J. Appl. Res. Ind. Eng. Vol. 7, No. 3 (2020) 238–266

Journal of Applied Research on Industrial Engineering
www.journal-aprie.com

Design a Green Closed Loop Supply Chain Network by Considering Discount under Uncertainty

Javid Ghahremani-Nahr[1], Hamed Nozari[2, *], Seyyed Esmaeil Najafi[3]
[1]*Faculty member of ACECR, Development and Planning Institute, Tabriz, Iran.*
[2]*Department of Industrial Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran.*
[3]*Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.*

PAPER INFO

Chronicle:
Received: 11 May 2020
Reviewed: 29 May 2020
Revised: 28 July 2020
Accepted: 17 August 2020

Keywords:
Green Closed Loop Supply Chain.
Discount.
Meta-Heuristic Algorithms.
Robust Possibilistic Optimization Approach.
Uncertainty.

ABSTRACT

Mathematical model of a multi-product multi-period multi-echelon closed-loop supply chain network design under uncertainty is designed in this paper. The designed network consists of raw material suppliers, plants, warehouses, distribution centers and customer zones in forward chain and collection centers, repair centers, recovery/decomposition center and disposal center in reverse chain. The goal of the model is to determine the quantities of products and raw material transported between the supply chain entities in each period by considering different transportation mode, the number and locations of the potential facilities, the shortage of products in each period, and the inventory of products in warehouses and plants with considering discount and uncertainty parameters. The robust possibilistic optimization approach was used to control the uncertainty parameter. At the end to solve the proposed model, five meta-heuristic algorithms include genetic algorithm, bee colony algorithm, simulated annealing, imperial competitive algorithm and particle swarm optimization are utilized. Finally, some numerical illustrations are provided to compare the proposed algorithms. The results show the genetic algorithm is efficient algorithm for solve the designed model in this paper.

1. Introduction

Because of cost and environmental concerns, technology development, global market competition, companies in the regular integration in all manufacturing processes and transportation of raw materials to customer needs. Nowadays, Supply Chain Management (SCM) has received a lot of attention in several organizations [1]. SCM is described as the design, production, organizing, execution, control, and testing regarding supply chain activities with the goal of creating net value, minimizing the logistics cost, creating a competitive infrastructure, synchronizing demand with supply, leveraging worldwide supply chain, as well as measuring effectiveness globally [2].

Ghahremani-Nahr, J., Nozari, H., & Najafi, S. E. (2020). Design a green closed loop supply chain network by considering discount under uncertainty. *Journal of applied research on industrial engineering*, *7*(3), 238-266.

* Corresponding author
E-mail address: ham.nozari.eng@iauctb.ac.ir
10.22105/jarie.2020.251240.1198

An efficient and effective supply chain is a competitive advantage for companies and plants and helps them to cope with the pressure of the global market. There are two kinds of supply chain: forward supply chains and reverse supply chains [3]. The forward supply chain is defined as the activities converting raw material to products, storing and distributing products to customers, while the reverse supply chain, including of a series of activities of the collection, inspection, repair, recovery and dispose of used products. The integration of reverse and forward supply chain creates a Closed-Loop Supply Chain (CLSC) [4]. In the other words, there are both forward and reverse supply chain network in closed-loop supply chain networks. Network design is one of the most important strategic decisions in supply chain management. In general, supply chain network design decisions include determining the numbers and locations of facilities and the quantity of flow between them. In recent years, a few of articles have focused on integrated forward and reverse network design that, this type of integration can prevent the sub-optimality, increases the level of network performance and coordination between forward and reverse processes [5]. Due to increased environmental impacts and their important role in human life, reduction of impacts made by human has attracted more attention, recently. Green supply chains are among the most effective issues related to environmental impacts and increased number of studies in this area verifies this opinion [6].

Additionally, environmental regulations and customers' increasing expectations leave companies no choice but applying the return policy. In this regard, designing a supply chain network incorporating the reverse flow can provide environmental requirements in addition to economic objectives [7].

This paper offers a Mixed-Integer Nonlinear Programming (MINLP) mathematical model for closed-loop supply chain network design. In the designed model, tactical decisions, including quantity discount and back order shortage is considered. In this paper, against the other papers, selected facilities in each period, can be opened and closed by considering respective costs. Also, the designed model, taking into account the different transportation modes, is trying to minimize logistics costs. To solve the designed model, five meta-heuristic algorithms includes Genetic Algorithm (GA), Bee Colony Algorithm (ABC), Simulated Annealing (SA), Imperial Competitive Algorithm (ICA) and Particle Swarm Optimization (PSO) are utilized and robust possibilistic optimization approach is used to control the uncertainty parameters.

2. Literature Review

In recent years, with regard to the rising importance of reverse and closed-loop supply chain network design, numerous articles in this area are published. Govindan et al. [8] presented a literature review for the closed-loop and reverse supply chain. They classified the 382 papers published between 2007_2013 based on the year of release, solving approach, objective functions and so on [8]. They stated, 49.7% of the total supply chain problem related to the closed-loop supply chain network design and 39.7% of them related to the reverse supply chain network design. Also in this study, 12.4% of published papers related to the single-objective and 87.6% of them related to the multi-objective models.

One of the most important research considered in the scope of reverse and closed-loop supply chain, is supply chain network configuration. This field, includes the coordination and integration of key business activities, including activities from the purchase of raw materials to distribution of finished products to customers. Many authors, have used the facility location models for the formulation of closed-loop supply chain networks.

Fleischmann et al. [9] proposed a Mixed-Integer Linear Programming (MILP) model for the design closed-loop supply chain network. Copier remanufacturing and paper recycling are utilized to show the efficiency of the model.

Üster et al. [10] considered a multi-product closed-loop supply chain network design problem. The model can determine the optimal numbers and locations of collection centers and remanufacturing facilities to minimize the total network costs include processing, transportation, and fixed location costs [10]. Lee and Dong [11] developed a two stage stochastic programming model which is based on Simulated annealing based heuristic algorithm. It provides an efficient framework for identifying and statistically solving the large scale dynamic network problems. The model minimizes total logistics costs [11]. Lee et al. [12] developed a mixed-integer linear mathematical model for the design closed-loop supply chain network. The proposed heuristics based GA is applied as a solution methodology. Also the model can determine the optimal numbers of potential facilities [12].

Kannan et al. [13] proposed a multi-stage, multi-period, multi-product integrated forward/reverse logistics network model for returned-product which is based on genetic algorithm based heuristic algorithm. Khajavi et al. [14] developed an integrated forward/reverse logistics network optimization model for multi-echelon capacitated supply chain network. The objectives are minimization of the total network costs and maximization of responsiveness of the network [14]. Das and Chowdhury [15] presented a new mathematical model for reverse supply chain network. They used an approach in the proposed model that uses retail outlets as a two-way channel for making new products, collecting the returned-products and recovered products as a way of promoting an effective product recovery system in supply chain operation and optimizing costs [15].

Mahmoudi et al. [16] developed an integer linear programming model for multi-layer, multi-product reverse supply chain to minimize the total costs of products and parts transportation costs among center. Dönmez [17] developed a mixed-integer linear mathematical model for the design of reverse supply chain network to minimize total recycling costs. Ramezani et al. [18] proposed a multi-objective stochastic mathematical model for a closed-loop supply chain network design with responsiveness and quality level. Özceylan et al. [19] developed a mixed integer nonlinear programming model that optimizes the tactical decisions on balancing decomposition lines in the reverse supply chain and the strategic decisions related to the quantity of products flowing on the closed-loop supply chains. Soleimani et al. [20] proposed a stochastic integer linear programming model for design multi-product closed-loop supply chain when there are uncertain demand, the purchase price and the rate of return. Rezaee et al. proposed a stochastic supply chain model that demand considered uncertain [21].

Alavi et al. [22] investigated the integration of location, allocation, inventory, and production decisions in the design of supply chain network. The model presented by them was a nonlinear mixed-integer programming to reduce the cost of network design. In this paper, the model aimed to determine the optimal number of potential facilities and the optimal allocation of the product flow between facilities. They used Imperialist Competitive Algorithm (ICCA) and Tabu Search (TS) algorithm to solve their nonlinear model [22]. Nobil and Taleizadeh [23] proposed a distribution-production and inventory planning model under uncertainty by maximizing the net present value in the supply chain network. They designed a closed-loop supply chain network with multiple suppliers, multiple manufacturing plants, and customers, which aimed to reduce the costs of network design taking into account the increase in the net present value. They used a fuzzy programming approach to control their uncertain parameters [23]. Talaei et al. [24] designed a sustainable supply chain network under uncertainty to reduce the total costs of the network and reduce CO2 emissions. They used fuzzy-robust optimization

to control uncertain parameters such as demand and transportation costs. The model developed by them was implemented on a case study of the printing industry, and the outputs of the paper indicated the relative efficiency of the fuzzy-robust method in the optimization of the problem [24]. Zhalechian et al. [25] designed a sustainable closed-loop supply chain network under uncertainty. They aimed to determine the optimal location-routing and inventory variables to reduce CO2 emissions. They also used possibilistic robust programming method to solve the problem [25].

Ciccullo et al. [26] examined the types of lean supply chain networks, agile supply chains, and sustainable supply chains. They evaluated 73 articles waiting until 2017 to demonstrate the aspects, similarities, and differences of each of the supply chains [26]. Das designed a multi-objective sustainable lean supply chain network [27]. He used four conflicting objectives in his model, which include maximizing the overall sustainability index, maximizing supply chain profits, reducing greenhouse gas emissions, and reducing total energy consumed. Ultimately, to solve his model, he used goal programming [27]. Haddadsisakht et al. [28] designed a supply chain network under different transportation modes. In this paper, they conceived demand as an uncertain parameter and solved the problem using a scenario-based approach with considering different probabilities. They used Cplex software to solve the problem [28].

Farrokh et al. [29] designed a closed-loop supply chain network under uncertainty. The model designed by them sought to reduce the total cost of designing the supply chain network. To achieve this, they sought to optimize the number and optimal location of the facilities as well as the optimal rate of flow of products between the facilities. They also used robust-fuzzy programming method to control the uncertain parameters of their model and represented that the efficacy of this approach in the control of uncertain parameters is better than other methods [29]. Darbari et al. [30] designed a supply chain network model for a laptop manufacturer company in India. In this model, they considered the demand parameter as an uncertain parameter and controlled the parameter by the fuzzy programming approach. The purpose of their paper was to maximize the profits of the entire network and reduce Co2 emissions [30]. Ghahremani et al. [31] designed a closed-loop supply chain network that could open and close facilities at different periods. They used robust-fuzzy optimization method to control their uncertain parameters and solved their NP-Hard model with Whale Optimization Algorithm [31]. Samuel et al. developed a deterministic mathematical model and its robust variant are proposed to investigate the effects of the quality of returns on the CLSC network under the Carbon Cap (CC) and Carbon Cap-And-Trade (CCT) policies [32]. Gholizadeh et al. [33] designed a closed loop supply chain network and used robust optimization method to control the uncertainty parameters for disposal appliances.

The remainder of this paper is as follows: In Section 3, mathematical formulation of the designed model is presented, in Section 4 the solution approaches are explained. Section 5 contains numerical results for a set of design problems of different sizes. Finally, the conclusions and future researches of this article are given in Section 6.

3. Model Description and Formulation

The designed closed-loop supply chain network in this paper is a multi-product, multi-period, multi-echelon network including raw material suppliers, plants, customer zones, warehouses and distribution centers in forward chain and collection centers, repair centers, recovery/decomposition center and disposal center in reverse chain, as well as all facilities with limited capacities. As it is illustrated in *Fig. 1*, the raw materials shipped from raw material supplier to plants for production, then products shipped

from plants to warehouses. The products are shipped to customer through distribution centers. The used products are collected in collection centers and after inspection the repairable products are shipped to repair center, and recoverable products are shipped to recovery/decomposition centers. In the repair center, the repaired products are shipped to distribution centers or warehouses. The recoverable products are collected and inspected in recovery/decomposition centers and after testing and decomposition the usable raw materials are shipped to plants and scrapped raw materials are shipped to disposal centers.

