

Hindawi Publishing Corporation
ISRN Artificial Intelligence
Volume 2013, Article ID 795752, 13 pages
<http://dx.doi.org/10.1155/2013/795752>



Research Article

Multiobjective Stochastic Programming for Mixed Integer Vendor Selection Problem Using Artificial Bee Colony Algorithm

Mostafa Ekhtiari¹ and Shahab Poursafary²

¹ Department of Industrial Management, Management and Accounting, Shahid Beheshti University, Tehran, Iran

² Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran

Correspondence should be addressed to Mostafa Ekhtiari; m_ekhtiari@yahoo.com

Received 21 September 2013; Accepted 13 October 2013

Academic Editors: R.-C. Hwang, P. Kokol, and Q. K. Pan

Copyright © 2013 M. Ekhtiari and S. Poursafary. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It has been always critical and inevitable to select and assess the appropriate and efficient vendors for the companies such that all the aspects and factors leading to the importance of the select process should be considered. This paper studies the process of selecting the vendors simultaneously in three aspects of multiple criteria, random factors, and reaching efficient solutions with the objective of improvement. Thus, selecting the vendors is introduced in the form of a mixed integer multiobjective stochastic problem and for the first time it is converted by CCGC (min-max) model to a mixed integer nonlinear single objective deterministic problem. As the converted problem is nonlinear and solving it in large scale will be time-consuming then the artificial bee colony (ABC) algorithm is used to solve it. Also, in order to better understand ABC efficiency, a comparison is performed between this algorithm and the particle swarm optimization (PSO) and the imperialist competitive algorithm (ICA) and Lingo software output. The results obtained from a real example show that ABC offers more efficient solutions to the problem solving in large scale and PSO spends less time to solve the same problem.

1. Introduction

In a competitive environment, to select and evaluate vendors are among the most important issues ahead of manufacturing companies as the expenses of procuring raw materials largely contribute to the final cost of a product. Vendor selection is one of the most important activities of a purchasing department and the selection of an appropriate vendor could largely result in decreases in purchase costs, delivery time, and an increase in customers' satisfaction and company's competition power. Companies are better off selecting as many appropriate vendors as they could commensurate with the production capacity of all potential vendors and establishing long lasting and useful relationships with them.

What brings about desirable results for companies managers in their decisions on selecting vendors is the selection of vendors based on three aspects of multiple criteria, random factors, and the quality of the obtained results simultaneously. The selection process and the evaluation of vendors are a multiple objective issue indeed so that more than one

criterion could be considered during this process. Moreover, decisions related to the selection of vendors become more intricate when different criteria are to be simultaneously dealt with in the decision making process. In the various evaluation methods proposed in the available literature, price, delivery performance, and quality are the most common criteria in evaluating suppliers [1].

Several studies have been so far conducted on the selection of vendor problem with multiple objectives/criteria some of which are dealt with as follows.

Weber and Current [2] presented a multiobjective approach to systematically analyze the inherent trade-offs involved in multiple criteria vendor selection problems. Yahya and Kingsman [3] proposed a new approach based on the use of analytic hierarchy process (AHP) method and applied it into vendor rating for a government sponsored entrepreneur development program in Malaysia. Lam and Tang [4] proposed a new integrated supply chain model for vendor allocation in a multiechelon supply chain. This model takes into account the usual cost objective and other

important criteria in a multiechelon supply chain ranging from the most upstream suppliers' quality to end customers' satisfaction level through a large-scale multi-objective linear programming (MOLP).

Lin et al. [5] proposed a novel hybrid multiple criteria decision making (MCDM) technique to cope with the complex and interactive vendor evaluation and selection problem which can determine the structural relationships and the interrelationships amongst all the evaluation's dimensions and support the analytic network process (ANP) method to arrange appropriate weightings to each dimension and criterion in the evaluation model by summarizing the opinions of the experts. Hsu et al. [6] proposed how the best selection to conduct the recycled materials can be implemented for enhancing and increasing the efficiency of using resources in the manufacturing process through recycled materials vendor selection. They used the MCDM model combining DEMATEL-based on ANP (called DANP) with VIKOR to solve the recycled materials vendor selection problems of multiple dimensions and criteria that were interdependent. Furthermore, some of the latest works concerning multiple criteria decision making in vendor selection problem can be searched in Zanjirani Farahani and Fadaei [7], Li et al. [8], Zhang et al. [9], and Arunkumar et al. [10].

One of the issues to be always accounted for in selecting vendors is to consider random factors and conditions that are likely to occur due to the existence of changing conditions and a dearth of information. To this end, some conventional techniques and methods, which consider the stochastic conditions of problem, should be used to select vendors. Stochastic programming deals with a class of optimization models and algorithms, in which all or some of the parameters may be subject to significant uncertainty. Stochastic programming is capable of inserting random factors in selecting and evaluating vendors.

Leung et al. [11] illustrated the production planning problem in supply chain management, with additional constraints, such as production plant preference selection. To deal with the uncertain demand data, they proposed a stochastic programming approach to determine optimal medium-term production loading plans under an uncertain environment. Talluri et al. [12] presented a chance-constrained data envelopment analysis (CCDEA) approach in the presence of multiple performance measures of vendors that were uncertain. Xu and Ding [13] presented a class of chance-constrained multi-objective linear programming models with birandom coefficients for vendor selection problem. They designed a genetic algorithm (GA) based on bi-random simulation for solving a birandom multiobjective vendor selection problem. Zhimin et al. [14] developed a multiple objective mixed integer stochastic programming model for the vendor selection problem (VSP) with stochastic demand under multiproducts purchases. Kasilingam and Lee [15] proposed a mixed integer programming model to select vendors and determine the order quantities. The model considers the stochastic nature of demand, the quality of supplied parts, the cost of purchasing and transportation, the fixed cost for establishing vendors, and the cost of receiving poor quality parts. Their proposed model also considers the lead time requirements for the parts.

Alonso-Ayuso et al. [16] presented a two-stage stochastic one-zero modeling and a related algorithmic approach for supply chain management under uncertainty. Zang et al. [17] developed a new chance-constrained programming model for supplier selection problem. In their proposed optimization problem, costs, quality, and lead times were characterized by random variables.

A review of the researches conducted in this regard shows that the selection of vendors has been modeled in the form of zero and one variables. For example, Keskin et al. [18] proposed a mixed integer nonlinear model for integrated vendor selection under capacity constraints. On the other hand, the issues that are modeled on a large scale and based on zero and one increase the time to achieve a solution. Hence, the researchers make use of metaheuristic algorithms to overcome this problem. Many algorithms such as GA, ABC, and PSO have been introduced so far. In the literature, there are several researches on the use of these algorithms in optimizing the vendor selection problem as follows.

He et al. [19] developed a class of special chance-constrained programming models and a GA designed for the vendor selection problem. They considered quality and service as uncertain parameters. Taleizadeh et al. [20] presented a multibuyer multivendor supply chain problem, in which the demand of each product was stochastic and was following a uniform distribution. The model of this problem was of an integer nonlinear programming type and in order to solve it a harmony search algorithm was employed. Also, to validate the solution and to compare the performance of the proposed algorithm, a GA was utilized as well. Huang et al. [21] used PSO algorithm to solve partner selection problem under uncertainty. Kuo et al. [22] developed an intelligent vendor decision support system by the collection of quantitative data such as profit and productivity, a PSO-based fuzzy neural network to derive the rules for qualitative data.

ABC is one of the new metaheuristic algorithms that can be used for solving nonlinear problems on a large scale. The algorithm was first introduced by Karaboga [23]. This paper studies the performance of ABC in solving the stochastic problem of selecting vendors on a large scale. It is conducted with the objective to improve the vendors' selection and evaluation process simultaneously at three aspects, that is, multiple criteria, random factors, and achievement of efficient solutions. For this purpose, the problem of vendor selection is introduced in the form of a mixed integer multi-objective stochastic problem as converted by CCGC (min-max) [24] into a mixed integer nonlinear single objective deterministic problem. The converted problem is nonlinear and its solution on the larger scale could be more time-consuming; therefore, in order to solve it, not only Lingo software; but also ABC, PSO, and ICA have been used and their performance is compared and evaluated from the view of solutions quality and the speed of reaching solutions.

