Evolutionary Intelligence (2023) 16:1985–1998
https://doi.org/10.1007/s12065-023-00816-4

SPECIAL ISSUE

Cloud-based solution approach for a large size logistics network planning

Ehsan Yadegari[1] · Elham Jelodari Mamaghani[2] · Maryam Afghah[3] · Mohsen Abdoli[4] · Amir Daneshvar[4,5]

Received: 5 January 2021 / Revised: 14 February 2022 / Accepted: 31 December 2022 / Published online: 9 March 2023

Abstract
During the last two decades, due to environmental laws and the competitive environment, the formulation of effective closed-loop supply chain networks has attracted researchers' attention. On the other hand, although there are many metaheuristics applied for these NP-hard problems, applying more efficient and effective algorithms with tailor-made local searches and solution representation is inevitable. In this paper, mixed-integer linear programming is assumed to deliver the final product to customers in the forward direction from suppliers through manufacturers and distribution centers (DCs). Simultaneously, collecting recycled products from customers and entering them into the recovery or landfilling cycle is examined. Mathematical modeling of this problem aims to minimize both the costs of opening facilities at potential locations as well as the optimal flow of materials across the network layers. Due to the NP-hard nature of the problem, a cloud-based simulated annealing algorithm (CSA) has been applied for the first time in this area. Moreover, a spanning tree-based method which occupies the least number of arrays, regarding the other methods of the literature has been adopted. To analyze the accuracy and the speed of the investigated algorithm, we have compared its performance with the genetic algorithm (GA) and the simulated annealing (SA) algorithm (which were applied in the literature). The results, regarding cost function, show that the CSA algorithm provides more effective results than the other two ones. Moreover, regarding CPU time, although the CSA shows better results than GA, statistically, it failed to show more efficient results than SA.

Keywords Cloud-based simulated annealing · Spanning tree · Logistics network planning · Closed-loop

1 Introduction

The ever-expanding competitive environment and globalization of the product market has led organizations to make a significant effort in line with supply, procurement, production, and distribution of their company's products in order to survive and to meet customers' diverse needs in a timely and least expensive manner. In recent years, with the intensification of the competitive environment, this issue, which has been recognized as an effective element in economic and industrial life, has been considered as a significant issue more than ever before.

The issue of supply chain network design (SCND) involves strategic decisions that refer to supply chain configuration and as an infrastructure issue in supply chain management, it has long-lasting effects on other tactical and operational decisions of the company. In general, the network design project faces identifying locations and capacities needed for new facilities and planning to purchase, production, distribution, and maintenance of products.

Totally, SCND includes many kinds of operational and strategic decisions. Order assignment and Supplier selection

✉ Ehsan Yadegari
ehsan.yadegari@emse.fr

Elham Jelodari Mamaghani
ejelodarimamaghani@univ-catholyon.fr

Maryam Afghah
maryamafghah@alumni.ut.ac.ir

Mohsen Abdoli
m.abdoli@mail.sbu.ac.ir

Amir Daneshvar
a_daneshvar@iauec.ac.ir

[1] Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, 42023 Saint-Etienne, France

[2] Institute of Sustainable Business and Organizations Sciences and Humanities Confluence Research Center- UCLY, ESDES, Lyon, France

[3] Industrial Management Group, University of Tehran, Tehran, Iran

[4] Department Industrial Management, Shahid Beheshti University, Tehran, Iran

[5] Department of Information Technology Management, Electronic Branch, Islamic Azad University, Tehran, Iran

are among the most important problems in SCND, especially influencing delivery time and the total cost. The location-inventory-routing (LIR) model is another sensitive model that highly has an impact on the total green gas emission and inventory costs in the SCND. Likely, considering pickup and delivery methods and cross-docking operations can directly reduce transportation costs of a supply chain. Moreover, forward and reverse flow is another commonly considered method for decreasing emissions and increasing sustainability [1].

On the other hand, Metaheuristic algorithms form a wide grouping of optimization methods, which are often inspired by the natural, or non-natural procedure. So, some of the firstly developed technics for metaheuristic algorithms were introduced about the middle of the twentieth century [2].

In the next section, we are going to focus more on solution approaches applied to different SCND problems.

2 Literature review

As the first work in this field, Fleischmann [3] proposed a mixed-integer linear programming model for integrated forward and reverse supply chain design. In the investigated network, the consumed products are first collected from the customers and redistributed after recovery. Although this is a closed-loop network and has both forward and reverse flow, the production and distribution of new products have not been taken into consideration. Therefore, we can loosely call this model an integrated design with the definition provided.