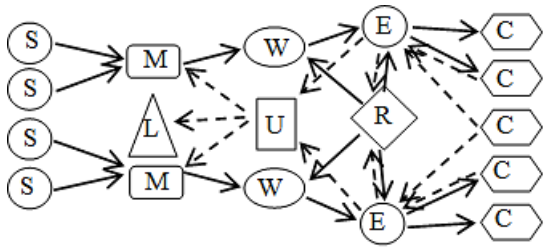

Fig. 1. *The proposed closed-loop supply chain network.*

To specify the study scope, ten assumptions are provided in the designed model as follows:

- Plants, purchasing the raw materials required to produce products at a discount from the raw material suppliers.
- All the warehouses and plants have limited capacities to store products and raw materials.
- All the facilities have limited and identified capacities.
- Shortage in the form of back order can happen at customer zones.
- All the transportation mode have unlimited capacities.
- Customer demand must be satisfied until the last period.
- Distribution and collection centers are considered a hybrid centers.
- Demand, Transportation Costs are considered uncertainty.
- To control the uncertainty parameter robust box method is used.
- The total amount of greenhouse gas emissions from product transfers is limited.

With the above assumptions in mind, the main issues to be addressed by this study are to choose the location and determine the number of a raw material supplier, plant, potential warehouse, distribution/collection, repair, recovery/decomposition and disposal centers that represent the degree of centralization of the network, and then determine the quantity of flows between each pair of network facilities.

3.1. Sets

S	Set of raw material suppliers $(s = 1,2,\dots,S)$.
M	Set of plants $(m = 1,2,\dots,M)$.
W	Set of potential warehouses $(w = 1,2,\dots,W)$.
E	Set of distribution/collection centers $(e = 1,2,\dots,E)$.
C	Set of customer zone $(c = 1,2,\dots,C)$.
R	Set of repair centers $(r = 1,2,\dots,R)$.
U	Set of recovery/decomposition centers $(u = 1,2,\dots,U)$.
L	Set of disposal centers $(l = 1,2,\dots,L)$.

T	Set of periods $(t = 1,2,\dots,T)$.
P	Set of products $(p = 1,2,\dots,P)$.
I	Set of raw materials $(i = 1,2,\dots,I)$.
H	Set of discount level $(h = 1,2,\dots,H)$.
N	Set of Transportation mode $(n = 1,2,\dots,N)$.

Let $\vartheta^f = (G^1, A', A'')$ and $\vartheta^r = (G^2, A''', A'''')$, where G is the graph of nodes and A is the graph arcs . The following definitions, parameters can be defined.

$G^1 = \{S \cup M \cup W \cup E \cup C\}$;

$A' = \{(j,j')|(i \in M, j \in W) \cup (i \in W, j \in E) \cup (i \in E, j \in C)\}$;

$A'' = \{(j,j')|(i \in S, j \in M)\}$;

$G^1 = \{C \cup E \cup R \cup U \cup L\}$;

$A''' = \left\{(j,j')\middle|\begin{matrix}(i \in C, j \in E) \cup (i \in E, j \in R) \cup (i \in E, j \in U) \\ \cup (i \in R, j \in E) \cup (i \in R, j \in W)\end{matrix}\right\}$;

$A'''' = \{(j,j')|(i \in U, j \in L) \cup (i \in U, j \in M)\}$;

$G = \{G^1 \cup G^2\} - C$;

$A_1 = \{A'' \cup A''''\}$;

$A_2 = \{A' \cup A'''\}$.

3.2. Parameters

fi_{jt}	The fixed cost of facility $j \in G$ in period t.
op_{jt}	The opening cost of facility $j \in$ in period t.
cl_{jt}	The closing cost of facility $j \in G$ in period t.
$\widetilde{tc}_{jj'in}$	Unit transportation cost of raw material i between facilities $(j,j') \in A_1$ with transportation mode n.
$\widetilde{tc}_{jj'pn}$	Unit transportation cost of product p between facilities $(j,j') \in A_2$ with transportation mode n.
h_{mit}	Unit inventory holding cost of raw material i by plant m in period t.
h'_{wpt}	Unit inventory holding cost of product m by potential warehouse w in period t.
pr_{shit}	Unit purchase cost of raw material i by raw material supplier s with discount level h in period t.
va_{shit}	Lower bound of interval discount raw material i by raw material supplier s with discount level h in period t.
$c1_{mpt}$	Unit production of product p by plant m in period t.
$c2_{ept}$	Unit distribution cost of product m by distribution/collection center e in period t.
$c3_{ept}$	Unit collection cost of returned product p by distribution/collection center e in period t.
$c4_{rpt}$	Unit repair cost of product p by repair center r in period t.
$c5_{upt}$	Unit recovery/decomposition cost of product p by recovery/decomposition center u in period t.

$c6_{lit}$	Unit disposal cost of raw material i by disposal center l in period t.
π_{cpt}	Unit shortage cost of product p in supplying demand of customer zone c in period t.
δ_{ip}	The number raw material i needed to produce a unit of product p.
$\widetilde{De}_{cpt}$	Demand of product p of customer zone c in period t.
α_{cpt}	Fraction of returned product p from customer zone c in period t.
β_{pt}	Fraction of repairable product p in period t.
γ_{pt}	Fraction of repaired product p in period t that sends to distribution/collection center.
θ_{it}	Fraction of usable raw material i in period t.
cap_{si}	Capacity of raw material supplier s of raw material i.
cap_{mi}	Capacity of plant m of raw material i.
cap_{mp}	Capacity of plant m of product p.
cap_{wp}	Capacity of potential warehouse w of product p.
cap_{ep}	Capacity of distribution center e of product p.
cap'_{ep}	Capacity of distribution center e of product p.
cap_{rp}	Capacity of repair center r of product p.
cap_{up}	Capacity of recovery/decomposition center u of product p.
cap_{li}	Capacity of disposal center l of raw material i.
B_j	Maximum needed facilities $j \in G$ in any period.
$co2_{jj'n}$	Greenhouse gas emissions between facilities $(j,j') \in A_1 and\ A_2$ with transportation mode n.
$Tco2$	Total allowable greenhouse gas emissions.

3.3. Decision Variable

$X_{jj'itn}$	Quantity of raw material i shipped between facilities $(j,j') \in A_1$ with transportation mode n in period t.
$X_{jj'ptn}$	Quantity of product p shipped between facilities $(j,j') \in A_2$ with transportation mode n in period t.
VQ_{mit}	Quantity of raw material i stored at plant m in period t.
IQ_{wpt}	Quantity of product p stored at potential warehouse w in period t.
sh_{cpt}	Quantity of non-satisfied demand of proudest p of customer c in period t.
Y_{jt}	1 if facilities $j \in G$ is open in period t; 0 otherwise.

3.4. Model Formulation

The mathematical model of the problem can be presented as follows.

$$\min z = \sum_{j\in G}\sum_{t\in T}\left(fi_{jt}\,Y_{jt} + op_{jt}\,Y_{jt}\,(1-Y_{j,t-1}) + cl_{jt}\,Y_{jt}\,(1-Y_{j,t+1})\right) +$$
$$\sum_{(j,j')\in A_1}\sum_{i\in I}\sum_{t\in T}\sum_{n\in N}\widetilde{tc}_{jj'in}X_{jj'itn} + \sum_{(j,j')\in A_2}\sum_{p\in P}\sum_{t\in T}\sum_{n\in N}\widetilde{tc}_{jj'pn}X_{jj'ptn} +$$
$$\sum_{j\in W}\sum_{p\in P}\sum_{t\in T}h'_{jit}IQ_{jpt} + \sum_{j\in m}\sum_{i\in I}\sum_{t\in T}h_{jit}VQ_{jit} \tag{1}$$
$$+\sum_{j\in S}\sum_{j'\in M}\sum_{i\in I}\sum_{t\in T}\sum_{n\in N}\sum_{h\in H}pr_{jhit}A_{jhit}X_{jj'itn}$$

$$+\sum_{j\in M}\sum_{j'\in W}\sum_{p\in P}\sum_{t\in T}\sum_{n\in N} C_{1_{jpt}}X_{jj'ptn} + \sum_{j\in E}\sum_{j'\in C}\sum_{p\in P}\sum_{t\in T}\sum_{n\in N} C_{2_{jpt}}X_{jj'ptn} +$$

$$\sum_{j\in E}\sum_{j'\in C}\sum_{p\in P}\sum_{t\in T}\sum_{n\in N} C_{3_{jpt}}X_{j'jptn} + \sum_{j\in E}\sum_{j'\in R}\sum_{p\in P}\sum_{t\in T}\sum_{n\in N} C_{4_{j'pt}}X_{jj'ptn} +$$

$$\sum_{j\in E}\sum_{j'\in U}\sum_{p\in P}\sum_{t\in T}\sum_{n\in N} C_{5_{j'pt}}X_{jj'ptn} + \sum_{j\in U}\sum_{j'\in L}\sum_{i\in I}\sum_{t\in T}\sum_{n\in N} C_{6_{j'it}}X_{jj'itn} +$$

$$\sum_{j\in C}\sum_{p\in P}\sum_{t\in T} \pi_{jpt}sh_{jpt}$$

s.t.

$$A_{jhit}\, VA_{jhit} \le Q_{jit}, \quad \forall\, j \in S, h, i, t \tag{2}$$

$$\sum_{h\in H} A_{jhit} = Y_{jt}, \quad \forall\, j \in S, i, t \tag{3}$$

$$Q_{jit} = \sum_{j'\in M}\sum_{n\in N} X_{jj'itn}, \quad \forall\, j \in S, i, t \tag{4}$$

$$\sum_{j\in S}\sum_{n\in N} X_{jj'itn} + \sum_{j\in U}\sum_{n\in N} X_{jj'itn} + VQ_{j'i,t-1} - VQ_{j'it} = \sum_{j\in W}\sum_{n\in N}\sum_{p\in P} X_{j'jptn}\delta_{ip}, \quad \forall\, j' \in M, i, t \tag{5}$$

$$\sum_{j\in M}\sum_{n\in N} X_{jj'ptn} + \sum_{j\in R}\sum_{n\in N} X_{jj'ptn} + IQ_{j'p,t-1} - \sum_{j\in E}\sum_{n\in N} X_{j'jptn} = IQ_{j'pt}, \quad \forall\, j' \in W, p, t \tag{6}$$

$$\sum_{j\in W}\sum_{n\in N} X_{jj'ptn} + \sum_{j\in R}\sum_{n\in N} X_{jj'ptn} = \sum_{j\in C}\sum_{n\in N} X_{j'jptn}, \quad \forall\, j' \in E, p, t \tag{7}$$

$$\sum_{j\in E}\sum_{n\in N} X_{jj'ptn} = \widetilde{De}_{j'p,t} + sh_{j'p,t-1} - \underset{t<T}{sh_{j'pt}}, \quad \forall\, j' \in C, p, t \tag{8}$$

$$\alpha_{j'pt}\sum_{j\in E}\sum_{n\in N} X_{jj'p,t-1,n} = \sum_{j\in E}\sum_{n\in N} X_{j'jptn}, \quad \forall\, j' \in C, p, t \tag{9}$$

$$\beta_{pt}\sum_{j\in C}\sum_{n\in N} X_{jj'ptn} = \sum_{j\in R}\sum_{n\in N} X_{j'jptn}, \quad \forall\, j' \in E, p, t \tag{10}$$

$$(1-\beta_{pt})\sum_{j\in C}\sum_{n\in N} X_{jj'ptn} = \sum_{j\in U}\sum_{n\in N} X_{j'jptn}, \quad \forall\, j' \in E, p, t \tag{11}$$

$$\gamma_{pt}\sum_{j\in E}\sum_{n\in N} X_{jj'ptn} = \sum_{j\in E}\sum_{n\in N} X_{j'jptn}, \quad \forall\, j' \in R, p, t \tag{12}$$

$$(1-\gamma_{pt})\sum_{j\in E}\sum_{n\in N} X_{jj'ptn} = \sum_{j\in W}\sum_{n\in N} X_{j'jptn}, \quad \forall\, j' \in R, p, t \tag{13}$$

$$\theta_{it}\sum_{j\in E}\sum_{n\in N}\sum_{p\in P} X_{jj'ptn}\delta_{ip} = \sum_{j\in M}\sum_{n\in N}\sum_{p\in P} X_{j'jitn}, \quad \forall\, j' \in U, i, t \tag{14}$$