The rest of this paper is organized as follows. Section 2 introduces the main structure of mixed integer multi-objective stochastic problem for vendor selection and the CCGC (min-max) model is reviewed in Section 3. In Section 4, some basic concepts on the ABC, the PSO, and the ICA are briefly introduced, respectively, for optimization

of mixed integer nonlinear problem of vendor selection. Section 5 provides the background information for the case study problem and obtains the results. Also in this section, comparative evaluations are made to contrast the performances of these algorithms and discussions. Finally, conclusion remarks are drawn in Section 6.

2. The Mixed Integer Multiobjective Stochastic of Vendor Selection Problem

A general model for the multi-objective stochastic problem of vendor selection can be stated as follows:

$$\begin{aligned}
 \max \quad & Z_k(\mathbf{x}) = \sum_{j=1}^n \tilde{c}_{ki} x_i y_i, \quad k = 1, \dots, K, \\
 \min \quad & Z_r(\mathbf{x}) = \sum_{j=1}^n \tilde{c}_{ri} x_i y_i, \quad r = K + 1, \dots, R, \\
 \text{subject to} \quad & \sum_{i=1}^n y_i \leq N_y, \\
 & \mathbf{x} \in S, \\
 & y_i \in \{0, 1\}, \quad i = 1, \dots, n, \\
 & x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, n.
 \end{aligned} \tag{1}$$

The objectives of Program (1) can be generally divided into two categories as follows. Some objectives are positive, and the aim is to maximize them. Some of these objectives are profit and/or quality, so that the aim is to select vendors, which increase these cases. The other category of the objectives is of negative type with the aim of minimizing. Some of these objectives are cost, wastes, and/or lead time. Therefore, our objective is to select vendors that decrease these objectives. The constraints of the model are such total demand volume from the vendors, minimum and maximum goods vendors being capable of providing and/or maximum vendors selected (N_y), where have been in brief displayed with $\mathbf{x} \in S$ in Program (1). In Program (1), y_i , $i = 1, \dots, n$, is a one-zero variable, where if y_i is equal to 1, that is to say, the vendor i is selected; otherwise, vendor i is not selected.

Also, vector \mathbf{x} takes positive integer or zero values and if $\mathbf{x} \in S$ are system constraints that make solution space of the model and \tilde{c}_{ki} and \tilde{c}_{ri} are normal random parameters with known means and variances, then the objective is to maximize $\sum_{i=1}^n \tilde{c}_{ki} x_i y_i$ (for $k = 1, \dots, K$) and minimize $\sum_{i=1}^n \tilde{c}_{ri} x_i y_i$ (for $r = K + 1, \dots, R$).

Program (1) can be solved by an appropriate optimization technique. Next section introduces the CCGC (min-max) model to solve Program (1) and multi-objective stochastic problems.

3. The CCGC (Min-Max) Model

The CCGC (min-max) model was proposed by Ekhtiari and Ghoseiri [24]. They introduced their deterministic equivalent programming model for solving multi-objective stochastic

problems such as Program (1). The CCGC (min-max) model, which is equivalent to Program (1), can be stated as

$$\begin{aligned}
 \min \quad & v, \\
 \text{subject to} \quad & v \geq w_k \times \left(U_k - \sum_{j=1}^n E(\tilde{c}_{ki}) x_i y_i \right. \\
 & \quad \left. + \Phi^{-1}(1 - \alpha_k) \right. \\
 & \quad \left. \times \sqrt{\sum_{i=1}^n \text{Var}(\tilde{c}_{ki}) x_i^2 y_i} \right), \\
 & \quad \quad \quad k = 1, \dots, K, \\
 & v \geq w_r \times \left(\sum_{i=1}^n E(\tilde{c}_{ri}) x_i y_i - U_r \right. \\
 & \quad \left. + \Phi^{-1}(1 - \alpha_r) \right. \\
 & \quad \left. \times \sqrt{\sum_{i=1}^n \text{Var}(\tilde{c}_{ri}) x_i^2 y_i} \right), \\
 & \quad \quad \quad r = K + 1, \dots, R, \\
 & \sum_{i=1}^n y_i \leq N_y, \\
 & \mathbf{x} \in S, \\
 & y_i \in \{0, 1\}, \quad i = 1, \dots, n, \\
 & x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, n, \\
 & v \geq 0,
 \end{aligned} \tag{2}$$

where w_k and w_r , respectively, are preference weights of objective k ($k = 1, \dots, K$) and r ($r = K + 1, \dots, R$), and $\sum_{k=1}^K w_k + \sum_{r=K+1}^R w_r = 1$ ($w_k, w_r > 0$, for $k = 1, \dots, K$ and $r = K + 1, \dots, R$). Also in Program (2), α_k and α_r , respectively, are the threshold value of objective k ($k = 1, \dots, K$) and $\Phi(z) = \text{prob}(N(0, 1) \leq z)$ represents the probability distribution function of a standard normal distribution. Meanwhile, r ($r = K + 1, \dots, R$) and U_k and U_r , respectively, are the utopia value of objective k ($k = 1, \dots, K$) and r ($r = K + 1, \dots, R$).

Let us consider c_{ki}^* as the maximum value observed for objective k , variable i for all the state of nature $\omega(c_{ki}^* = \max_{\omega} c_{ki})$. In Program (2), U_k is the best solution of the objective function $\sum_{i=1}^n c_{ki}^* x_i y_i$ subject to system constraints. In other words,

$$\begin{aligned}
 U_k = \max \quad & \sum_{i=1}^n c_{ki}^* x_i y_i, \quad k = 1, \dots, K, \\
 \text{subject to} \quad & \mathbf{x} \in S, \\
 & y_i \in \{0, 1\}, \quad i = 1, \dots, n, \\
 & x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, n.
 \end{aligned} \tag{3}$$

Also, let us consider c_{ri}^* as the minimum value observed for objective r , variable i for all the state of nature $\omega(c_{ri}^* = \min_{\omega} \tilde{c}_{ri})$. In Program (2), U_r is the best solution of the objective function $\sum_{i=1}^n c_{ri}^* x_i y_i$ subject to system constraints. In other words,

$$U_r = \min \sum_{i=1}^n c_{ri}^* x_i y_i, \quad r = K + 1, \dots, R,$$

subject to $\mathbf{x} \in S$ (4)

$$y_i \in \{0, 1\}, i = 1, \dots, n,$$

$$x_i \in \{0, \mathbf{Z}^+\}, i = 1, \dots, n.$$

Program (2) is a nonlinear programming model whose solution by Lingo software results in local solutions. Therefore, in order to obtain effective solutions, one may make use of metaheuristic algorithms, which, in addition to original ABC, PSO, and ICA, will be reviewed in brief for solving the problem of selecting vendors.

4. Metaheuristic Algorithms

One of the common methods for solving optimization problems is metaheuristic algorithms. In some problems with structural complication, mathematical methods and software packages based on mathematical methods are not capable of solving them or their solution time is too lengthy. Some metaheuristic algorithms can be recommended for such problems. The structure of such algorithms is that, at first, they produce primary random population in the searching space, and then they make use of latent calculation intelligence in their structure; they move the solutions in such a way as to be directed towards optimal point.

The main advantages of metaheuristic algorithms are as follows [25].

- (1) *Being robust to dynamic changes*: traditional methods of optimization are not robust to dynamic changes in the environment and they require a complete restart for providing a solution. In contrary, evolutionary computation can be used to adapt solutions to the changing circumstances.
- (2) *Broad applicability*: metaheuristic algorithms can be applied to any problems that can be formulated as function optimization problems.
- (3) *Hybridization with other methods*: metaheuristic algorithms can be combined with more traditional optimization techniques.
- (4) *Solves problems that have no solutions*: the advantages of metaheuristic algorithms includes the ability to address problems for which there is no human expertise. Even though human expertise should be used when it is needed and available, it often proves less adequate for automated problem-solving routines.

