



# A new bi-objective model for a closed-loop supply chain problem with inventory and transportation times

F. Forouzanfar<sup>a</sup>, R. Tavakkoli-Moghaddam<sup>b,\*</sup>, M. Bashiri<sup>c</sup> and A. Baboli<sup>d</sup>

- a. *Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.*  
 b. *School of Industrial Engineering and Center of Excellence for Intelligence Based Experimental Mechanics, College of Engineering, University of Tehran, Tehran, Iran.*  
 c. *Department of Industrial Engineering, Faculty of Engineering, Shahed University, Tehran, Iran.*  
 d. *DISP Laboratory, INSA-Lyon, 69621 Villeurbanne Cedex, France.*

Received 12 October 2014; received in revised form 7 February 2105; accepted 25 April 2015

## KEYWORDS

Closed-loop supply chain;  
 Integer nonlinear programming;  
 Transportation;  
 Inventory;  
 Bee algorithm.

**Abstract.** This paper considers a closed-loop supply chain design problem including several producers, distributors, customers, collecting centers, recycle centers, revival centers, raw materials customers considering several periods, existing inventory and shortage in distribution centers, and transportation cost and time. This problem is formulated as a bi-objective integer nonlinear programming model. The aim of this model is to determine numbers and locations of supply chain elements, their capacity levels, allocation structure, mode of transportation between them, amount of transported products between them, amount of existing inventory, and shortage in distribution centers in each period to minimize the sum of system costs and transportation time in the network. To validate this model and show the applicability of it for small-sized problems, GAMS software is used. Because this given problem is NP-hard, a Bee Colony Optimization (BCO) algorithm is proposed to solve medium and large-sized problems. Furthermore, to examine the efficiency of the proposed BCO algorithm, the associated results are compared with the results obtained by the Genetic Algorithm (GA). Finally, the conclusion is provided.

© 2016 Sharif University of Technology. All rights reserved.

## 1. Introduction

Nowadays, rapid economic changes and competitive pressures in current global markets force firms to invest and focus on efficient management of their logistics system. Return policies, environmental concerns, and emphasis on servicing and reusing pieces lead to improvement in forwarding traditional supply chain, where Reverse Logistics (RLs) components have been combined. Precise, on time, and accurate transfer of

useless materials, items, and products from the end point and ultimate consumer to suitable and relevant unit through supply chain has been described by the term ‘reverse logistics’ [1]. Using reverse logistics not only saves inventory transportation costs, waste material transportation costs, and disposal costs, but also ensures future sale and customer satisfaction. Global competitive conditions, legal obligations, and especially environmental concerns oblige organizations to collect their returned products, so that they revive, recycle, and dispose these products for the sake of environmental conservation. Collecting products after consumption by customers and returning them into supply chain or disposing them have raised the issue of closed loop supply chain, which considers integrated management of forwarding and reverse streams in this

\*. *Corresponding author.*

*E-mail addresses:* ie\_forouzanfar@yahoo.com (F. Forouzanfar); tavakoli@ut.ac.ir (R. Tavakkoli-Moghaddam); bashiri.m@gmail.com (M. Bashiri); armand.baboli@insa-lyon.fr (A. Baboli)

chain. Real design of supply chain network structure leads firms to gain more competitive advantages. Therefore, closed-loop supply chain designing that simultaneously considers the forward and reverse chains can be efficient in gaining more advantages. Organizations must reduce time and cost of supplying customer demands to survive in global markets. Time for doing the order depends on several factors including a transportation state. Different transportation states include a reverse relation between time and cost. Undoubtedly, when this value is slight, it is taken as value added by which one could reach long term and short term competitive advantages in market. On the other hand, decisions regarding the amount of inventory or shortage in Distribution Centers (DC's) in each period, amount of transported products between levels in each phase during each period, and cost of establishing centers with a specific capacity level depend on their costs.

In this paper, we propose a mathematical model for a closed-loop supply chain including several producers, distributors, customers, collecting centers, recycle centers, revival centers, and raw materials customers. Additionally, we consider several periods, escalating factor of cost, existing inventory and shortage in distribution centers, several levels of capacity and mode of transportation between centers in each period, and transportation cost and time in the modeling as novel innovations. Furthermore, a solution approach based on the artificial bee colony is developed. To the best of our knowledge, this paper is among the first publications that consider the bee colony optimization algorithm to solve a closed-loop supply chain network problem. The results of its solution are compared with the results obtained by the Genetic Algorithm (GA). The rest of this paper is organized as follows. In Section 2, the given problem is defined and a mathematical model is proposed. In Section 3, to solve this model, the bee colony optimization algorithm is developed and a genetic algorithm is also applied. In Section 4, some experiments and results are discussed. Finally, conclusion is provided in Section 5.

## 2. Literature review

Some studies related to a closed-loop supply chain are mentioned in this section in order to clarify the necessity of this study. Some recent studies on a closed-loop supply chain and characteristics of this paper are illustrated in Table 1.

Tavakkoli-Moghaddam et al. [11] presented a three-level multi-period supply chain model that minimizes the time and cost of transportation along the chain. They considered existing inventory or shortage in distribution centers and also took into account different transportation states between two centers in two different levels.

Guide and Van Wassenhove [12] used five phases to describe the evolution of the closed-loop supply chain research. Krikke et al. [13] surveyed a wide study in the basic closed-loop supply chain about return practices. They analyzed these practices and provided some recommendations for converting value destruction into value creation. Stindt and Sahamie [14] reviewed the research on closed-loop supply chain management in a process industry. They investigated the main characteristic of CLSC planning in the process industry to determine the evolution and gaps of this current research and to explore the topic area and methodology in this field. Govindan et al. [15] presented a universal literature review of recent papers in a RL/CLSC and suggested future directions and opportunities of related research. According to the reviewed papers, no paper has considered minimization of the cost and transportation time throughout the closed-loop supply chain, simultaneously, so far.

Nature-inspired algorithms are very useful in solving multi-variable optimization problems. The Bee Algorithm (BA) [16] is one of the well-known group algorithms, which simulates the foraging behavior of honey bee colonies. We use the BA for optimizing a closed-loop supply chain network and compare the results with the Genetic Algorithm (GA). The GA [17] is a specific kind of evolution algorithms using biological techniques, such as inheritance and mutation. It is an innovative population based algorithm. This paper is among the first publications that consider the bee colony algorithm to solve a closed-loop supply chain network problem. Soleimani and Kannan [18] applied a hybrid Particle Swarm Optimization (PSO) and GA for solving a closed-loop supply chain network design problem in large-scale networks. Kannan et al. [19] used a GA to solve a closed-loop supply chain model. Min et al. [20] proposed a mixed-integer nonlinear model and GA to provide a minimum cost solution for the closed-loop supply chain network design problem involving the spatial and temporal consolidations of product returns. To the best of our knowledge, there is no paper that considers the bee colony optimization algorithm to solve a closed-loop supply chain problem.

## 3. Problem definition

This problem consists of several producers, distributors, customers, collecting centers, recycle centers, revival centers, and raw materials customers (see Figure 1) considering several periods, existing inventory and shortage in distribution centers, and transportation cost and time. Trade-off between cost and time creates a bi-objective problem. One criterion tries to minimize the fixed cost for opening centers with a certain capacity level, transportation and allocation costs, construction, process, restructuring and sepa-

**Table 1.** Some recent studies on a closed-loop supply chain.

Ref.	Modeling						Objective function						Period		Contribution
	LP	MILP	MINLP	RMIP	MIP	ILP	Max		Min			Single	Multiple		
							Profit	Weights of suppliers	Defect rates	Cost				Time	
										Shortage	Other				
[2]		•					•							•	Considering uncertainty on products' demand and prices, different decision scenarios in the planning model.
[3]		•												•	Developing a model to determine the raw material level, production level, distribution and inventory level, disposal level, and recycling level at different facilities. A case of battery recycling is considered.
[4]				•						•	•		•		A robust optimization model (for handling the inherent uncertainty of input data).
[5]	•												•		The model captures the trade-offs between various costs with considering a set of available transportation modes. That is scenario-based.
[6]					•									•	Considering setup, production and inventory cost for new products and remanufactured products in all periods. Using a Lagrangian relaxation based approach for the capacitated lot sizing problem.
[7]		•					•	•	•				•		Considering supplier selection and using the fuzzy sets theory (to overcome the uncertainty in assessment of eligible suppliers).
[8]		•											•		Considering uncertainty in demand, facility location and using stochastic programming. The model is scenario-based.
[9]						•							•		Considering a single-product and constant demand. The utility of the proposed PRISM algorithm.
[10]			•											•	Describing a model which optimizes the strategic and tactical Decisions. Considering escalating factor of cost and a single-product. Using one type of vehicles between two nodes and selecting one capacity for each DCs.
This paper			•							•	•	•		•	

LP: Linear Programming;

MINLP: Mixed-Integer Nonlinear Programming;

MIP: Mixed-Integer Programming;

MILP: Mixed-Integer Linear Programming;

RMIP: Relaxed Mixed-Integer Programming;

ILP: Integer Linear Programming.