$$(1-\theta_{it})\sum_{j\in E}\sum_{n\in N}\sum_{p\in P} X_{jj'ptn}\delta_{ip} = \sum_{j\in L}\sum_{n\in N}\sum_{p\in P} X_{j'jitn}, \quad \forall j' \in U, i, t \tag{15}$$

$$\sum_{j\in M}\sum_{n\in N} X_{jj'itn} \le cap_{ji}Y_{jt}, \quad \forall j \in S, i, t \tag{16}$$

$$VQ_{jit} \le cap_{ji}Y_{jt}, \quad \forall j \in M, i, t \tag{17}$$

$$\sum_{j'\in W}\sum_{n\in N} X_{jj'ptn} \le cap_{jp}Y_{jt}, \quad \forall j \in M, p, t \tag{18}$$

$$IQ_{jpt} \le cap_{jp}Y_{jt}, \quad \forall j \in W, p, t \tag{19}$$

$$\sum_{j\in w}\sum_{n\in N} X_{jj'ptn} + \sum_{j\in R}\sum_{n\in N} X_{jj'ptn} \le cap_{j'p}Y_{j't}, \quad \forall j' \in E, p, t \tag{20}$$

$$\sum_{j\in C}\sum_{n\in N} X_{jj'ptn} \le cap'_{j'p}Y_{j't}, \quad \forall j' \in E, p, t \tag{21}$$

$$\sum_{j\in E}\sum_{n\in N} X_{jj'ptn} \le cap_{j'p}Y_{j't}, \quad \forall j' \in U, p, t \tag{22}$$

$$\sum_{j\in E}\sum_{n\in N} X_{jj'ptn} \le cap_{j'p}Y_{j't}, \quad \forall j' \in R, p, t \tag{23}$$

$$\sum_{j\in U}\sum_{n\in N} X_{jj'itn} \le cap_{j'i}Y_{j'i}, \quad \forall j' \in L, i, t \tag{24}$$

$$\sum_{t\in T} Y_{jt} \le B_j, \quad \forall\, j \in G \tag{25}$$

$$\sum_{(j,j')\in A_1}\sum_{i\in I}\sum_{t\in T}\sum_{n\in N} Co2_{jj'n}X_{jj'itn} + \sum_{(j,j')\in A_2}\sum_{p\in P}\sum_{t\in T}\sum_{n\in N} Co2_{jj'n}X_{jj'ptn} \le Tco2 \tag{26}$$

$$X_{jj'itn}, VQ_{mit}, Q_{sit} \ge 0, \quad \forall (j, j') \in A_1, i, n, t, s, m \tag{27}$$

$$X_{jj'ptn}, IQ_{wpt}, sh_{cpt} \ge 0, \quad \forall (j, j') \in A_2, p, n, t, w, c \tag{28}$$

$$Y_{jt} \in \{0, 1\}. \quad \forall j \in G, t \tag{29}$$

Objective function (1) minimizes the total logistics cost, which includes fixed, opening and closing costs, transportation costs between facilities, inventory holding costs, operation cost in any facilities (production costs, distribution costs, collection costs, repair costs, recovery /decomposition costs, disposal costs) and shortage costs, respectively.

Constraint (2) specifies that the total purchased raw material, from raw material supplier of discount levels. *Constraint (3)* ensures that if a raw material supplier is selected, only one discount level can be used. *Constraint (4)* states that the total purchased raw material, from raw material suppliers shipped to plants for production. *Constraint (5)* assures the flow balance at the plants, part of raw materials after production, stored in the plant. *Constraints (6) & (7)* assure the flow balance at warehouses and distribution centers. *Constraint (8)* ensures that customer demand must be satisfied until the last period. *Constraint (9)* guarantees the percentage of the used product by the customer, through the collection center, to be collected. *Constrains (10) & (11)* specify that the collection, after collection and inspection of returned products, the repairable product transfers to repair center, and recoverable product transfers to recovery/ decomposition centers. *Constraints (12) & (13)* state in the repair center, the repaired products are shipped to distribution centers or warehouses. *Constraint (14) & (15)* assure the recoverable products are collected and inspected in recovery/decomposition centers and after testing and decomposition the usable raw materials are transferred to plants and scrapped raw materials are transferred to disposal centers. *Constraint (16) – (24)* reflect the capacity constraints, in that no facilities

can transfer more than the pre specified quantity of raw material or product in each period. *Constraint (25)* shows the maximum number of facilities needed in each period. *Constraint (25)* shows the Greenhouse gas emission limit. *Constraints (27) – (29)* express the non-negativity and binary decision variables.

3.5. Robust Possibilistic Optimization Approach

Given the dynamic and swaying nature of some of the important parameters (including demand and transportation costs) determination of which is beyond planning, and the inaccessibility and even unattainability of the historical data required at the design stage, these parameters are mainly estimated based on comments and mental experiences of experts; accordingly, the ambiguous parameters above are formulated as uncertain data in the form of trapezoidal fuzzy numbers as follows (*Fig. 2*).

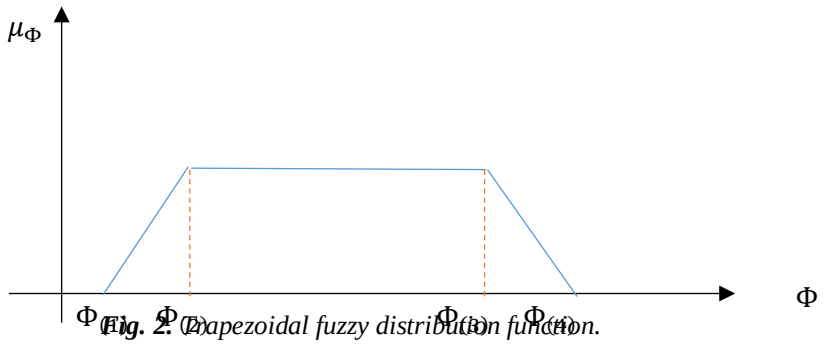

Fig. 2. *Trapezoidal fuzzy distribution function.*

$$\widetilde{De}_{cpt} = \widetilde{tc}_{jj'in} = \widetilde{tc}_{jj'pn} = \left(\Phi_{(1)}, \Phi_{(2)}, \Phi_{(3)}, \Phi_{(4)}\right).$$

It is worth pointing out that for long-term decisions, evaluating a deterministic demand is hard and even sometimes not feasible. Even if one can estimate a probabilistic distribution function for these parameters, these parameters may not have the same behavior with past data. Therefore, the demand for each product in each period, along with the transportation costs that changes in a short-term planning horizon, is considered as fuzzy data.

In addition, the Possibilistic Chance Constrained Programming (PCCP) approach is usually used to deal with uncertain constraints (probabilistic) in which uncertain data is on the left or right side of the equation. If this approach is used, in order to control the level of certainty of imposing these uncertain constraints, the concept of decision can achieve the minimum level of confidence as a confidence level for each of these constraints [34]. To do this, two fuzzy standard methods are commonly used with possibility (Pos) and necessity (Nec) measurment. It is worth noting that Pos indicates the level of optimistic probability of an uncertain event involving uncertain parameters, while the Nec represents a pessimistic decision about an event that is uncertain. However, it is more conservative to use pessimistic fuzzy [35], that is, assuming that decision making has a pessimistic (conservative) attitude to create uncertain limits; therefore, Nec has been used to ensure that uncertainties are established.

At present, based on the obscure parameters mentioned and the use of the expected value for the objective function and the Nec for uncertain constraints, the obvious equivalent of the basic uncertain model can be formulated. To do this, first consider the abbreviation for the proposed model:

$$\begin{aligned}
&\mathrm{Min}Z_1 = Fy + Cx \\
&\mathrm{s.t.} \\
&Ax \geq d, \\
&Bx \leq Sy, \\
&y \in \{0,1\}, x \geq 0.
\end{aligned} \tag{30}$$

Where the vectors C and F represent the variable cost and fixed cost, the vector θ shows Indicates the failure rate of the products and the vector T shows the Sustainable performance indicators. Also, the vectors d and S represent the demand and facility capacities, respectively. At the end A, and B are the matrix of coefficients and finally, x and y the continuous and zero and one variables, respectively. It is assumed that the vectors F and d are presented in the model as non-deterministic parameters. Regarding the general form of non-deterministic limited programming, the expected value of the objective and pessimistic fuzzy functions is to be taken in order to deal with the objective function and the uncertain limit. Now, with the abbreviation, the PCCP base model is as follows:

$$\begin{aligned}
&\mathrm{Min}Z1 = E[Z_1] = Ey + E[\tilde{C}]x, \\
&\mathrm{s.t.} \\
&\mathrm{NEC}\{Ax \geq \tilde{d}\} \geq \alpha, \\
&Bx \leq Sy, \\
&y \in \{0,1\}, x \geq 0.
\end{aligned} \tag{31}$$

Where α controls the minimum degree of certainty of establishing an uncertainty constraint with a Nec approach. Regarding the distribution of the trapezoid probability for uncertainty parameters, the general form of *model (31)* is as follows:

$$\begin{aligned}
&\mathrm{Min}Z1 = E[Z_1] = Fy + \left(\frac{C^1 + C^2 + C^3 + C^4}{4}\right)x, \\
&\mathrm{s.t.} \\
&Ax \geq (1-\alpha)d^3 + \alpha d^4, \\
&Bx \leq Sy, \\
&y \in \{0,1\}, x \geq 0.
\end{aligned} \tag{32}$$

In PCCP models, the minimum level of confidence to establish a non-deterministic constraint should be determined in terms of decision preferences. As can be seen, in the proposed model, the objective function is not sensitive to the deviation from its expected value, which means that achieving robust solutions in the PCCP model is not guaranteed. As a result, a deterministic model of the lean supply chain network can be formulated with location-routing-allocation decisions as follows:

$$\begin{aligned}
&\mathrm{Min}Z1 = E[Z_1] + \xi\left(Z_{1(\max)} - Z_{1(\min)}\right) + \eta_1\left(d^4 - d^3 - \alpha(d^4 - d^3)\right), \\
&\mathrm{s.t.} \\
&Ax \geq (1-\alpha)d^3 + \alpha d^4, \\
&Bx \leq Sy, \\
&y \in \{0,1\}, x \geq 0.
\end{aligned} \tag{33}$$

Where $Z_{1(max)}$ and $Z_{1(\min)}$ can be expressed as follows:

$$f_{1(\max)} = C^4 x. \tag{34}$$

$$f_{1(\min)} = C^1 x. \tag{35}$$

In the *Objective Function (33)*, the first expression refers to the expected value of the first objective function using the mean values of the model uncertainty parameters. The second statement refers to the penalty cost for deviating more than the expected value of the first objective function (robust optimization). The third and fourth sentences also show the total cost of the deviation from the demand (uncertainty parameter). Therefore, the parameter ξ shows the weighting factor objective function, η_1, represent the penalty cost for not estimating demand. The parameter α as the correction coefficients of the fuzzy numbers that must be numeric between 0.5 and 1.

4. Solution Methods

In this section, five meta-heuristic algorithms including GA, ICA, PSO, ABC and SA are proposed for solving the developed closed-loop supply chain network problem. The first section is about the proposed algorithms, and the second section explains the decoding method.

4.1. Genetic Algorithm

Genetic algorithm begins by creating a random initial population of chromosomes while it satisfies the limits or restrictions of the problem. In other words, the chromosomes are strings of values for the decision variables of the problem and each represent a possible problem solution. The chromosomes are derived from the successive iterations called generations. During each generation the chromosomes are evaluated according to objective of the optimization and chromosomes that respond better to the problem are given a greater chance to reproduce the problem solutions. Formulating the chromosomes' evaluation function that increases the speed of convergence of computing to the optimal solution help is important because in the GA the analysis function value should be calculated and since in many cases we are faced with a significant number of chromosomes, time-consuming evaluation function can make the use of genetic algorithm functionally impossible; therefore based on the obtained values of the objective function in a population of strings, each string obtains a fitness value. This fitness value will determine the probability of selection for each string. Based on this probability, first a set of strings is selected. The next generation of new chromosomes called offspring is created by the union of two chromosomes using the crossover operator or through modifying chromosomes using mutation operator. Thus the initial population is replaced by the new strings to make the number of string in computing repetitions constant. Thus the random mechanisms that act on the selection and removal of the strings are such that the strings with higher fitness are more likely to mix and produce new strings and they are more resistive in the replacement stage. Thus the population of sequence in a competition is completed based on the objective function over the different generations and string population is increased by the objective function value so that after several years, the algorithm converges to the best chromosomes which hopefully represent the optimal solution or sub- optimal solution for a given problem. In general, in this algorithm while in each computing iteration new points of solution space are searched by genetic operators, they explore the search process of the areas of space where there is higher average statistical objective function. Usually the new population, which replaces the former population, has higher fitness. This means that the population improves from generation to generation. If we reach the maximum possible generation, gain convergence or the stopping criteria in met, the

search will be successful and as a result the best chromosome obtained from the last generation is selected as the approximate optimal solution or the optimal solution of the problem.