Pishvaee et al. [4] have classified the integration of SCNDinto two categories: (1) Vertical integration and (2) Horizontal integration. Vertical integration is defined as the integrated decisions at strategic (long-term), tactical (mid-term), and operational (short-term) levels in network design. Designing the supply chain network is in the nature of a strategic decision that typically involves determining the location of facilities, their capacities, the number of categories in the chain, and how the facilities are related. Therefore, it should be borne in mind that integrating lower-level decisions in network design must be accompanied by maintaining strategic-level decisions. By integrating these decisions with tactical-level decisions, for instance, inventory management or operational-level decisions such as routing, sub-optimality can be avoided. We refer readers to the literature review [5] for further study of this type of integration. This type of integration is called vertical integration since it integrates the levels of decision-making in a vertical way.

Ko and Evans [6] have presented a nonlinear mixed-integer programming model for integrated SCNDfor third-party supply chain service providers. In order to tackle the uncertainty in these conditions, it is suggested that the model be run in different periods so it can possess the required dynamics, hence the parameters of the problem are determined for each period, and then the model is resolved again for the new parameters. In this paper, compound facilities are used as well. Similar work has been proposed by Min and Ko [7].

Another study that has properly addressed the integrated design of the forward and reverse supply chain networks is the paper by Lee and Dong [8]. In this paper, a type of compound facility that plays both the role of distribution centers (warehouses) in the forward flow and the role of collection centers in reverse flow has been used to design supply chain networks of computer products. This problem is modeled using mixed-integer linear programming and due to its high complexity, it is solved using a hybrid heuristic method with tabu search meta-heuristic method. Nonetheless, this paper has some weaknesses, such as simplifying assumptions like a specified number of compound facilities, or the use of only one manufactory. This paper also focuses only on the recovery procedure in the reverse supply chain flow.

Wang and Hsu [9] presented a model for designing a closed-loop supply chain network that includes suppliers, manufactories, distribution centers, and landfilling. In this paper, the authors have converted the proposed model to linear integer programming, considering the number of recovered products to be non-integer, which is assumed to be a coefficient of customer demands.

In this regard, according to the literature review [5, 10], the 2010 to 2017 studies in the network design area have been investigated in terms of modeling type, in which categorizing of this field in terms of modeling and solution methods of such problems have been discussed.

Numerous approaches have been developed with regards to the methodologies for representing network design solutions. We will discuss two of the top ones here. The first is the Prufer Numbers, which uses the spanning tree concept as a representation of the meta-heuristic algorithms solutions that were first introduced by Syarif [11]. In this approach, they developed the method on a genetic algorithm by mixing the concept of the Prufer codes and the spanning tree. In this method, an n-vertex tree contains n-2 nodes and can be generated by a simple iterative algorithm.

The next is the priority-based encoding method, in which non-duplicate numbers are used to select the valuation ordering of consecutive nodes in the supply chain. The advantage of this method over the method used in this paper is its simplicity and clarity of encoding and decoding of the solutions. Another strength of this method is that there is no need to apply a repair mechanism to the solutions after applying the other operators of the algorithm. In contrast, the method used in this paper for spanning trees executes the encoding with the lowest number of elements in a solution matrix, which plays an important role in reducing the running time of the algorithms. However, the disadvantage of the spanning tree method used in the repair mechanism operation is after applying some operators, which results in more difficulty for the coder.

In Table 1 the research background has been classified in terms of solving approaches as well as the methods of representing meta-heuristic solutions.

As can be seen, despite the rich literature in the field of supply chain network design, exact solution methods for solving models on small and medium scales have been utilized in most models presented. Also, considering the high speed of the spanning tree method, which uses the least possible arrays in the solution matrix, it has been used in only two papers so far. One of the most important reasons for not using this method is the difficulty of coding as well as the repair mechanism operations of the solutions after applying some operators. In this study, after describing a nonlinear closed-loop supply chain model, we attempt to linearize the model, thus identifying optimal locations for opening manufactories, distribution centers, and collection centers for returned products. Next, we use a cloud-based simulated annealing algorithm for network design problems for the first time. The spanning-tree method has also been used to represent the solutions and the results have been compared to other papers in this field.

