

A fuzzy-random programming for integrated closed-loop logistics network design by using priority-based genetic algorithm

Emad Roghanian and Keyvan Kamandanipour*

Department of industrial engineering, K.N Toosi University of Technology, Tehran, Iran

ARTICLE INFO

Article history:

Received 18 July 2012
Received in revised format
14 August 2012
Accepted September 10 2012
Available online
15 September 2012

Keywords:

Integrated logistics network
Closed-loop logistics
Genetic algorithm (GA)
Priority-based encoding
Fuzzy-random programming

ABSTRACT

Recovery of used products has steadily become interesting issue for research due to economic reasons and growing environmental or legislative concern. This paper presents a closed-loop logistics network design based on reverse logistics models. A mixed integer linear programming model is implemented to integrate logistics network design in order to prevent the sub-optimality caused by the separate design of the forward and reverse networks. The study presents a single product and multi-stage logistics network problem for the new and return products not only to determine subsets of logistics centers to be opened, but also to determine transportation strategy, which satisfies demand imposed by facilities and minimizes fixed opening and total shipping costs. Since the deterministic estimation of some parameters such as demand and rate of return of used products in closed loop logistics models is impractical, an uncertain programming is proposed. In this case, we assume there are several economic conditions with predefined probabilities calculated from historical data. Then by means of expert's opinion, a fuzzy variable is offered as customer's demand under each economic condition. In addition, demand and rate of return of products for each customer zone is presented by fuzzy-random variables, similarly. Therefore, a fuzzy-random programming is used and a priority-based genetic algorithm is proposed to solve large-scale problems.

© 2012 Growing Science Ltd. All rights reserved

1. Introduction and literature review

1.1. Closed loop logistics network design

During the past few decades, there has been increasing growing interests on final products or after sales services for a variety of economic, environmental or legislative reasons. Reverse logistics is the process of moving goods from their typical final destination to another point, for the purpose of capturing value or for the proper disposal of the products (Du & Evans, 2008). According to the American Reverse Logistics Executive Council, Reverse Logistics is defined as: “The process of planning, implementing, and controlling the efficient, cost effective flow of raw materials, in-process inventory, finished goods and related information from the point of consumption to the point of origin for the purpose of recapturing value or proper disposal.” A reverse logistics system comprises a series of activities, which

* Corresponding author. +98 21 88192570
E-mail: k1.kamandanipour@gmail.com (K. Kamandanipour)

form a continuous process to treat return-product items until they are properly recovered or disposed. These activities include collection, cleaning, disassembly, test and sorting, storage, transport, and recovery operations. The latter can also be represented as one or a combination of several main recovery options, like reuse, repair, refurbishing, remanufacturing, cannibalization and recycling (Lu & Bostel, 2007). Reverse logistics is getting increasingly important as a profitable and sustainable business strategy. There are a number of situations for products to be placed in a reverse flow. Normally, return flows are classified into commercial, warranty and end-of-use returns, reusable container returns, etc. (Du & Evans, 2008). Implementation of reverse logistics especially in product returns not only saves inventory carrying, transportation, and waste disposal costs due to returned products, but also it improves customer loyalty and futures sales (Ko & Evans, 2007). Reverse logistics systems are more complex than forward logistics systems, which stems from a high degree of uncertainty due to the quantity and quality of the products (Kara & Rugrungruang, 2007). Reverse logistics has attracted much attention recently due to growing environmental or legislative concerns and economic opportunities for cost savings or revenues from returned products.

Barros et al. (1998) proposed a mixed integer linear programming (MILP) model based on a multi-level capacitated warehouse location problem for sand and proposed a heuristic method to solve the resulted model. The model determined the optimal number, capacities, and locations of the depots and cleaning facilities. Kirkke et al. (1998) presented a MILP model based on a multi-level uncapacitated warehouse location model. They presented a case study, which deals with a reverse logistics network for the returns, processing, and recovery of discarded copiers. The model was used to determine the locations and capacities of the recovery facilities as well as the transportation links connecting various locations. Jayaraman et al. (1999) proposed another MILP model to determine optimal numbers and locations of distribution/remanufacturing facilities for electronic equipment.

Jayaraman et al. (2003) developed the other MILP model and solution procedure for a reverse distribution problem focused on the strategic level. The model determines whether each remanufacturing facility is open considering the product return flow. Min et al. (2005) proposed a Lagrangian relaxation heuristics to design the multi-commodity, multi-echelon reverse logistics network. Kim et al. (2006) proposed a general framework for remanufacturing environment and a mathematical model to maximize the total cost saving. The model determines the quantity of products/parts processed in the remanufacturing facilities/subcontractors and the amount of parts purchased from the external suppliers while maximizing the total remanufacturing cost saving. Min et al. (2006) proposed a nonlinear mixed integer programming model and a genetic algorithm, which could solve the reverse logistics problem involving product returns. Their study proposed a mathematical model and GA, which aims to provide a minimum-cost solution for the reverse logistics network design problem involving product returns.

Ko and Evans (2007) presented a mixed integer nonlinear programming model for the design of a dynamic integrated distribution network to account for the integrated aspect of optimizing the forward and return network, simultaneously. They proposed a genetic algorithm-based heuristic for solving this problem. Lee et al. (2009) proposed a multi-stage, multi-product, MILP model for minimizing the total of costs to reverse logistics shipping and fixed opening cost of facilities. They also proposed a hybrid genetic algorithm for solving this problem. Lee and Dong (2008) developed a MILP model for integrated logistics network design for computer products. They considered a network with a given number of hybrid distribution-collection facilities to be opened and a single production center. Pishvae et al. (2009) developed a stochastic programming model for an integrated forward/reverse logistics network design under several uncertain scenarios. They assumed some hybrid facilities to reduce implementation costs. Later Pishvae et al. (2010) developed their last works by a bi-objective mixed integer programming formulation to minimize the total costs and maximize the responsiveness of a logistics network. They used a new dynamic search strategy by employing three different local searches.

1.2. Hybrid programming

Uncertainty is one of the main characteristics of reverse logistics network problem, which further increases the complexity of the problem. The degree of uncertainty in terms of the capacities, demands and quantity of products exists in reverse logistics parameters. An important issue, when manufacturing centers' demands as well as recycling centers' demand are random variables in reverse logistics network design problem, is to find the network strategy, which could achieve the objective of minimization of total shipping cost and fixed opening expenditures of the disassembly and the processing centers. This paper proposes a probabilistic MILP model for the design of a reverse logistics network. This probabilistic model is first converted into an equivalent deterministic model.