Altiparmak et al. [36] developed a new technique called Priority based genetic algorithm by encrypting and decrypting a dihedral transportation problem and developed a new crossover operator called WMX. In this method, a gene on a chromosome includes two types of information: 1) The gene location (the location of genes in chromosome structure) and 2) the gene (i.e. the amount the gene is embedded in chromosome structure). They used gene location to present a node (source/warehouse) and the value of it to show that node's priority to build transportation tree. Priority based coding was used for planning resource constraints project [37], for multi-level logistics network design [38], for reverse logistics network design [39], for the mixed model of assembly line balancing and sequencing [40] and for routing the shortest path [41].

4.1.1. The method of selecting parents

In selection step a pair of chromosomes is chosen to be combined. The operator selects the relationship between the two generations and transfers some members of the current generation to the next generation. After the selection the selection operators are implemented on the selected members. A member selection criterion is their compliance value while they have a random mode. In evaluation algorithms we deal with the genes. A member function with low compliance, even though it is not a good member of his generation, it may contain good genes and if its selection chance is zero, these good Genes cannot be transferred to the next generation.

Roulette Wheel Selection. The probability of selection of each chromosome is proportional to the amount of fitness. In this method the chromosomes with higher fitness have a greater chance of selection. First the cumulative level of chromosomes' fitness is achieved. Then a random number between zero and total fitness value is produced and thus the chromosome selection depends on the placement of the value within the location of cumulative fitness of chromosome among other chromosomes.

4.1.2. Crossover operator

The most important operator of genetic algorithm is the crossover. Crossover is a process in which the older generation of chromosomes is mixed to create new generations of chromosomes. Pairs which are selected as a parent exchange genes and create new members. Crossing in the genetic algorithm reduces dispersion or genetic diversity because it allows good genes find each other.

Single-Point Crossover. In this method select a point on the two parents. Up to this point the order of the first parent is copied and then the rest of genes that do not exist in the first parent are transmitted by the second parent and the first offspring us generated. In this method each crossover generated two offsprings.

4.1.3. Mutation operator

Mutation is another operator that generated other possible solutions. In the genetic algorithm, after creating a new member in the new population, any gene with the mutation probability is mutated. In the mutation a gene might be removed from the gene population or a gene that has not been in the population might be added to it.

Two-Point Inversion. In this method first two points are randomly considered on the selected parent. Then the values of the genes are inversely written between these two cells.

A flowchart of the GA is included in *Fig. 3*. That presents the overall procedure for solving the problem.

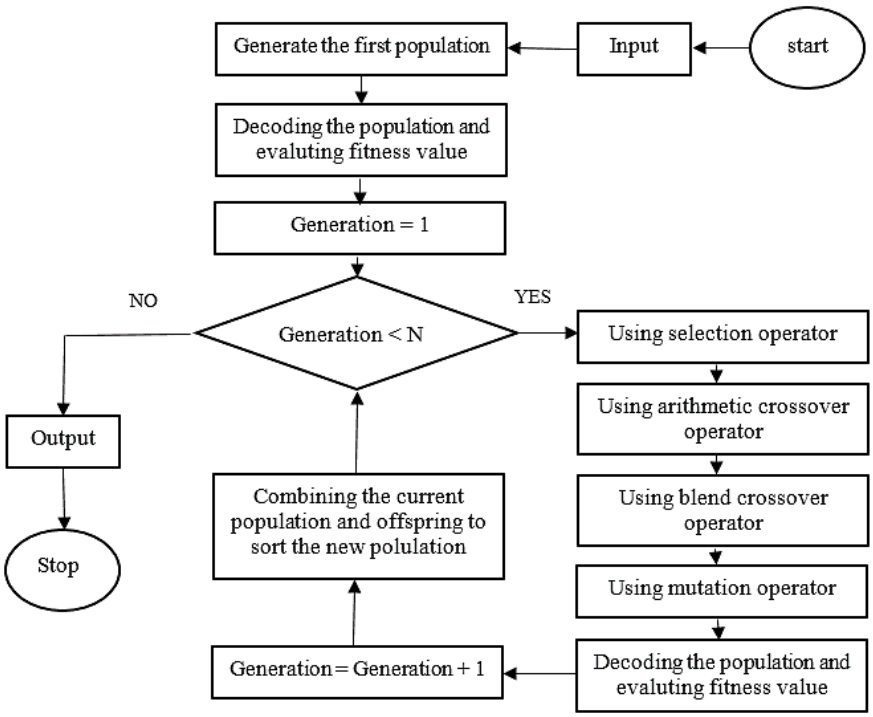

Fig. 3. *Flowchart of the overall procedure of the GA for solving the problem.*

4.2. Imperial Competitive Algorithm

It is a method in the field of evolutionary computation to find the optimal solution of optimization problems and was first presented by Atashpaz-Gargari, and has been widely used by researchers to optimize various problems [42]. The algorithm through socio-political evolution process by mathematical modeling presents an algorithm for solving mathematical problems. In terms of application, this algorithm is in the category of evolutionary optimization algorithms. Similar to all algorithms embedded in this category, this algorithm is started by a number of initial populations. First an array of variables that must be optimized is created. This array is called chromosome in GA and it is called country in this algorithm. In an optimization problem the next N_{var} is presented as an array $1 * N_{var}$.

$$\text{Country} = [p_1, p_2, p_3, \dots, p_N]. \tag{36}$$

In fact, in solving an optimization problem introduced by the algorithm, we look for the best country. Finding this country means finding the best variable that produces the least amount of cost function. Variable values in a country are presented as decimal numbers. To start the algorithm, $N_{country}$ is created. N_{imp} of the best members of the population (the countries with the lowest cost function) are selected as imperialist. The remaining countries as $N_{country}$ form the colonial countries each of which belonging to

an empire. In order to divide the initial colonials among imperialists, each imperialist is given colonies based on its power, the power of an imperialist is defined as the power of the imperialist power, plus a percentage of the total power of the colonies. Each empire that fails to increase its power and loses its competitiveness is removed from the imperialist competition. This removal is done gradually. This means that over the time the weak empires lose their colonies and powerful empires take these colonies and add to their power. In Imperialist Competitive Algorithm the empire being removed is the weakest empire. Thus, in the iteration of the algorithm one or more of the weakest colonies remove the weakest empire and a competition is formed to take these colonials. Colonies are not taken necessarily by the most powerful empire but the stronger empires are more likely to possess them. With the formation of the initial empires, imperialist competition starts between them. Each empire that fails to succeed in imperialist competition and adds to its power (or least reduce the influence) it will be removed from the colonial competition; therefore survival of an empire depends on its ability to attract rival empires colonies and possessing them. As a result, in the imperialist competition, the power of larger empires is gradually increased and the weak empires will be removed. Empires to increase their power would be forced to develop their colonies.

A flowchart of the ICA is included in *Fig. 4*. That presents the overall procedure for solving the problem.

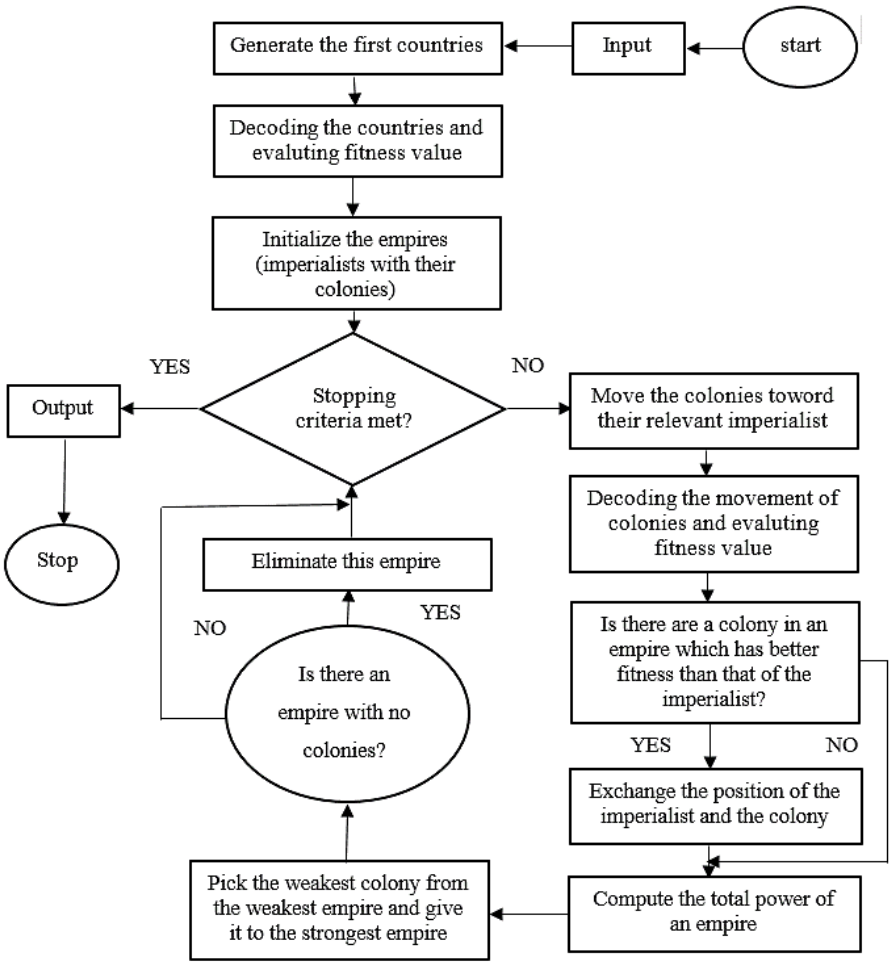

Fig. 4. *Flowchart of the overall procedure of the ICA for solving the problem.*

4.3. Particle Swarm Optimization

Eberhart and Kennedy proposed a method called particle motion by modeling the movement of the birds in the air, finding a rational relationship between the change in direction and speed of birds and using the knowledge of physics [43]. Later these scientists found the dependence of these motions on each other and found that the movement of a bird is caused by the information received from other birds. Therefore they completed the proposed method and called it swarm motion. In general, particle swarm algorithm is similar to ants or genetic algorithms but it has some serious differences with them that makes the difference and the simplicity of the algorithm. For example this algorithm does not use operators such as crossover and mutation thus this algorithm does not require the use of sequences of numbers and decoding stage and it is much easier than genetic algorithms. This algorithm divides the solution space into fragmented routes by a quasi-likelihood function and these routes are formed by the individual motion of particles in space. The motion of a particle group is made up of two main components: A) Deterministic component and B) Probable component. Each particle is interested to move towards the best current solution x^* or the best solution obtained so far g^*. For each moving particle in space, regardless of the fact that it follows the swarm intelligence or not, there are location and velocity vectors. Now for particle *i* (bird) that moves by the swarm intelligence, if the current vector equals x_i, its velocity vector is presented as v_i, According to the equation 30:

$$\mathrm{v}_\mathrm{i}^{\mathrm{t}+1} = \mathrm{v}_\mathrm{i}^{\mathrm{t}} + \alpha\epsilon_1 \odot [\mathrm{g}^* - \mathrm{x}_\mathrm{i}^{\mathrm{t}}] + \beta\epsilon_2 \odot [\mathrm{x}_\mathrm{i}^* - \mathrm{x}_\mathrm{i}^{\mathrm{t}}]. \tag{37}$$

In this equation ϵ_2 and ϵ_1 are random vectors that their elements' values are real numbers between zero and one. Also $\odot$ presents the inner multiplication between two matrixes. The parameters α and β are considered as learning and acceleration parameters. The initial location of particles should be distributed uniformly throughout the space, so that they could be found in most places, i.e. position of particles must be produced with uniform distribution. In addition the velocity of initial change of direction should be considered equal to zero ($v_i^{t=0} = 0$). According to the velocity vector defined in equation 30 the new location vector of each particle will be based on the *Eq. (31)*.

$$\mathrm{x}_\mathrm{i}^{\mathrm{t}+1} = \mathrm{x}_\mathrm{i}^{\mathrm{t}} + \mathrm{v}_\mathrm{i}^{\mathrm{t}+1}. \tag{38}$$

In this equation v_i can take any value in the range $[0, v_{max}]$.