Some of meta-heuristic algorithms as introduced are reviewed in brief as follows.

4.1. The ABC. ABC is one of the metaheuristic algorithms recently introduced. The stages of original ABC are as follows [26].

In ABC algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population. At the first step, the ABC generates randomly distributed initial population P ($C = 0$) of SN solutions (food source positions), where SN denotes the size of employed bees or onlooker bees. Each solution x_i ($i = 1, \dots, SN$) is a D -dimensional vector. Here, D is the number of optimization parameters. After initialization, the population of the positions (solutions) is subject to repeated cycles, $C = 1, \dots, MCN$, of the search processes of the employed bees, the onlooker bees, and the scout bees. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) and tests the nectar amount (fitness value) of the new source (new solution). If the nectar amount of the new one is higher than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one in her memory. After all employed bees complete the search process, they share the nectar information of the food sources and their position information with the onlooker bees. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. As in the case of the employed bee, she produces a modification on the position in her memory and checks the nectar amount of the candidate source. If the nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one.

The main steps of the algorithm are as follows.

- (1) Initialize population.
- (2) *repeat*.
- (3) Place the employed bees on their food sources.
- (4) Place the onlooker bees on the food sources depending on their nectar amounts.
- (5) Send the scouts to the search area for discovering new food sources.
- (6) Memorize the best food source found so far.
- (7) *until* requirements are met.

In ABC algorithm, each cycle of the search consists of three steps: sending the employed bees onto their food sources and evaluating their nectar amounts; after sharing the nectar information of food sources, selecting food source regions by the onlookers and evaluating the nectar amount of the food sources; determining the scout bees and then sending them randomly onto possible new food sources. At the initialization stage, a set of food sources is randomly selected by the bees and their nectar amounts are determined. At the first step of the cycle, these bees come into the hive and share the nectar information of the sources with

the bees waiting on the dance area. A bee waiting on the dance area for making decision to choose a food source is called onlooker and the bee going to the food source visited by herself just before is named as employed bee. After sharing their information with onlookers, every employed bee goes to the food source area visited by herself at the previous cycle since that food source exists in her memory and then chooses a new food source by means of visual information in the neighborhood of the one in her memory and evaluates its nectar amount. At the second step, an onlooker prefers a food source area depending on the nectar information distributed by the employed bees on the dance area. As the nectar amount of a food source increases, the probability of that food source chosen also increases. After arriving at the selected area, the bee chooses a new food source in the neighborhood of the one in the memory depending on visual information as in the case of employed bees. The determination of the new food source is carried out by the bees based on the comparison process of food source positions visually. At the third step of the cycle, when the nectar of a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and replaced with the abandoned one. In our model, at each cycle at most one scout goes outside for searching a new food source, and the number of employed and onlooker bees is selected to be equal to each other. These three steps are repeated through a predetermined number of cycles called maximum cycle number (MCN) or until a termination criterion is satisfied. An artificial onlooker bee chooses a food source depending on the probability value associated with that food source, p_i , calculated by

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{SN} \text{fit}_n}, \quad (5)$$

where fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i and SN is the number of food sources which is equal to the number of employed bees or onlooker bees.

In order to produce a candidate food position from the old one in memory, the ABC uses

$$v_{ij} = x_{ij} + \varphi_{ij} (x_{ij} - x_{kj}), \quad (6)$$

where $k \in \{1, \dots, SN\}$ and $j \in \{1, \dots, D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i . φ_{ij} is a random number between $[-1, 1]$. It controls the production of neighbor food sources around x_{ij} and represents the comparison of two food positions visually by a bee. As can be seen from (6), as the difference between the parameters of the x_{ij} and x_{kj} decreases, the perturbation on the position x_{ij} gets decreased, too. Thus, as the search approaches the optimum solution in the search space, the step length is adaptively reduced.

If a parameter value produced by this operation exceeds its predetermined limit, the parameter can be set to an acceptable value. In this work, the value of the parameter exceeding its limit is set to its limit value.

The food source of which the nectar is abandoned by the bees is replaced with a new food source by the scouts.

In ABC, this is simulated by producing a position randomly and replacing it with the abandoned one. In ABC, if a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called "limit" for abandonment. Assume that the abandoned source is x_i and $j \in \{1, \dots, D\}$; then the scout discovers a new food source to be replaced with x_i . This operation can be defined as

$$x_i^j = x_{\min}^j + \text{rand}[0, 1] (x_{\max}^j - x_{\min}^j). \quad (7)$$

After each candidate source position v_{ij} is produced and then evaluated by the artificial bee, its performance is compared with that of its old one. If the new food source has an equal or better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained in the memory. In other words, a greedy selection mechanism is employed as the selection operation between the old and the candidate one.

Totally, ABC algorithm employs four different selection processes: (1) a global probabilistic selection process, in which the probability value is calculated by (5), used by the onlooker bees for discovering promising regions, (2) a local probabilistic selection process carried out in a region by the employed bees and the onlookers depending on the visual information such as the color, shape and fragrance of the flowers (sources) (bees will not be able to identify the type of nectar source until they arrive at the right location and discriminate among sources growing there based on their scent) for determining a food source around the source in the memory in a way described by (6), (3) a local selection called greedy selection process carried out by onlooker and employed bees in which if the nectar amount of the candidate source is better than that of the present one, the bee forgets the present one and memorizes the candidate source produced by (6), otherwise, the bee keeps the present one in the memory, and (4) a random selection process carried out by scouts as defined in (7).

It is clear from the above explanation that there are three control parameters in the basic ABC: the number of food sources which is equal to the number of employed or onlooker bees (SN), the value of limit, and the maximum cycle number (MCN).

In the case of honeybees, the recruitment rate represents a measure of how quickly the bee colony finds and exploits a newly discovered food source. Artificial recruiting could similarly represent the measurement of the speed with which the feasible solutions or the good quality solutions of the difficult optimization problems can be discovered. The survival and progress of the bee colony are dependent upon the rapid discovery and efficient utilization of the best food resources. Similarly, the successful solution of difficult engineering problems is connected to the relatively fast discovery of good solutions especially for the problems that need to be solved in real time. In a robust search process, exploration and exploitation processes must be carried out together. In the ABC algorithm, while onlookers and employed bees carry

out the exploitation process in the search space, the scouts control the exploration process. Detailed pseudocode of the ABC algorithm is given as follows.

- (1) Initialize the population of solutions $x_i, i = 1, \dots, SN$.
- (2) Evaluate the population.
- (3) cycle = 1.
- (4) repeat.
- (5) Produce new solutions v_i for the employed bees by using (6) and evaluate them.
- (6) Apply the greedy selection process for the employed bees.
- (7) Calculate the probability values p_i for the solutions x_i by (5).
- (8) Produce the new solutions v_i for the onlookers from the solutions x_i selected depending on p_i and evaluate them.
- (9) Apply the greedy selection process for the onlookers.
- (10) Determine the abandoned solution for the scout, if exists, and replace it with a new randomly produced solution x_i by (7).
- (11) Memorize the best solution achieved so far.
- (12) cycle = cycle + 1.
- (13) until cycle = MCN.

4.2. The PSO. This subsection will introduce the original PSO algorithm developed by Kennedy and Eberhart [27] for optimization of continuous nonlinear functions. PSO was inspired by the motion of a flock of birds searching for food. During the search, each bird, called a particles adjust, its searching direction according to two factors, its own best previous experience (pbest) and the experience of all other members (gbest).

Mathematically, assume that the searching space is D -dimensional. Let $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t)$ be the particle i in D -dimensional vector, where X_i^t is treated as a potential solution that explores the search space by the rate of position change called velocity. The velocity is denoted as $V_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{id}^t)$. Let $P_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{id}^t)$ be the best particle i obtained until iteration t and let $P_g^t = (p_{g1}^t, p_{g2}^t, \dots, p_{gd}^t)$ be the global best in the population at iteration t . The basic procedure for implementing original PSO is described as follows [28].

