STOCHASTIC VENDOR SELECTION PROBLEM

Baohua Wang^(a), Shiwei He^(b), Sohail S. Chaudhry^(c)

^(a)IBM China Research Lab, Beijing, China 100094

^(b)Department of Transportation Management Engineering, College of Traffic and Transportation, Beijing Jiaotong

University, Beijing, China 100044

^(c)Department of Management and Operations/International Business, Villanova School of Business, Villanova University, Villanova, PA USA 19085

^(a) <u>baohuaw@cn.ibm.com</u>, ^(b) <u>shwhe@bjtu.edu.cn</u>, ^(c) <u>sohail.chaudhry@villanova.edu</u>

ABSTRACT

We study the vendor selection problem in which capacity, quality level, service level, and lead time associated with each vendor are considered to be stochastic. The problem is modeled as a stochastic dependent-chance programming model. As stochastic programming models are difficult to solve by traditional methods, a hybrid adaptive genetic algorithm, which embeds the neural network and stochastic simulation, was designed and implemented. To further improve the performance of the algorithm, the adaptive genetic algorithm was adjusted by varying the crossover probability and mutation rate according to the stage of evolution and fitness of the population. The solution procedure was tested on several randomly generated problems with varying parameters. Our extensive computational experience on these problems indicates that the hybrid adaptive genetic algorithm has strong adaptability on the tested problems as the algorithm converged more rapidly than the simple genetic algorithm.

Keywords: Vendor selection problem, stochastic dependent-chance programming, genetic algorithm

1. INTRODUCTION

For most manufacturing firms, the purchasing of raw materials and component parts from outside vendors constitutes a major expense. In a vendor selection problem (aka the supplier selection problem in literature), the purchasing manager must choose order quantities to place among vendors in a multi-sourcing network to satisfy the demand of the firms under a variety of conditions. Generally, the selected vendors need to be evaluated on more than a single criterion. Dickson (1966) studied the vendor selection problem and reported that there are 23 factors that are important to purchasing managers when selecting vendors whereas Dempsey (1978) identified 18 criteria. Among these criteria, it was found that price, delivery, quality, and capacity were ranked at the top of the list of purchasing managers when they selected vendors. A similar conclusion was reported by Weber et al. (1991)

using a review of 74 related papers for the vendor selection problem.

The focus of this paper is to analyze the vendor selection problem under stochastic environment. We propose a stochastic dependent-chance programming model which aims to maximize the probability that the demand of the firm can be satisfied while minimizing the expected cost. Since stochastic programming model is hard to solve by traditional methods, a hybrid intelligent algorithm, which integrates stochastic simulation, neural network into genetic algorithm, is designed to solve the problem.

The paper is organized as follows. In the next section, a literature review is presented. In the following section, we present some basic concepts related to dependent-chance programming based on Liu and the stochastic dependent-chance (1997)programming model for vendor selection problem. This is followed by design of the hybrid genetic algorithm for the model. To further improve the performance of the algorithm, an adaptive genetic algorithm is also presented in this section. In the penultimate section, we present the application of the hybrid genetic algorithm to a series of randomly generated problem instances. The conclusions and future research direction are presented in the final section.

2. LITERATURE REVIEW

During the past 50 years or more, many different methods have been proposed to solve a variety of different the vendor selection problems. These methods can be grouped into whether the technique focuses on qualitative or quantitative factors that are relevant in the vendor selection problem. However, in the recent years, researchers have developed solution approaches that based are based on two or more of these methodologies.

Wind and Robinson (1968), Mazurak et al. (1985), Cooper (1977) and others have used a weighted linear method of multiple criteria for this problem. Timmerman (1986) and Gregory (1986) linked this approach to a matrix representation of data and Narasimhan (1983) employed the analytical hierarchical process to generate weights for such models. Recently, Micheli (2008) investigated supplier selection problem as a way to mitigate the overall supply risk. A risk efficiency-based supplier selection approach was developed for critical supplies that allowed a decision maker to consider the procurement-related "risk" and "investment" for mitigation/exploitation interventions. Kirytopoulos et al. (2008) analyzed the supply chain processes within the pharmaceutical industry in Greece in which analytic network process based method for the selection was used to solve the problem. Saen (2008) addressed the supplier ranking in a volume discount environment and introduced an innovative approach which was based on the super-efficiency analysis.