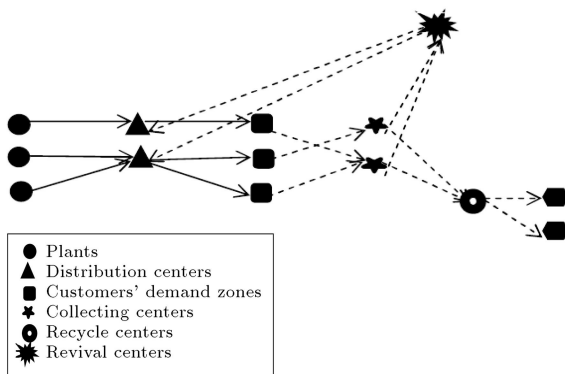


Figure 1. Closed-loop supply chain network.

ration costs, and inventory and shortage costs in the closed-loop supply chain. The other criterion reduces transportation time along the chain.

### 3.1. Assumptions

The main assumptions of the presented model are as follows:

1. This problem is single product [21];
2. Several capacity levels are considered for centers and finally one capacity will be chosen for each center;
3. There are several available states for transportation between two consecutive levels [21];
4. Only one kind of transportation vehicles is used between two knots among levels in each period [21];
5. Faster transportation vehicle is costly [21];
6. There is a balance between inputs and outputs of each center;
7. Several periods are considered along the planning horizon;
8. At the end of the last period, inventory or shortage is supposed to be zero;
9. Centers allocation has been done at the beginning of the first period and until the end of the planning horizon, it will not change;
10. Shortage in DCs in each period should be given in the next period;
11. The analysis just considers inventory of DCs.

### 3.2. Sets

$I$	Set of plants;
$J$	Set of DCs;
$K$	Set of demand zones of customers;
$L$	Set of collecting centers;
$M$	Set of revival centers;
$N$	Set of recycle centers;

$D_i$	Set of capacity levels available to plant $i$ ( $i \in I$ );
$D_j$	Set of capacity levels available to $DC_j$ ( $j \in J$ );
$D_l$	Set of capacity levels available to collecting center $l$ ( $l \in L$ );
$D_m$	Set of capacity levels available to revival center $m$ ( $m \in M$ );
$D_n$	Set of capacity levels available to recycle center $n$ ( $n \in N$ );
$T$	Time period.

### 3.3. Parameters

$B_{ijl_1t}$	Time for transporting any quantity of product from plant $i$ to $DC_j$ using transportation mode $l_1$ in period $t$ ;
$B_{jkl_2t}$	Time for transporting any quantity of product from $DC_j$ to customer's demand zone $k$ using transportation mode $l_2$ in period $t$ ;
$B_{kl_3t}$	Time for transporting any quantity of product from customer's demand zone $k$ to collecting center $l$ using transportation mode $l_3$ in period $t$ ;
$B_{lnl_4t}$	Time for transporting any quantity of product from collecting center $l$ to recycle center $n$ using transportation mode $l_4$ in period $t$ ;
$B_{nel_5t}$	Time for transporting any quantity of product from recycle center $n$ to material customer $e$ using transportation mode $l_5$ in period $t$ ;
$B_{lm_6t}$	Time for transporting any quantity of product from collecting center $l$ to revival center $m$ using transportation mode $l_6$ in period $t$ ;
$B_{mjl_7t}$	Time for transporting any quantity of product from revival center $m$ to $DC_j$ using transportation mode $l_7$ in period $t$ ;
$F_i^d$	Yearly fixed cost for opening and operating plant $i$ with capacity level $d$ ( $d \in D_i$ ), ( $i \in I$ );
$F_j^d$	Yearly fixed cost for opening and operating $DC_j$ with capacity level $d$ ( $d \in D_j$ ), ( $j \in J$ );
$F_m^d$	Yearly fixed cost for opening and operating revival center $m$ with capacity level $d$ ( $d \in D_m$ ), ( $m \in M$ );
$F_n^d$	Yearly fixed cost for opening and operating recycle center $n$ with capacity level $d$ ( $d \in D_n$ ), ( $n \in N$ );

$A_{ijl_1t}$	Unit cost of transportation from plant $i$ to DC $_j$ using transportation mode $l_1$ in period $t$ ;
$A_{jkl_2t}$	Unit cost of transportation from DC $_j$ to customer's demand zone $k$ using transportation mode $l_2$ in period $t$ ;
$A_{kll_3t}$	Unit cost of transportation from customer's demand zone $k$ to collecting center $l$ using transportation mode $l_3$ in period $t$ ;
$A_{lnl_4t}$	Unit cost of transportation from collecting center $l$ to recycle center $n$ using transportation mode $l_4$ in period $t$ ;
$A_{nel_5t}$	Unit cost of transportation from recycle center $n$ to material customer $e$ using transportation mode $l_5$ in period $t$ ;
$A_{lm_l_6t}$	Unit cost of transportation from collecting center $l$ to revival center $m$ using transportation mode $l_6$ in period $t$ ;
$A_{mjl_7t}$	Unit cost of transportation from revival center $m$ to DC $_j$ using transportation mode $l_7$ in period $t$ ;
$P_i$	Production cost of each unit of product in plant $i$ ;
$P_j$	Process cost of each unit of product in DC $_j$ ;
$P_l$	Process cost of each unit of product in collecting center $l$ ;
$P_m$	Restructuring cost of each unit of product in revival center $m$ ;
$P_n$	Separation cost of each unit of product in recycle center $n$ ;
$V_j$	Holding cost of each unit of inventory in DC $_j$ ;
$Z_j$	Cost of each unit of shortage in DC $_j$ ;
$e_a$	Cost increase factor for production of each unit of product in plant $i$ ;
$e_b$	Cost increase factor for process of each unit of product in DC $_j$ ;
$e_c$	Cost increase factor for process of each unit of product in collecting center $l$ ;
$e_f$	Cost increase factor for restructuring each unit of product in revival center $m$ ;
$e_g$	Cost increase factor for separation of each unit of product in recycle center $n$ ;
$e_v$	Cost increase factor for holding each unit of inventory in DC $_j$ ;

$e_z$	Cost increase factor for shortage of each unit in DC $_j$ ;
$C_{et}$	Material customers' demand $e$ in period $t$ ;
$\gamma_{lt}$	Un-reviving percentage of collected products in each collection center $l$ in period $t$ ;
$w_i$	Capacity of plant $i$ ;
$w_j$	Capacity of DC $_j$ ;
$w_l$	Capacity of collecting center $l$ ;
$w_m$	Capacity of revival center $m$ ;
$w_n$	Capacity of recycle center $n$ ;
$\alpha_{jt}$	Disrepair percentage of products sent from DC $_j$ in period $t$ ;
$\eta_{kt}$	Percentage of supplied demand during period $t$ which is returned by customer demand of zone $k$ ;
$H_{kt}$	Amount of customer's demand of zone $k$ in period $t$ ;
$L_j$	Initial inventory level in DC $_j$ .

### 3.4. Decision variables

$X_{ijl_1t}$	1 if plant $i$ is allocated to DC $_j$ during period $t$ via transportation mode $l_1$ ; and 0, otherwise;
$X_{jkl_2t}$	1 if DC $_j$ is allocated to customer's demand zone $k$ during period $t$ via transportation mode $l_2$ ; and 0, otherwise;
$X_{kll_3t}$	1 if customer's demand zone $k$ is allocated to collecting center $l$ during period $t$ via transportation mode $l_3$ ; and 0, otherwise;
$X_{lnl_4t}$	1 if collecting center $l$ is allocated to recycle center $n$ during period $t$ via transportation mode $l_4$ ; and 0, otherwise;
$X_{nel_5t}$	1 if recycle center $n$ is allocated to material customer $e$ during period $t$ via transportation mode $l_5$ ; and 0, otherwise;
$X_{lm_l_6t}$	1 if collecting center $l$ is allocated to revival center $m$ during period $t$ via transportation mode $l_6$ ; and 0, otherwise;
$X_{mjl_7t}$	1 if revival center $m$ is allocated to DC $_j$ during period $t$ via transportation mode $l_7$ ; and 0, otherwise;
$U_i^d$	1 if plant $i$ is opened with capacity level $d$ ; and 0, otherwise;
$U_j^d$	1 if DC $_j$ is opened with capacity level $d$ ; and 0, otherwise;

$U_l^d$	1 if collecting center $l$ is opened with capacity level $d$ ; and 0, otherwise;
$U_m^d$	1 if revival center $m$ is opened with capacity level $d$ ; and 0, otherwise;
$U_n^d$	1 if recycle center $n$ is opened with capacity level $d$ ; and 0, otherwise;
$O_{ijt}$	Amount of product transported from plant $i$ to DC $_j$ during period $t$ ;
$O_{jkt}$	Amount of product transported from DC $_j$ to customer's demand zone $k$ during period $t$ ;
$O_{klt}$	Amount of product transported from customer's demand zone $k$ to collecting center $l$ during period $t$ ;
$O_{lmt}$	Amount of product transported from collecting center $l$ to revival center $m$ during period $t$ ;
$O_{ln t}$	Amount of product transported from collecting center $l$ to recycle center $n$ during period $t$ ;
$O_{mjt}$	Amount of product transported from revival center $m$ to DC $_j$ during period $t$ ;
$O_{net}$	Amount of product transported from recycle center $n$ to material customer $e$ during period $t$
$Q_{jt}$	Inventory level of DC $_j$ in period $t$ ;
$E_{jt}$	Shortage of DC $_j$ in period $t$ ;
$Z_{kt}$	Amount of supplied demand of customer in zone $k$ in period $t$ .