- (1) Create population of particles with random positions and velocities on the searching space.
- (2) For each particle, evaluate the desired optimization fitness function and compare the evaluated fitness with its pbest. If the current particle is better than pbest, then set pbest to the current particle.
- (3) Update particle velocities according to the following equation:

$$v_{id}^t = wv_{id}^{t-1} + c_1 \text{rand}(\cdot) (p_{id}^t - x_{id}^t) + c_2 \text{rand}(\cdot) (P_{gd}^t - x_{id}^t), \quad (8)$$

where c_1 is the cognition learning factor, c_2 is the social learning factor, $\text{rand}(\cdot)$ are random numbers uniformly distributed in $U(0, 1)$, and w is the inertia weight.

- (4) Particles are changed to their new positions according to the following equation:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^t. \quad (9)$$

- (5) Stop the algorithm if the stopping criterion is satisfied; return to Step 2 otherwise.

The velocity update can improve the diversification of the search. To assure that the velocity would not lead the particles to move beyond boundaries, a maximum (V_{\max}) is set to limit the velocity range; any velocity tending to exceed it is brought back to it [29]. An inertia weight w is used to balance between global and local searches when updating the velocity in (8). The swarm population sizes ranging from 10 to 30 are the most common ones, and it has been learned that PSO requires a smaller population than is generally used in genetic algorithms to search for high quality solution [30].

4.3. The ICA. ICA is a novel global search strategy and inspired by the imperialistic competition based on the human's sociopolitical evolution. This algorithm was first introduced by Atashpaz-Gargari and Lucas [31]. Imperialistic competition forms the core of the algorithm. This causes all countries to converge to an absolute minimum cost function. The original ICA starts with some countries as initial population, which is classified into two groups. Some of the countries with more power are selected to be the imperialist and all other countries with less power than the imperialists form colonies of them. The imperialist countries absorb the colonies based on their power using the absorption policy. The total power of an empire depends on both of its constituents, the imperialist country and the colonies. The mathematical expression of this relationship is defined as a power made up of the power of the imperialist country plus a percentage of the average power of the colonies.

After the initial imperialists were formed, the imperialist competition step starts between them. Each imperialist that cannot act successfully in this competition to increase its power (or at least to prevent the loss of its influence) will collapse. Therefore, the survival of an imperialist depends directly on its ability to absorb the rival imperialists' colonies and its ability to rule them out. Consequently, during the imperialist competition, gradually the power of larger imperialist is added and weaker imperialists will be removed. Those empires that want to increase their power will be forced to develop their own colonies. Thus, over time, colonies will be closer to imperialists, and a convergence will be seen. The convergence condition will be achieved when a single imperialist is created along with the colonies whose statuses are very close to the imperialist country.

In the next section, original ABC performance in solving a stochastic problem of selecting vendors in the form of a real example is reviewed in comparison to original PSO and ICA and Lingo software.

5. Case Study

In this section, the performance of the proposed methodology is validated on a real problem. For this purpose, a real data set is obtained from a home appliances manufacturer as a case study in the manufacturing industry in Iran. This company manufactures a wide range of home appliances in its factory near Tehran in Iran. Besides, the company also markets its products in various cities.

In order to procure some materials and components, the company is always in need of keeping contact with vendors. At present, the company works with ten vendors only. They are its first priority. What the company is looking for is a planning based on two issues: first, if the company wants to continue its activities with these ten vendors according to certain criteria and limits, which vendors could meet how much demand of the company? Second, if the company wants to increase its potential vendors for its future development, which vendor can meet how much need of the company?

As the problem propounded by the company revolves around the selection of vendors and determination of the good number they provide, it can be defined in the form of a mixed integer multi-objective stochastic programming problem by taking some random parameters into account. They will be described in detail as follows.

5.1. Objectives

(i) *Objective Function of Purchase Cost.* If c_i ($i = 1, \dots, n$) is fixed cost of purchase of a good unit from vendor i , n is number of vendors, x_i is number of products bought from vendor i in such a way that its value is an integer, y_i is a zero and one variable, which is equal to 1 if it is bought from vendor i and zero if not, and Z_1 is objective function of the purchase cost from vendors, then we have

$$\min Z_1 = \sum_{i=1}^n c_i x_i y_i. \quad (10)$$

(ii) *Objective Function of Wastes.* If \tilde{y}_i is a normal random variable of waste percentage for the vendor i and Z_2 is the objective function of wastes for vendors, then we have

$$\min Z_2 = \sum_{i=1}^n \tilde{y}_i x_i y_i. \quad (11)$$

The best value of the random variable of \tilde{y}_i is the least waste percentage observed for the vendor i in accordance with the historical data. Meanwhile, mean and variance of this random variable are calculated based on its relevant historical data.

(iii) *Objective Function of Lead Time.* If \tilde{t}_i is normal random variable for lead time of vendor i and Z_3 is the objective function for lead time of vendors, then we have

$$\min Z_3 = \sum_{i=1}^n \tilde{t}_i x_i y_i. \quad (12)$$

The best value for \tilde{t}_i is the least lead time observed for vendor i in accordance with the historical data. Meanwhile, mean and variance of random variable can be calculated based on its relevant historical data.

(iv) *Objective Function of Quality.* If \tilde{q}_i is normal random variable for quality percentage of products provided by vendor i and if Z_4 is quality objective function of vendors, then we have

$$\max Z_4 = \sum_{i=1}^n \tilde{q}_i x_i y_i. \quad (13)$$

The best value of \tilde{q}_i is the highest percentage observed by vendor i in accordance with historical data. Meanwhile, mean and variance of this random variable can be calculated based on its relevant historical data.

5.2. Constraints

(i) *Demand Volume.* If X stands for total purchase demand volume from vendors, then we have

$$\sum_{i=1}^n x_i y_i = X. \quad (14)$$

(ii) *Selection of Maximum Number of Vendors.* If N_y is the maximum number of vendors to whom we can refer to purchase goods, then we have

$$\sum_{i=1}^n y_i \leq N_y. \quad (15)$$

(iii) *Minimum and Maximum Numbers of Products Each Provider Can Provide.* If \underline{x}_i and \bar{x}_i are minimum and maximum numbers of products that vendor i can provide, then we have

$$\underline{x}_i y_i \leq x_i \leq \bar{x}_i y_i, \quad i = 1, \dots, n. \quad (16)$$

Therefore, the general form of mixed integer multi-objective stochastic model for vendor selection can be as follows

$$\begin{aligned} \min \quad & Z_1 = \sum_{i=1}^n c_i x_i y_i, \\ \min \quad & Z_2 = \sum_{i=1}^n \tilde{y}_i x_i y_i, \\ \min \quad & Z_3 = \sum_{i=1}^n \tilde{t}_i x_i y_i, \\ \max \quad & Z_4 = \sum_{i=1}^n \tilde{q}_i x_i y_i, \end{aligned}$$

$$\begin{aligned}
\text{subject to } & \sum_{i=1}^n x_i y_i = X, \\
& \sum_{i=1}^n y_i \leq N_y, \\
& \underline{x}_i y_i \leq x_i \leq \bar{x}_i y_i, \quad i = 1, \dots, n, \\
& x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, n, \\
& y_i \in \{0, 1\}, \quad i = 1, \dots, n.
\end{aligned} \tag{17}$$