Moreover, there are other kinds of metaheuristics that has been applied in other eareas and researchers shows the efficiency of them for example Marine Predators Algorithm (MPA) in optimal design of renewable CHP systems [12], antlion optimization algorithm in Energy consumption scheduleing [13], Barnacles Mating Optimization (BMO) algorithm and Sunflower Optimization Algorithm (SFO) in parameter selecting [14, 15], krill herd optimization algorithm (CKH) in hybrid selection of energy systems.

Table 1 Summary of Supply Chain Network Design Literature with Emphasis on Model Solving Approaches

Networks	Solution representation	Solution methodology
Forward networks		
Jayaraman and Pirkul [16]	–	Lagrangian Relaxation
Jayaraman, Gupta [17]	Other	Simulated annealing algorithm
Li, Chen [18]	–	Exact solution
Tsiakis and Papageorgiou [19]	–	Exact solution
Pishvaee and Razmi [4]	–	Interactive fuzzy Solution approach
Syarif, Prugel-Bennett [11, 20]	Other	Genetic algorithm
Elhedhli and Merrick [21]	–	Lagrangian Relaxation
Reverse networks		
Krikke, van Harten [22]	–	Exact solution
Min and Ko [7]	Other	Genetic algorithm
Aras and Crowther [23]	Other	Tabu search algorithm
Gírio, Fonseca [24]	–	Exact solution
Govindan, Khodaverdi [25]	–	Exact solution
Closed-loop networks		
Fleischmann, Beullens [3]	–	Exact solution
Lu and Bostel [26]	–	Lagrangian Relaxation
Salema, Póvoa [27]	–	Branch and bound algorithm
Pishvaee and Razmi [4]	–	Interactive fuzzy solution approach
Wang and Hsu [9]	Spanning tree	Genetic algorithm
Devika, Jafarian [28]	Priority-based	Hybrid Meta-heuristics
Yadegari, Ekhtiari [29]	Other	Immune algorithm
Yadegari, Najmi [30]	Discrete multipath	Memetic algorithm
Yadegari, Zandieh [31]	Spanning tree	Hybrid Meta-Heuristics
Ghayebloo, Tarokh [32]	–	Exact solution
Kaya and Urek [33]	Priority-Based	Hybrid Heuristics
Huang [34]	Other	Genetic algorithm
This paper	Spanning tree	Cloud theory-based simulated Annealing algorithm

3 Problem definition

Some of the characteristics and conditions of the problem are as follows:

(1) The location of suppliers, customers, and landfilling are known and fixed.
(2) Potential locations for the opening manufactories, distribution/collection centers, and dismantlers are known and discrete.
(3) The flow is only allowed to be transferred between two consecutive stages and also inter-facility interference in one stage does not exist.
(4) The capacity of the network facilities is limited.
(5) The demand is presumed to be decisive.
(6) The recovery percentage is presumed as a constant percentage of demand.
(7) The timing of the procurements is presumed to be known and constant.
(8) The quality of the repaired products is equal to the quality of the manufactured products.
(9) Collection centers collect all recovered products from all customers by the end of each period.

Figure 1 illustrates the configuration of the discussed network. As we can see in this figure raw materials are delivered from suppliers to manufacturers and after producing the products, they are delivered to customer zones through distribution centers in the forward flow. In the backward network, the used products are recovered in DCs and then are classified and are decided to go back to the forward network or safely do the disposal activities on them. To better understand the proposed mathematical model, we first explain the model in the verbal form:

The objective function in verbal form:

Minimum cost = Fixed costs of reopening + Shipping costs.

Model constraints in the verbal form:

- Satisfy all the demands in forward and reverse direction
- The balance of the flow between nodes
- Capacity constraints
- Logical constraints related to different capacity levels
- Nonnegative constraints and binary

The following describes the sets, parameters, and variables of the model.

3.1 Sets

I Set of fixed locations of suppliers $i = 1, 2 \ldots., I$

jSet of potential locations for manufactories $j = 1, 2 \ldots., J$

kSet of potential locations for distribution/collection centers $k = 1, 2 \ldots., K$

lSet of fixed locations of customers $l = 1, 2 \ldots., L$

mSet of potential locations for dismantlers $m = 1, 2 \ldots., M$

3.2 Parameters

A_i Capacity of supplier i

b_j Capacity of manufactory j

Sc_k Total capacity of forward and reverse logistics in the DC k

Pc_l Recovery percentage of customer l

pl_m The landfilling rate of dismantler m

d_l Demand of customer l

e_m Capacity of dismantler m

s_{ij} Unit cost of production in manufactory j using materials from supplier i

t_{jk} Unit cost of transportation from each manufactory j to each DC k

u_{kl} Unit cost of transportation from DC k to customer l

v_{km} Unit cost of transportation from DC k to dismantlers m

w_{mj} Unit cost of transportation from dismantlers m to manufactory j

Ru_{lk} Unit cost of recovery in DC k from customer l

f_j Fixed cost of opening manufactory in potential location j

g_k Fixed cost of opening distribution/collection centers in potential location k