### 3.5. Mathematical model

$\text{Min}f_1$  is calculated as shown in Box I, and  $\text{Min}f_2$  is obtained by the following equation:

$$\begin{aligned}
 \text{Min}f_2 = & \sum_{d \in D_i} \sum_i F_i^d U_i^d + \sum_{d \in D_j} \sum_j F_j^d U_j^d \\
 & + \sum_{d \in D_l} \sum_L F_l^d U_l^d + \sum_{d \in D_m} \sum_M F_m^d U_m^d \\
 & + \sum_{d \in D_n} \sum_N F_n^d U_n^d + \sum_i \sum_j \sum_{l_1} \sum_t A_{ijl_1 t} X_{ijl_1 t} \\
 & + \sum_j \sum_k \sum_{l_2} \sum_t A_{jkl_2 t} X_{jkl_2 t} \\
 & + \sum_k \sum_l \sum_{l_3} \sum_t A_{kll_3 t} X_{kll_3 t} \\
 & + \sum_L \sum_M \sum_{l_6} \sum_t A_{lm l_6 t} X_{lm l_6 t} \\
 & + \sum_M \sum_j \sum_{l_7} \sum_t A_{mjl_7 t} X_{mjl_7 t} \\
 & + \sum_L \sum_N \sum_{l_4} \sum_t A_{ln l_4 t} X_{ln l_4 t} \\
 & + \sum_N \sum_e \sum_{l_5} \sum_t A_{nel_5 t} X_{nel_5 t} \\
 & + \sum_i \sum_j \sum_t P_i O_{ijt} (1 + e_a)^t \\
 & + \sum_j \sum_k \sum_t P_j O_{jkt} (1 + e_b)^t \\
 & + \sum_L \sum_M \sum_t P_l O_{lmt} (1 + e_c)^t
 \end{aligned}$$

$$\min f_1 = \max_{k,t} \left\{ \begin{aligned} & \max_{j,l_2} \{X_{jkl_2 t} T_{jkl_2 t}\} + \left[ \sum_j \sum_{l_2} X_{jkl_2 t} \right] \times \max_{i,l_1} \{X_{ijl_1 t} T_{ijl_1 t}\} \\ & + \max_{l,l_3} \{X_{kll_3 t} T_{kll_3 t}\} + \left[ \sum_l \sum_{l_3} X_{kll_3 t} \right] \\ & \times \max_l \left\{ \begin{aligned} & \left[ \max_{N,l_4} \{X_{ln l_4 t} T_{ln l_4 t}\} + \left[ \sum_N \sum_{l_4} X_{ln l_4 t} \right] \times \max_{e,l_5} \{X_{nel_5 t} T_{nel_5 t}\} \right], \\ & \left[ \max_{M,l_6} \{X_{lm l_6 t} T_{lm l_6 t}\} + \left[ \sum_M \sum_{l_6} X_{lm l_6 t} \right] \times \max_{j,l_7} \{X_{mjl_7 t} T_{mjl_7 t}\} \right] \end{aligned} \right\} \end{aligned} \right\},$$

Box I

$$+ \sum_L \sum_N \sum_t P_l O_{ln\ t} (1 + e_c)^t \quad \sum_{d \in D_l} U_l^d \leq 1 \quad \forall l, \quad (14)$$

$$+ \sum_M \sum_j \sum_t P_m O_{mj\ t} (1 + e_f)^t \quad \sum_{d \in D_m} U_m^d \leq 1 \quad \forall m, \quad (15)$$

$$+ \sum_N \sum_e \sum_t P_n O_{ne\ t} (1 + e_g)^t \quad \sum_{d \in D_n} U_n^d \leq 1 \quad \forall n, \quad (16)$$

$$+ \sum_j \sum_t Q_{jt} V_j (1 + e_v)^t + \sum_j \sum_t E_{jt} Z_j (1 + e_z)^t, \quad \sum_i \sum_{l_1} O_{ij\ t} X_{ij\ l_1\ t} + \sum_M \sum_{l_7} O_{mj\ t} X_{Mj\ l_7\ t} + Q_{j(t-1)}$$

s.t.:

$$\sum_L \sum_{l_3} X_{kl\ l_3\ t} = 1 \quad \forall k, t, \quad (1)$$

$$\sum_N \sum_{l_4} X_{ln\ l_4\ t} \leq 1 \quad \forall l, t, \quad (2)$$

$$\sum_M \sum_{l_6} X_{lm\ l_6\ t} \leq 1 \quad \forall l, t, \quad (3)$$

$$\sum_j \sum_{l_2} X_{jk\ l_2\ t} = 1 \quad \forall k, t, \quad (4)$$

$$\sum_N \sum_{l_5} O_{ne\ t} X_{ne\ l_5\ t} \geq C_{et} \quad \forall e, t, \quad (5)$$

$$\sum_L \sum_{l_3} O_{kl\ t} X_{kl\ l_3\ t} \geq \eta_{kt} Z_{kt} \quad \forall k, t, \quad (6)$$

$$\sum_{l_3} \sum_k \gamma_{lt} O_{kl\ t} X_{kl\ l_3\ t} = \sum_N \sum_{l_4} O_{ln\ t} X_{ln\ l_4\ t} \quad \forall l, t, \quad (7)$$

$$\sum_{l_3} \sum_k (1 - \gamma_{lt}) O_{kl\ t} X_{kl\ l_3\ t} = \sum_M \sum_{l_6} O_{lm\ t} X_{lm\ l_6\ t} \quad \forall l, t, \quad (8)$$

$$\sum_j \sum_{l_7} O_{mj\ t} X_{mj\ l_7\ t} = \sum_L \sum_{l_6} O_{lm\ t} X_{lm\ l_6\ t} \quad \forall m, t, \quad (9)$$

$$\sum_{l_5} \sum_e O_{ne\ t} X_{ne\ l_5\ t} \geq \sum_L \sum_{l_4} O_{ln\ t} X_{ln\ l_4\ t} \quad \forall n, t, \quad (10)$$

$$\sum_j \sum_{l_1} O_{ij\ t} X_{ij\ l_1\ t} \leq \sum_{d \in D_i} U_i^d w_i \quad \forall i, t, \quad (11)$$

$$\sum_{d \in D_i} U_i^d \leq 1 \quad \forall i, \quad (12)$$

$$\sum_{d \in D_j} U_j^d \leq 1 \quad \forall j, \quad (13)$$

$$\leq \sum_{d \in D_j} U_j^d w_j \quad \forall j, t, \quad (17)$$

$$\sum_{d \in D_j} U_j^d w_j \geq \sum_k \sum_{l_2} O_{jk\ t} X_{jk\ l_2\ t} \quad \forall j, t, \quad (18)$$

$$\sum_k \sum_{l_3} O_{kl\ t} X_{kl\ l_3\ t} \leq \sum_{d \in D_l} U_l^d w_l \quad \forall l, t, \quad (19)$$

$$\sum_L \sum_{l_6} O_{lm\ t} X_{lm\ l_6\ t} \leq \sum_{d \in D_m} U_m^d w_m \quad \forall m, t, \quad (20)$$

$$\sum_L \sum_{l_4} O_{ln\ t} X_{ln\ l_4\ t} \leq \sum_{d \in D_n} U_n^d w_n, \quad \forall n, t, \quad (21)$$

$$\sum_M \sum_{l_7} X_{mj\ l_7\ t} = \sum_{d \in D_j} U_j^d \quad \forall j, t, \quad (22)$$

$$\begin{aligned} & \sum_M \sum_{l_7} O_{mj\ t} X_{mj\ l_7\ t} + \sum_i \sum_j O_{ij\ t} X_{ij\ l_1\ t} + Q_{j(t-1)} \\ &= \sum_k \sum_{l_2} O_{jk\ t} X_{jk\ l_2\ t} + Q_{jt}, \quad \forall j, t, \end{aligned} \quad (23)$$

$$\begin{aligned} & \sum_k \sum_{l_2} (1 - \alpha_{jt}) O_{jk\ t} X_{jk\ l_2\ t} \geq E_{j(t-1)} \\ &+ \sum_{l_2} \sum_k \eta_{k(t-1)} Z_{k(t-1)} X_{jk\ l_2\ t} \quad \forall j, t, \end{aligned} \quad (24)$$