Besides qualitative methods, there is an abundance of published research that utilizes quantitative methods, which may also be integrated with qualitative methods to solve this problem. The earliest papers that utilized quantitative methods to solve the vendor selection can be traced back to Stanley et al. (1954) and Gainen (1955) in which linear programming was used for awarding contracts to contractors at the Department of Defense. Bender et al. (1985) proposed a mixed integer programming (MIP) model for vendor selection problem. This approach was used at IBM to select vendors and their order quantities with the objective to minimize purchasing, inventory, and transportation costs; however, the specific mathematical formulation was not presented. Pan (1989) developed a single item linear programming model to allocate order quantities among suppliers in which the objective was to minimize aggregate price that was restricted by the constraints on quality, service level, and lead-time. Sharma et al. (1989) suggested a goal programming formulation that considered price, quality and lead-time goals with demand and budget constraints.

Chaudhry et al. (1993), Degraeve et al. (2000), De Boer et al. (2001), and Aissaoui et al. (2007) also have provided a well-structured literature survey on the application of different techniques to the vendor selection problem. More recently, Chen and Huang (2007) related product characteristics to supply chain strategy and adopted supply chain operations reference (SCOR) performance metrics as the decision criteria. A scheme integrated analytic hierarchy process (AHP) with bi-negotiation agents based on the multi-criteria decision-making approach and software agent techniqueis then developed to take into account both qualitative and quantitative attributes in supplier selection. Ting and Cho (2008) developed a two-step decision-making procedure utilizing analytic hierarchy process and multi-objective linear programming in which analytic hierarchy process was used for select candidate supplier and multi-objective linear programming was used to allocate the quantities among the selected suppliers

However, in solving practical vendor selection and purchasing plans, businesses are faced with some uncertain factors. For example, the quantity supplied by the vendors, the quality level, and the service level of the vendors sometimes can be considered to be random variables with known distribution function. Kasilingam and Lee (1996) considered the stochastic nature of demand and propose a mixed-integer programming model to select vendors and determine the order quantities. Shiromaru et al. (2000) treated coal purchase planning in a real electric power plant and applied a fuzzy satisfying method to deal with the vagueness of the goals. Kumar et al. (2005) presented a fuzzy multi-objective integer programming model and discussed the corresponding crisp equivalence for optimization.

In a recent work, Rezaei and Davoodi (2006) formulated a fuzzy mixed integer programming model of a multi-period inventory lot sizing problem with supplier selection. Amid et al. (2006) firstly developed a fuzzy multiobjective model in which different weights can be considered for various objectives. This fuzzy model enabled the purchasing managers not only to consider the imprecision of information but also take into consideration the limitations of buyer and supplier into account in order to calculate the order quantity assigned to each supplier. Liao and Rittscher (2006) considered demand quantities and timing the uncertainties into consideration and proposed a multiobjective supplier selection model. A genetic algorithm was utilized to handle this model. Sevkli et al. (2007) proposed an analytical hierarchy process weighted fuzzy linear programming model for supplier selection and compared this new model with the classical analytic hierarchy process. Amid et al. (2007) developed a fuzzy multi-objective model for the supplier selection problem under price breaks and presented a weighted additive method to generate an optimal solution in the fuzzy environment. Chan et al. (2008) discussed the fuzzy based analytic hierarchy process to efficiently tackle both quantitative and qualitative decision factors involved in selection of global supplier in current Olson Wu and business scenario. (2008)considered three of vendor selection types methodologies in supply chains with risk, which are chance constrained programming, data envelopment analysis, and multi-objective programming models. The Monte-Carlo simulation was applied to these three methodologies. He et al. (2008) developed a class of special stochastic chance-constrained programming models and presented a genetic algorithm for vendor selection problem under stochastic environment.