There are some mathematical programming which deals with the theory and methods for the solution of conditional extremum problems under incomplete information about the random parameters called "stochastic programming". Fuzziness, on the other hand, is associated with the unsharp boundaries of the parameters of the model, and can be traced to sources of uncertainty such as modeling choices, parameter choices, application of expertise, boundary conditions, and lack of knowledge. Thus, randomness is more an instrument of a normative analysis, which focuses on the future and fuzziness and it is more a tool for a descriptive analysis to reflect the past and its implications. Clearly, randomness and fuzziness are complementary. One important aspect of this relationship is the fuzzy-random variable (FRV), which is a measurable function from a probability space to the set of fuzzy variables (Shapiro, 2009). A simple type of programming, which simultaneously deals with random and fuzzy nature of the uncertain models is called "hybrid programming" and the special type of this programming, which solves the resulted model including FRVs is "fuzzy-random programming".

Unfortunately, the SCN design problem is subject to many sources of uncertainty besides random uncertainty and fuzzy uncertainty (Huang, 2006) and in many practical decision-making issues, we deal with a hybrid uncertain environment. To deal with twofold uncertainties, fuzzy random variable was proposed by Kwakernaak (1978) to depict the phenomena in which fuzziness and randomness appear, simultaneously. As mentioned earlier, in SCN design problem, uncertainty in some parameters such as demand of customer zones and rate of return of used products from customers is reasonable. Although there are some approaches to solve models with hybrid and particularly fuzzy random parameters (such as Liu (2002a, 2002b) and Liu and Liu(2003)), there are only a few researches about SCN design with hybrid parameters. This point is more critical for integrated logistics network designs. Nonetheless there are some papers in facility location allocation in hybrid environment, for example Wen and Imamura (2008), Wen and Kang (2011). In this paper, we develop an integrated logistics network design model with fuzzy-random demands and rate of returned products.

1.3. Priority-based genetic algorithm

GA's are stochastic search techniques based on the mechanism of natural selection and natural genetics (Jaramillo et al., 2002). As one of the Evolutionary Computation (EC) techniques, the GA has been receiving great attention and it has successfully been applied for combinatorial optimization problems (Gen et al., 2006). GA is very useful when a large search space with little knowledge of how to solve the problem is presented. It belongs to the class of heuristic optimization techniques, which includes simulated annealing (SA), Tabu search, and evolutionary strategies. It has been with great success in providing optimal or near optimal solution for many diverse and difficult problems (Gen & Syarif, 2005). Representation is one of the important issues, which reflects the performance of Gas and normally different problems have various data structures or genetic representations. Tree-based representation is known to be one way for representing network problems. There are three ways of encoding tree: (1) edge-based encoding, (2) vertex-based encoding and (3) edge-and-vertex encoding (Gen & Cheng, 2000). Michalewicz et al. (1991) used matrix-based representation GA, which belongs to edge-based encoding to handle linear and non-linear transportation/distribution problems. Let m and

n be the number of sources and depots, respectively, then the dimension of matrix will be $m*n$. Although representation is straightforward, this approach requires special crossover and mutation operators for obtaining feasible solutions.

Gen and Cheng (2000) introduced spanning tree GA (st-GA) for solving network problems. They used Prüfer number representation for solving transportation problems and developed feasibility criteria for Prüfer number to be decoded into a spanning tree. Syarif et al. (2002) proposed spanning tree-based genetic algorithm by using Prüfer's number representation for solving a single product, three-stage supply chain network (SCN) problem. Xu et al. (2008) applied spanning tree-based genetic algorithm (st-GA) by the Prüfer number representation to locate the SCN to meet the demand imposed by customers with minimum total cost and maximum customer services for a multi objective SCN design problem. Although Prüfer number developed to encode of spanning trees, has been successfully applied to transportation problems, it needs some modifications to reach feasible solutions after classical genetic operators.

In this study, to escape from these repair mechanisms in the search process of GA, we concentrate on the priority-based encoding method. Gen et al. (2006) used priority-based encoding for a single-product, two-stage transportation problem. Altıparmak et al. (2006) applied priority-based representation to a single- product, single-source, and three-stage SCN problem, Altıparmak et al. (2009) proposed this encoding to a single-source, multi-product, multi-stage SCN problem. Lee et al. (2009) proposed a hybrid genetic algorithm with priority-based encoding method.

In this paper, we propose single-product, multi-stage integrated logistics network problem, which consider the minimizing of total shipping cost and fixed opening costs of the hybrid production/recovery centers, hybrid distribution/collection and disposal centers. In fact, this type of network design problem belongs to the class of NP-hard problems, so that priority based genetic algorithm could be presented to solve large scale problems. Finally, we apply the proposed model for an example problem and present some numerical results.

The remainder of this paper is organized as follows: The problem definition and mathematical model of integrated closed-loop logistics network design with hybrid parameters are introduced in section 2. Then in section 3, solution approach including priority-based genetic algorithm and fuzzy-random programming is discussed. Illustrative numerical examples and the results for evaluation are given in section 4. Finally, concluding remarks and future researches are outlined in section 5.

2. Problem definition and formulation

2.1 Problem definition

There are many modeling approach for supply chain network design of reverse and forward logistics flows. The logistics network discussed in this paper is an integrated logistics one, which models forwards and reverses flows. The mentioned integrated logistics network, which is based on Pishvae et al. (2009), is a single product, single period, multi-stage logistics network. There are four kinds of logistics node including customer zones, production, recovery, distribution, collection centers and disposal centers. As stated by Pishvae et al. (2009) in such an integrated logistics network, hybrid logistics facilities offer more advantages compared with separate distribution or collection centers. The advantages include both cost savings and pollution reduction as results of sharing material handling equipment and infrastructures (Shen, 2007).

As discussed earlier, in many real systems, the randomness and fuzziness of the customers' demands or rate of returned products are often mixed up with each other. Often estimation of these parameters can be considered by means of historical data. However, sometimes obtaining these data by means of

experiments is difficult. Instead, expert opinion is used to provide the estimations. Unfortunately sometimes using fuzzy variables cannot overcome on this kind of uncertainty, properly. Lack of attention to hidden aspect of randomness in these parameters could be a reason of this problem. In this case, fuzziness and randomness of the customers' demands are mixed up. The primary assumption is that there are several economic conditions with predefined probabilities, which can be calculated from historical data. Then, by means of expert opinions, a fuzzy variable is offered as demand of the customer under each economic condition. So the demand of each customer zone is presented by a fuzzy-random variable. Similarly, another fuzzy-random variable is offered for rate of return of products. As shown in Fig.1, in the forward flow in a pull system, products are shipped from hybrid production/recovery centers through hybrid distribution/collection centers to customer zones to meet customers' demand. In the reverse flow through a push system, returned products are collected by distribution-collection centers and after inspection the recoverable products are shipped to production/recovery centers, and scrapped products are shipped to disposal centers.

After the processes of recovery, the recovered products are used as new products in forward flows. A predefined fuzzy-random rate of demand and a predefined fuzzy-random percentage of demand of each customer zone are assumed as rate of returned products and a predefined value is determined as an average disposal rate. High quality returned products are more capable for recovery processes and low quality ones should be disposed in a safe process.