A flowchart of the PSO is included in *Fig. 5*. That presents the overall procedure for solving the problem.

4.4. Bee Colony Algorithm

Bees live in the community in a colony and they store the honey required by colony in their hive. Bees communicate with each other through means such as pheromones secretion or vagel dance. For example, risk warning bees alert other bees by certain pheromones secretion at the time of danger. In addition when a bee is faced by a food source, it takes some of its nectar to the hive then using vagel dance that determined the state, direction and quality of the resource, makes the other ones informed about the food. In the case of understanding the instinctive behavior of bees it is possible to present a different modeling to solve optimization problems. In bees algorithms the employed bees (forager bees) are specialized to maximize the ultimate nectar by finding the food location (place of flowers). In order to optimize the performance of bees allocating the forager bees to the food resources is similar to the

allocation of web-host servers on the Internet that due to this similarity, the issue of allocation was the first problem solved by Nakarani and Tovey with algorithms derived from the behavior of bees [44].

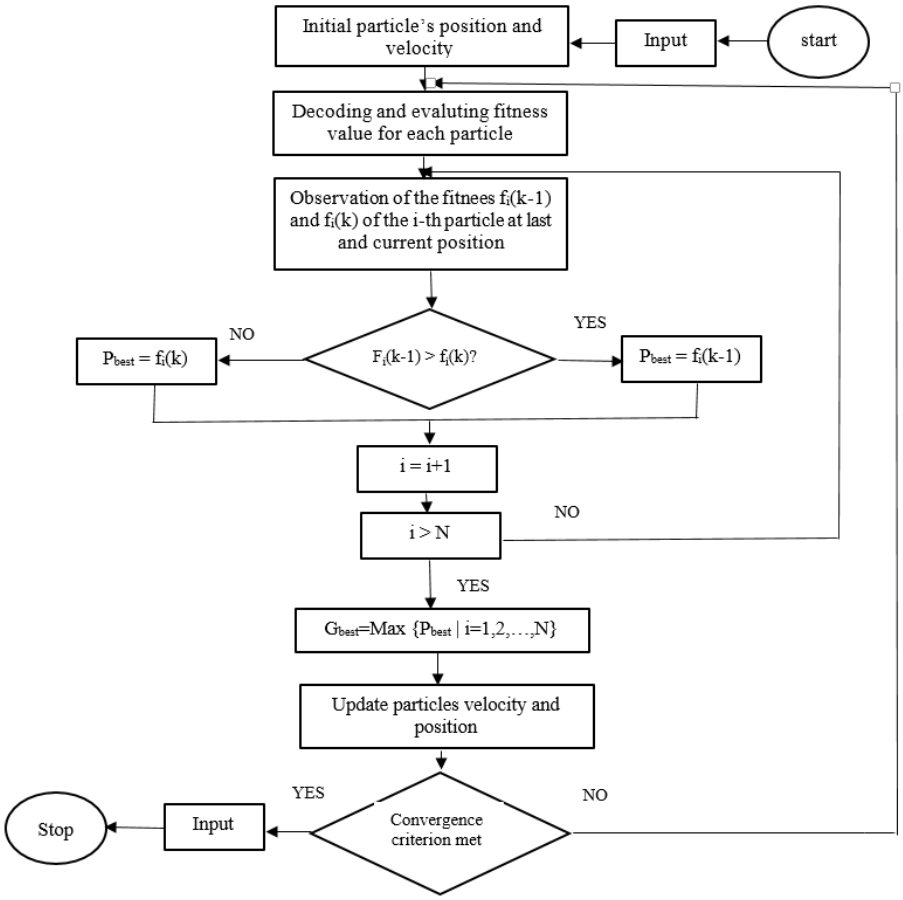

Fig. 5. *Flowchart of the overall procedure of the ICA for solving the problem.*

If $w(j)_i$ presents the power and richness of the movement of bee i at *t=j* the possibility that the Onlooker Bees (observer bees) follow the bee *i* to reach the food source can be calculated by different methods based on the type of problem and one of the easiest way is the following equation:

$$P_i = {w_i^j}\Big/{\sum_{i=1}^{n_f} w_i^j}. \tag{39}$$

In this definition n_f is the numbers of foraging bees involved in the search process and present their vagel dance in the current repetition, so if the total number of bees is N, $N - n_f$ is the number of observer bees. T present the simulation time. When bee algorithm is used to solve problems with discrete space such as work scheduling, forager bees perform their vagel dance with the length of $L = \alpha f_p$. Here f_p presents the richness and profitability (fitness) of food source and α is the adjustment factor. It should be noted that f_p is a function of the problem objective that should reach its maximum.

A flowchart of the ABC is included in *Fig. 6*. that presents the overall procedure for solving the problem.

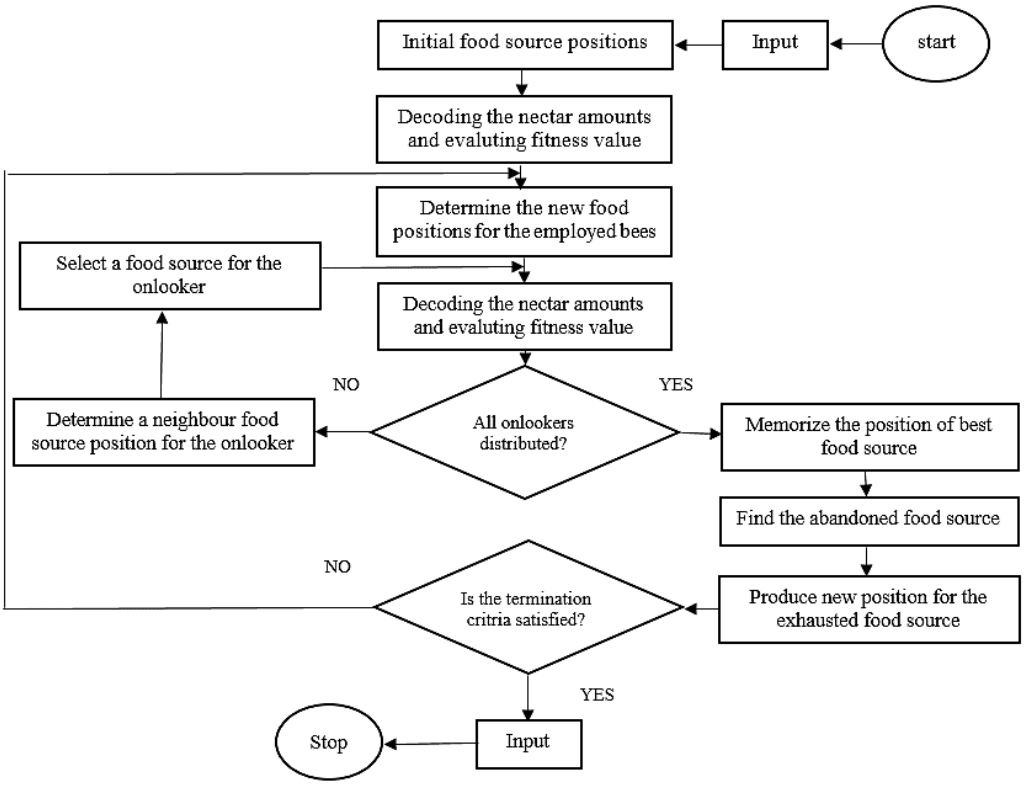

Fig. 6. *Flowchart of the overall procedure of the ABC for solving the problem.*

4.5. Simulated Annealing

Annealing means smelting the object, but in practice it means a physical process to raise the temperature of the object to the melting point and then cooling it down gradually over the specified conditions and during this process the energy of the body is minimized. Metropolice provided an algorithm to evaluate the solid body temperature changes [45]. At first he increased the object's temperature to form a molten and then displaced the atoms to reduce its internal energy. This displacement is made between two atoms. Then another atom is selected in the neighborhood of the atom and displaced with this atom, choosing the atoms for displacement is totally random and there is no order in this regard. At this temperature numerous displacements occur and when no change is made in the level of energy, the object's energy is reduced. Before reducing the object's temperature, the balance test is performed. If under displacement the object's energy is reduced, it is accepted; otherwise it is accepted under one probability. Later in 1983, Kirkpatrick by simulating this algorithm used it to solve and optimization problem through minimizing the cost function of one problem and cooling an object until reaching that basic energy [45]. Through this displacement he and his colleagues introduced an algorithm called simulated annealing. The physical interpretation of is related to the simulation of annealing in solids. Simulated annealing process that led to the loss of energy in a solid is defined as follows. At each stage, the atom is slightly displaced which leads to changes in the energy system presented as ΔE. If this amount is less than zero, the displacement of atoms is accepted and the structure of the displaced atom is used as the starting point for the next stage. If it is greater than zero, the probability approach is used, which means that the probability that a solid structure shall be accepted is determined using the

following equation, where, T and K_b represent the temperature and Bolteznman constant, respectively [45].

$$P(\Delta E) = e^{\frac{\Delta E}{K_b T}}. \tag{40}$$

Here a random number is selected based on uniform distribution within the range [0 1] and compared with P (ΔE). If the resulting number is less than *P (ΔE)* the new structure is accepted and used for the next step, otherwise the new structure is rejected. This process continues until it reaches an equilibrium level, where the temperature is reduced in accordance with the annealing process. This process continues until the system reaches a solid state.

At each temperature annealing should be such that a sufficient number of displacements are performed to reach an equilibrium condition. The simulated annealing and other heuristic algorithms start working with an initial response that is created initiatively. Then a neighbor response that creates improvement in the objective function is selected and continues until there is no improvement in the objective function and usually the optimization algorithms that start with an initial response and optimized in later procedures may reach the local optimum point which is sometimes too far from the final answer. The difference between annealing algorithms and optimization algorithm is that in the local optimization algorithm a solution is created in the neighborhood of the previous solution, if the objective function is optimized due to the new solution, the new solution is accepted otherwise it is rejected. This may result in being placed in the local optimal point and the lack of possibility to exit it. But in simulated annealing procedure, stopping at local optimum area is avoided and its passes it transiently. This state is performed by accepting bad solutions so that it could exit the local optimum point. This probability is equal to $e^{\frac{-df}{T}}$ where, *df* presents the change in the objective function. If this probability is higher than a uniform random number between [0, 1], the inappropriate answer is accepted.

A flowchart of the SA is included in *Fig. 7*. That presents the overall procedure for solving the problem.

In this paper, due to the complexity of the proposed model, the new decoding based on real numbers to locate the facility is used. A proper chromosome, can lead to efficiency and effectiveness of the evolutionary algorithm. For more explanation, in this paper a chromosome consists of two sections; the first section is organized according to the total number of periods (T), the second section is all potential facilities that includes the raw material supplier (S), plants (M), warehouses (W), distribution/collection centers (E), repair centers (R), recovery/ decomposition centers (U) and disposal centers (L), respectively. The real-based encoding is represented by a $|T|.(|S|+|M|+|W|+|E|+|R|+|U|+|L|)$ matrix, where T is a number of periods. Therefore the designed chromosome includes T string, each string with $.(|S|+|M|+|W|+|E|+|R|+|U|+|L|)$ gens that are generated randomly number between 0 and 1. A primitive encoding solution is shown in Figure (8). For three periods and six raw material suppliers that during the three periods, maximum 3 suppliers can be selected. *Fig. 8* shows the decoding of the chromosome.