3. STOCHASTIC DEPENDENT-CHANCE PROGRAMMING MODEL FOR VENDOR SELECTION PROBLEM

In practice, the decision-maker may want to maximize the chance functions of some events (i.e., the probabilities of satisfying the events) under stochastic environment. In order to model this type of stochastic decision system, Liu (1999) provided a new type of stochastic programming, called dependent-chance programming. Dependent-chance programming involves maximizing chance functions of events in an uncertain environment.

According to Liu (1999), an uncertain environment signifies the following stochastic constraint, $g_i(x,\xi) \le 0$, $j = 1, 2, 3 \cdots p$, where x is a decision vector, and ξ is a stochastic vector. event Also, let an be represented by $h_{k}(x,\xi) \leq 0, k = 1, 2, 3 \cdots q$, whose chance function is defined as the probability measure of the event, $f(x) = \Pr\{h_k(x,\xi) \le 0, k = 1, 2, 3 \cdots q\}$, subject to the uncertain environment as defined above.

In deterministic model, expected value model, and chance-constrained programming, the feasible set is essentially assumed to be deterministic after the real problem is modeled. That is, an optimal solution is given regardless of whether it can be performed in practice. However, the given solution may be impossible to perform if the realization of uncertain parameter is unfavorable. Thus dependent-chance programming theory never assumes that the feasible set is deterministic. In fact, it is constructed in an uncertain environment.

Formally, a typical dependent-chance programming model can be represented as maximizing the chance function of an event subject to an uncertain environment in the following way:

$$\max_{k \in \mathbb{N}} \Pr \left\{ h_{k} \left(x, \xi \right) \le 0, k = 1, 2, 3 \dots q \right\}$$

s.t.
$$g_{j} \left(x, \xi \right) \le 0, j = 1, 2, 3 \dots p,$$

where x is an n-dimensional decision vector, ξ is a random vector of parameters, the system $h_k(x,\xi) \le 0, k = 1,2,3\cdots q$, represents an event, and the constraints $g_j(x,\xi) \le 0, j = 1,2,3\cdots p$ are an uncertain environment.

3.1. Notation and mathematical model

Let the decision variable x_i represent the percentage of the quantity to be ordered from vendor *i*. In addition, let the parameters be defined as:

D Total demand of the item;

$$Ω$$
 Set of vendors competing for selection, $Ω = \{1, 2, 3 \cdots N\}, i ∈ Ω;$

 C_i Unit cost of purchasing plus transportation from vendor i;

d Unit cost due to receiving poor quality items;

e Unit cost due to receiving late delivered items;

 ξ_i Upper limit of the quantity available for vendor i, random variable;

 λ_i Percentage of good items supplied by vendor i, random variable;

 η_i Percentage of items receiving good after service offered by vendor *i*, random variable;

 τ_i Percentage of the late delivered items by the vendor i, random variable;

S Minimum allowable aggregate quantity of items receiving good after service (Required service level);

L Maximum allowable aggregate quantity of late delivered items (Required lead-time level);

W Minimum allowable aggregate quantity of good items (Required quality level);

B Budget constraint.

In our model we assume that quantity discounts are not allowed. There is only one item to be considered. However, multi-item vendor selection problem can be simplified into several single-item vendor selection problems. The maximum number of vendors which can be selected is not restricted. Finally, all the random variables are independent.

3.2. Dependent-chance programming model

Given the definitions, assumptions and notations above, the vendor selection problem can be formulated as the following dependent-chance integer goal programming model.

1) Constraints

$$D \cdot x_i \le \xi_i \text{ for all } i$$
 (1)

. .