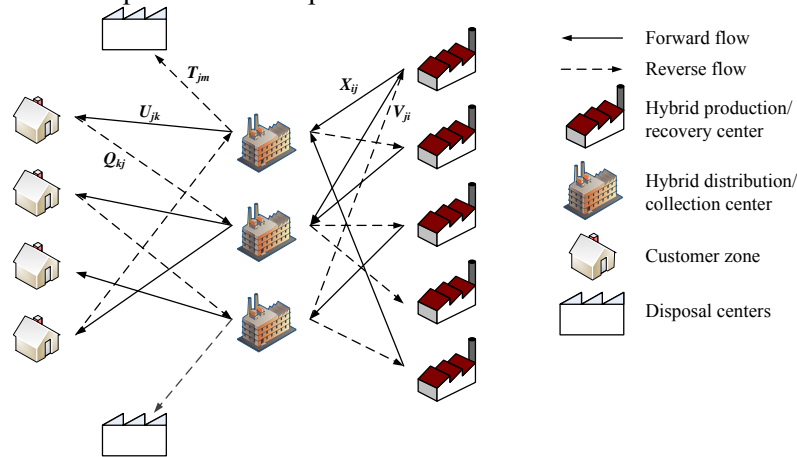


Fig. 1. The integrated forward-reverse logistics network

2.2 Mathematical formulation

Based on mentioned assumptions, the main issues to be addressed by this paper are to select the location and to determine the number of hybrid distribution/collection, hybrid production/recovery and disposal centers and to quantify the products shipped among logistics facilities. The notations used for the considered problem are listed below:

Parameters:

- I Set of potential production/recovery center locations $i \in I$
- J Set of potential hybrid distribution/collection center locations $j \in J$
- K Fixed locations of customer zones $k \in K$
- M Set of potential disposal center locations $m \in M$
- ξ_k Fuzzy-random demand of customer zone k
- η_k Fuzzy-random rate of return of used products from customer zone k
- s Average disposal fraction
- f_j Fixed cost of opening hybrid distribution/collection center j
- g_j Fixed cost of opening disposal center m

h_m	Shipping cost per unit of products from production/recovery center i to hybrid distribution/collection center j
c_{ij}	Shipping cost per unit of products from hybrid distribution/collection center j to customer zone k
a_{jk}	Shipping cost per unit of returned products from customer zone k to hybrid distribution/collection center j
b_{kj}	Shipping cost per unit of recoverable products from hybrid distribution/collection center j to production/recovery center i
e_{ji}	Shipping cost per unit of scrapped products from hybrid distribution/collection center j to disposal center m
p_{jm}	Manufacturing/recovery cost per unit of product at production/recovery center i
cw_i	Capacity of production for production/recovery center i
cy_j	Capacity of handling products in forward flow at hybrid distribution/collection center j
cz_m	Capacity of handling scrapped products at disposal center m
cyr_j	Capacity of handling returned products in reverse flow at hybrid distribution/collection center j
cwr_i	Capacity of recovery for production/recovery center i

Variables:

X_{ij}	Quantity of products shipped from production/recovery center i to hybrid distribution/collection center j
U_{jk}	Quantity of products shipped from hybrid distribution/collection center j to customer zone k
Q_{kj}	Quantity of returned products shipped from customer zone k to hybrid distribution/collection center j
V_{ji}	Quantity of recoverable products shipped from hybrid distribution/collection center j to production/recovery center i
T_{jm}	Quantity of scrapped products shipped from hybrid distribution/collection center j to disposal center m
$W_i = \begin{cases} 1 & \text{if a production/recovery center is opened at location } i, \\ 0 & \text{otherwise} \end{cases}$	
$Y_j = \begin{cases} 1 & \text{if a hybrid distribution/collection center is opened at location } j, \\ 0 & \text{otherwise} \end{cases}$	
$W_m = \begin{cases} 1 & \text{if a disposal center is opened at location } m, \\ 0 & \text{otherwise} \end{cases}$	

The problem can be formulated as follow:

$$\min \sum_{i \in I} f_i W_i + \sum_{i \in I} g_j Y_j + \sum_{m \in M} h_m Z_m + \sum_{i \in I} \sum_{j \in J} c_{ij} X_{ij} + \sum_{i \in I} \sum_{j \in J} a_{jk} U_{jk} + \sum_{k \in K} \sum_{j \in J} b_{kj} Q_{kj} + \sum_{j \in J} \sum_{i \in I} e_{ji} V_{ji} + \sum_{j \in J} \sum_{m \in M} p_{jm} T_{jm} \quad (1)$$

subject to

$$\sum_{j \in J} U_{jk} \geq \xi_k \quad \forall k \in K \quad (2)$$

$$\sum_{j \in J} Q_{kj} \geq \eta_k \xi_k \quad \forall k \in K \quad (3)$$

$$\sum_{j \in J} X_{ij} - \sum_{k \in K} U_{jk} = 0 \quad \forall j \in J \quad (4)$$

$$\sum_{i \in I} V_{ji} - (1-s) \sum_{k \in K} Q_{kj} = 0 \quad \forall j \in J \quad (5)$$

$$\sum_{m \in M} T_{jm} - s \sum_{k \in K} Q_{kj} = 0 \quad \forall j \in J \quad (6)$$

$$\sum_{j \in J} V_{ji} - \sum_{j \in J} X_{ij} \leq 0 \quad \forall i \in I \quad (7)$$

$$\sum_{j \in I} X_{ji} \leq cw_i W_i \quad \forall i \in I \quad (8)$$

$$\sum_{i \in I} X_{ij} \leq cy_j Y_j \quad \forall j \in J \quad (9)$$

$$\sum_{k \in K} Q_{kj} \leq cyr_j Y_j \quad \forall j \in J \quad (10)$$

$$\sum_{i \in I} V_{ji} \leq cwr_i W_i \quad \forall i \in I \quad (11)$$

$$\sum_{j \in I} T_{jm} \leq cz_m Z_m \quad \forall m \in M \quad (12)$$

$$W_i, Y_j, Z_m \in \{0, 1\} \quad \forall i \in I, \forall j \in J, \forall m \in M \quad (13)$$

$$X_{ij}, U_{jk}, Q_{kj}, V_{ji}, T_{jm} \geq 0 \quad \forall i \in I, \forall j \in J, \forall m \in M, \forall k \in K \quad (14)$$

where:

$$\xi_k = \begin{cases} (d_{11}, d_{12}, d_{13}, d_{14}) & \text{with probability } p_1 \\ (d_{21}, d_{22}, d_{23}, d_{24}) & \text{with probability } p_2 \\ (d_{31}, d_{32}, d_{33}, d_{34}) & \text{with probability } p_3 \end{cases} \quad (15)$$

$$\eta_k = \begin{cases} (r_{11}, r_{12}, r_{13}, r_{14}) & \text{with probability } p_1 \\ (r_{21}, r_{22}, r_{23}, r_{24}) & \text{with probability } p_2 \\ (r_{31}, r_{32}, r_{33}, r_{34}) & \text{with probability } p_3 \end{cases} \quad (16)$$

The objective function (1) minimizes the total cost of the forward and reverse logistics. It consists of the logistics shipping cost and fixed opening cost of the facilities. Constraints (2) and (3) ensure that the demands of all customers are satisfied and the returned products from all customers are collected. Constrains (4)-(7) balance flows between production/recovery and distribution/collection centers in forward and reverse flows. Constrains (8)-(12) explain about capacity of each facilities. Constraint (13) imposes the binary restriction on the decision variables W_i , Y_j , Z_m and constraint (14) imposes the non-negativity restriction on the other decision variables.