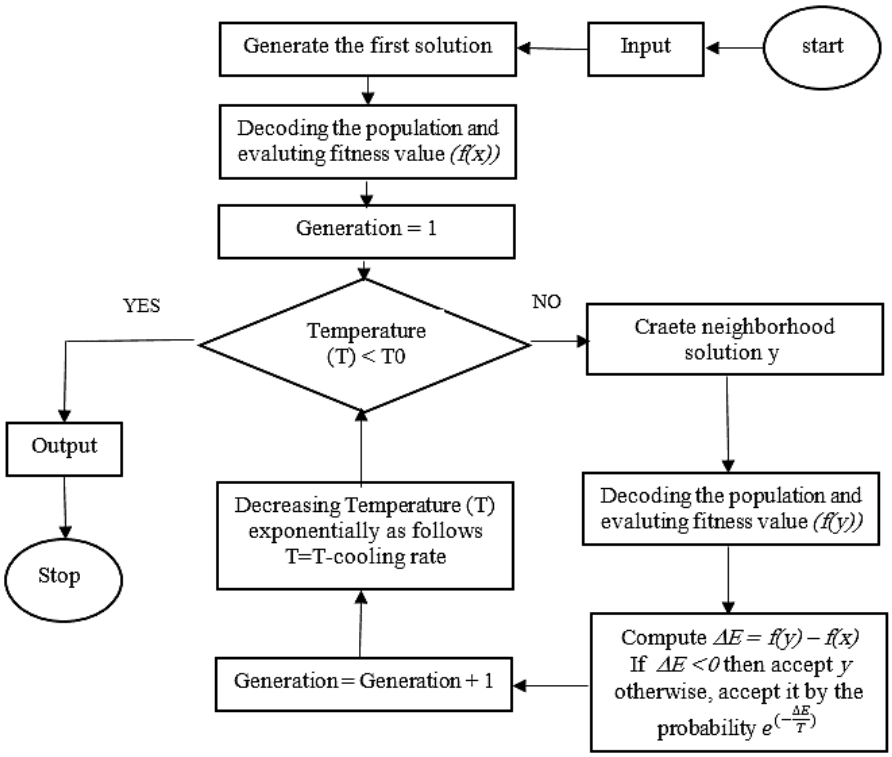

Fig. 7. *Flowchart of the overall procedure of the SA for solving the problem.*

Period(T)

0.11	*0.33*	*0.27*	*0.04*	*0.23*	*0.84*
0.49	*0.10*	*0.73*	*0.07*	*0.60*	*0.47*
0.14	*0.38*	*0.61*	*0.47*	*0.94*	*0.04*

Supplier (S)

Fig. 8. *An example of the display of the chromosomes in the supply chain.*

Period(T)

0.84	*0.33*	*0.27*	*0.23*	*0.11*	*0.04*
0.73	*0.60*	*0.49*	*0.47*	*0.10*	*0.07*
0.94	*0.61*	*0.47*	*0.38*	*0.14*	*0.04*

Supplier (S)

Period(T)

6	*2*	*3*	*0*	*0*	*0*
3	*5*	*1*	*0*	*0*	*0*
5	*3*	*4*	*0*	*0*	*0*

Selected Supplier (S)

Fig. 9. *The decoding process.*

In *Fig. 8*, in each period, the numbers from largest to smallest, are arranged, then the maximum number of required facilities, facility location is determined, at the end Products based on random data relative to demand, are allocated to the selected facility.

5. Numerical Results

In order to assess the performance of the proposed meta-heuristic algorithms in terms of the objective-function value and required CPU time, several numerical experiments of different sizes are implemented and related results are reported in this section. To this aim, 10 test problem with different sizes are designed with different combinations of the parameter values. The size of the designed problem are shown in *Table 1*. The ranges of the parameters are shown in *Table 2*. Furthermore, all the parameters of test problem are randomly generated based on Uniform distributions in pre-specified intervals.

Table 1. *Test problems.*

	$(\mathbf{S} \times \mathbf{M} \times \mathbf{W} \times \mathbf{E} \times \mathbf{C} \times \mathbf{R} \times \mathbf{U} \times \mathbf{L} \times \mathbf{T} \times \mathbf{P} \times \mathbf{I} \times \mathbf{N} \times \mathbf{H})$
Small Scale	*Prob.1* $(6 \times 6 \times 6 \times 6 \times 10 \times 4 \times 4 \times 4 \times 6 \times 2 \times 2 \times 3 \times 3)$
Medium Scale	*Prob.2* $(10 \times 10 \times 10 \times 10 \times 15 \times 6 \times 6 \times 6 \times 10 \times 3 \times 3 \times 4 \times 3)$
Large Scale	*Prob.3* $(15 \times 15 \times 15 \times 15 \times 20 \times 10 \times 10 \times 10 \times 18 \times 4 \times 4 \times 5 \times 3)$

Table 2. *Pre-specified intervals to generate parameters based on Uniform distributions.*

Parameters Intervals	
fi_{jt}	*~U(40000,45000)*
cl_{jt}	*~U(400000,6000000)*
op_{jt}	*~U(200000,4000000)*
$c1_{mpt}, c2_{ept}, c3_{ept}, c4_{rpt}, c5_{upt}, c6_{lit}$	*~U(0.5,1.5)*
va_{shit}	*~U(4000,10000)*
$\widetilde{De}_{cpt}$	*~U([100,150],[150,200],[250,300],[350,400])*
$\widetilde{tc}_{jj'in}, \widetilde{tc}_{jj'pn}$	*~U([5,10],[15,20],[25,30],[35,40])*
$co2_{jj'n}$	*~U(2.5,4)*
h_{mit}, h'_{wpt}	*~U(0.8,1.2)*
pr_{shit}	*~U(1,1.5)*
π_{cpt}	*~U(150,200)*
δ_{ip}, B_j	*~U(1,3)*
$\alpha_{cpt}, \beta_{pt}, \gamma_{pt}, \theta_{it}$	*~U(0.2,0.3)*
$cap_{si}, cap_{li}, cap_{mi}$	*~U(1000,1600)*
$cap_{mp}, cap_{wp}, cap_{ep}, cap'_{ep}, cap_{rp}, cap_{up}$	*~U(800,1000)*
$Tco2$	*~U(1000,1200)*

For the reason that acquired results through using proposed meta-heuristic algorithms are sensitive to their initial parameters, a small change can affect the accuracy best solution obtained. Therefore, the Taguchi tuning method is used for the parameters in order to find best solutions. In the Taguchi method, first, the appropriate factors (initial parameters) are determined, and the level of each factor is selected, then the design of experiments for this control factor is specified. After specifying the experimental design, the proposed algorithm is used in order to find the best combination of factors. In this paper, for

each factor 3 level is considered. In *Table 3*, population size (Npop), arithmetic crossover rate (Pa), blend crossover rate (Pb), mutation rate (Pm) in GA, particle size (Nparticle), velocity rate (D), velocity coefficients rate (C1,C2) in PSO, the number of countries (Ncoun), the number of empires (Nimp), revolution rate (Rev rate), deflection rate (Def rate) in ICA, temprature (T0), colling rate (α) in SA and the number of bees (Nbee) and the maximum motion each bee (limite), as the initial parameters are considered.

For each algorithm, according to the number of factors and number of levels, the experimental design is employed. To this aim, the experiment is repeated 5 times for each size. Average results are reported as the final value. For example, the best values of the proposed parameter for genetic algorithm in prob.1 to *Table 4*. Are 200, 0.1, 0.3 and 0.5 for population size, arithmetic crossover rate, and blend crossover and mutation rate, respectively.

Table 3. *Proposed parameter level.*

Algorithm	Parameter	Low	MID	High
GA	*Npop*	*50*	*100*	*200*
	Pa	*0.1*	*0.3*	*0.5*
	Pb	*0.1*	*0.3*	*0.5*
	Pm	*0.1*	*0.3*	*0.5*
ICA	*Ncoun*	*100*	*150*	*200*
	Nimp	*20*	*30*	*40*
	RevRate	*0.1*	*0.3*	*0.5*
	DefRate	*0.1*	*0.3*	*0.5*
PSO	*Nparticle*	*50*	*100*	*200*
	D	*0.1*	*0.3*	*0.5*
	C1	*1.5*	*2*	*3*
	C2	*1.5*	*2*	*3*
ABC	*Nbee*	*100*	*150*	*200*
	Limite	*50*	*100*	*150*
SA	*Temperature*	*1000*	*1500*	*2000*
	ReductionRate	*0.8*	*0.9*	*0.9*

Table 4. *The best parameter values for the proposed algorithms.*

Algorithm	Parameter	Prob.1	Prob.2	Prob.3
GA	*Npop*	*200*	*200*	*200*
	Pa	*0.1*	*0.3*	*0.5*
	Pb	*0.3*	*0.1*	*0.5*
	Pm	*0.5*	*0.3*	*0.3*
ICA	*Ncoun*	*100*	*200*	*200*
	Nimp	*20*	*20*	*20*
	RevRate	*0.3*	*0.5*	*0.5*
	DefRate	*0.5*	*0.3*	*0.5*
PSO	*Nparticle*	*200*	*200*	*200*
	D	*0.5*	*0.3*	*0.5*
	C1	*1.5*	*2*	*1.5*
	C2	*1.5*	*2*	*2*
ABC	*Nbee*	*150*	*100*	*200*
	Limite	*150*	*50*	*150*
SA	*Temperature*	*2000*	*2000*	*2000*
	ReductionRate	*0.8*	*0.9*	*0.8*

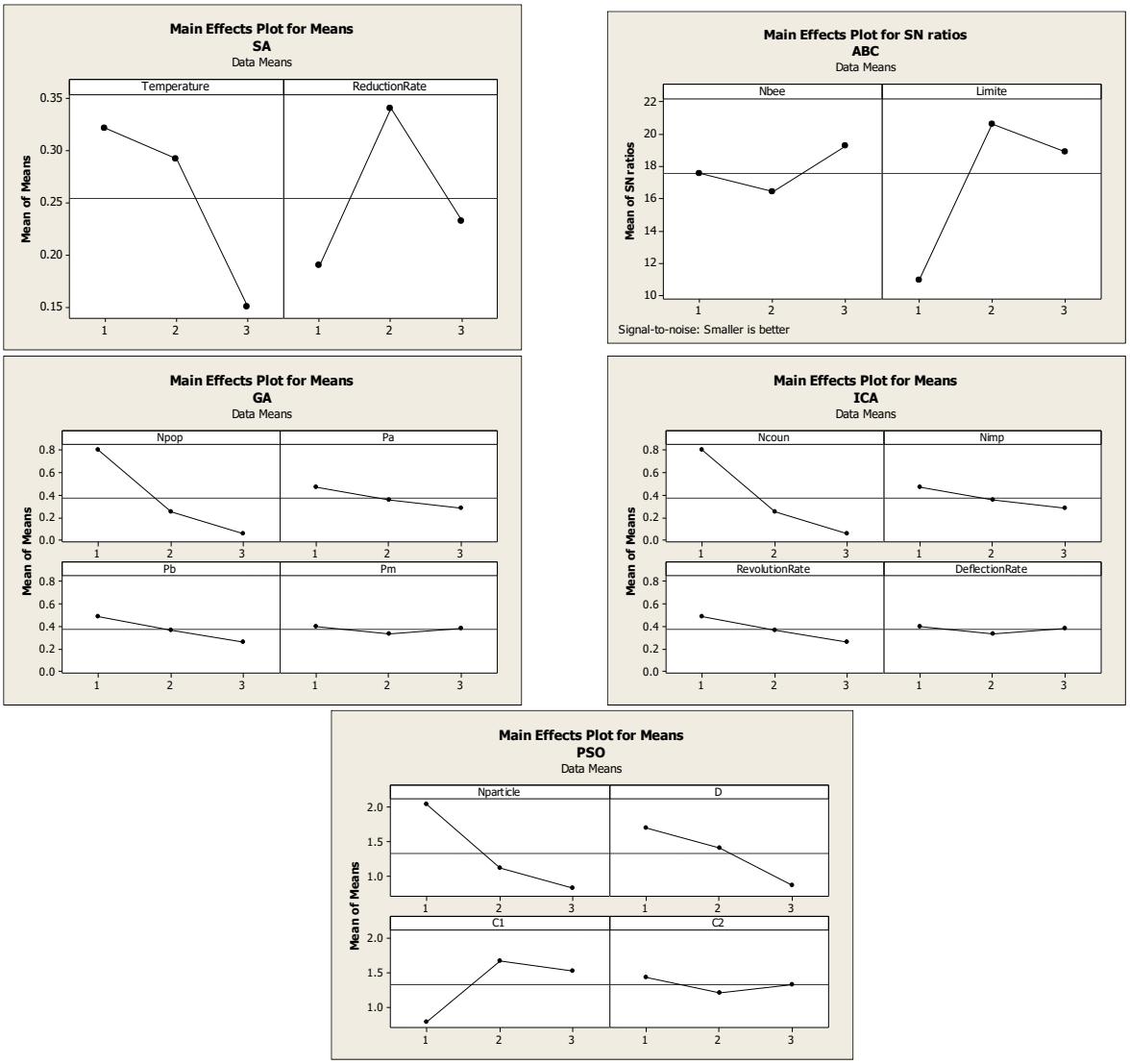

Fig. 10. *The mean of means graph.*

Based on the results in *Fig. 10*. In mean of means graph, if the parameter is placed in the lower, algorithms result in a more efficient response. As the results in *Table 4* and *Fig. 10*, in the genetic algorithm, population size, arithmetic crossover rate and blend crossover are placed in the level 3, and the mutation rate is placed in the level 2.