Due to the existence of varying conditions and inadequate information, Program (17) is a mixed integer multi-objective stochastic model whose some of its parameters are normal random variables with known mean and variance. Therefore, based on Program (2), Program (17) can be converted into a deterministic equivalent model like Program (18):

$$\begin{aligned}
\min \quad & v, \\
\text{subject to } \quad & v \geq w_1 \times \left(\sum_{i=1}^n c_i x_i y_i - U_1 \right), \\
& v \geq w_2 \times \left(\sum_{i=1}^n E(\tilde{y}_i) x_i y_i - U_2 + \Phi^{-1} \right. \\
& \quad \left. \times (1 - \alpha_2) \sqrt{\sum_{i=1}^n \text{Var}(\tilde{y}_i) x_i^2 y_i} \right), \\
& v \geq w_3 \times \left(\sum_{i=1}^n E(\tilde{t}_i) x_i y_i - U_3 + \Phi^{-1} \right. \\
& \quad \left. \times (1 - \alpha_3) \sqrt{\sum_{i=1}^n \text{Var}(\tilde{t}_i) x_i^2 y_i} \right), \\
& v \geq w_4 \times \left(U_4 - \sum_{j=1}^n E(\tilde{q}_i) x_i y_i + \Phi^{-1} \right. \\
& \quad \left. \times (1 - \alpha_4) \sqrt{\sum_{i=1}^n \text{Var}(\tilde{q}_i) x_i^2 y_i} \right), \\
& \sum_{i=1}^n x_i y_i = X, \\
& \sum_{i=1}^n y_i \leq N_y, \\
& \underline{x}_i y_i \leq x_i \leq \bar{x}_i y_i, \quad i = 1, \dots, n, \\
& x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, n, \\
& y_i \in \{0, 1\}, \quad i = 1, \dots, n, \\
& v \geq 0.
\end{aligned} \tag{18}$$

5.3. Data Set. At present, the company has 10 vendors, which are considered its top priority. The whole set of data related to the performance of each of them is presented in Table 1.

The information presented in Table 1 is in accordance with the two-year performance of 10 main vendors of the company. As it can be seen in this table, waste percentage, lead time, and the quality are normal random variables with known means and variances that have been calculated based on each vendor's historical data. Meanwhile, as the costs of the purchases from the vendors have been fixed, the purchase cost is considered as certain parameter. Given the data of Table 1, the demand volume from all vendors being 200 and the maximum selection of vendors' number being 5, Program (18) was solved by Lingo software package and its results have been shown in Table 2 (the complete shape of the model inserted in Lingo software is presented in appendix section of the paper). The preference weights, w_k (for $k = 1, 2, 3, 4$), have been equally considered to be 0.25 and the values of α_2 , α_3 , and α_4 are determined to be 0.1, 0.025 and 0.05, respectively. As the problem modeled in Program (18) is nonlinear and the solution obtained is local, therefore, in order to increase the accuracy of the results, this problem in the presence of 10 main vendors was repeated for 25 times. All results have been similarly saved for the solution achievement time at each execution time.

Considering the results presented in Table 2, four of the main vendors have been selected only, so that the optimal value of the objective function of Program (18) is equal to $v = 145.94$ and the average time for obtaining solution is some 1.4 seconds. As the problem of selecting vendors as commercial partners is of high importance for the company and as the model inserted in Lingo software is nonlinear, the results shown in Table 2 are local and better solutions are likely. For this purpose, three metaheuristic algorithms, ABC, PSO, and ICA, which are among the most updated metaheuristic algorithms, are used to solve the problem of Program (18).

5.4. Parameters Tuning. For tuning the ABC for the mixed integer vendor selection stochastic problems, extensive experiments were conducted with differing sets of parameters in a competence against PSO, and ICA. At the end, the following sets were found to be effective in terms of solutions quality. Table 3 presents the parameters tuned by ABC, PSO and ICA for solving the problem of selecting vendors.

5.5. Solving the Problem of Main Vendors by ABC, PSO and ICA. In order to omit random consequences of the results, the problem of Program (18) was repeated for 25 times in the presence of data related to 10 main vendors, and here the mean value of the objective function, v , and the mean CPU time obtained by ABC, PSO, and ICA are considered as the basis of comparisons. In this research, the said algorithms were executed by Matlab 7.12.0 under Microsoft Windows 7 in a personal computer with Dual Core CPU, 2.2 GHz, and a 4 GB RAM. The results obtained by ABC, PSO, and ICA in 25 repetitions are shown in Figure 1. Number of population for

TABLE 1: The primal and historical data related to 10 main vendors.

i	1	2	3	4	5	6	7	8	9	10
\underline{x}_i	40	53	65	45	40	33	28	46	55	63
\bar{x}_i	50	63	75	55	50	43	38	56	65	73
c_i (\$)	13	14	6	15	11	6	8	11	15	15
$E(\bar{\gamma}_i)$ (%)	0.037	0.03	0.023	0.05	0.023	0.023	0.042	0.023	0.05	0.05
$\text{Var}(\bar{\gamma}_i)$	452×10^{-9}	721×10^{-9}	252×10^{-9}	8×10^{-7}	613×10^{-9}	252×10^{-9}	111×10^{-9}	613×10^{-9}	7×10^{-7}	825×10^{-9}
$E(\bar{t}_i)$ (day)	3.1	3.2	3.1	3.55	3.37	3.4	3.87	3.35	3.59	3.41
$\text{Var}(\bar{t}_i)$	0.61	0.62	0.59	0.61	0.91	0.59	0.68	0.95	0.61	0.69
$E(\bar{q}_i)$ (%)	0.83	0.84	0.825	0.85	0.87	0.825	0.88	0.87	0.841	0.85
$\text{Var}(\bar{q}_i)$	53×10^{-5}	54×10^{-5}	6×10^{-4}	55×10^{-5}	535×10^{-6}	6×10^{-4}	58×10^{-5}	53×10^{-5}	45×10^{-5}	55×10^{-5}

TABLE 2: Results obtained from solving Program (18) with 10 main vendors by Lingo software.

										x_i									
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}										
0	53	65	0	40	42	0	0	0	0										
										y_i									
y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}										
0	1	1	0	1	1	0	0	0	0										

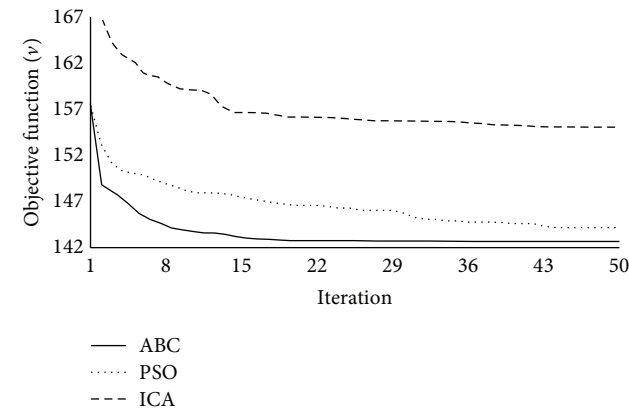


FIGURE 1: Changes trend of objective function for selection problem of 10 main vendors.

all three algorithms is 10 and maximum repetition number for each algorithm is considered to be 50.

As specified in Figure 1, ABC shows better results as compared to PSO and ICA. ABC practically reaches the final optimal solution from iteration 23 on, while PSO and ICA do not reach the solution in all 25 repetitions. Table 4 presents the mean time for achieving the solution to the problem of Program (18) in the presence of 10 main vendors as well as the least value of ν obtained via ABC, PSO, and ICA and Lingo software. From the time view, ABC with the mean time of 0.517 seconds enjoys better status considering Lingo software with the mean time of 1.4 seconds. Of course, in a general comparison, PSO takes lower mean time as compared to ABC and ICA and Lingo software, and this result is not surprising considering its simple structure. Considering that the problem modeled by Program (17) is of multi-objective

problems, the best solution is the nearest solution to utopia point (where considering the space solution, each and every one of the objectives has its own best value). This issue can be obtained through the least value of ν in Program (18). In this study, the most efficient solution is the one which presents the least possible value for ν . Hence, from the viewpoint of the quality and efficiency of the solutions obtained here, ABC with the least possible value for ν presents the best solution compared to PSO and ICA and Lingo. Also it should be noted that ICA is ranked last in obtainment of the least possible value for ν .

As seen already, ABC is ranked first in achieving an efficient solution by obtaining the least possible value for ν compared to PSO and ICA and Lingo software. Therefore, Table 5 presents the results obtained from solving Program (18) by ABC in the presence of 10 main vendors.