3. Solution approach

We propose priority-based GA to solve the deterministic integrated closed-loop logistics network design model. Then it is developed for solving uncertain model with fuzzy-random parameters.

3.1. Priority-based genetic algorithm

In priority-based encoding, a gene in a chromosome is characterized by two factors: locus, the position of the gene within the structure of chromosome, and allele, the value the gene takes. In priority-based encoding, the position of a gene is used to represent a node (source/depot in transportation network), and the value is used to represent the priority of corresponding node for constructing a tree among candidates (Gen & Cheng, 2000). For a transportation problem, a chromosome consists of priorities of sources and depots to obtain transportation tree and its length is equal to total number of sources $|K|$ and depots $|J|$, i.e. $|K| + |J|$. The transportation tree corresponding to a given chromosome is generated by sequential arc appending between sources and depots. At each step, only one arc is added to tree selecting a source (depot) with the highest priority and connecting it to a depot (source) considering minimum cost (Altıparmak et al., 2006). The decoding algorithm of the priority-based encoding is presented below.

Algorithm 1: Priority-based decoding:**Inputs:**

- K: set of sources
 J: set of depots
 b_j : demand on depot $j, \forall j \in J$
 a_k : capacity of source $k, \forall k \in K$
 c_{kj} : transportation cost of one unit of product from source k to depot $j, \forall k \in K, \forall j \in J$
 $v(k + j)$: chromosome, $\forall k \in K, \forall j \in J$

Outputs:

x_{kj} : the amount of product shipped from source k to depot j

While $\sum_{j=1}^J b_j \geq 0$

Step 1: $x_{kj} = 0, \forall k \in K, \forall j \in J$

Step 2: select a node based on $l = \arg \max \{v(t), t \in |k| + |j|\}, \forall k \in K, \forall j \in J$

Step 3: **if** $l \in K$ **then** a source is selected $k^* = l$,
 $j^* = \arg \min \{c_{kj} | v(j) \neq 0, j \in J\}$, Select a depot with minimum cost
else $j^* = l$ a depot is selected
 $k^* = \arg \min \{c_{kj} | v(j) \neq 0, k \in K\}$, Select a source with minimum cost

Step 4: $x_{k^*j^*} = \min\{a_{k^*}, b_{j^*}\}$

Update demands and capacities

$$a_{k^*} = a_{k^*} - x_{k^*j^*} \quad b_{j^*} = b_{j^*} - x_{k^*j^*}$$

Step 5: **if** $a_{k^*} = 0$ **then** $v(k^*) = 0$

if $b_{j^*} = 0$ **then** $v(j^*) = 0$

Step 6: **if** $v(k + j) = 0, \forall j \in J$ **then** output x_{kj} and calculate transportation cost,
else return step 1

End

Fig. 2 represents a transportation tree with 5 sources and 4 depots, its cost matrix and priority based encoding. Table 1 gives trace table of the decoding procedure to obtain transportation tree in Fig. 2

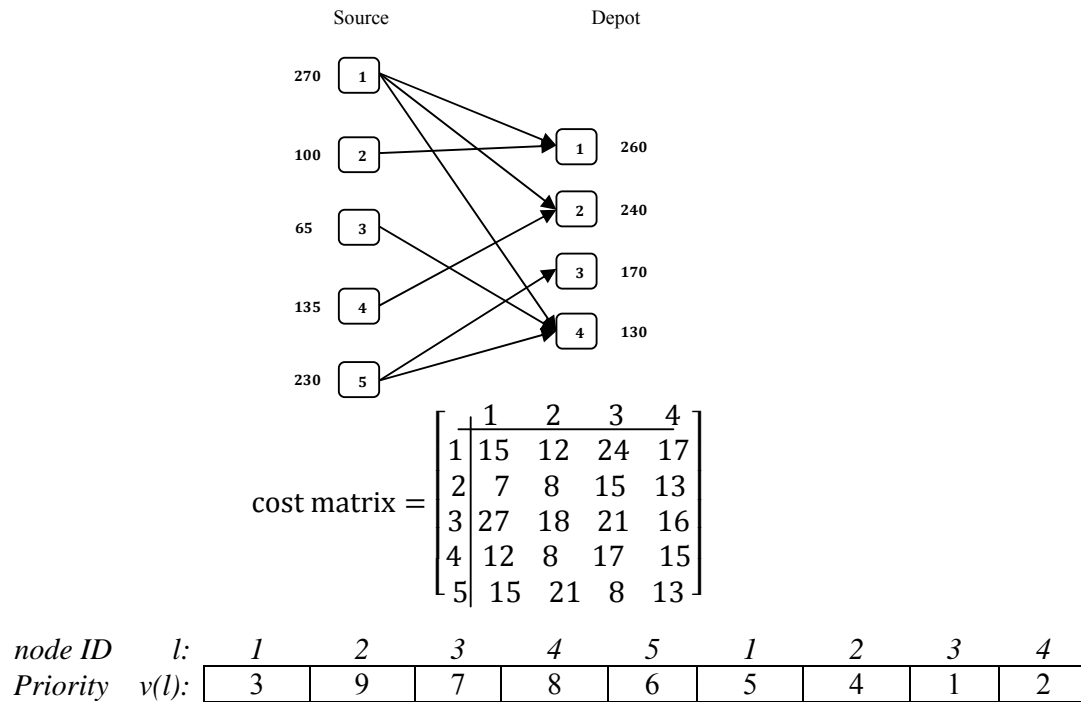


Fig. 2. A sample of transportation tree and its encoding

Table 1

Trace table of decoding procedure.