After tuning the parameters of the algorithms in each problem instance, the results obtained proposed algorithms are compared to the results obtained GAMS software. For this purpose, each algorithm is employed 20 times to solve the closed-loop supply chain network problem. The means of the 20 runs for each of the 3 test problem instances solved by proposed algorithms are shown in *Table 4*. Respectively *Fig. 11* shows the graph of the solutions generated using proposed algorithms on the prob.1 with the same input parameters of the closed-loop supply chain.

Table 5. *The means of the objective values.*

Test Problem	GA	ICA	PSO	ABC	SA	GAMS
1	***14485116.9***	14490096.5	14516960.9	14505556.4	14492921.6	14301344.0
	14489157.6	14498542.6	14543943.2	14508081.9	14501363.7	14358137.6
	14148818.3	14152983.3	14226720.4	14155081.1	14156243.9	14024170.2
	14616522.5	***14612908.1***	14648623.1	14624808.2	14616377.1	14588111.2
	14189028.9	***14186060.8***	14278453.1	14216069.1	14193125.8	14062490.1
	14342359.1	14348424.9	***14338102.6***	14351235.9	14347601.0	14216932.0
	14072419.2	14076983.6	14229511.3	14090702.1	14076726.6	13851372.5
	14788195.8	14801171.4	14900101.4	14824035.1	14789225.9	14658085.1
	14436302.4	***14434541.4***	14670135.8	14489339.4	14445054.2	14333818.4
	14948443.3	14953797.7	15040253.9	14971422.1	14954246.3	14679723.3
2	28418535.6	28462758.9	28785403.9	28525598.3	***28408222.8***	27539829.2
	28483786.1	28567653.9	28825630.1	28863360.1	28488930.3	27600701.9
	28257590.7	28333243.3	28384009.2	28639569.5	28276837.2	27757355.5
	28272552.5	28298704.1	28365000.8	28579114.3	28279724.5	27401886.5
	28302971.1	28372724.1	28528543.9	28730721.1	28325273.2	27931708.9
	28421751.9	28462012.5	28723363.4	28510936.9	28423309.4	28262990.6
	28756152.7	28816003.6	28950474.8	28928773.9	28765179.3	28037319.8
	28498229.8	28528301.8	28812659.5	28893111.4	28540939.6	27790428.0
	28838576.1	28917021.2	29018786.7	29096205.0	28849712.9	28386933.2
	28180657.8	28249580.6	28802464.2	28776221.4	28186765.3	27713143.8
3	***51503416.7***	51884150.5	52258598.5	52329448.5	51516651.0	49939904.3
	51884694.3	52145120.1	52937311.0	52960300.5	52058815.4	49792082.1
	51681793.2	52154618.9	51891132.5	52620851.7	51777922.4	50203506.5
	52167927.0	52597494.6	52990516.4	53311811.0	52252476.8	50027578.6
	51202276.8	51723313.0	52138196.5	52169518.3	51325476.8	50029552.0
	52038649.3	52411705.4	53040328.2	52415880.0	52163918.9	49777803.7
	50799110.8	50964203.8	51484781.6	53132401.3	***50796129***.8	50476705.4
	51271170.7	51715761.6	52493419.7	52575324.2	51387201.6	50326206.6
	51294321.4	51960763.7	52311357.3	52621011.7	51369044.9	49958729.9
	50995260.3	51405605.7	51564102.5	52175897.7	51137338.6	49693221.5

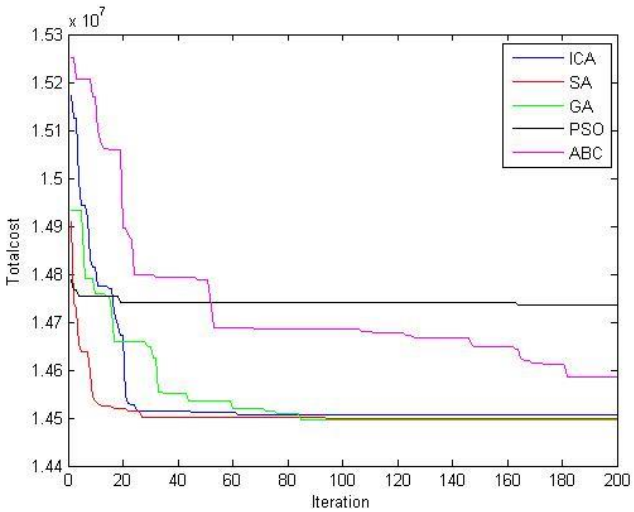

Fig. 11. *The graph of the solutions generated using proposed algorithms.*

To compare the optimal solutions obtained by GAMS software with the results obtained of the proposed algorithms, a quality criterion, the gap of the solutions, is defined according to the following equation:

$$\%gap = \frac{(algorithm_{answer}) - (GAMS_{answer})}{GAMS_{answer}} \times 100 \,.$$

In *Table 6*, the means of the gap of solutions are presented. As the results show (see *Table 6*), the solutions gaps vary from 0.16% to 6.56% for all test problem, also the maximum gap for largest test problem is less than 6.56%. In the most problems solved, the solution gaps of a genetic algorithms much lower than other proposed algorithms. This means that the efficiency of genetic algorithms in solving the developed closed-loop supply chain is very high. Also in *Fig. 12* the average of CPU time obtained of the different size problem is illustrated.

Table 6. *The means of the gap.*

Test Problem	GA	ICA	PSO	ABC	SA
1	*1.285004*	*1.319824*	*1.507669*	*1.427925*	*1.339578*
	0.912514	*0.977878*	*1.294079*	*1.044316*	*0.997526*
	0.888809	*0.918508*	*1.444294*	*0.933466*	*0.941758*
	0.194757	*0.169980*	*0.414803*	*0.251554*	*0.193760*
	0.899832	*0.878726*	*1.535738*	*1.092118*	*0.928966*
	0.882237	*0.924903*	*0.852298*	*0.944676*	*0.919108*
	1.595847	*1.628800*	*2.729974*	*1.727840*	*1.626944*
	0.887638	*0.976160*	*1.651077*	*1.132140*	*0.894665*
	0.714980	*0.702695*	*2.346321*	*1.084994*	*0.776037*
	1.830552	*1.867027*	*2.455977*	*1.987087*	*1.870083*
2	*3.190675*	*3.351254*	*4.522812*	*3.579431*	*3.153228*
	3.199499	*3.503360*	*4.438033*	*4.574732*	*3.218137*
	1.802172	*2.074721*	*2.257613*	*3.178307*	*1.871510*
	3.177394	*3.272832*	*3.514774*	*4.296156*	*3.203568*
	1.329178	*1.578905*	*2.136765*	*2.860592*	*1.409023*
	0.561729	*0.704178*	*1.628889*	*0.877283*	*0.567239*
	2.563843	*2.777312*	*3.256927*	*3.179527*	*2.596038*
	2.546927	*2.655137*	*3.678358*	*3.967853*	*2.700612*
	1.591024	*1.867366*	*2.22586*	*2.498586*	*1.630256*
	1.686976	*1.935676*	*3.930699*	*3.836005*	*1.709014*
3	*3.130788*	*3.893172*	*4.642969*	*4.784839*	*3.157288*
	4.202701	*4.725727*	*6.316725*	*6.362896*	*4.552397*
	2.944589	*3.886407*	*3.361570*	*4.815092*	*3.136068*
	4.278337	*5.136999*	*5.922609*	*6.564844*	*4.447343*
	2.344064	*3.385521*	*4.214798*	*4.277404*	*2.590319*
	4.541875	*5.291318*	*6.554175*	*5.299704*	*4.793533*
	0.638721	*0.965789*	*1.997112*	*5.261231*	*0.632815*
	1.877678	*2.761096*	*4.306331*	*4.469078*	*2.108236*
	2.673390	*4.007375*	*4.709142*	*5.328962*	*2.822960*
	2.620154	*3.445911*	*3.764862*	*4.996006*	*2.906065*

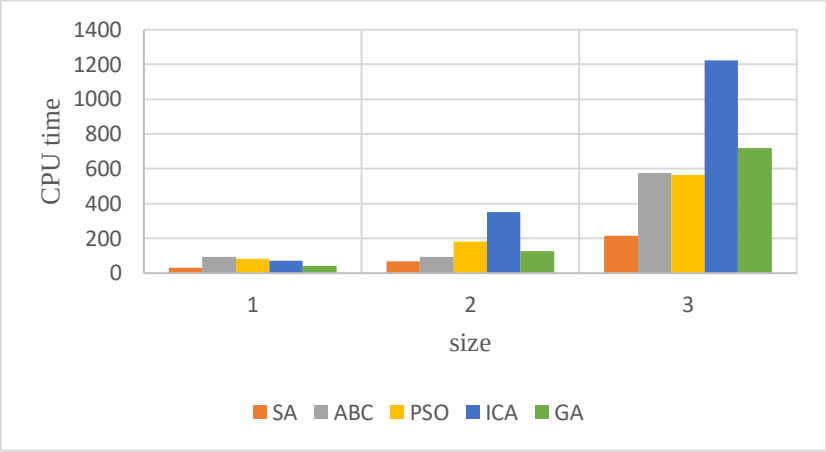

Fig. 12. *The average of CPU time.*

Although the CPU time for simulated annealing algorithms is significantly lower than other algorithms, the genetic algorithm provides a better performance solution. According to results, CPU time obtained by GAMS software for medium and large sizes are much higher than the CPU time obtained of the proposed algorithms, while the solution gaps for all the algorithms of the amount not exceed 6.65%. In *Table 5*, the average of the objective function obtained of the meta-heuristic algorithms in each test problem are shown. To compare the results obtained, T-Test at a 95% confidence level was used for comparing the significant difference of the means of each index. Hence, if the P-value of the test statistic for each index is less than 0.05, H_0 is rejected and indicates that there is a significant difference between the means of the index. If the P-value of the test statistic is greater than 0.05, H_1 is rejected and indicates no significant difference in the means of the index. *Table 7* illustrates the results of the T-Test on the objective function.

Table 7. *Results of the T-test on the objective function.*

Algorithm	Diff Average	95% confidence interval	T-Value	P-Value
GA-ICA	*772699*	*(-7126045 8671444)*	*0.20*	*0.845*
GA-PSO	*1229021*	*(-6786740 9244781)*	*0.31*	*0.760*
GA-ABC	*1132570*	*(-6839720 9104861)*	*0.28*	*0.777*
GA-SA	*893483*	*(-7041957 8828922)*	*0.23*	*0.822*
ICA-PSO	*456321*	*(-7709636 8622278)*	*0.11*	*0.911*
ICA-ABC	*359871*	*(-7763420 8483162)*	*0.09*	*0.930*
ICA-SA	*120783*	*(-7966344 8207911)*	*0.03*	*0.976*
PSO-ABC	*96450*	*(-8333568 8140668)*	*0.02*	*0.981*
PSO-SA	*335538*	*(-7865919 8536995)*	*0.08*	*0.935*
ABC-SA	*239088*	*(-7919888 8398064)*	*0.06*	*0.953*

According to *Table 7* and the P-value obtained, it is observed that there is no significant difference between the averages of the objective function of the meta-heuristic algorithms. Therefore, the genetic algorithm is selected as the most efficient algorithm with the lowest objective function value and computational time.