$$\begin{aligned} & Q_{j(t-1)} + \sum_i \sum_{l_1} O_{ij\ t} X_{ij\ l_1\ t} + \sum_M \sum_{l_7} O_{mj\ t} X_{mj\ l_7\ t} \\ &- \sum_k \sum_{l_2} H_{kt} X_{jk\ l_2\ t} - E_{j(t-1)} \\ &- \sum_{l_2} \sum_k \eta_{k(t-1)} Z_{k(t-1)} X_{jk\ l_2\ t} = Q_{jt} - E_{jt}, \end{aligned}$$

$$\forall j, t, \quad (25)$$

$$Q_{jt} \times E_{jt} = 0 \quad \forall j, t, \quad (26)$$

$$Q_{jT} = 0 \quad \forall j, t, \quad (27)$$

$$E_{jT} = 0 \quad \forall j, t, \quad (28)$$

$$Q_{jT_0} = L_{0j} \quad \forall j, t, \quad (29)$$

$$Q_{jt} \leq \sum_{d \in D_j} U_j^d w_j \quad \forall j, t, \quad (30)$$

$$\begin{aligned} \sum_{l_3} \sum_k O_{kl} X_{kl_3t} &= \sum_M \sum_{l_6} O_{lmt} X_{lml_6t} \\ &+ \sum_N \sum_{l_4} O_{ln} X_{lnl_4t} \quad \forall l, t \end{aligned} \quad (31)$$

$$\sum_{l_1} X_{ijl_1t} \leq 1 \quad \forall i, j, t, \quad (32)$$

$$\sum_{l_2} X_{jkl_2t} \leq 1 \quad \forall j, k, t, \quad (33)$$

$$\sum_{l_3} X_{kll_3t} \leq 1 \quad \forall k, l, t, \quad (34)$$

$$\sum_{l_4} X_{lnl_4t} \leq 1 \quad \forall l, n, t, \quad (35)$$

$$\sum_{l_5} X_{nel_5t} \leq 1 \quad \forall n, e, t, \quad (36)$$

$$\sum_{l_6} X_{lml_6t} \leq 1 \quad \forall l, m, t, \quad (37)$$

$$\sum_{l_7} X_{mjl_7t} \leq 1 \quad \forall m, j, t, \quad (38)$$

$$\sum_j X_{jkl_2t} = Z_{kt} \quad \forall k, t, \quad (39)$$

$$Z_{kt} \leq H_{kt} \quad \forall k, t, \quad (40)$$

$$\begin{aligned} &X_{ijl_1t}, X_{jkl_2t}, X_{kll_3t}, X_{lnl_4t}, X_{nel_5t}, X_{lml_6t}, X_{mjl_7t}, \\ &U_i^d, U_j^d, U_l^d, U_m^d, U_n^d \in [0, 1], \end{aligned} \quad (41)$$

$$O_{ijt}, O_{jkt}, O_{lmt}, O_{lnl_4t}, O_{mjt}, O_{net}, Q_{jt}, E_{jt} \geq 0. \quad (42)$$

In this model, the first objective function tries to find the minimum time to transport products along any path in the closed-loop supply chain. The second objective function minimizes the fixed cost for opening centers with a certain capacity level, transportation and allocation costs, construction, process, restructuring and separation costs, and inventory and shortage costs in a closed-loop supply chain. Constraint (1) shows that each customer's demand zone is allocated to

a collecting center. Constraints (2) and (3) suggest that each collecting center is allocated to a recycle center (in Constraint 2) and a revival center (in Constraint 3) when a collecting center is established. Constraint (4) emphasizes that each customer's demand zone takes service from one DC. Constraint (5) suggests that customer demand for materials is supplied during each period. Constraint (6) shows the amount of returned product from customer  $k$  during period  $t$ . Constraints (7) and (8) show that the returned products from customer  $k$  to collecting center  $l$  are sent to revival center  $m$  and recycle center  $n$ . Constraint (9) shows that inputs and outputs of the revival center are equal. Constraint (10) indicates that recycled product constituents are sold to customers after recycle process. Constraint (11) relates to the plant capacity. Constraints (12) to (16) show that if a center has been established, a capacity level will be allocated to it. Constraints (17) and (18) relate to capacity of  $DC_j$ . Constraints (19) to (21) relate to the capacities of collecting, revival, and recycle centers, respectively. Constraint (22) shows whether  $DC_j$  is established with the capacity level and allocated to a revival center.

Constraint (23) suggests that the summation of period  $(t - 1)$  inventory in  $DC_j$  and products received from recycle centers and plants during period  $t$  is equal to the summation of period  $t$  inventory and products sent to the customers' demand zones. Constraint (24) emphasizes that shortage in  $DC_j$  and impaired sent products should be compensated in the next phase. Constraints (25) and (26) show that it is impossible to have inventory and shortage, simultaneously, in  $DC_j$  in each phase. Constraints (27) and (28) show that in the last period, we do not have inventory and shortage in  $DC_j$ . Constraint (29) suggests that we consider an initial inventory level for DCs. Constraint (30) relates to  $DC_j$  capacity. Constraint (31) indicates that a revivable part of the collected products from customer  $k$  in collecting center  $l$  is allocated to a revival center and a recyclable part of it is allocated to a recycle center. Constraints (32) to (38) relate to using one type of transportation vehicles between two centers at two levels. Constraint (39) shows the amount of supplied demand of customer  $k$ . Constraint (40) says that this amount cannot be greater than the real demand of customer  $k$ . Finally, Constraints (41) and (42) define variables.

#### 4. Bee colony optimization and genetic algorithms

To show applicability and validity of the presented model, we solve several small-scale problems through a branch-and-bound module in GAMS. This software is a robust tool for solving and analyzing mathematical models of linear and nonlinear optimization problems.



To solve large-scale problems, we use the bee algorithm. In order to show its efficiency, the associated results are compared with the results obtained from the genetic algorithm.

#### 4.1. Bee colony optimization algorithm

The proposed structure of the bee colony optimization algorithm is as follows (please see Figure 2):

Purposed bee colony optimization algorithm  
 {Initialization:

Initialize the algorithm parameter.

Generate N feasible solutions as the initial population.

While criterion is meet

Calculate the fitness for each solution in current population.