Constraint (1) puts restrictions due to the maximum capacity of the vendors.

$$\sum_{i} (D \cdot x_{i}) \cdot \lambda_{i} \ge W$$
(2)

Constraint (2) means that the required quality level should be achieved.

$$\sum_{i} (D \cdot x_i) \tau_i \le L \tag{3}$$

Constraint (3) means that the required lead-time level should be achieved.

$$\sum_{i} (D \cdot x_i) \eta_i \ge S \tag{4}$$

Constraint (4) means that the required service level should be achieved.

$$\sum_{i} (D \cdot x_i) \cdot c_i + (D \cdot x_i) \cdot (1 - \lambda_i) \cdot d + (D \cdot x_i) \cdot e \cdot \tau_i \le B \quad (5)$$

Constraint (5) puts restrictions on the budget.

$$x_i \ge 0$$
 for all i (6)

Constraint (6) ensures the non-negativity of the solution.

2) Objective Function

$$\max \Pr\left\{\sum_{i} x_{i} = 1\right\}$$
(7)

min
$$E\left(\sum_{i} (D \cdot x_{i}) \cdot c_{i} + (D \cdot x_{i}) \cdot d \cdot (1 - \lambda_{i}) + (D \cdot x_{i}) \cdot e \cdot \tau_{i}\right)$$
 (8)

The objectives are to maximize the probability that the demand can be satisfied and minimize the total expected cost under stochastic environment, which is characterized by constraint (1) – (6). The probability can also be considered to the reliability, or the risk, of the purchasing plan. The reason why we take the total expected cost into consideration is that with the same probability there can be more than a single purchasing plan and the decision-maker would like to know which one is the best. We define μ as the weight coefficient. We also assume the priority of objective (7) is higher than objective (8). So μ should be a sufficiently large positive number. The two objectives can be integrated into one objective by the following equation:

$$\max \mu \cdot \Pr\left\{\sum_{i} x_{i} = 1\right\} - E\left(\sum_{i} (D \cdot x_{i}) \cdot c_{i} + (D \cdot x_{i}) \cdot d \cdot (1 - \lambda_{i}) + (D \cdot x_{i}) \cdot e \cdot \tau_{i}\right)$$
(9)

When some management targets are given, the objective function may minimize the deviations, positive, negative, or both, with a certain priority structure set by the decision-maker. In this paper, if we let α denote the chance of meeting the demand given by the decision-maker, then the objective function of the dependent-chance goal programming model can be formulated as follow, where d^- and d_2^+ is to be minimized:

$$\min \left\{ d_{1}^{-}, d_{2}^{+} \right\}$$

$$\Pr \left\{ \sum_{i} x_{i} = 1 \right\} + d_{1}^{-} - d_{2}^{+} = \alpha \qquad (10)$$

$$E \left(\sum_{i} (D \cdot x_{i}) \cdot c_{i} + (D \cdot x_{i}) \cdot d \cdot (1 - \lambda_{i}) + (D \cdot x_{i}) \cdot e \cdot \tau_{i} \right) + d_{2}^{-} - d_{2}^{+} = B \qquad (11)$$

4. HYBRID GENETIC ALGORITHM

Generally speaking, stochastic programming models are difficult to solve by traditional methods. It has been shown that a good way to solve these difficult problems is to design hybrid intelligent algorithms (Liu and Iwamura 1997, Liu 1997, Liu 2000). In this section, we integrate the neural network, stochastic simulation, and genetic algorithm to produce a hybrid intelligent algorithm for solving stochastic dependent-chance programming models of vendor selection problem, which is formulated by equations (1) - (7).