5.6. *Company's Future Development Strategy with the Policy to Increase the Number of Vendors.* Given the reports obtained from R&D unit of the company, one of the important strategies used by company is future development, increase in products, and further acquisition of competitive market share. Therefore, considering the constraint of minimum and maximum good number company can purchase from vendors, the company intends to also consider the policy of increasing the number of vendors commensurate with increased production volume. For this purpose, Table 6 presents the scenarios defined for the company's future strategy within the framework of changes in production volume and maximum number of vendors required.

To review the scenarios defined in Table 6, we act according to the following steps.

- (1) The problem modeled in Program (18) is repeated by Lingo software and ABC, PSO, and ICA for each and every one of the scenarios defined in Table 6, another 9 times, so that, with each repetition, the data related to 10 vendors is added to the previous problem. In other words, the first problem is modeled with the data of 20 vendors and the last problem with that of 100 vendors. It should also be noted that the data related to the volume of products and maximum number of vendors required changes commensurate with each scenario. Meanwhile, other data inserted

TABLE 3: Parameters tuned by ABC, PSO, and ICA for solving the problem of Program (18) in the presence of 10 main vendors.

ABC		PSO			ICA			
Food number	Limit	w	c_1	c_2	Number of initial imperialists	Revolution rate	β	γ
5	15	0.7298	1.4962	1.4962	2	0.4	2	0.1

TABLE 4: The average solution achievement time and the least value for ν in ABC, PSO, and ICA and Lingo software.

Algorithm	Average of min ν	Average of CPU time (seconds)
Lingo	145.9443	1.4
ABC	142.6703	0.517
PSO	144.2132	0.43
ICA	152.5845	0.565

TABLE 5: The results obtained from solving Program (18) by ABC in the presence of 10 main vendors.

x_i									
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
49	53	65	0	0	33	0	0	0	0
y_i									
y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}
1	1	1	0	0	1	0	0	0	0

for each vendor comply with the available historical data of their performances during the past cycles.

- (2) For each scenario, the value of utopia of each and everyone of the objectives is calculated separately.
- (3) Optimization of each and everyone of the problems of Step 1 is repeated 25 times so that the average of the results is finally reported for each algorithm.

It should be noted that in all scenarios, w_k (for $k = 1, 2, 3, 4$) are considered to be equal to 0.25, and α_2 , α_3 , and α_4 are 0.1, 0.025, and 0.05, respectively. Table 7 illustrates a summary of the results, which are obtained by Lingo software and ABC, PSO, and ICA according to the above steps.

One of the important parameters in the presented problem is the maximum number of vendors required. The existence of this parameter increases not only the complexity of the problem but also its solution time by the algorithms. In fact, the relation between the parameter and the company's purchase volume and the volume any vendor could provide could result in increased complexity; therefore, the population number and maximum iterations (maxiter) for algorithms are based on the complexity rate of these problems so that the solution time is lower than that of Lingo. The results presented in Table 7 imply that from the view of reaching an efficient solution, ABC provides better performance compared to other algorithms and Lingo, to the extent that only ABC presents the lowest value for ν in all scenarios. Meanwhile, PSO is ranked first from the view of the least solution achievement time.

6. Conclusion

In this paper, the problem of selecting vendors was modeled within the framework of a mixed integer multi-objective stochastic programming problem. With the objective to improve, the paper studied selection and evaluation process of vendors from the view of multiple criteria, random factors and achievement to efficient solutions. To this end, the problem of selecting vendors was presented as a mixed integer multi-objective stochastic problem its for solution; CCGC (min-max) model was used. As the presented problem was of nonlinear and mixed integer type, the original ABC was described to achieve efficient solutions. The proposed methodology was illustrated in the form of a real example so that the problem of selecting vendors was modeled on a large scale, and apart from original ABC and Lingo software, meta-heuristic algorithms PSO and ICA were also used for solving it. The results obtained indicated that the performance of ABC was better compared to PSO, ICA and Lingo output from the view of achieving efficient solutions; and PSO has a better performance compared to other algorithms from the view of the speed of reaching solution.

Appendix

Consider

$$\begin{aligned}
 \min \quad & \nu, \\
 \text{subject to} \quad & \nu \geq 0.25 \times (13x_1y_1 + 14x_2y_2 + 6x_3y_3 \\
 & \quad + 15x_4y_4 + 11x_5y_5 + 6x_6y_6 \\
 & \quad + 8x_7y_7 + 11x_8y_8 + 15x_9y_9 \\
 & \quad + 15x_{10}y_{10} - U_1), \\
 & \nu \geq 0.25 \times \left(0.037x_1y_1 + 0.03x_2y_2 \right. \\
 & \quad + 0.023x_3y_3 + 0.05x_4y_4 \\
 & \quad + 0.023x_5y_5 + 0.023x_6y_6 \\
 & \quad + 0.042x_7y_7 + 0.023x_8y_8 \\
 & \quad + 0.05x_9y_9 + 0.05x_{10}y_{10} \\
 & \quad \left. - U_2 + \Phi^{-1}(1 - \alpha_2) \right) \\
 & \quad \times \left(452 \times 10^{-9}x_1^2y_1 \right. \\
 & \quad \quad + 721 \times 10^{-9}x_2^2y_2 \\
 & \quad \quad + 252 \times 10^{-9}x_3^2y_3 \\
 & \quad \quad + 8 \times 10^{-7}x_4^2y_4 \\
 & \quad \quad \left. + 613 \times 10^{-9}x_5^2y_5 \right)
 \end{aligned}$$

TABLE 6: Scenarios defined for the company’s future development.

Scenario	1	2	3	4	5	6	7	8	9
Number of available vendors	20	30	40	50	60	70	80	90	100
Purchase demand volume (X)	400	600	1050	1200	2200	3000	3500	4000	4500
Maximum number of vendors required (N_y)	6	9	15	20	30	40	45	50	60

TABLE 7: The results obtained by ABC, PSO, and ICA and Lingo considering the scenarios related to the company’s future development.

Scenario	1	2	3	4	5	6	7	8	9
Lingo									
Average of min v	272.08	391.07	617.56	678.05	1212.78	1629.56	1856.94	2166.92	2423.74
Average of CPU time (seconds)	1.2	1.3	171	51.3	70.9	47.9	55	15.9	89
ABC									
Average of min v	267.82	375.68	610.34	675.74	1226.78	1598.91	1857.32	2097.85	2338.65
Average of CPU time (seconds)	0.95	1.05	9.56	39.54	43.51	39.65	47.31	14.58	43.41
Population	20	10	40	50	120	140	100	70	120
Maxiter	50	20	50	100	50	50	50	30	50
PSO									
Average of min v	274.324	389.86	643.16	714.3	1538.44	1678.22	1965.17	2258.42	2463.15
Average of CPU time (seconds)	0.923	0.99	6.76	10.29	18.44	34.17	35.54	8.65	24.15
Population	20	10	40	50	120	140	100	70	120
Maxiter	50	20	50	100	50	50	50	30	50
ICA									
Average of min v	278.35	384.88	636.92	708.24	1235.99	1646.22	1928.48	2206.17	2340.42
Average of CPU time (seconds)	1.17	1.22	12.33	43.54	65.56	61.1	71.93	21.46	60.8
Population	20	10	40	50	120	140	100	70	120
Maxiter	50	20	50	100	50	50	50	30	50