Iteration	v(k+j)	A	b	k	j	x _{kj}
0	[3 9 7 8 6 5 4 1 2]	(270,100,65,135,230)	(260,240,170,130)	2	1	100
1	[3 0 7 8 6 5 4 1 2]	(270, 0, 65,135,230)	(160,240,170,130)	4	2	135
2	[3 0 7 0 6 5 4 1 2]	(270, 0,65,0,230)	(160,105,170,130)	3	4	65
3	[3 0 0 0 6 5 4 1 2]	(270,0,0,0,230)	(160,105,170,65)	5	3	170
4	[3 0 0 0 0 5 4 1 2]	(270,0,0, 0,60)	(160,105,0,65)	1	1	160
5	[3 0 0 0 0 0 4 1 2]	(110,0,0, 0,60)	(0,105,0,65)	1	2	105
6	[3 0 0 0 0 0 0 1 2]	(5,0,0,0,60)	(0,0,0,65)	1	4	5
7	[0 0 0 0 0 0 0 1 2]	(0,0,0,0,60)	(0,0,0,60)	5	4	60
8	[0 0 0 0 0 0 0 1 0]	(0,0,0,0,0)	(0,0,0,0)			

For the proposed problem of this paper, we use a chromosome consists of five segments, in which each one is associated with one echelon of the integrated closed-loop logistics network. We use segment *J-M* to represent the transportation pattern from hybrid distribution/collection centers to disposal centers. Segment *J-I* is used to represent the transportation pattern from hybrid distribution/collection centers to hybrid production/recovery centers. Segment *K-J* represents the transportation pattern from customer zones to hybrid distribution/collection centers. Segment *I-J* shows the transportation pattern from hybrid production/recovery centers to hybrid distribution/collection centers, segment determines *J-K* the transportation pattern from hybrid distribution/collection centers to customer zones (see Fig. 3). The chromosome of integrated closed-loop logistics network is decoded on a special sequence.

	First segment J-M				Second segment J-I				Third segment K-J				Forth segment I-J				Fifth segment J-K			
<i>Node</i>	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
<i>priority</i>	2	1	4	7	5	3	6	6	7	3	5	1	4	2	6	8	3	1	5	2

Fig. 3. An illustration of integrated closed-loop logistics network model chromosome.

The decoding algorithm of an integrated closed-loop logistics network chromosome is presented in Algorithm 2.

Algorithm 2: Integrated closed-loop logistics network decoding algorithm

Inputs: $\xi_k, \mathcal{N}_k, S, f_i, g_j, h_m, c_{ij}, a_{jk}, b_{kj}, e_{ji}, cw_i, cy_j, cz_m, cyr_j$.

Outputs: $X_{ij}, U_{jk}, Q_{kj}, V_{ji}, T_{jm}, W_i, Y_j, Z_m$.

Step 1: calculate $U_{jk}, Y_j, \forall j \in J, k \in K$ using Algorithm 1,

Step 2: calculate $X_{ij}, W_i, \forall i \in I, j \in J$ using Algorithm 1,

Step 3: calculate $Q_{kj}, \forall k \in K, j \in J$ using Algorithm 1,

Step 4: calculate $V_{ji}, \forall j \in J, i \in I$ using Algorithm 1,

Step 5: calculate $T_{jm}, Z_m, \forall j \in J, m \in M$ using Algorithm 1.

The proposed GA solution procedure used four genetic operators described below.

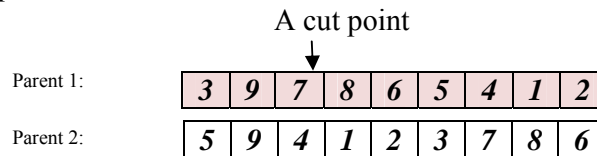
3.1.1. Parent selection operator

The parent selection operator is an important process, which directs a GA search towards promising regions in a search space. Two parents are selected from the solutions of a particular generation by selection methods, which assign reproduction opportunities to each individual parent in the population. There are a number of different selection methods, such as roulette wheel selection, tournament selection, rank selection, elitism selection, and random selection (Gen & Cheng, 2000). In this study, we implement tournament selection method where two teams of individuals are chosen from the population, randomly and each team consists of two chromosomes. The two best chromosomes taken from one of the two teams are chosen for crossover operations.

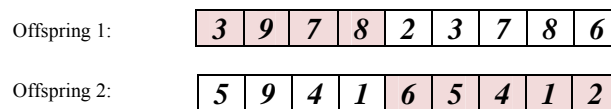
3.1.2. Crossover operator

The crossover operator generates new offspring by combining information contained in the chromosomes of the parents so that new chromosomes will have the best parts of the parent's chromosomes. The crossover is performed to explore new solution space and crossover operator corresponds to exchanging parts of strings between selected parents. Several crossover operators have been proposed for permutation representation, such as partially mapping crossover (PMX), order crossover (OX), position-based crossover (PX), cycle crossover (CX), weight mapping crossover (WMX), Heuristic crossover etc. (Lee et al., 2009). In this paper, we have used weight mapping crossover (WMX) operator, which is one-cut point crossover for permutation representation. As one point crossover, two chromosomes (parents) chooses a random cut-point and generates the offspring using segment of own parent to the left of the cut-point, then remapping the right segment based on the weight of other parent of right segment (Fig. 4).

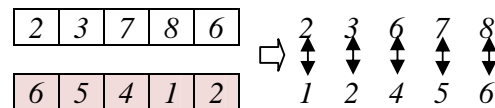
Step 1: select a cut-point



Step 2: exchange substrings between parents



Step 3: mapping the weight of the right segment



Step 4: generate offspring with mapping relationship

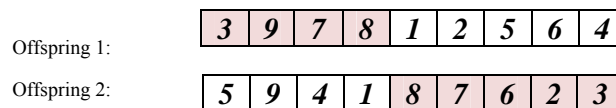


Fig. 4. Illustration of the WMX

3.1.3. Mutation operator

After recombination, some children undergo mutation, similar to crossover; mutation is executed to prevent the premature convergence and explores new solution space. However, unlike crossover, mutation is usually performed by modifying gene within a chromosome. In this study, we have used insert mutation where a digit is randomly selected and it is inserted a randomly selected new position in chromosome. Fig. 5 represents insert mutation.

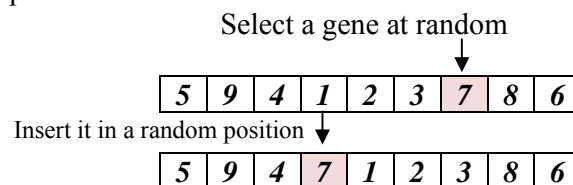


Fig. 5. Illustration of the insert mutation

3.1.4. Evaluation operator

The evaluation aims to associate each individual with a fitness value so that it can reflect the goodness of fit for an individual. The evaluation process is intended to compare one individual with other individuals in the population. The choice of fitness function is also very critical because it has to accurately measure the desirability of the features described by the chromosome. The function should be computationally efficient since it is used many times to evaluate each solution (Gen & Cheng, 2000). In this study, the objective function has been taken as fitness function.

3.3. Fuzzy-random programming

In this section, we shall recall some basic concepts and results on random fuzzy variable. These results are crucial for the remainder of this paper. An interested reader may consult Liu (2009) where important properties of random fuzzy variables are recorded.