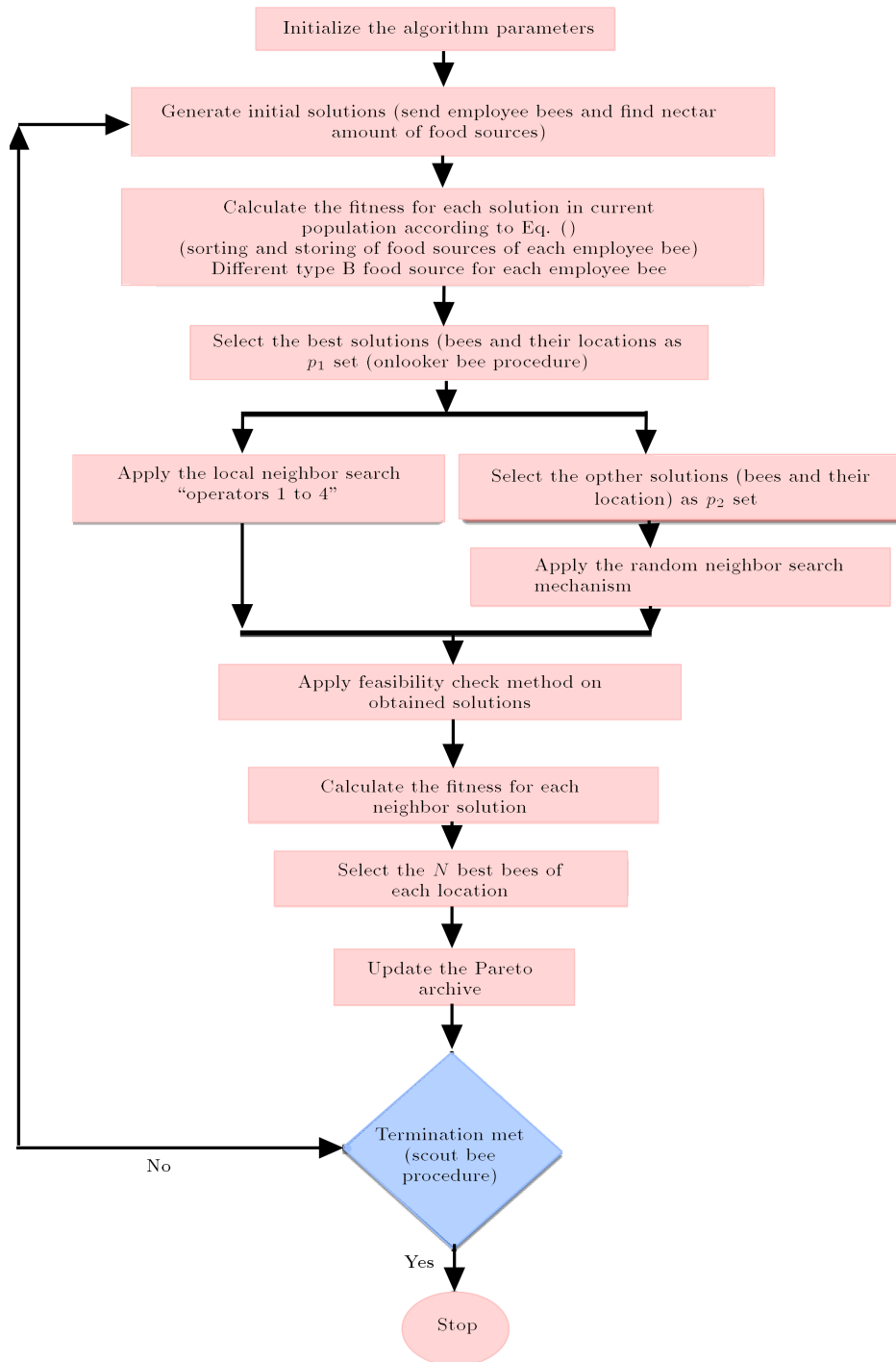


Figure 2. Structure of bee algorithm.

Select the best bees and their location as  $p_1$  set.  
 Select the other bees and their location as  $p_2$  set.  
 Apply neighborhood search operator to  $p_1$  set.  
 Apply feasibility check method to the obtained solutions.  
 Assign some bees to the obtained solutions and calculate their fitness.  
 Apply random neighborhood search operator on  $p_2$ .  
 Apply feasibility check method on obtained solutions.  
 Calculate their fitness.  
 Select the  $N$  best bees of each location.  
 Apply improvement method on selected solutions and take  
 output of this method as population of the next generation.  
 Update Pareto archive  
 End while  
 Return the Pareto archive.  
 }

#### 4.1.1. Solution representation

We use a matrix to present each solution. Each solution includes several matrices which are designed in accordance with model outputs. For example, for variable  $X_{ijl_1t}$ , we define a four-dimensional matrix with dimensions  $I \times J \times l_1 \times T$ . In the same way, we define a matrix for other outputs.

#### 4.1.2. Solution initialization method

Given that the Bee Colony Optimization (BCO) algorithm is a population-based algorithm, at the beginning of the algorithm, we need a population of solutions as the initial population. This paper uses a random process to produce possible initial solutions. Available solutions indicated by  $N$  in each repeat of the BCO process are supposed to be fixed during the optimization process. To produce  $N$  possible initial solutions, the designed process should be repeated  $N$  times.

#### 4.1.3. Applicability of solutions

Since during performing the algorithm new solutions are produced, an approach has been designed to check the applicability of solutions. The algorithm make the infeasible solutions to feasible ones where it checks out all the limitations of the produced solutions. If one or more limitations have been violated in that solution, it tries to make the solution feasible.

#### 4.1.4. Fitness function calculation

In this paper, since the proposed model is a multi-objective one, we calculate the fitness index of each solution, which classifies solutions and calculates the crowding distance criterion of solutions [22]. The fitness value of individual  $c$ , considering its rank and

objective values, can be calculated by [23]:

$$\text{Fitness}_c = \frac{1}{\sum_{j=1}^D \left[ \frac{f_j(c)}{\sum_{i=1}^{N_{\text{rank}(c)}} f_j(i)} \right]} + (\text{rank}(c) - 1) \times D, \quad (43)$$

where  $D$  is the number of objectives,  $f_j(i)$  is the value of the  $j$ th objective of the  $i$ th individual, and  $N_{\text{rank}(c)}$  is the number of solutions in  $\text{rank}(c)$ .

#### 4.1.5. Local search (bee $P_1$ group)

To solve the studied problem, a new process is designed based on local search.  $P_1$  group solutions population is the input of this process. This process bases upon neighborhood search. In other words, this process receives group of solutions as input and tries to reach suitable neighbor solutions by recovering each solution. In the present paper, we use a multi-operator local search process as well as a repeatable local search process with guided mutation to design the above process [24]. By combining these two local search processes, we try to present a new process of local search. A general structure of the proposed operator is presented below:

```
{
Step 1: Get input solutions ( $p_1$ ).
Step 2:
    For each solution in  $p_1$  set, use the multi-operator
    search to enhance solutions.
    Construct the pool of trial solutions generated in
    the step before.
}
```

In this approach, a multi-operator search process is executed on each available solution in  $P_1$  group and the result is a set of local optimized solutions, which are in the neighborhood of the aforesaid solution. All local optimized solutions obtained from execution of this process are pooled. For recovering solutions, their neighborhood is searched and analyzed. Neighborhood search operators used here include an exchange operator that will be executed on four location-related matrices. These operators work as follows:

**Operator 1:** Two indices  $i$  and  $j$  are constructed randomly during consistent periods  $\{1, \dots, n\}$  (i.e.,  $n$  factories) and  $\{1, \dots, m\}$  (i.e.,  $m$  distribution centers); index  $t$  is produced during period  $\{1, \dots, T\}$ , and if square  $X_{ijl_1t}$  is equal to zero, it will be considered 1 and if it is 1, it will be considered zero; also, a stream of materials between factory  $i$  and distributor  $j$  during period  $t$  by vehicle  $l_1$  is changed according to capacities. Then, a modification process is executed on the solution matrices and it modifies them according to model limitations.

**Operator 2:** Two indices  $k$  and  $l$  are constructed randomly during consistent periods  $\{1, \dots, k\}$  (i.e.,  $k$  customer regions) and  $\{1, \dots, l\}$  (i.e.,  $l$  collecting centers); index  $t$  is produced during period  $\{1, \dots, T\}$  and if square  $X_{kll_3t}$  is equal to zero, it will be considered 1 and if it is 1, it will be considered zero; also a stream of materials between customer  $k$  and collecting center  $l$  during period  $t$  by vehicle  $l_3$  is changed according to capacities. Then, a modification process is executed on the solution matrices and it modifies them according to model limitations.

**Operator 3:** Two indices  $l$  and  $m$  are constructed randomly during consistent periods  $\{1, \dots, l\}$  (i.e.,  $l$  collecting centers) and  $\{1, \dots, m\}$  (i.e.,  $m$  revival centers); index  $t$  is produced during period  $\{1, \dots, T\}$ , and if square  $X_{lm_l6t}$  is equal to zero, it will be considered 1 and if it is 1, it will be considered zero; also stream of materials between collecting center  $l$  and revival center  $m$  during period  $t$  by vehicle  $l_6$  is changed according to capacities. Then, a modification process is executed on the solution matrices and it modifies them according to model limitations.

**Operator 4:** Two indices  $n$  and  $l$  are constructed randomly during consistent periods  $\{1, \dots, n\}$  (i.e.,  $n$  recycle centers) and  $\{1, \dots, l\}$  (i.e.,  $l$  collecting centers), index  $t$  is produced during period  $\{1, \dots, T\}$ , and if square  $X_{ln_l4t}$  is equal to zero, it will be considered 1 and if it is 1, it will be considered zero; also stream of materials between centers  $l$  and  $n$  during period  $t$  by vehicle  $l_4$  is changed according to capacities. Then, a modification process is executed on the solution matrices and it modifies them according to model limitations.

Each of these operators search a part of neighbor solutions and other parts may be searched by other operators; thus, using one neighborhood search operator may lead to losing some potential neighbor solutions. Therefore, this study uses a multi-operator search process, which will be explained later. In a multi-operator search process, we use the above four operators to produce neighbor solutions. A multi-operator search process algorithm is presented below:

{Multi operator search framework

1. For each input solution  $x$ , set  $p_{\text{approx}} = \{x\}$
2. Repeat
3. Randomly select some  $x$  in  $p_{\text{approx}}$  for which  $nh(x)$  has not been investigated yet
4. For all  $NR = \{nh_1, nh_2, \dots, nh_r\}$   
Generate neighborhood  $nh_i(x)$
5. Update  $p_{\text{approx}}$  with all elements  $x_{nh}$  in  $nh_i(x)$
6. If  $x$  in  $p_{\text{approx}}$ , then
7. Mark  $x$  as ‘investigated’

8. End if
  9. Until no element as  $x$  in  $p_{\text{approx}}$  with  $x$  still to be investigated
  10. Return  $p_{\text{approx}}$
- }

In the defined search process, as can be seen in its algorithm, a set of solutions ( $P_{\text{approx}}$ ) is constructed, which includes initial solutions. In each repeat of this process, one member that has not been analyzed so far is chosen randomly and neighbor solutions are produced by the aforementioned four neighborhood search operators. Then, approx set  $P$  is updated through comparing solutions with non-dominated relations and this cycle will be repeated until all members are analyzed. At the end of the algorithm, approx set  $P$  that includes a local optimized solution in the neighborhood of an initial solution is returned as the algorithm result.