6. Conclusion and Future Research

In this paper, a novel mathematical formulation for a multi-echelon multi-product multi-period closed-loop supply chain network was developed. It was formulated to obtain a single-objective mixed-integer nonlinear programming. To solve this model, five meta-heuristics algorithm proposed. Three problem instances of small, medium and large was designed to evaluate the application of the proposed algorithms as well as to compare the quality between algorithms. After the parameter tuning based on Taguchi method, 10 examples, the algorithms were employed to solve the test problems, each 20 times. As the results, the solutions gaps vary from 0.16% to 6.56% for all test problem, also the maximum gap for largest test problem is less than 6.56 % in the most problems solved, the solution gaps of a genetic algorithms much lower than other proposed algorithms. This means that the efficiency of genetic algorithms in solving the developed closed-loop supply chain is very high. Although the CPU time for simulated annealing algorithms is significantly lower than other algorithms, the genetic algorithm provides a better performance solution. According to results, CPU time obtained by GAMS software for medium and large sizes are much higher than the CPU time obtained of the proposed algorithms, while the solution gaps for all the algorithms of the amount not exceed 6.65%. At the end to compare the results obtained, T-Test at a 95% confidence level was used for comparing the significant difference of the means of each index. The results show no significant difference between the averages of the objective function of the meta-heuristic algorithms.

References

[1] Fazli-Khalaf, M., Mirzazadeh, A., & Pishvaee, M. S. (2017). A robust fuzzy stochastic programming model for the design of a reliable green closed-loop supply chain network. *Human and ecological risk assessment: an international journal*, *23*(8), 2119-2149.

[2] Ghahremani Nahr, J., Pasandideh, S. H. R., & Niaki, S. T. A. (2020). A robust optimization approach for multi-objective, multi-product, multi-period, closed-loop green supply chain network designs under uncertainty and discount. *Journal of industrial and production engineering*, *37*(1), 1-22.

[3] Jabbarzadeh, A., Haughton, M., & Khosrojerdi, A. (2018). Closed-loop supply chain network design under disruption risks: A robust approach with real world application. *Computers & industrial engineering*, *116*, 178-191.

[4] Mohtashami, Z., Aghsami, A., & Jolai, F. (2020). A green closed loop supply chain design using queuing system for reducing environmental impact and energy consumption. *Journal of cleaner production*, *242*. https://doi.org/10.1016/j.jclepro.2019.118452

[5] Prakash, S., Kumar, S., Soni, G., Jain, V., & Rathore, A. P. S. (2020). Closed-loop supply chain network design and modelling under risks and demand uncertainty: an integrated robust optimization approach. *Annals of operations research*, *290*(1), 837-864.

[6] Sadeghi, A., Mina, H., & Bahrami, N. (2020). A mixed integer linear programming model for designing a green closed-loop supply chain network considering location-routing problem. *International journal of logistics systems and management*, *36*(2), 177-198.

[7] Safaei, A. S., Roozbeh, A., & Paydar, M. M. (2017). A robust optimization model for the design of a cardboard closed-loop supply chain. *Journal of cleaner production*, *166*, 1154-1168.

[8] Govindan, K., Soleimani, H., & Kannan, D. (2015). Reverse logistics and closed-loop supply chain: A comprehensive review to explore the future. *European journal of operational research*, *240*(3), 603-626.

[9] Fleischmann, M., Beullens, P., BLOEMHOF-RUWAARD, J. M., & Van Wassenhove, L. N. (2001). The impact of product recovery on logistics network design. *Production and operations management*, *10*(2), 156-173.

[10] Üster, H., Easwaran, G., Akçali, E., & Çetinkaya, S. (2007). Benders decomposition with alternative multiple cuts for a multi-product closed-loop supply chain network design model. *Naval research logistics (NRL)*, *54*(8), 890-907.

[11] Lee, D. H., & Dong, M. (2009). Dynamic network design for reverse logistics operations under uncertainty. *Transportation research part E: logistics and transportation review*, *45*(1), 61-71.

[12] Lee, J. E., Gen, M., & Rhee, K. G. (2009). Network model and optimization of reverse logistics by hybrid genetic algorithm. *Computers & industrial engineering*, *56*(3), 951-964.

[13] Kannan, G., Sasikumar, P., & Devika, K. (2010). A genetic algorithm approach for solving a closed loop supply chain model: A case of battery recycling. *Applied mathematical modelling*, *34*(3), 655-670.

[14] Khajavi, L. T., Seyed-Hosseini, S. M., & Makui, A. (2011). An integrated forward/reverse logistics network optimization model for multi-stage capacitated supply chain. iBusiness 3(2). DOI:10.4236/ib.2011.32030

[15] Das, K., & Chowdhury, A. H. (2012). Designing a reverse logistics network for optimal collection, recovery and quality-based product-mix planning. *International journal of production economics*, *135*(1), 209-221.

[16] Mahmoudi, H., Fazlollahtabar, H., & Mahdavi, I. (2013). Mathematical modeling for minimizing costs in a multilayer multi-product reverse supply chain. *Industrial engineering & management*, *2*(2), 6.

[17] Dönmez, İ. (2013). Design of reverse logistics network for waste batteries with an application in Turkey. *Chemical engineering transactions*. Retrieved from https://d1wqtxts1xzle7.cloudfront.net/43286668/Design_of_Reverse_Logistics_Network_for_20160302-1112-2hyem.pdf?1456963233=&response-content-disposition=inline%3B+filename%3DDesign_of_reverse_logistics_network_for.pdf&Expires=1609333993&Signature=UnomU6E0H~gyPwvKaPdxBn261~518QLiB0DZdCPbT3PWXn6HDE7IQR9gFd7Elbdcn-8ij8Sg8rKfjqbxzyeyqm0RPfWaCxJ5XTGsZKLh6-cMk3UWuNLJjW~MRdOWAfV2kiz7gRTrEhR7PxbMPkF2hD-isDa6St3L4326fHpW3Xab9UPGc-rWAU1ROZfIodyIMVbL520s2P2ejeOU9pMPaLcoKeVV86bBmqrQSweT4Le88J43EvhQiENiqQtY1qXU6JI7hi12m0R6aqhJHm-pAi63nDOGXuPcFEb8UCz9qgxVyN63lkCc08~DSSD3yqesQ-kWaWqDiqSGyBIOQ4NlIg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[18] Ramezani, M., Bashiri, M., & Tavakkoli-Moghaddam, R. (2013). A new multi-objective stochastic model for a forward/reverse logistic network design with responsiveness and quality level. *Applied mathematical modelling*, *37*(1-2), 328-344.

[19] Özceylan, E., Paksoy, T., & Bektaş, T. (2014). 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*, 142-164.

[20] Soleimani, H., & Govindan, K. (2014). Reverse logistics network design and planning utilizing conditional value at risk. *European journal of operational research*, *237*(2), 487-497.

[21] Rezaee, A., Dehghanian, F., Fahimnia, B., & Beamon, B. (2017). Green supply chain network design with stochastic demand and carbon price. *Annals of operations research*, *250*(2), 463-485.

[22] Alavi, S., Azad, N., Heydar, M., & Davoudpour, H. (2016). Integrated production, inventory, and location-allocation decisions in designing supply chain networks. *International journal of information systems and supply chain management (IJISSCM)*, *9*(4), 22-42.

[23] Nobil, A. H., & Taleizadeh, A. A. (2016). Analysing a fuzzy integrated inventory-production-distribution planning problem with maximum NPV of cash flows in a closed-loop supply chain. *International Journal of Inventory Research*, *3*(1), 31-48.

[24] Talaei, M., Moghaddam, B. F., Pishvaee, M. S., Bozorgi-Amiri, A., & Gholamnejad, S. (2016). A robust fuzzy optimization model for carbon-efficient closed-loop supply chain network design problem: a numerical illustration in electronics industry. *Journal of cleaner production*, *113*, 662-673.

[25] Zhalechian, M., Tavakkoli-Moghaddam, R., Zahiri, B., & Mohammadi, M. (2016). Sustainable design of a closed-loop location-routing-inventory supply chain network under mixed uncertainty. *Transportation research part E: logistics and transportation review*, *89*, 182-214.

[26] Ciccullo, F., Pero, M., Caridi, M., Gosling, J., & Purvis, L. (2018). Integrating the environmental and social sustainability pillars into the lean and agile supply chain management paradigms: A literature review and future research directions. *Journal of cleaner production*, *172*, 2336-2350.

[27] Das, K. (2018). Integrating lean systems in the design of a sustainable supply chain model. *International Journal of Production Economics*, *198*, 177-190.

[28] Haddadsisakht, A., & Ryan, S. M. (2018). Closed-loop supply chain network design with multiple transportation modes under stochastic demand and uncertain carbon tax. *International journal of production economics*, *195*, 118-131.

[29] Farrokh, M., Azar, A., Jandaghi, G., & Ahmadi, E. (2018). A novel robust fuzzy stochastic programming for closed loop supply chain network design under hybrid uncertainty. *Fuzzy sets and systems*, *341*, 69-91.

[30] Darbari, J. D., Kannan, D., Agarwal, V., & Jha, P. C. (2019). Fuzzy criteria programming approach for optimising the TBL performance of closed loop supply chain network design problem. *Annals of operations research*, *273*(1-2), 693-738.

[31] Ghahremani-Nahr, J., Kian, R., & Sabet, E. (2019). A robust fuzzy mathematical programming model for the closed-loop supply chain network design and a whale optimization solution algorithm. *Expert systems with applications*, *116*, 454-471.

[32] Samuel, C. N., Venkatadri, U., Diallo, C., & Khatab, A. (2020). Robust closed-loop supply chain design with presorting, return quality and carbon emission considerations. *Journal of Cleaner Production*, *247* . https://doi.org/10.1016/j.jclepro.2019.119086

[33] Gholizadeh, H., Tajdin, A., & Javadian, N. (2020). A closed-loop supply chain robust optimization for disposable appliances. *Neural computing and applications*, *32*(8), 3967-3985.

[34] Pishvaee, M. S., Rabbani, M., & Torabi, S. A. (2011). A robust optimization approach to closed-loop supply chain network design under uncertainty. *Applied mathematical modelling*, *35*(2), 637-649.

[35] Pishvaee, M. S., & Torabi, S. A. (2010). A possibilistic programming approach for closed-loop supply chain network design under uncertainty. *Fuzzy sets and systems*, *161*(20), 2668-2683.

[36] Altiparmak, F., Gen, M., Lin, L., & Karaoglan, I. (2009). A steady-state genetic algorithm for multi-product supply chain network design. *Computers & industrial engineering*, *56*(2), 521-537.

[37] Kim, K. W., Gen, M., & Yamazaki, G. (2003). Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling. *Applied soft computing*, *2*(3), 174-188.

[38] Zegordi, S. H., Abadi, I. K., & Nia, M. B. (2010). A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain. *Computers & industrial engineering*, *58*(3), 373-381.

[39] Hu, C., & Li, S. (2009). Two-phase interactive satisfying method of fuzzy multiple objective optimization with linguistic preference. *International journal of information technology & decision making*, *8*(03), 427-443.

[40] Hamzadayi, A., & Yildiz, G. (2012). A genetic algorithm based approach for simultaneously balancing and sequencing of mixed-model U-lines with parallel workstations and zoning constraints. *Computers & industrial engineering*, *62*(1), 206-215.

[41] Chitra, C., & Subbaraj, P. (2012). A nondominated sorting genetic algorithm solution for shortest path routing problem in computer networks. *Expert systems with applications*, *39*(1), 1518-1525.

[42] Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *2007 IEEE congress on evolutionary computation* (pp. 4661-4667). IEEE.

[43] Eberhart, R., & Kennedy, J. (1995, October). A new optimizer using particle swarm theory. *Proceedings of the sixth international symposium on micro machine and human Science* (pp. 39-43). IEEE.

[44] Nakrani, S., & Tovey, C. (2004). On honey bees and dynamic server allocation in internet hosting centers. *Adaptive behavior*, *12*(3-4), 223-240.

[45] Lin, L., Gen, M., & Wang, X. (2009). Integrated multistage logistics network design by using hybrid evolutionary algorithm. *Computers & industrial engineering*, *56*(3), 854-873.