According to Liu (1997)'s study, equation (7) is equivalent to the uncertain function as follow:

$$f_{1}(x) = \Pr \begin{cases} \sum_{i} x_{i} = 1 \\ D \cdot x_{i} \leq \xi_{i} \text{ for all } i \\ \sum_{i} (D \cdot x_{i}) \cdot \lambda_{i} \geq W \\ \sum_{i} (D \cdot x_{i}) \cdot \tau_{i} \leq L \\ \sum_{i} (D \cdot x_{i}) \cdot \eta_{i} \geq S \\ \sum_{i} (D \cdot x_{i}) \cdot c_{i} + (D \cdot x_{i}) \cdot d \cdot (1 - \lambda_{i}) + (D \cdot x_{i}) \cdot e \cdot \tau_{i} \leq B \\ x_{i} \geq 0 \text{ for all } i \end{cases}$$

Equation (8) is equivalent to the uncertain function as follow:

$$f_2(x) = E\left[\sum_i (D \cdot x_i) \cdot c_i + (D \cdot x_i) \cdot d \cdot (1 - \lambda_i) + (D \cdot x_i) \cdot e \cdot \tau_i\right]$$

Given a certain

$$\mathbf{x} = \left\{ \left(x_1, x_2 \cdots x_N \right) \middle| \sum_i x_i = 1, x_i \ge 0 \text{ for all } i \right\}, \text{ the}$$

value of $f_1(x)$ and $f_2(x)$ may be estimated by the following stochastic simulation.

4.1. Algorithm (Stochastic Simulation)

Step 1. Set N' = 0, Cost = 0.

Step 2. Generate ξ_i , λ_i , τ_i , η_i according to the their distribution function.

Step 3. $Cost = Cost + E[\bullet]$. If constraints (2) – (5) can be satisfied, then N' + +.

Step 4. Repeat the second to fourth steps for N times, where N is a sufficiently large number.

Step 5. $f_1(x) = N' / N$, $f_2(x) = Cost / N$.

Although stochastic simulations are able to compute the chance functions, we need relatively simple functions to approximate the uncertain functions because the stochastic simulations are a time-consuming process. In order to speed up the solution process, a neural network is employed to approximate the chance functions since the neural network has the ability to approximate the uncertain functions by using the training data, it can compensate for the error of training data (all input-output data obtained by stochastic simulation are clearly not precise), and has the high speed of operation after they are trained. Hence, the hybrid genetic algorithm is presented next.

4.2. Algorithm (Hybrid Genetic Algorithm)

Step 1. Generate training input-output data for the chance function $f_1(x)$ and $f_2(x)$ by stochastic simulation (Algorithm 4.1).

Step 2. Train a neural network to approximate the chance function according to the generated training data.

Step 3. In the paper, we use the floating vector to represent a solution in which each chromosome vector is coded as a vector of floating numbers, of the same length as the solution vector. Let $V = (x_1, x_2 \cdots x_N)$ be the chromosome representing the solution $x = (x_1, x_2 \cdots x_N)$. We assume all the vendors have the same priority. Then the chromosomes should be initialized by the following manner.

Step 3.1 Define Total = 0;

Step 3.2 Choose a vendor *i* randomly. The quantity purchased from vendor *i*, x_i , is initialized by generating a random number *q* in $(0, E(\xi_i))$.

Step 3.3 If $x_i = 0$, $x_i = x_i + q$. Otherwise, if $x_i + q > E(\xi_i)$, $x_i = E(\xi_i)$;

Step $3.4 Total = Total + x_i$. If Total > D, $x_i = x_i - (Total - D)$; Otherwise, if

Total < D, go to Step 3.2.

Repeat the algorithm above *pop_size* times, we can obtain *pop_size* chromosomes.

Step 4. Compute the fitness of all chromosomes $V_k, k = 1, 2 \cdots pop_size$ by the trained neural network to rearrange them from best to worse according to their objective function values.

Step 5. Select the chromosomes by spinning the roulette wheel.

Step 6. Renew the chromosomes $V_k, k = 1, 2 \cdots pop_size$ by crossover operation.

We define a parameter P_c of a genetic system as the probability of crossover. This probability gives us the expected number $P_c \cdot pop_size$ of chromosomes undergoing the crossover operation. In order to determine the parents for crossover operation, we can generate a random real number r from the interval [0,1]. If $r < P_c$, the chromosome V_i is selected as a parent.