$$\begin{aligned}
 & + 252 \times 10^{-9} x_6^2 y_6 \\
 & + 111 \times 10^{-9} x_7^2 y_7 \\
 & + 613 \times 10^{-9} x_8^2 y_8 \\
 & + 7 \times 10^{-7} x_9^2 y_9 + 825 \\
 & \times 10^{-9} x_{10}^2 y_{10} \Big)^{1/2} \Big), \\
 v \geq & 0.25 \times \left(3.1x_1y_1 + 3.2x_2y_2 + 3.1x_3y_3 \right. \\
 & + 3.55x_4y_4 + 3.37x_5y_5 + 3.4x_6y_6 \\
 & + 3.87x_7y_7 + 3.35x_8y_8 + 3.59x_9y_9 \\
 & + 3.41x_{10}y_{10} - U_3 + \Phi^{-1} (1 - \alpha_3) \\
 & \left. \left(0.61x_1^2y_1 + 0.62x_2^2y_2 + 0.59x_3^2y_3 \right. \right. \\
 & \left. \left. + 0.61x_4^2y_4 + 0.91x_5^2y_5 \right. \right. \\
 & \left. \left. + 0.59x_6^2y_6 + 0.68x_7^2y_7 \right. \right. \\
 & \left. \left. + 0.95x_8^2y_8 + 0.61x_9^2y_9 \right. \right. \\
 & \left. \left. + 0.69x_{10}^2y_{10} \right)^{1/2} \Big), \\
 & v \geq 0.25 \times \left(U_4 - (0.83x_1y_1 + 0.84x_2y_2 \right. \\
 & \left. + 0.825x_3y_3 + 0.85x_4y_4 + 0.87x_5y_5 \right. \\
 & \left. + 0.825x_6y_6 + 0.88x_7y_7 + 0.87x_8y_8 \right. \\
 & \left. + 0.841x_9y_9 + 0.85x_{10}y_{10}) + \Phi^{-1} \right. \\
 & \left. \times (1 - \alpha_4) \right. \\
 & \left. \times \left(53 \times 10^{-5} x_1^2y_1 + 54 \times 10^{-5} x_2^2y_2 \right. \right. \\
 & \left. \left. + 6 \times 10^{-4} x_3^2y_3 + 55 \times 10^{-5} x_4^2y_4 \right. \right. \\
 & \left. \left. + 535 \times 10^{-6} x_5^2y_5 \right. \right. \\
 & \left. \left. + 6 \times 10^{-4} x_6^2y_6 \right. \right. \\
 & \left. \left. + 58 \times 10^{-5} x_7^2y_7 \right. \right. \\
 & \left. \left. + 53 \times 10^{-5} x_8^2y_8 \right. \right. \\
 & \left. \left. + 45 \times 10^{-5} x_9^2y_9 \right. \right. \\
 & \left. \left. + 55 \times 10^{-5} x_{10}^2y_{10} \right)^{1/2} \Big),
 \end{aligned}$$

$$\begin{aligned}
& \sum_{i=1}^{10} x_i y_i = 200, \\
& \sum_{i=1}^{10} y_i \leq 5, \\
& 40y_1 \leq x_1 \leq 50y_1, \quad 53y_2 \leq x_2 \leq 63y_2, \\
& 65y_3 \leq x_3 \leq 75y_3, \quad 45y_4 \leq x_4 \leq 55y_4, \\
& 40y_5 \leq x_5 \leq 50y_5, \quad 33y_6 \leq x_6 \leq 43y_6, \\
& 28y_7 \leq x_7 \leq 38y_7, \quad 46y_8 \leq x_8 \leq 56y_8, \\
& 55y_9 \leq x_9 \leq 65y_9, \quad 63y_{10} \leq x_{10} \leq 73y_{10}, \\
& x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, 10, \\
& y_i \in \{0, 1\}, \quad i = 1, \dots, 10, \\
& v \geq 0,
\end{aligned} \tag{A.1}$$

where

$$\begin{aligned}
U_1 : \min & \quad 13x_1y_1 + 14x_2y_2 + 6x_3y_3 + 15x_4y_4 + 11x_5y_5 \\
& \quad + 6x_6y_6 + 8x_7y_7 + 11x_8y_8 \\
& \quad + 15x_9y_9 + 15x_{10}y_{10}, \\
\text{subject to} & \quad \sum_{i=1}^{10} x_i y_i = 200, \\
& \quad \sum_{i=1}^{10} y_i \leq 5, \\
& \quad 40y_1 \leq x_1 \leq 50y_1, \quad 53y_2 \leq x_2 \leq 63y_2, \\
& \quad 65y_3 \leq x_3 \leq 75y_3, \quad 45y_4 \leq x_4 \leq 55y_4, \\
& \quad 40y_5 \leq x_5 \leq 50y_5, \quad 33y_6 \leq x_6 \leq 43y_6, \\
& \quad 28y_7 \leq x_7 \leq 38y_7, \quad 46y_8 \leq x_8 \leq 56y_8, \\
& \quad 55y_9 \leq x_9 \leq 65y_9, \quad 63y_{10} \leq x_{10} \leq 73y_{10}, \\
& \quad x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, 10, \\
& \quad y_i \in \{0, 1\}, \quad i = 1, \dots, 10,
\end{aligned} \tag{A.2}$$

$$\begin{aligned}
U_2 : \min & \quad 0.037x_1y_1 + 0.03x_2y_2 + 0.023x_3y_3 + 0.05x_4y_4 \\
& \quad + 0.023x_5y_5 + 0.023x_6y_6 + 0.042x_7y_7 \\
& \quad + 0.023x_8y_8 + 0.05x_9y_9 + 0.05x_{10}y_{10}, \\
\text{subject to} & \quad \sum_{i=1}^{10} x_i y_i = 200, \\
& \quad \sum_{i=1}^{10} y_i \leq 5, \\
& \quad 40y_1 \leq x_1 \leq 50y_1, \quad 53y_2 \leq x_2 \leq 63y_2, \\
& \quad 65y_3 \leq x_3 \leq 75y_3, \quad 45y_4 \leq x_4 \leq 55y_4,
\end{aligned}$$

$$\begin{aligned}
& 40y_5 \leq x_5 \leq 50y_5, \quad 33y_6 \leq x_6 \leq 43y_6 \\
& 28y_7 \leq x_7 \leq 38y_7, \quad 46y_8 \leq x_8 \leq 56y_8, \\
& 55y_9 \leq x_9 \leq 65y_9, \quad 63y_{10} \leq x_{10} \leq 73y_{10} \\
& x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, 10, \\
& y_i \in \{0, 1\}, \quad i = 1, \dots, 10,
\end{aligned} \tag{A.3}$$

$$\begin{aligned}
U_3 : \min & \quad 3.1x_1y_1 + 3.2x_2y_2 + 3.1x_3y_3 + 3.55x_4y_4 \\
& \quad + 3.37x_5y_5 + 3.4x_6y_6 + 3.87x_7y_7 \\
& \quad + 3.35x_8y_8 + 3.59x_9y_9 + 3.41x_{10}y_{10}, \\
\text{subject to} & \quad \sum_{i=1}^{10} x_i y_i = 200, \\
& \quad \sum_{i=1}^{10} y_i \leq 5,
\end{aligned}$$

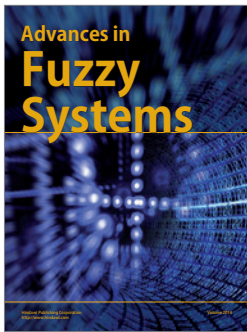
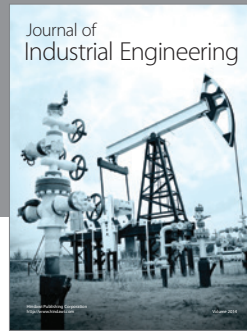
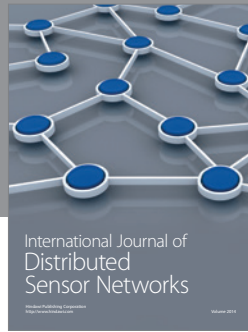
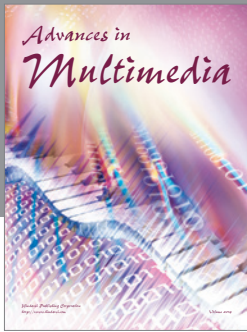
$$\begin{aligned}
& 40y_1 \leq x_1 \leq 50y_1, \quad 53y_2 \leq x_2 \leq 63y_2 \\
& 65y_3 \leq x_3 \leq 75y_3, \quad 45y_4 \leq x_4 \leq 55y_4, \\
& 40y_5 \leq x_5 \leq 50y_5, \quad 33y_6 \leq x_6 \leq 43y_6, \\
& 28y_7 \leq x_7 \leq 38y_7, \quad 46y_8 \leq x_8 \leq 56y_8, \\
& 55y_9 \leq x_9 \leq 65y_9, \quad 63y_{10} \leq x_{10} \leq 73y_{10} \\
& x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, 10, \\
& y_i \in \{0, 1\}, \quad i = 1, \dots, 10,
\end{aligned} \tag{A.4}$$