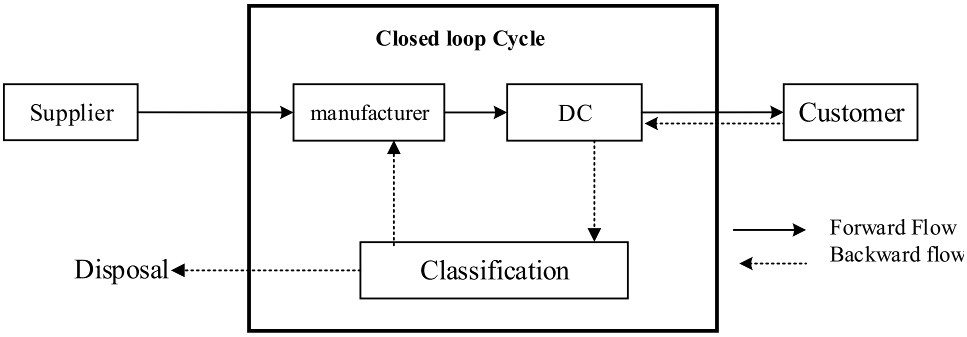


Fig. 1 The overall structure of SCND and collection of investigated products

h_m Fixed cost of opening dismantlers in potential location m

3.3 Decision variables

3.3.1 Continuous decision variables

X_{ij} Quantity produced at manufactory j using raw materials from supply i

Y_{jk} Amount shipped from manufactory j to DC k

Z_{kl} Amount shipped from DC k to customer l

O_{km} Amount shipped from DC k to dismantler m

Rd_{mj} Amount shipped from dismantler m to manufactory j

Rz_{lk} Quantity recovered at DC k from customer l

3.4 Binary decision variables

$$\alpha_j\begin{cases}1: & \text{if plant is opened in location } j\\ 0: & \text{otherwise}\end{cases}$$

$$\beta_k\begin{cases}1: & \text{if DC is opened in location } k\\ 0: & \text{otherwise}\end{cases}$$

$$\delta_m\begin{cases}1: & \text{if dismantler is opened in location } m\\ 0: & \text{otherwise}\end{cases}$$

4 Mathematical model

According to the aforementioned symbols, the mixed-integer linear programming model for the design of a forward/reverse logistics network to minimize costs is presented as follows:

$$\text{Opening costs}: \sum_j f_j\alpha_j + \sum_k g_k\beta_k + \sum_m h_m\delta_m \tag{1}$$

$$\text{Transportation costs}: \sum_i\sum_j s_{ij}x_{ij} + \sum_j\sum_k t_{jk}y_{jk} + \sum_k\sum_i u_{kl}z_{kl} + \sum_k\sum_m v_{km}o_{km} + \sum_m\sum_j w_{mj}Rd_{mj} + \sum_l\sum_k Ru_{lk}Rz_{lk} \tag{2}$$

$$\text{Landfilling costs}: \varphi\sum_m OP_m \tag{3}$$

4.1 Constraints

$$\sum_j x_{ij} \le a_i, \forall i \tag{4}$$

$$\sum_k y_{jk} \le b_j\alpha_j, \forall j \tag{5}$$

$$\sum_l z_{kl} + \sum_m o_{km} \le Sc_k\beta_k, \forall k \tag{6}$$

$$\sum_m o_{km} \le SP_k\beta_k, \forall k \tag{7}$$

$$\sum_i x_{ij} + \sum_m Rd_{mj} = \sum_k y_{jk}, \forall j \tag{8}$$

$$\sum_j y_{ik} = \sum_l z_{kl}, \forall k \tag{9}$$

$$\sum_l Rz_{lk} = \sum_m o_{km}, \forall k \tag{10}$$

$$\sum_k o_{km} + \sum_j Rd_{mj} + OP_m, \forall m \tag{11}$$

$$\sum_l z = d_l, \forall l \tag{12}$$

$$OP_m \le pl_m\sum_k o_{km}, \forall m \tag{13}$$

$$OP_m \ge pl_m\sum_k o_{km} - \varepsilon, \forall m \tag{14}$$

$$\alpha_j, \beta_k, \delta_m \in \{0,1\}, \forall j, k, m \tag{15}$$

$$x_{ij}, y_{jk}, z_{kl}, o_{km}, Rd_{mj}, Rz_{lk}, OP_m \in N \cup \{0\}, \tag{16}$$

