameter setting in evolutionary algorithms, pp. 241-257.

Deb, K., and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. IEEE Trans. *Evolutionary Computation*, Vol. 18, pp. 577-601.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, Vol. 6, pp. 182-197.

Ding, S., Chen, C., Xin, B., and Pardalos, P.M. (2018). A bi-objective load balancing model in a distributed simulation system using NSGA-II and MOPSO approaches. *Applied Soft Computing*, Vol. 63, pp. 249-267.

Eberhart, R., and Kennedy, J. (1995). A new optimizer using particle swarm theory. In Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on (IEEE), pp. 39-43.

Fahimnia, B., Farahani, R.Z., Marian, R., and Luong, L. (2013). A review and critique on integrated production–distribution planning models and techniques. *Journal of Manufacturing Systems*, Vol. 32, pp. 1-19.

Foroozesh, N., and Tavakoli-Moghadam, R. (2017). Sustainable Supplier Selection by a New Hybrid Support Vector-model based on the Cuckoo Optimization Algorithm. *International Journal of Engineering*, Vol.30, pp. 867-875.

Gen, M., and Cheng, R. (2000). Genetic algorithms and engineering optimization, Vol 7 (John Wiley & Sons).

Gupta, H., and Barua, M.K. (2017). Supplier selection among SMEs on the basis of their green innovation ability using BWM and fuzzy TOPSIS. *Journal of Cleaner Production*, Vol. 152, pp. 242-258.

Kim, I.Y., and De Weck, O. (2006). Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. S*tructural and multidisciplinary optimization*, Vol. 31, pp. 105-116.

Manerba, D., and Perboli, G. (2019). New solution approaches for the capacitated supplier selection problem with total quantity discount and activation costs under demand uncertainty. *Computers & Operations Research*, Vol. 101, pp. 29-42.

Najjartabar, M., Shetaban, S., and Malmir, B. (2016). An Integrated location-inventory model for supply chain network with correlated demand. In Proceedings of the 2016 international conference on information systems, logistics and supply chain, June, pp. 1-4.

Negash, Y.T., Kartika, J., Tseng, M.-L., and Tan, K. (2020). A novel approach to measure product quality in sustainable supplier selection. *Journal of Cleaner Production,* Vol. 252, 119838-119852.

Parkouhi, S.V., and Ghadikolaei, A.S. (2017). A resilience approach for supplier selection: Using Fuzzy Analytic Network Process and grey VIKOR techniques. *Journal of Cleaner Production,* Vol. 161, pp. 431-451.

Peng, W., and Wei, Y. (2008). PSO for solving RCPSP. In Control and Decision Conference, 2008. CCDC 2008. Chinese (IEEE), pp. 818-822.

Roy, R.K. (2010). A primer on the Taguchi method (Society of Manufacturing Engineers).

Sarrafha, K., Rahmati, S.H.A., Niaki, S.T.A., and Zaretalab, A. (2015). A bi-objective integrated procurement, production, and distribution problem of a multi-echelon supply chain network design: A new tuned MOEA. *Computers & Operations Research,* Vol. 54, pp. 35-51.

Sarvestani, H.K., Zadeh, A., Seyfi, M., and Rasti-Barzoki, M. (2019). Integrated order acceptance and supply chain scheduling problem with supplier selection and due date assignment. *Applied Soft Computing,* Vol. 75, pp. 72-83.

Tirkolaee, E.B., Mardani, A., Dashtian, Z., Soltani, M., and Weber, G.-W. (2020). A novel hybrid method using fuzzy decision making and multi-objective programming for sustainable-reliable supplier selection in two-echelon supply chain design. *Journal of Cleaner Production,* Vol. 250 , 119517-119532.

Triantaphyllou, E. (2000). Multi-criteria decision making methods. In Multi-criteria decision making methods: A comparative study (Springer), pp. 5-21.

Türk, S., Özcan, E., and John, R. (2017). Multi-objective optimisation in inventory planning with supplier selection. *Expert Systems with Applications*, Vol. 78, pp. 51-63.

Valipour Parkouhi, S., Safaei Ghadikolaei, A., and Fallah Lajimi, H. (2019). Resilient supplier selection and segmentation in grey environment. *Journal of Cleaner Production,* Vol. 207, pp. 1123-1137.