Let Θ be a nonempty set, (Θ) the power set of Θ , and Pos a possibility measure. Then the triplet $(\Theta, P(\Theta), \text{Pos})$ is called a possibility space. A fuzzy variable ξ is defined as a function from a possibility space $(\xi, P(\xi), \text{Pos})$ to the set of real numbers. The possibility, necessity, and credibility of a fuzzy event $\{\xi \leq r\}$ can be represented by

$$\text{Pos}\{\xi \leq r\} = \sup_{x \leq r} \mu(x), \quad (17)$$

$$\text{Nec}\{\xi \leq r\} = 1 - \sup_{x > r} \mu(x), \quad (18)$$

$$\text{Cr}\{\xi \leq r\} = \frac{1}{2} (\text{Pos}\{\xi \leq r\} + \text{Nec}\{\xi \leq r\}), \quad (19)$$

respectively.

Definition 1. Let ξ be a fuzzy variable. Then the expected value of ξ is defined by

$$E[\xi] = \int_0^{\infty} \text{Cr}\{\xi \geq r\} dr - \int_{-\infty}^0 \text{Cr}\{\xi \leq r\} dr, \quad (20)$$

provided that at least one of the two integrals is finite.

Definition 2. A fuzzy random variable is a function ξ from a probability space $(\Omega; \Pr)$ to the set of fuzzy variables such that $\text{Cr}\{\xi(\omega) \in B\}$ is a measurable function of ω for any Borel set of B □ .

Definition 3. Let ξ be a random fuzzy variable, and B a Borel set of R . Then the chance of random fuzzy event $\xi \in B$ is a function from $[0,1]$ to $[0,1]$, defined as

$$\text{Ch}\{\xi \in B\}(\alpha) = \sup_{\text{Cr}\{A\} \geq \alpha} \inf_{\theta \in A} \{\xi(\theta) \in B\}. \quad (21)$$

There are several hybrid programming models for solving an optimization model including fuzzy-random variables. Expected value model (EVM), chance-constrained programming (CCP), dependent-chance programming (DCP) are the generic models of them. Chance-constrained programming, which was initialized by Charnes and Cooper (1961), offers a powerful means for modeling uncertain decision systems. Liu and Iwamura (1998) extended the chance-constrained programming to fuzzy decision

systems. Liu (2001) initialized a general framework of fuzzy random chance-constrained programming. The essential idea of chance-constrained programming is to optimize some critical value with a given confidence level subject to some chance constraints. The general form of this programming is:

$$\begin{aligned} & \max \quad \bar{f} \\ & \text{subject to} \\ & \text{Ch}\{f(x, \xi) \geq \bar{f}\} \geq \alpha \end{aligned} \quad (22)$$

$$\text{Ch}\{g_i(x, \xi) \leq 0\} \geq \alpha, \quad j=1, 2, \dots, p \quad (23)$$

where x is decision vector, ξ is a hybrid vector, $f(x, \xi)$ objective function, $g_j(x, \xi)$, $j=1, 2, \dots, p$ constraint functions, α confidence level and $\text{Ch}\{\cdot\}$ is a chance measure. In the discussed model in section 2.2 the objective function does not include any hybrid variable but to calculate the chance function for uncertain constraints (2) and (3), a fuzzy-random simulation is used. Then the mentioned chance constraints are added to the model. The chance constraints related to the integrated logistics network design are:

$$\text{Ch}\{\xi_k - \sum_{j \in J} U_{jk} \leq 0\} \geq \alpha \quad \forall k \in K \quad (24)$$

$$\text{Ch}\{\eta_k \xi_k - \sum_{j \in J} Q_{kj} \leq 0\} \geq \alpha \quad \forall k \in K \quad (25)$$

The fuzzy-random simulation algorithm to calculate chance constraints is as following (Liu, 2006):

Algorithm 1. Fuzzy Random Simulation

Step 1. Generate $\omega_1, \omega_2, \dots, \omega_N$ from Ω according to the probability measure Pr.

Step 2. Compute the credibilities $\beta_k = \text{Cr}\{g_j(x, \xi(\omega_k)) \leq 0, j=1, 2, \dots, p\}$ for $k=1, 2, \dots, N$ by the fuzzy simulation (see algorithm 2).

Step 3. Set N' as the integer part of αN .

Step 4. Return the N' th largest element in $\{\beta_1, \beta_2, \dots, \beta_N\}$.

In order to compute uncertain functions $\text{Cr}\{g_j(x, \xi(\omega_k)) \leq 0, j=1, 2, \dots, p\}$ a fuzzy simulation can be used.

The fuzzy simulation algorithm is as following (Liu, 2006):

Algorithm 2. Fuzzy Random Simulation

Step 1. Randomly generate θ_k from Θ such that $\text{Cr}\{\theta_k\} \geq \varepsilon / 2$, for $k=1, 2, \dots, N$.

Step 2. Write $v_k = (2\text{Cr}\{\theta_k\} \wedge 1)$ and produce $\xi_k = \xi(\theta_k)$ for $k=1, 2, \dots, N$ where ε is a sufficiently small number.

Step 3. The credibility $\text{Cr}\{g_j(x, \xi(\omega_k)) \leq 0, j=1, 2, \dots, p\}$ can be estimated by the following formula

$$\frac{1}{2} \left(\max_{1 \leq k \leq N} \left\{ v_k \mid g_j(x, \xi_k) \leq 0 \right\} + \min_{1 \leq k \leq N} \left\{ 1 - v_k \mid g_j(x, \xi_k) > 0 \right\} \right) \quad (26)$$

4. Numerical examples and results

To illustrate previous parts and efficiency evaluation, the proposed method is applied to some hypothetical problems. All the computations for this part were coded by the Visual Basic 6 software

and were run on a core i5 Intel CPU with 4 GB of RAM. First, we assume a small-scaled integrated logistics network design with deterministic parameters to evaluate the priority-based genetic algorithm. Tables 2 to 5 give information about this problem and we assume that $s=0.5$.

Table 2

Demand and rate of return for each customer zone

Customer zone	d_k	r_k
k=1	1	0.3
k=2	1	0.5
k=3	4	0.2
k=4	9	0.7

Table 3

Fixed cost for each facility

Facility no.	f_i	g_i	h_m
1	2	7	7
2	3	8	10
3	4	9	
4	5		
5	6		

Table 4

Transportation cost between facilities

c_{ij}	b_{ki}			p_{ik}						
	j=1	j=2	j=3							
i=1	1	2	3	k=1	4	7	9	j=1	6	3
i=2	7	12	10	k=2	10	6	3	j=2	7	7
i=3	9	5	4	k=3	5	7	1	j=3	8	5
i=4	3	6	7	k=4	2	4	7			
i=5	10	5	2							

a_{jk}	e_{ii}									
	k=1	k=2	k=3	k=4						
j=1	8	1	2	2	i=1	2	8	9	3	1
j=2	9	3	10	4	i=2	0	6	0	7	4
j=3	2	5	9	9	i=3	0	7	0	7	4

Table 5

Capacities of each facility

cw_i	j=1	j=2	j=3	j=4	j=5	cy_i	j=1	j=2	j=3	cz_m	m=1	m=2
	cwr_i	81	90	12	91		63	4	13		12	6
	31	38	26	1	33	2	8	18				

Table 6

Binary decision variables

W_i	Y_j	Z_m	4	3	2	1
			0	0	0	1
	1	1	1			
		0	1			

First the model is solved by lingo 11, and a global optimum is reached in about 1 second. The related solution is shown in Table 6 and 8. Lingo uses a branch and bound approach and the objective value of the global optimum is 160.65.