The objective function of the problem is to minimize costs, which include transportation costs within the network and the fixed costs of opening units in potential locations. The constraints generally contain two types: facilities capacity constraints and flow interaction constraints in nodes. Terms (4) and (5) are related to the capacity constraints of suppliers and manufacturers in the forward flow. Constraint (7) indicates that the total flow in the forward and reverse path that passes through the distribution/collecting center k must not be greater than the capacity of that

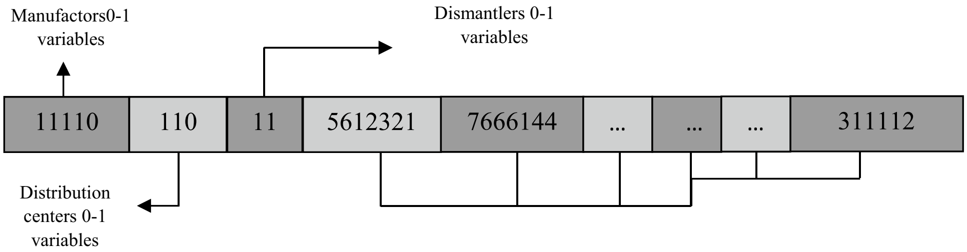


Fig. 2 Representation of spanning tree solution

center. Constraints (9) and (13) ensure that in reverse flow the flows from the distribution/collection and dismantlers will not exceed their capacity. Constraint (11) indicates the relationship between the amount shipped from dismantler to manufactory and the quantity recovered from the customer.

Constraints (6), (8), (10), and (14) are flow constraints within the network indicating that the amount of input to each node must be equal to the amount of output from the same node. Constraint (12) ensures that all customer demands are met. Constraint (15) relates to the binary variables of the problem. Constraint (16) satisfies the requirement of non-negativity in continuous variables.

In the presented closed-loop logistic model we have constraint as many as ($I+2\ J+4\ k+4\ M$) and the number of variables equals to ($I\times J+J\times k+k\times L\times L\times k+k\times M+M+J+J\times k+2\ M$) from which as many as ($I+k+M$) are binary variables. From this amount, $2\ M$ variables and M additional constraints are the contribution of the model linearization procedure.

Since the problem of integrated logistics network design includes the problem of relocation of facilities with different capacities known as NP-Completed problems, the problem presented is one of the NP-Hard problems. Exact methods cannot be utilized in real dimensions to solve such problems. Hence, in this study, the simulated annealing algorithms and finally a hybrid algorithm are used.

5 Solution approach

5.1 Solution encoding using the modified spanning tree method

Since the closed-loop supply chain design problem is a capacitated location-allocation problem and also can be viewed as a multiple-choice Knapsack problem, it is known to be an NP-hard problem [3, 34–36].

The spanning-tree encoding method is generally used in problems where there is no loop and since the problem of the integrated network design contains a loop, by dividing it into smaller parts, where there is no loop, this can be resolved.

In the integrated logistics network design problem under study, the first two levels, namely suppliers' level and manufactories level, are represented by the splay tree encoding method. A solution to the problem on hand is represented by a list of parameters called the solution representation or genome. Genomes are generally represented as a simple sequence of data. In the simulated annealing algorithm, the same method for representing the encoded solution which uses the tree method is applied to represent the genomes. Thus, each genome (solution) as illustrated in Fig. 2 includes 12 segments and contains $2\times(I+J+J+K+K+L+L+M+L+I)$ arrays. Both consecutive segments represent the flow between two consecutive levels in the supply chain network.□□□□□□

3	*4*	*2*	*5*	*8*	*5*	*3*	*9*

Fig. 3 Solution matrix using the spanning tree method

2	*3*	*4*	*5*	*6*	*7*	*8*	*9*

Fig. 4 The position of each allele in each solution matrix

5.2 Decoding the solutions

By using an example, we will explain how to decode, convert and analyze a solution into clear forms. Suppose the matrix in Fig. 3 is a sample solution that shows the relationship of 9 nodes to one another.