#### 4.1.6. Random neighborhood search ( $P_2$ group)

To execute random neighborhood search for the second bee group, this study uses a parallel neighborhood search operator. This process uses four operators mentioned in the previous section simultaneously or in a parallel way. If we indicate those operators with  $ls_1, ls_2, ls_3, ls_4$  symbols, the structure of parallel process is as follows:

{For input solution  $s$ :

- $S_1 = ls_1(s)$ ;
- $S_1$ =Apply feasibility check method on  $s_1$ ;
- $S_2 = ls_2(s)$ ;
- $S_2$ =Apply feasibility check method on  $s_2$ ;
- $S_3 = ls_3(s)$ ;
- $S_3$ =Apply feasibility check method on  $s_3$ ;
- $S_4 = ls_4(s)$ ;
- $S_4$ = Apply feasibility check method on  $s_4$ ;
- Output solution = acceptance( $s, s_1, s_2, s_3, s_4$ ).

}

As can be seen in the above structure, for each existing solution in  $P_2$ , each operator will be executed separately on solution and, ultimately, among outputs and inputs of the four operators, one will be chosen, given non-dominated relations, and will be reported as process output.

#### 4.1.7. Selection

In each repeat, the algorithm needs a population of solutions. In this study, for choosing the next repeat population, the existing solutions in the current repeat population and new solutions produced by the algorithm are pooled and after classification and calculation of the crowding distance criterion for each solution given its level, through Deb’s formula [22],  $N$  solutions

which have the highest quality and highest variance will be chosen as the next repeat population of the algorithm.

#### 4.1.8. Improvement structure

In the proposed structure of the bee colony optimization algorithm, an improvement structure is designed which is executed on the selected solutions in the previous repeat to improve them. Output solutions of the improvement structure are chosen as the next repeat population of the algorithm. Execution of the improvement process bases upon Variable Neighborhood Search (VNS), as will be explained later. The VNS structure uses four neighborhood search structures. These structures are local search operators, explained in the previous section. These four structures are used in the context of VNS, whose general structure (i.e., pseudo-code) is as follows:

```

{For each input solution
  K = 1
  While stopping criterion is met, do
    New solution=Apply NSS type k
    If new solution is better than K = 1
      Else,
        K = k + 1
        If k = 5 then
          K = 1
        End if
      End if
    End if
  End while
}
```

The VNS algorithm receives all the available solutions in a population and gives an output solution. Then, a recovery process is executed on other solution matrices and after recovery, it replaces the input solution. In fact, a general structure of the recovery process is as follows:

```

Improvement method
{For each  $s_i$  in input population
   $S_i$ =apply VNS procedure on  $s_i$ ;
   $S_i$ =check feasibility method.
}
```

## 4.2. Genetic algorithm

The proposed general structure of the genetic algorithm is as follows:

```

Genetic algorithm
{Initialization:
  Initialize algorithm parameters;
  Generate  $N$  feasible solution as the initial population.
  While stopping criterion is met, do
```

```

    Calculate fitness for solution in the current population
    Calculate number of solutions ( $nc$ ) for crossover operator.
    Counter=0;
    While counter $\leq nc$  do
      Select two parents.
      Counter=counter+2.
      Apply crossover operator on the selected parents.
    End while
    Calculate number of solutions ( $nm$ ) for mutation operator
    Counter=0;
    While counter $\leq nm$  do
      Select one solution that has not been selected so far.
      Apply mutation operator on the selected solution.
      Counter=counter+1;
    End while
    Apply reproduction operator on the other solutions in population that has not been selected so far.
    Select  $N$  best solutions as the population of the next generation.
  End while
  Return the best solution.
}
```

#### 4.2.1. Establishing an initial population

Initial solutions are produced randomly.

#### 4.2.2. Parent selection method

This study uses a roulette wheel method for selecting parents. The probability of parallel selection with each chromosome is calculated in terms of its fitness. If  $f_k$  is the fitness value of chromosome  $k$ , the probability of parallel survival with that chromosome is as follows:

$$P_k = \frac{f_k}{\sum_{i=1}^n f_i}. \quad (44)$$

Now, we arrange chromosomes according to  $P_k$  and  $q_k$ , which are a cumulative frequency of  $P_k$ , calculated by:

$$q_k = \sum_{i=1}^k P_i. \quad (45)$$

This method simulates a roulette wheel in order to determine which members have a chance to reproduce. Each member in terms of its conformity receives some parts of the rolling wheel. Then, in each phase, one member is selected and this process is subjected to repeat until enough pairs are selected for producing the next generation.

#### 4.2.3. Mutation operator

For executing the mutation operator used in the genetic algorithm, this study uses a parallel neighborhood search structure explained in the bee algorithm.

#### 4.2.4. Crossover operator

The crossover operator designed in this algorithm is a single point crossover operator, which is executed on all solution matrices.

#### 4.2.5. Fitness function

Calculation of the fitness value of each solution is similar to the bee algorithm.

#### 4.2.6. Selecting a population for the next repeat

At the end of each repeat, among the solutions of that repeat and the new produced solutions, we select  $n$  solutions with the highest fitness value as a population of the next repeat (or generation).

### 5. Designing experiment and results

Given the considered hypotheses and parameters of the proposed mathematical model, we define several small, medium, and large-scale problems, randomly. In order to analyze efficiency of the proposed algorithm, we execute it in the MATLAB software environment and results of the execution on experimental problems are compared with the results of GAMS software precise calculations. Comparisons base upon a criterion, which is the gap between target function and execution time of each process.

#### 5.1. Comparison criteria

To solve the presented model, we propose and execute the bee colony algorithm and the genetic algorithm. However, this model is solved in the GAMS software environment. Given that this model is bi-objective, in order to solve the model with GAMS software, we consider the weight composition of targets. To compare the results obtained by the algorithms and GAMS software, we use comparison criterion that shows the gap between target functions. This criterion is explained later.

Eqs. (46) and (47), representing the distance between the proposed optimization algorithms, are used as efficiency criterion. This criterion shows validity of the developed algorithms.

$$\text{error}(\%) = \frac{(\text{ans.BCO} - \text{ans.GAMS})}{\text{ans.GAMS}}, \quad (46)$$

$$\text{error}(\%) = \frac{(\text{ans.GA} - \text{ans.GAMS})}{\text{ans.GAMS}}. \quad (47)$$

This criterion is defined as the gap between the objective function values (i.e., weight composition of model objectives) of the algorithm and GAMS software. Eq. (46) calculates the gap between the bee colony algorithm and GAMS software. In this formula, we have:

- ans. BCO: objective function values of the bee colony algorithm;

- ans. GAMS: objective function values of solving the model by GMS software;
- % error: the gap between two objective function values of the bee colony algorithm and GAMS software.

Eq. (47) calculates the gap between the objective function values of the genetic algorithm and GAMS software. In this formula, we have:

- ans. BCO: objective function values of the genetic algorithm;
- ans. GAMS: objective function values of solving the model by GMS software;
- % error: the gap between two objective function values of the genetic algorithm and GAMS software.

In addition to the above criterion and weight composition of objectives, the results of these algorithms are compared based on comparison indices of multi-objective problems on the basis of Pareto archive. There are many different indices for analyzing quality and variance of multi-objective innovative algorithms. This study considers three comparison criteria presented as follows [25]:

**Quality metric:** This metric compares the quality of Pareto solutions in each process. Indeed, quality metric classifies all the obtained Pareto solutions in both algorithms and indicates what percentage of high level solutions belongs to each process. Higher percentage shows that algorithm has a high quality.

**Spacing metric:** This metric analyzes the consistency of Pareto solution distribution around solutions border. This index is defined by:

$$s = \frac{\sum_{i=1}^{N-1} |d_{\text{mean}} - d_i|}{(N-1) \times d_{\text{mean}}}.$$

In this relation,  $d_i$  indicates the Euclidean distance between two adjacent non-subdued solutions and  $d_{\text{mean}}$  shows the average  $d_i$  values.

**Diversification metric:** This metric is used to determine non-subdued solution rate on the optimized border and is defined by:

$$D = \sqrt{\sum_{i=1}^N \max(\|x_t^i - y_t^i\|)}.$$

In this relation,  $\|x_t^i - y_t^i\|$  indicates the Euclidean distance between two adjacent solutions  $x_t^i$  and  $y_t^i$  on the optimized border.