Table 7

Transportation decision variables

X_{ij}	Q_{ki}			T_{ik}						
	j=1	j=2	j=3							
i=1	4	10.75	0	k=1	0.5	0	0	j=1	0.3	0
i=2	0	0	0	k=2	0	0	0.69	j=2	0.345	0
i=3	0	0	0	k=3	0	0	1.12	j=3	0.27	0
i=4	0	0	0	k=4	1.5	2.3	0			
i=5	0	0	1.75							

U_{jk}	V_{ii}									
	k=1	k=2	k=3	k=4						
j=1	0	0	4	0	i=1	1.7	0	0	0	0
j=2	0	1.25	0	9.5	i=2	1.95	0	0	0	0
j=3	1.25	0	0.5	0	i=3	1.54	0	0	0	0

As mentioned earlier, for large-scaled problems a metaheuristic algorithm is needed because the integrated closed-loop network design of this paper is NP-hard. Therefore, two genetic algorithm encoding is used: classic encoding (with usual operators) and priority-based encoding. To compare this 3 solving method Table 8 depicts best objective values and computational time for each method.

Table 8
Comparison of 3 solving methods

Problem no.	I	J	K	M	No. of variables	No of constraints	Lingo		Classic GA		Priority-based GA	
							Best Obj.	Time (sec.)	Best Obj.	Time (sec.)	Best Obj.	Time (sec.)
1	5	3	4	2	70	40	161.65	1	332	15	161.65	0.48
2	15	13	14	12	950	150	440.15	12	503	22	442.75	8.87
3	60	50	45	55	13415	575	1510	114	1930	142	912	89.42
4	120	100	90	110	53330	1150	*	*	4735	394	2743.9	233.53

In problem no. 1 and 2 Lingo solver reached some optimal solution, which can be a basis to evaluate the two other methods. In problem no. 4 Lingo could not reach to a solution in reasonable amount of time. Therefore, it seems, priority-based genetic algorithm is a better efficiency choice for large-scaled cases. Now we solve a similar example but with hybrid parameters. Here, we assume that there are three scenarios (for example economic conditions) and estimate demand and rate of return under each scenario in terms of form of fuzzy variables based on expert's opinions. Table 9 gives information about demand (ξ_k) and rate of return (η_k). Probability of occurrence of each scenario (P_1, P_2, P_3) which are predicted by historical data is shown in this table. Other information about this example is similar to previous problem.

Table 9
Fuzzy-random variables related to demand and rate of return of each customer zone

Customer zone	Scenario 1 $P_1=0.2$		Scenario 2 $P_2=0.5$		Scenario $P_3=0.5$	
	ξ_k	η_k	ξ_k	η_k	ξ_k	η_k
k=1	(0.5,1,1.5)	(0.2,0.3,0.5)	(1,2,3)	(0.3,0.5,0.9)	(2,4,6)	(0.4,0.6,0.9)
k=2	(0.5,1,1.5)	(0.05,0.4,0.5)	(1,3,4)	(0.1,0.4,0.7)	(3,4,5)	(0.2,0.5,0.7)
k=3	(3,4,5)	(0.1,0.4,0.7)	(3,6,8)	(0.3,0.6,0.8)	(5,8,10)	(0.4,0.7,0.8)
k=4	(8,9,10)	(0.2,0.3,0.5)	(9,11,13)	(0.3,0.5,0.9)	(10,12,16)	(0.4,0.6,0.9)

Now we solve this problem by the proposed fuzzy-random programming under several confidence level (α) with this assumption that $N=200$. The results are shown in Table 10. As it seems, the objective function gets worse when we increase α .

Table 10
Results of hybrid programming for several confidence levels

Problem no.	confidence level	No. of variables	No of constraints	Objective value	function	Computational time (sec.)
1	$\alpha=0.1$	70	40	109.55		208
2	$\alpha=0.3$	70	40	109.55		338
3	$\alpha=0.5$	70	40	140.53		89
4	$\alpha=0.7$	70	40	140.53		150
5	$\alpha=0.9$	70	40	203.22		418

In this part, we solve the problem with a predefined confidence level and several number of customer zones to evaluate efficiency of the proposed hybrid programming under various scales. Table 11 illustrates the results and Fig. 6 depicts computational time versus each number of customer zones (K). Results from Table 11 represent that the proposed hybrid programming for integrated closed-loop logistics network design with fuzzy-random parameters is efficient for various scales.

Table 11
Results of hybrid programming for several number of customer zones (K)

Problem no.	I	J	M	K	confidence level	No. of variables	No of constraints	Objective value	function	Computational time (sec.)
1	5	3	2	4	$\alpha=0.8$	70	40	124.39		16
2	5	3	2	10	$\alpha=0.8$	106	52	240.09		24
3	5	3	2	20	$\alpha=0.8$	166	72	677.39		146
4	5	3	2	30	$\alpha=0.8$	226	92	1142.26		479
5	5	3	2	40	$\alpha=0.8$	286	112	1805.79		1954

As it seems in Fig. 6, by increasing the number of customer zones (K), computational times of solving the model increase, exponentially.

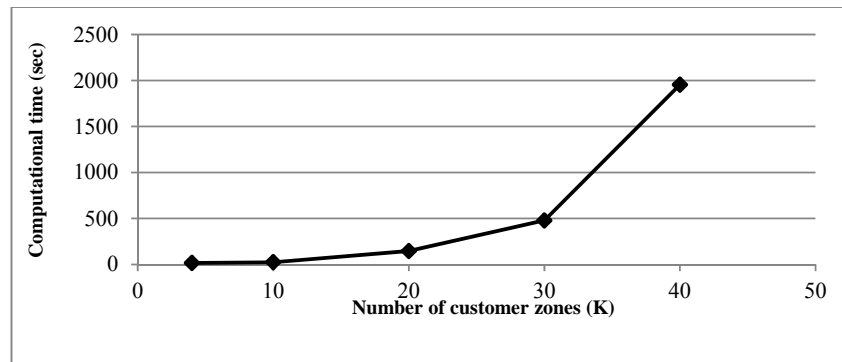


Fig. 6. Computational time versus each number of customer zones (K)

5. Conclusion

In this paper, we have considered a fuzzy-random mixed integer linear programming model for the design of an integrated forward/reverse logistics network with hybrid facilities. We have assumed that there are several scenarios (for example economic conditions) and estimated demand and rate of return under each scenario in the form of fuzzy variables by expert's opinion. Probability of occurrence of each scenario predicted by historical data and as a result, the demand and rate of return of customer zones have been regarded as fuzzy-random variables. In fact, this type of network design problem belongs to the class of NP-hard problems, so that we have utilized the priority-based genetic algorithm known to be an efficient and robust method for this kind of problem. A hybrid programming approach by fuzzy-random simulation has been applied to solve the uncertain model. Some examples are introduced in several scales and results showed that the proposed method solves them efficiently. Develop the hybrid model for multi-product or multi-period, using multi-objective programming to consider customer satisfaction for this problem can be subjects for future researches.