The limit for the values in solution matrix is between 1 to the total nodes of the start and end point, which is between 1 to 9 in this example. Since the C matrix consists of 8 parts, we can understand that there are 9 nodes in the problem because in this method the number of matrix components is always one unit less than the number of nodes in the problem. To understand the relationship between nodes, we first need to create an 8×1 matrix containing the numbers 2 to 9, thus the C' matrix is displayed as follows:

By putting these two matrices together (Fig. 3 and 4 shown in Fig. 5 together) the relationship between the nodes is shown. So, we remove the first component from matrix C' and the second component from matrix C. Thus, the relationships can be expressed as follows:

(2,3), (3,4), (4,2), (5,5), (6,8), (7,5), (8,3), (9,9)

The generated tree is illegal and needs some operations to be repaired (Fig. 5). As can be seen, there are some kinds of infeasible solutions in this kind of representation of the SCND problem. First node 1 and node 9 have no connection with any of the other nodes. Second, there is an illegal cycle in nodes 2, 3, and 4 and third, there is no connection to nodes 5 and 7.

To resolve these unjustified conditions in representing the solutions they must:

(1) Avoid duplication in a sequence of solutions, i.e. for those researchers who use the MATLAB coding program, apply the randperm function for generating the solutions.
(2) After the valuation, check that no values of any corresponding alleles of matrices C and C' are the same.
(3) If the number "one" is not examined in any allele of the answer, randomly replace one of the answer components with the number "one".

As we can see from Fig. 2, our initial solution has two parts: The first part deals with the openness or closure of each facility, and the second part deals with the flow rate among the nodes of the network, which we will discuss below.

To form the first part of the matrix whose length is ($j+k+M$), that is, the sum of the potential locations for the opening manufactories, distribution centers, and dismantlers, first their assigned houses should be filled in randomly with zero and one numbers. The number one represents openness and zero is an indication of the facility being closed. The second step is to see if the total capacity of the facilities can meet customer needs.

5.3 Flow rate in solution representation matrix

In the second part of the genome, according to the proposed model assumptions, products should flow only at unequal and sequential levels of the network. Therefore, we need to restrict portions of the matrix to specific numbers in the solution representation matrix, thereby fixing the 2 problems mentioned therein. For example, if I is the number of suppliers and J is the number of manufactories, the solution matrix length will be $I+J\text{-}1$. For the first $I\text{-}1$ houses of the matrix we have to randomly put $I+1$ to $I+J$ numbers inside the matrix and for the second J houses of the matrix we have to put the numbers 1 to I. That way we can overcome both of the problems mentioned in the representation of the solution matrix.

Representing the encoded solutions using the splay tree method involves $I+J\text{-}1$ bridges between suppliers and manufactories, while the actual number of the connecting lines is $I\times J$. In fact, this method enables us to save time on the algorithm and avoid the costs resulting from it.

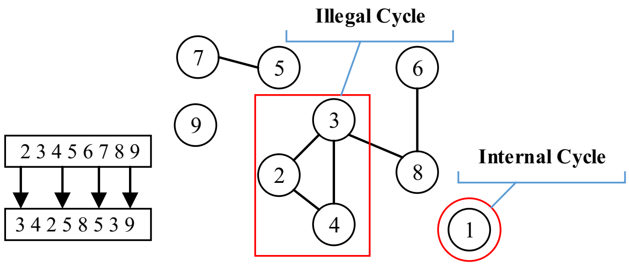


Fig. 5 Illegal generated tree

5.4 Valuing orders in solution matrix

There are two types of transport systems in a closed-loop logistics network: push system and pull system. The push system is used on the forward side of the network and delivers materials from suppliers to customers. The pull system inverts materials from customers to manufactories or landfilling. Valuation ordering in the solution matrix should first start from the pull system in the reverse direction and then continue with the push system in the forward direction. This way we can accurately determine the number of manufactories' demands from suppliers and all costs can be reasonably calculated.

5.5 Determining the flow rate and the value of the objective function

Solution matrices are evaluated using the objective function (fitness function). Real problems are usually accompanied by limitations that appear in modeling. To reach the overall optimum, the feasible space of the problem must be searched extensively. For this purpose, there must be a balance between feasible and non-feasible solutions, otherwise, there will be no variation in the solutions, and we will focus only on a limited portion of the feasible space. The optimal result will therefore be local. The procedure is described in detail in the decoding section of the solution matrix.