**Table 2.** Sizes of some existing problems in the literature.

References	No. of plants	No. of DCs	No. of customer's centers	No. of collecting centers	No. of revival centers
[26]	-	-	100	40	30
[27]	20	30	500	-	-
[28]	-	-	120	35	-
[29]	3	-	-	80	20
[30]	30	40	100	40	30

**Table 3.** Small, medium and large-scale problems.

Size	Instance	No. of plants	No. of DCs	No. of customer's centers	No. of collecting centers	No. of revival centers	No. of recycle centers	No. of period
Small	1	2	4	5	3	2	3	1
	2	2	4	5	3	2	3	2
	3	2	4	5	4	2	3	1
	4	2	4	7	4	2	3	2
	5	2	4	7	4	2	3	3
Medium	1	3	7	7	7	5	4	1
	2	3	7	7	7	5	4	2
	3	3	7	7	7	5	4	3
	4	6	8	10	8	6	5	1
	5	6	8	10	8	6	5	2
	6	6	8	10	8	6	5	3
	7	7	9	15	9	7	7	1
	8	7	9	15	9	7	7	2
	9	7	9	15	9	7	7	3
Large	1	10	20	30	16	7	6	1
	2	10	20	30	16	7	6	2
	3	10	20	30	16	7	6	3
	4	15	40	70	35	12	10	1
	5	15	40	70	35	12	10	2
	6	15	40	70	35	12	10	3
	7	15	45	90	40	15	13	1
	8	15	45	90	40	15	13	2
	9	15	45	90	40	15	13	3

### 5.2. Design of experiments

This study presents different experimental and real problems with different sizes in the area of direct and reverse logistics network design. In order to design and construct experimental problems, we analyze some problems in the literature that are reported in Table 2.

Given the existing sizes in the literature, we consider three groups of problems with small, medium, and large sizes to analyze the efficiency of the proposed algorithms that are presented in Table 3. For designing experimental problems groups, we try to define the

problem size according to the existing area in the previous studies as shown in Table 3.

### 5.3. Parameter setting

- The population size in both algorithms is set to 150;
- Repeat number (stopping criterion) in both algorithms is set to 500;
- In GA, crossover rate is considered equal to 0.7 and mutation rate is set to 0.2;

- Products transportation time between all centers  $\sim U[1, 100]$ ;
- Annual cost of opening centers  $\sim U[1, 40]$ ;
- Transportation cost of each unit between all centers  $\sim U[50, 100]$ ;
- Costs of production, process, rebuild, separation in their related centers  $\sim U[1, 20]$ ;
- Shortage and inventory holding costs in DCs  $\sim U[1, 40]$ ;
- Customers' demand for material e  $\sim U[1, 20]$ ;
- Customers' demand  $\sim U[20, 100]$ ;
- Initial inventory level  $\sim U[1, 500]$ ;
- Capacity of centers in all levels  $\sim U[1, 1000]$ .

#### 5.4. Solution results

As can be seen in Table 4, in the first sample with a small size, error is low and equal to 0.07 per cent for the GA and 0.02 per cent for the BCO algorithm.

Regarding quality, as can be seen, the BCO algorithm can produce higher quality solutions than the GA and GAMS. Regarding time in a proposed structure of the BCO algorithm and using certain search algorithms, time of the BCO algorithm is more than that of the GA and GAMS.

As illustrated in Table 4, one can see that the given problem is more complicated than the existing problems in the literature due to numerous variables. Therefore, GAMS software can solve only small-size problems within an acceptable time, while in large-sized calculation of the exact solution during the reasonable time it is not possible. For medium- and large-sized problems, as can be seen in the BCO algorithm, it produces higher quality, but needs much time for solutions as compared to GA. Given the executive results, we conclude that for solving the presented model, the proposed BCO algorithm is more efficient and robust than the GA. As it was mentioned, the results of both algorithms are compared on the basis of Pareto archive using quality, diversification, and space

**Table 4.** Results of small, medium and large-scale problems.

Size	Instance	OFV			CPU time (sec.)			Gap (%)	
		BCO	GA	GAMS	BCO	GA	GAMS	BCO	GA
Small	1	1181491	1252640	1152521	395.7	365.17	1	0.02	0.07
	2	1226598	1355365	1134969	640.06	578.26	5	0.07	0.16
	3	1794968	2028732	1147734	487.5	410.08	42	0.36	0.43
	4	1831084	3852185	1041230	1129.8	654.33	780	0.43	0.72
	5	3465977	7734957	1701200	1328.05	1012.6	815	0.51	0.78
Medium	1	1387675	1634129	-	1713.8	1467.3	>10800	-	-
	2	1698359	1895608	-	2011.5	1630.09	>10800	-	-
	3	1539384	1845281	-	1949.04	1427.3	>10800	-	-
	4	1787620	1899429	-	2109.01	1511.08	>10800	-	-
	5	2043612	2403157	-	2017.05	1741.7	>10800	-	-
	6	1987630	2134252	-	2314.01	1917.06	>10800	-	-
	7	2529438	2726481	-	2575.1	2113.09	>10800	-	-
	8	2376948	2859374	-	2719.5	2284.2	>10800	-	-
	9	2765888	3098100	-	2638.1	2165.01	>10800	-	-
Large	1	3487397	3728139	-	3542.1	2871.4	>10800	-	-
	2	3295429	3295429	-	3792.8	2659.2	>10800	-	-
	3	3745584	4056711	-	3087.07	2112.3	>10800	-	-
	4	4082775	5417927	-	4172.7	2978.4	>10800	-	-
	5	4107439	5672473	-	4587.2	3891.06	>10800	-	-
	6	4028569	5372876	-	4296.3	3968.5	>10800	-	-
	7	7907703	9173296	-	5819.09	4275.02	>10800	-	-
	8	8013319	9826382	-	5569.6	4922.7	>10800	-	-
	9	7899971	9900763	-	5761.4	5034.06	>10800	-	-

**Table 5.** Results of small, medium and large-scale problems.

Size	Instance	Quality metric		Diversification metric		Spacing metric	
		BCO	GA	BCO	GA	BCO	GA
Small	1	100	0	987.34	503.3	1.22	0.77
	2	70.28	29.72	1002.01	749.09	0.69	0.45
	3	81.09	19.91	985.22	633.7	0.93	0.69
	4	100	0	1122.65	923.43	1.56	0.73
	5	92	8	1034.98	1039.06	0.99	1.01
Medium	1	70	30	1128.3	860.03	1.02	0.44
	2	89.22	10.78	1216.44	999.21	1.11	0.77
	3	91.24	8.76	1198.31	853.54	0.84	0.32
	4	100	0	1487.32	920.16	0.89	0.53
	5	100	0	1398.67	1039.91	0.79	0.49
	6	100	0	1502.65	1156.2	1.22	0.84
	7	77.98	22.02	1739.28	1065.3	0.98	0.61
	8	98.2	1.8	1792.67	1103.11	0.74	1.34
	9	100	0	1938.65	1384.6	0.94	0.79
Large	1	88.22	11.78	2769.9	1384.6	0.78	0.66
	2	100	0	2970.1	1269.5	0.93	0.78
	3	91.03	8.97	3186.4	1877.52	1.22	0.94
	4	100	0	3068.4	2310.21	1.32	1.21
	5	79.45	20.55	3748.14	2449.2	0.87	1.03
	6	80.6	19.4	3722.87	2706.06	0.69	0.44
	7	100	0	4194.5	2514.4	0.93	0.59
	8	100	0	4557.8	2963.7	0.95	0.84
	9	100	0	4937.54	3056.5	0.77	0.29

comparison metrics. The results of these comparisons based on these three metrics are reported in Table 5.

As illustrated in Table 4, in all cases, the proposed BCO algorithm can produce solutions with higher quality and higher diversification than the GA. Regarding the space metric, in most cases, the obtained solutions by the genetic algorithm are more consistent than the obtained solutions of the proposed BCO, which is the disadvantage of BCO.

## 6. Conclusion

This study has presented a new mathematical model to design a closed-loop supply chain considering several periods, inventory and shortage in distribution centers, and time and cost of transportation. This problem has been formulated as a bi-objective integer nonlinear programming model, whose applicability has been analyzed by solving several small-sized problems by GAMS software. Because this problem is NP-hard, for solving medium- and large-sized problems, a

Bee Colony Optimization (BCO) algorithm has been proposed and its results have been compared with the results obtained by the Genetic Algorithm (GA). The computational results have indicated that the proposed BCO algorithm produces higher quality solutions than the GA; however, its CPU time is not less than that of the GA. Some useful comparison metrics (i.e., quality, space, and diversity metrics) have been applied to validate the efficiency of the proposed BCO algorithm. The results have shown that the proposed BCO algorithm produces solutions with higher quality and diversification than the GA; therefore, it is more efficient. The experimental results have indicated that our proposed algorithm outperforms the GA. There are some areas for future research in this paper. A deterministic model has been developed in this research. It is valuable to consider uncertain parameters in the model and examine the effects of uncertainty on the results. Furthermore, it is worthwhile to apply other multi-objective solution approaches in the literature to solve the model and compare the results.



## Acknowledgements

The authors would like to thank the Editor-in-Chief of “Scientia Iranica” and anonymous referees for their helpful comments and suggestions, which greatly improved the presentation of this paper. Additionally, the second author would like to acknowledge the partial financial support of University of Tehran for this research under Grant No. 8106043/1/25.