References

- Altıparmak, F., Gen, M., Lin, L., & Karaoglan, I. (2009). A steady-state genetic algorithm for multi-product supply chain network design. *Computers & Industrial Engineering*, 56, 521–537.
- Altıparmak, F., Gen, M., Lin, L., & Paksoy, T. (2006). A genetic algorithm for multi-objective optimization of supply chain networks. *Computers & Industrial Engineering*, 51, 197–216.
- Barros, A. I., Dekker, R., & Scholten, V. (1998). A two-level network for recycling sand: a case study. *European Journal of Operational Research*, 110, 199–214.
- Charnes, A., & Cooper, W. (1961). *Management Models and Industrial Applications of Linear Programming*. Wiley, New York.
- Du, F., & Evans, G. W. (2008). A bi-objective reverse logistics network analysis for post-sale service. *Computers and Operations Research*, 35, 2617 – 2634.
- Gen, M., Altıparmak, F., & Lin, L. (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spectrum*, 28, 337–354.
- Gen, M., & Cheng, R. (2000). *Genetic algorithms and engineering optimization*. John Wiley and Sons, New York.
- Gen, M., & Syarif, A. (2005). Hybrid genetic algorithm for multi-time period production/distribution planning. *Computers & Industrial Engineering*, 48, 799–809.
- Huang, X. (2006). Optimal project selection with random fuzzy parameters, *International Journal of Production Economics*, 112-122.
- Jaramillo, J. H., Bhadury J, & Batta, R. (2002). On the use of genetic algorithms to solve location problems. *Computers and Operations Research*, 29, 761-779.

- Jayaraman, V., Guide, V. D., & Srivastava, R. (1999). A closed loop logistics model for remanufacturing. *Journal of the Operational Research Society*, 50, 497–508.
- Jayaraman, V., Patterson, R. A., & Rolland, E. (2003). The design of reverse distribution networks: Models and solution procedures. *European Journal of Operational Research*, 150, 128–149.
- Kara, S., Rugrungruang, F., & Kaebernick, H. (2007). Simulation modeling of reverse logistics networks. *International Journal of Production Economics*, 106, 61–69.
- Kim, K., Song, I., Kim, J., & Jeong, B. (2006). Supply planning model for remanufacturing system in reverse logistics environment. *Computers & Industrial Engineering*, 51, 279–287.
- Kirkke, H. R., Harten, A. V., & Schuur, P. C. (1999). Business case Oco: reverse logistic network redesign for copiers. *OR Spectrum*, 21, 381–409.
- Ko, H. J., & Evans, G. W. (2007). A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs. *Computers and Operations Research*, 34, 346–366.
- Kwakernaak, H. (1978). random variables: *Definitions and theorems*. 15, 1-29.
- Lee, D.H., & Dong, M. (2008). A heuristic approach to logistics network design for end-of-lease computer products recovery. *Transportation Research Part E*, 44, 455-474.
- Lee, J. E., Gen, M., & Rhee, K. G. (2009). Network model and optimization of reverse logistics by hybrid genetic algorithm. *Computers & Industrial Engineering*, 56, 951–964.
- Liu, B. (2001). Fuzzy random chance-constrained programming, *IEEE Transactions on Fuzzy Systems*, 9(5), 713-720.
- Liu, B. (2002a). Random fuzzy dependent-chance programming and its hybrid intelligent algorithm, *Information Sciences*, 141 (3–4), 259–271.
- Liu, B. (2002b). *Theory and Practice of Uncertain Programming*, Physica-Verlag, Heidelberg.
- Liu, B. (2006). *Theory and Practice of Uncertain Programming*. Beijing, China: Uncertainty Theory Laboratory- Department of Mathematical Sciences-Tsinghua University.
- Liu, B. (2009). *Theory and Practice of Uncertain Programming* (3rd ed.). Beijing, China: Uncertainty Theory Laboratory- Department of Mathematical Sciences-Tsinghua University.
- Liu, B., Iwamura, K. (1998). Chance constrained programming with fuzzy parameters, *Fuzzy Sets and Systems*, 94, 227–237.
- Liu, Y., & Liu, B. (2003). A class of fuzzy random optimization: Expected value models. *Information Sciences*, 155, 89-102.
- Liu, Y., & Liu, B. (2003). Expected value operator of random fuzzy variable and random fuzzy expected value models, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 11 (2), 195–215.
- Lu, Z., & Bostel, N. (2007). A facility location model for logistics systems including reverse flows: The case of remanufacturing activities. *Computers and Operations Research*, 34: 299–323.
- Michalewicz, Z., Vignaux, G. A., & Hobbs, M. (1991). A non-standard genetic algorithm for the nonlinear transportation problem. *ORSA Journal on Computing*, 3, 307–316.
- Min, H., Ko, H. J., & Ko, C. S. (2006). A genetic algorithm approach to developing the multi-echelon reverse logistics network for product returns. *Omega*, 34, 56 – 69.
- Min, H., Ko, H. J., & Park, B. I. (2005). A Lagrangian relaxation heuristic for solving the multi-echelon, multi-commodity, closed-loop supply chain network design problem. *International Journal of Logistics Systems and Management*, 1, 382–404.
- Pishvae, M. S., Jolai, F., & Razmi, J. (2009). A stochastic optimization model for integrated forward/reverse logistics A stochastic optimization model for integrated forward/reverse logistics. *Journal of Manufacturing Systems*, 107-114.
- Pishvae, M. S., Zanjirani Farahani, R., & Dullaert, W. (2010). A memetic algorithm for bi-objective integrated forward/reverse logistics network design. *Computers & Operations Research*, 37, 1100-1112.
- Shen, Z.M. (2007). Integrated supply chain design models: a survey and future research directions. *Journal of Industrial and Management Optimization*, 3(1), 1-27.
- Syarif, A., Yun, Y. S., & Gen, M. (2002). Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach. *Computers & Industrial Engineering*, 43, 299-314.
- Wen, M., & Iwamura, K. (2008). Facility location–allocation problem in random fuzzy environment: Using (α, β) -cost minimization model under the Hurewicz criterion. *Computers & Mathematics with Applications*, 55(4), 704-713.
- Xu, J., Liu, Q., & Wang, R. (2008). A class of multi-objective supply chain networks optimal model under random fuzzy environment and its application to the industry of Chinese liquor. *Information Sciences*, 178, 2022–2043.