The initial idea, which later became the basis of the simulated annealing algorithm approach, was first proposed by Metropolis in 1953 based on the process of cooling or annealing of materials in statistical thermodynamics science.

The use of the cooling process in optimization discussions was first suggested by Kirkpatrick in 1980 as simulated annealing. The SA approach is one of the local search (neighborhood) methods. This approach unlike other local search methods is not dependent on the starting point (initial solution) and can largely be freed from the trap of local optimizations. This is due to the acceptance of non-refinement motions of the objective function. The quality of the obtained solution will also depend on the rate of reduction of the temperature parameter (t). According to statistical thermodynamic rules, the probability of elevating the energy of the system to ΔE at temperature t equals to:

$$P(\Delta E) = e^{\frac{-\Delta E}{kt}} \tag{29}$$

In the simulated annealing, the probability of accepting non-refinement motions at any temperature is calculated according to the above equation. Such that ΔE represents the change in the value of the objective function for a change in the input value from the current solution x to the new solution x'. By lowering the temperature, the aforementioned probability also decreases, as it is necessary to spend more time at the lower temperatures to find the local optimum.

$$\Delta E = E(x') - E(x) \min EandE(x') > E(x) \tag{30}$$

In the general case, the following function is used to reduce the temperature, such that T_r represents the temperature in the r the iteration of the algorithm:

$$T_{(r+1)} = \alpha T_r 0 < \alpha < 1 \tag{31}$$

5.6 Definition of the symbols

NNumber of accepted solutions at each temperature (exit criteria from inner loop)

RMaximum temperature transfers (stopping and exit criteria from outer loop)

T_0 Initial temperature

α Cooling schedule

X Feasible solution

$f(X)$ Objective function value for every x

n Temperature transfer counter

$_r$Temperature transfer counter

Following the research activities regarding the development of simulated annealing, which it's general pseudocode is illustrated in Fig. 6, Lu, Yuan, and Zhang [37] presented a cloud theory-based approach that enabled better neighborhood search and obtained better solutions.

The cloud model is a model that incorporates qualitative concepts and quantitative representation that utilizes natural language for this purpose [38].

If the three digital characteristics (En, Ex, He) and a certain μ_0 are given, then a drop of cloud drop (X_i, μ_0) can be generated by a generator, which is called Y condition cloud generator. The procedure of genating Y condition in illustrated in Fig. 7.

```
$r = 0$ , $T = T_0$, $X_{best} = \emptyset$
Generate $X_0$
$X_{best} = X_0$
Do(Out side loop)
$n = 0$
while $T < 10^{-10}$
Do insert and swap mutation $X_n$ as : $\Delta C = C(X_{new}) - C(X_{best})$
If $\Delta C < 0$
$X_{best} = X_{new}$ ; $n = n + 1; X_n = X_{new}$
Else
Generate $y \to U(0,1)$ Randomly
Set $Z = Exp(-\Delta C / Tr)$
if $y < z$
$n = n + 1; X_n = X_{new}$
End
End
```

Fig. 6 Simulated Annealing Optimization Algorithm

```
INPUT :$(Ex, En, He)$
Output :$((x_{i,}\mu_0),\ldots,(x_{n,}\mu_0)$
for i = 1 : n
   $En' = \mathrm{randn}(En, He)$
   $X_i = Ex \pm En'\sqrt{-2\ln(\mu_0)}$
   $drop(x_i, \mu_0)$
Endfor
```

Fig. 7 Procedure of generator Y

The cloud-based simulated annealing algorithm follows the Metropolis rule and uses the Y condition normal cloud generator to produce nearly continuous annealing temperature, which influences the state generating and accepting process, and then complies with the physical laws better. The cloud based nature of this algorithm and its main characters are shown in Fig. 8 which is derived from [37]. Moreovere Fig. 9 illustrates the steps of running the algorithm according to the explained Y conditions.