We denote the selected parents by $V_1^{'}, V_2^{'}, V_3^{'}, ...$ and divide them into the following pairs: $((V_1^{'}, V_2^{'}), (V_3^{'}, V_4^{'}), (V_5^{'}, V_6^{'})$ Let us illustrate the crossover operator on each pair by using the pair $(V_1^{'}, V_2^{'})$. Initially, a random number *c* is generated from the open interval (0,1). Then, the crossover operator on V_1' and V_2' will produce two children X and Y as follows:

$$X = c \cdot V_1' + (1 - c) \cdot V_2', \quad Y = (1 - c) \cdot V_1' + c \cdot V_2'$$

We must check the feasibility of each child before accepting it, and only replace the parents with the feasible children.

Step 7. Update the chromosomes

 $V_k, k = 1, 2 \cdots pop_size$ by mutation operation.

We define a parameter P_m of a genetic system as the probability of mutation. This probability gives us the expected number of $P_m \cdot pop_size$ of chromosomes undergoing the mutation operations. The mutation operation will be carried out as the following manner, which is similar to Gaussian Mutation.

For each selected parent, denoted by $V = (x_1, x_2, ..., x_N)$, we randomly generated N real positive numbers, $r_1, r_2 \cdots r_N$, with the distribution $U(0, x_i)$.

Step 7.1 Randomly choose x_i for mutation.

Step 7.2 Randomly generated a number ω from (0,1).

If $\omega > 0.5$, $x_i = x_i + r_i$; else, $x_i = x_i - r_i$;

Step 7.3 If $\sum_{i} x_i > 1$, adjust x_i to make sure $\sum x_i = 1$. Then the mutation operation is

over; else, go to step 7.1.

Step 8. Repeat the third to sixth steps for a given number of cycles.

Step 9. Report the best chromosome as the optimal solution.

Based on the algorithm above, an adaptive genetic algorithm (AGA), in which the probability of the crossover and mutation operation will be adjusted according to the stage of evolution and fitness of the population, is used to improve the performance. In AGA, we define f_{avg} as the average fitness of the population, f as the fitness of the chromosome, $P_{c-\max}$ as the maximum probability of crossover, $P_{c-\min}$ as the minimum probability of crossover, $P_{m-\max}$ as the minimum probability of mutation, $P_{m-\min}$ as the minimum probability of mutation, MaxGen as the maximum generation of the algorithm, gen is the current generation of the algorithm.

So in every generation, the probability of crossover and mutation can be obtained by following equations:

$$P_{c} = \begin{cases} P_{c-\max} - \left(\frac{P_{c-\max} - P_{c-\min}}{MaxGen}\right) \bullet gen, f > f_{avg} \\ P_{c-\max}, f \le f_{avg} \end{cases}$$
(12)
$$P_{m} = \begin{cases} P_{m-\max} - \left(\frac{P_{m-\max} - P_{m-\min}}{MaxGen}\right) \bullet gen, f > f_{avg} \\ P_{m-\max}, f \le f_{avg} \end{cases}$$
(13)

5. COMPUTATIONAL EXPERIENCE AND DISCUSSION

In this section, we apply the hybrid genetic algorithm to a series of instances of the stochastic dependent-chance programming model for the allocation of order quantity among vendors. All computational analysis was performed on an AMD Turion 1.7 GHz notepad and the algorithm code is implemented in C++.

Please note that due to space limitations of 6 pages for a regular paper, the results and discussion as well as additional references can be requested from the corresponding author, <u>sohail.chaudhry@villanova.edu</u>.