$$\begin{aligned}
U_4 : \max & \quad 0.83x_1y_1 + 0.84x_2y_2 + 0.825x_3y_3 + 0.85x_4y_4 \\
& \quad + 0.87x_5y_5 + 0.825x_6y_6 + 0.88x_7y_7 \\
& \quad + 0.87x_8y_8 + 0.841x_9y_9 + 0.85x_{10}y_{10}, \\
\text{subject to} & \quad \sum_{i=1}^{10} x_i y_i = 200,
\end{aligned}$$

$$\begin{aligned}
& \sum_{i=1}^{10} y_i \leq 5, \\
& 40y_1 \leq x_1 \leq 50y_1, \quad 53y_2 \leq x_2 \leq 63y_2, \\
& 65y_3 \leq x_3 \leq 75y_3, \quad 45y_4 \leq x_4 \leq 55y_4, \\
& 40y_5 \leq x_5 \leq 50y_5, \quad 33y_6 \leq x_6 \leq 43y_6, \\
& 28y_7 \leq x_7 \leq 38y_7, \quad 46y_8 \leq x_8 \leq 56y_8 \\
& 55y_9 \leq x_9 \leq 65y_9, \quad 63y_{10} \leq x_{10} \leq 73y_{10}, \\
& x_i \in \{0, \mathbf{Z}^+\}, \quad i = 1, \dots, 10, \\
& y_i \in \{0, 1\}, \quad i = 1, \dots, 10.
\end{aligned} \tag{A.5}$$

References

- [1] G. W. Dickson, "An analysis of vendor selection systems and decisions," *Journal of Purchasing*, vol. 2, no. 1, pp. 5–17, 1966.
- [2] C. A. Weber and J. R. Current, "A multiobjective approach to vendor selection," *European Journal of Operational Research*, vol. 68, no. 2, pp. 173–184, 1993.
- [3] S. Yahya and B. Kingsman, "Vendor rating for an entrepreneur development programme: a case study using the analytic hierarchy process method," *Journal of the Operational Research Society*, vol. 50, no. 9, pp. 916–930, 1999.
- [4] S. W. Lam and L. C. Tang, "Multiobjective vendor allocation in multiechelon inventory systems: a spreadsheet model," *Journal of the Operational Research Society*, vol. 57, no. 5, pp. 561–578, 2006.
- [5] Y.-T. Lin, C.-L. Lin, H.-C. Yu, and G.-H. Tzeng, "A novel hybrid MCDM approach for outsourcing vendor selection: a case study for a semiconductor company in Taiwan," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4796–4804, 2010.
- [6] C.-H. Hsu, F.-K. Wang, and G.-H. Tzeng, "The best vendor selection for conducting the recycled material based on a hybrid MCDM model combining DANP with VIKOR," *Resources, Conservation and Recycling*, vol. 66, pp. 95–111, 2012.
- [7] R. Zanjirani Farahani and M. Fadaei, "A MCDM-based model for vendor selection: a case study in the particleboard industry," *Journal of Forestry Research*, vol. 23, no. 4, pp. 685–690, 2012.
- [8] W. Li, X. Zhang, and Y. Chen, "Information integration approach to vendor selection group decision making under multiple criteria," in *Advances in Neural Networks—ISNN 2009*, vol. 5551 of *Lecture Notes in Computer Science*, pp. 1138–1143, 2009.
- [9] H. Zhang, X. Li, and W. Liu, "An AHP/DEA methodology for 3PL vendor selection in 4PL," in *Computer Supported Cooperative Work in Design II*, vol. 3865 of *Lecture Notes in Computer Science*, pp. 646–655, 2006.
- [10] N. Arunkumar, L. Karunamoorthy, S. Anand, and T. Ramesh Babu, "Linear approach for solving a piecewise linear vendor selection problem of quantity discounts using lexicographic method," *International Journal of Advanced Manufacturing Technology*, vol. 28, no. 11–12, pp. 1254–1260, 2006.
- [11] S. C. H. Leung, Y. Wu, and K. K. Lai, "A stochastic programming approach for multi-site aggregate production planning," *Journal of the Operational Research Society*, vol. 57, no. 2, pp. 123–132, 2006.
- [12] S. Talluri, R. Narasimhan, and A. Nair, "Vendor performance with supply risk: a chance-constrained DEA approach," *International Journal of Production Economics*, vol. 100, no. 2, pp. 212–222, 2006.
- [13] J. Xu and C. Ding, "A class of chance constrained multiobjective linear programming with birandom coefficients and its application to vendors selection," *International Journal of Production Economics*, vol. 131, no. 2, pp. 709–720, 2011.
- [14] G. Zhimin, J. Zhihong, and Z. Baogang, "A multi-objective mixed-integer stochastic programming model for the vendor selection problem under multi-product purchases," *International Journal of Information and Management Sciences*, vol. 18, no. 3, pp. 241–252, 2007.
- [15] R. G. Kasilingam and C. P. Lee, "Selection of vendors—a mixed-integer programming approach," *Computers and Industrial Engineering*, vol. 31, no. 1–2, pp. 347–350, 1996.
- [16] A. Alonso-Ayuso, L. F. Escudero, A. Garín, M. T. Ortuño, and G. Pérez, "An approach for strategic supply chain planning under uncertainty based on stochastic 0-1 programming," *Journal of Global Optimization*, vol. 26, no. 1, pp. 97–124, 2003.
- [17] W. Zang, Y. Liu, and Z. Li, "Optimizing supplier selection with disruptions by chance-constrained programming," in *Advances in Swarm Intelligence*, vol. 7332 of *Lecture Notes in Computer Science*, pp. 108–116, 2012.
- [18] B. B. Keskin, H. Ster, and S. Etinkaya, "Integration of strategic and tactical decisions for vendor selection under capacity constraints," *Computers and Operations Research*, vol. 37, no. 12, pp. 2182–2191, 2010.
- [19] S. He, S. S. Chaudhry, Z. Lei, and W. Baohua, "Stochastic vendor selection problem: chance-constrained model and genetic algorithms," *Annals of Operations Research*, vol. 168, no. 1, pp. 169–179, 2009.
- [20] A. A. Taleizadeh, S. T. A. Niaki, and F. Barzinpour, "Multiple-buyer multiple-vendor multi-product multi-constraint supply chain problem with stochastic demand and variable lead-time: a harmony search algorithm," *Applied Mathematics and Computation*, vol. 217, no. 22, pp. 9234–9253, 2011.
- [21] B. Huang, C. Gao, and L. Chen, "Partner selection in a virtual enterprise under uncertain information about candidates," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11305–11310, 2011.
- [22] R. J. Kuo, S. Y. Hong, and Y. C. Huang, "Integration of particle swarm optimization-based fuzzy neural network and artificial neural network for supplier selection," *Applied Mathematical Modelling*, vol. 34, no. 12, pp. 3976–3990, 2010.
- [23] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
- [24] M. Ekhtiari and K. Ghoseiri, "Multi-objective stochastic programming to solve manpower allocation problem," *International Journal of Advanced Manufacturing Technology*, vol. 65, no. 1, pp. 183–196, 2013.
- [25] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*, Springer, Berlin, Germany, 2008.
- [26] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [27] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth WA, Australia, December 1995.
- [28] C.-J. Liao, E. Tjandradjaja, and T.-P. Chung, "An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem," *Applied Soft Computing Journal*, vol. 12, no. 6, pp. 1755–1764, 2012.
- [29] M. Clerc, *Particle Swarm Optimization*, ISTE, London, UK, 2006.
- [30] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, 2005.
- [31] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 4661–4667, Singapore, September 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