## References

- Blumberg, D.F. “Introduction to management of reverse logistics and closed loop supply chain processes”, *Business & Management*, CRC Press, Florida, USA (2005).
- Amaroa, A.C.S. and Barbosa-Povoa, A.P.F.D. “The effect of uncertainty on the optimal closed-loop supply chain planning under different partnerships structure”, *Computers and Chemical Engineering*, **33**(12), pp. 2144-2158 (2009).
- Kannan, G., Sasikumar, P. and Devika, K. “A genetic algorithm approach for solving a closed loop supply chain model: A case of battery recycling”, *Applied Mathematical Modeling*, **34**(3), pp. 655-670 (2010).
- Pishvae, M.S., Rabbani, M. and Torabi, S.A. “A robust optimization approach to closed-loop supply chain network design under uncertainty”, *Applied Mathematical Modeling*, **35**(2), pp. 637-649 (2011).
- Paksoy, T., Bektas, T. and Ozceylan, E. “Operational and environmental performance measures in a multi-product closed-loop supply chain”, *Transportation Research Part E: Logistics and Transportation Review*, **47**(4), pp. 532-546 (2011).
- Zhang, Z.-H., Jiang, H. and Pan, X. “A Lagrangian relaxation based approach for the capacitated lot sizing problem in closed-loop supply chain”, *Int. J. Production Economics*, **140**(1), pp. 249-255 (2012).
- Hassanzadeh Amin, S. and Zhang, G. “An integrated model for closed-loop supply chain configuration and supplier selection: multi-objective approach”, *Expert Systems with Applications*, **39**(8), pp. 6782-6791 (2012).
- Hassanzadeh Amin, S. and Zhang, G. “A multi-objective facility location model for closed-loop supply chain network under uncertain demand and return”, *Applied Mathematical Modeling*, **37**(6), pp. 4165-4176 (2013).
- Subramanian, P., Ramkumar, N., Narendran, T.T. and Ganesh, K. “PRISM: PRiority based simulated annealing for a closed-loop supply chain network design problem”, *Applied Soft Computing*, **13**(2), pp. 1121-1135 (2013).
- Ozceylan, E., Paksoy, T. and Bektas, T. “Modeling and optimizing the integrated problem of closed-loop supply chain network design and disassembly line balancing”, *Transportation Research Part E: Logistics and Transportation Review*, **61**, pp. 142-164 (2014).
- Tavakkoli-Moghaddam, R., Forouzanfar, F. and Ebrahimnejad, S. “A new multi-objective model for a three-level supply chain problem with inventory and transportation times”, *Proc. of the International Conference on Advances in Supply Chain and Manufacturing Management (ASCMM)*, pp. 16-18 (2011).
- Guide Jr., V.D. R. and Van Wassenhove, L.N. “The evolution of closed-loop supply chain research”, *Operations Research*, **57**(1), pp. 10-18 (2009).
- Krikke, H., Hofenkc, D. and Wang, Y. “Revealing an invisible giant: A comprehensive survey into return practices within original (closed-loop) supply chains”, *Resources, Conservation and Recycling*, **73**, pp. 239-250 (2013).
- Stindt, D. and Sahamie, R. “Review of research on closed loop supply chain management in the process industry”, *Flexible Services and Manufacturing Journal*, **26**(1-2), pp. 268-293 (2014).
- Govindan, K., Soleimani, H. and Kannan, D. “Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future”, *European Journal of Operational Research*, **240**(3), pp. 603-626 (2015).
- Pham, D.T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S. and Zaidi, M. “The bees algorithm - A novel tool for complex optimisation problems”, *2nd International Virtual Conference on Intelligent Production Machines and Systems*, pp. 454-459 (2006).
- Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989).
- Soleimani, H. and Kannan, G. “A hybrid particle swarm optimization and genetic algorithm for closed-loop supply chain network design in large-scale networks”, *Appl. Math. Modelling*, **39**(14), pp. 3990-4012 (2015).
- Kannan, G., Sasikumar, P. and Devika, K. “A genetic algorithm approach for solving a closed loop supply chain model: A case of battery recycling”, *Applied Mathematical Modelling*, **34**(3), pp. 655-670 (2010).
- Min, H., Ko, C.S. and Ko, H.-J. “The spatial and temporal consolidation of returned products in a closed-loop supply chain network”, *Computers & Industrial Engineering*, **51**(2), pp. 309-320 (2006).
- Tavakkoli-Moghaddam, R., Forouzanfar, F. and Ebrahimnejad, S. “Incorporating location, routing and inventory decisions in a bi-objective supply chain design problem with risk-pooling”, *Journal of Industrial Engineering International*, **9**(19), pp. 1-6 (2013).
- Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. “A fast and elitist multi objective genetic algorithm: NSGA-II”, *IEEE Transactions on Evolutionary Computation*, **6**(2), pp. 182-197 (2002).
- Enayatifar, R., Yousefi, M., Abdullah, A.H. and Darus, A.N. “MOICA: A novel multi-objective approach based on imperialist competitive algorithm”, *Applied Mathematics and Computation*, **219**(17), pp. 8829-8841 (2013).

24. Qingfu, Zh. And Jianyong, S. "Iterated local search with guided mutation", *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 924-929 (2006).
25. Tavakkoli-Moghaddam, R., Azarkish, M. and Sadeghnejad-Barkousaraie, A. "A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem", *Expert Systems with Applications*, **38**(9), pp. 10812-10821 (2011).
26. Jayaramann, V. and Srivastava, R. "A closed-loop logistics model for remanufacturing", *Journal of the Operational Research Society*, **50**(5), pp. 497-508 (1999).
27. Aras, N., Aksen, D. and Tanugur, A.G. "Locating collection centers for incentive-dependent returns under a pick-up policy with capacitated vehicles", *European Journal of Operational Research*, **191**(3), pp. 1223-40 (2008).
28. Gen, M., Altiparmak, F. and Lin, L. "A genetic algorithm for two-stage transportation problem using priority-based encoding", *Operations Research*, **28**(3), pp. 337-354 (2006).
29. Yeh, W.-C. "A hybrid heuristic algorithm for the multistage supply chain network problem", *Int. J. Adv. Manuf. Technol.*, **26**(5-6), pp. 675-685 (2005).
30. Jayaraman, V. and Ross, A. "A simulated annealing methodology to distribution network design and management", *European Journal of Operational Research*, **144**(3), pp. 629-645 (2003).

## Biographies

**Fatemeh Forouzanfar** is currently a PhD student under the supervision of Prof. Reza Tavakkoli-Moghaddam at the College of Industrial Engineering, Iran University of Science and Research in Tehran. She obtained her MSc in Industrial Engineering from the Islamic Azad University of South Tehran Branch in Iran (2011) and her BSc in Industrial Engineering from Iran University of Science and Technology, Behshahr Campus (2007). Her research interests include supply chain and closed-loop supply chain, mathematical modeling, and location-routing-inventory problems.

**Reza Tavakkoli-Moghaddam** is a Professor of Industrial Engineering at the College of Engineering, University of Tehran in Iran. He obtained his PhD in

Industrial Engineering from Swinburne University of Technology in Melbourne (1998), his MSc in Industrial Engineering from the University of Melbourne in Melbourne (1994), and his BSc in Industrial Engineering from Iran University of Science and Technology in Tehran (1989). He serves as a member of Editorial Board in five reputable academic journals. He is the recipient of the 2009 and 2011 Distinguished Researcher Awards and the 2010 and 2014 Distinguished Applied Research Awards at University of Tehran, Iran. He has been selected as the National Iranian Distinguished Researcher in 2008 and 2010 by the Ministry of Science, Research, and Technology (MSRT) in Iran. He obtained the outstanding rank as the top 1% scientist and researcher in the world elite group, reported by Thomson Reuters in 2014. He has published 4 books, 15 book chapters, and more than 600 journal and conference papers.

**Mahdi Bashiri** is an Associate Professor of Industrial Engineering at Shahed University. He holds a BS in Industrial Engineering from Iran University of Science and Technology, MS, and PhD in this field from Tarbiat Modares University. He is the recipient of the 2013 Young National Top Scientist Award from the Academy of Sciences of the Islamic Republic of Iran. His research interests are facilities planning, meta-heuristics, and multi-response optimization. He published about 19 books and more than 160 papers in reputable academic journals and conferences.

**Armand Baboli** is an Associate Professor in the Department of Industrial Engineering and DISP (Decision and Information for Production System) Laboratory of INSA (National Institute of Applied Sciences) Lyon at the University de Lyon, France. He received his MSc and PhD degrees from Institut National Polytechnique de Grenoble in France, in 1994 and 1999, respectively. His current researches focus on supply chain management (networks configuration, organization, and coordination) under uncertainty, facilities design (layout and location), dynamic cellular manufacturing system, production planning and inventory control, and intelligent manufacturing system with mobile robots. He has published more than 110 papers in journals and conferences.