REFERENCES

- Aissaoui, N., Haouari, M. and Hassini, E., 2007. Supplier selection and order lot sizing modeling: a review. *Computers and Operations Research*, 34(12), 3516-3540.
- Amid A., Ghodsypour, S.H. and O'Brien C., 2006. Fuzzy multiobjective linear model for supplier selection in a supply chain. *International Journal* of Production Economics, 104(2), 394-407.
- Amid A., Ghodsypour, S.H. and O'Brien C., 2007. A weighted additive fuzzy multiobjective model for the supplier selection problem under price breaks in a supply Chain. *International Journal of Production Economics*, In press, doi:10.1016/j.ijpe.2007.02.040.
- Bender, P.S., Brown, R.W., Issac, M.H. and Shapiro, J.F., 1985. Improving purchasing productivity at IBM with a normative decision support system. *Interfaces*, 15(3), 106–115.
- Charnes, A. and Cooper, W.W., 1959. Chanceconstrained programming. *Management Science*, 6(1), 73-79.
- Chan, F.T.S., Kumar, N., Tiwari, M.K., Lau, H.C.W. and Choy, K.L., 2008. Global supplier selection: a fuzzy-AHP approach. *International Journal of Production Research*, 46(14), 3825-2857.
- Chaudhry, S.S., Forst, F.G. and Zydiak, J.L., 1993. Vendor selection with price breaks. *European Journal of Operational Research*, 70(1), 52-66.
- Chen Y.M. and Huang, P.N., 2007. Bi-negotiation integrated AHP in supplier selection. *International*

Journal of Operations & Production Management, 27(11), 1254 - 1274.

- Crama, Y., R. Pascual J. and Torres, A., 2004. Optimal procurement decisions in the presence of total quantity discounts and alternative product recipes. *European Journal of Operational Research*, 159(2), 364–378.
- Cooper, S.D., 1977. A total system for measuring of performance. *Journal of Purchasing and Materials Management*, 13(3), 22–26.
- De Boer, L. and van der Wegen, L.L.M., 2003. Practice and promise of formal supplier selection: a study of four empirical cases. *Journal of Purchasing & Supply Management*, 9(3), 109-118.
- Degraeve, Z., Labro, E. and Roodhooft, F., 2000. An Evaluation of Vendor Selection Models from a Total Cost of Ownership Perspective. *European Journal of Operational Research*, 125(1), 34-58.
- Dempsey, W.A., 1978. Vendor selection and the buying process. *Industrial Marketing Management*, 7(4), 257–267.
- Dickson, G.W., 1966. An analysis of vendor selection systems and decisions. *Journal of Purchasing*, 2(1), 5-17.
- Gainen, L., 1955. Linear programming in bid evaluations. In: Proceedings of the 2nd Symposium of Linear Programming, Washington DC, 29-38.
- Gregory, R.E., 1986, Source selection: a matrix approach. Journal of Purchasing and Materials Management, 22(2), 24–29.
- Sevkli, M., Koh, S.C.L., Zaim, S., Demirbag, M. and Tatoglu, E., 2008. Hybrid analytical hierarchy process model for supplier selection. *Industrial Management & Data Systems*, 108(1), 122-142.
- He, S., Chaudhry, S.S., Zhonglin, L. and Wang, B., 2008. Stochastic vendor selection problem: chance-constrained model and genetic algorithms. *Annals of Operation Research*, In press, doi:10.1007/s10479-008-0367-5
- Kasilingam, R.G. and Lee, C.P., 1996. Selection of vendors - a mixed-integer programming approach. *Computers & Industrial Engineering*, 31(1-2), 347-350.
- Kumar, M., Vrat, P. and Shankar, R., 2006. A fuzzy programming approach for vendor selection problem in a supply chain. *International Journal of Production Economics*, 101(2), 273-285.
- Liao, Z. and Rittscher, J., 2007. A multi-objective supplier selection model under stochastic demand conditions. *International Journal of Production Economics*, 105(1), 150-159.
- Kirytopoulos, K., Leopoulos, V. and Voulgaridou, D., 2008. Supplier selection in pharmaceutical industry: an analytic network process approach. *Benchmarking: An International Journal*, 15(4), 494-516.
- Liu, B., 1997. Dependent-chance programming: a class of stochastic optimization. *Computers & Mathematics with Applications*, 34(12), 89-104.