The random change of annealing temperature after importing the cloud theory can increase the diversity of the searched individual and better avoid being trapped by a local minimum than a normal cloud-based simulated annealing algorithm. Moreover, the stable tendency of annealing temperature can faster detect better solutions and thus improve the efficiency of the simulated annealing algorithm. In fact, by revising the temperature change pattern in the SA algorithm, we are likely to see more speed and accuracy in solving the NP-hard problems. In

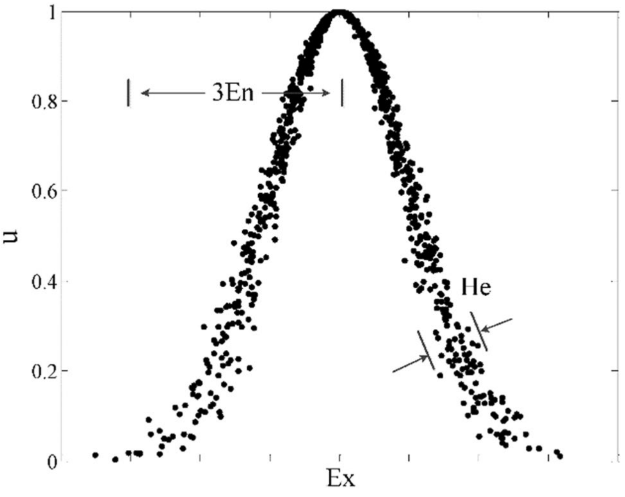


Fig. 8 Three digital characteristics of a normal cloud. [37]

```
$Let\ T_0 = 0.1$
$T = T_0, s = s_0, k = k_{0;}$
$While\ T > 10^{-10}$
$He = T; En = T, \mu = 1 - T;$
$En' = (En + He - 1/3rand(0,1));$
$T' = En'\sqrt{-2\ln(\mu_0)};$
$L_1 = Value\ of\ spanning\ tree\ based\ decoded\ fitness\ function;$
$s' = Do\ swap\ and\ insertion\ mutation\ randomlly$
$L_2 = Value\ of\ the\ new\ solution$
    $If\ L_2 \le L_1$
    $S_0 = S';$
    $Else$
    $with\ probability\ of\ P = e^{\frac{-\Delta}{T'}}$
    $End$
$k = k + 1$
$T = T_0\lambda^k$
$End\,While$
```

Fig. 9 Cloud-based simulated annealing optimization algorithm

Table 2 Problems sample

Problem	Suppliers	Manufactories	DCs	Customers	Dismantlers
1	3	5	3	4	2
2	6	10	6	8	4
3	12	20	12	16	8
4	18	30	18	24	12
5	21	35	21	28	14
6	24	40	24	32	16
7	27	45	27	36	18
8	30	50	30	40	20
9	36	60	36	48	24
10	40	66	40	53	26
11	46	76	46	61	30
12	49	81	49	65	32
13	55	91	55	73	36
14	58	96	58	77	38

Table 3 Parameters values

Parameter	Range
Capacity of Suppliers	$\sim uniform(350, 700)$
Capacity of manufactories	$\sim uniform(300, 600)$
Suppliers' fixed cost	$\sim uniform(800, 2300)$
Capacity of Distribution/Collection centers	$\sim uniform(600, 950)$
Distribution/Collection centers' fixed cost	$\sim uniform(800, 1800)$
Demand	$\sim uniform(300, 500)$
Capacity of dismantlers	$\sim uniform(350, 800)$
Dismantlers' fixed cost	$\sim uniform(800, 1200)$
pd_k, pc_l,pl_m	*10%*

Table 4 Range of shipping costs per level

Transportation cost	Range
Supplier–Manufactories	$\sim uniform(2, 10)$
Manufactories–Distributor	$\sim uniform(2, 9)$
Distributor–Customer	$\sim uniform(2, 8)$
Customer–Distributor	$\sim uniform(2, 8)$
Distributor–Dismantlers	$\sim uniform(1, 6)$
Dismantlers–Manufactories	$\sim uniform(2, 10)$

the following section, we will discuss the results of the closed-loop network design problem.

6 Computational results

For the integrated design problem of forward and reverse logistics networks, 20 test problem samples are generated from small to large sizes. These problems are generated with the help of the problems used in similar papers. Other problem data are also extracted from Wang and Hsu [7] and Yadegari et al. [25]. (Tables 2, 3 and 4).

In this section, the performance of the cloud-based simulated annealing algorithm is compared to the performance of two of the algorithms in the literature review that used the spanning tree representation method to solve the NP-hard problem. The first algorithm is the matching genetic algorithm developed in [7] and the second is the simulated annealing study developed in [25]. The three concerned

Table 5 The results obtained from the implementation of the algorithms regarding the objective function in 25%- and 100%-time intervals as well as the first time to reach the best solution

Algorithm	CSA			GA			SA		
size	First Ans	Last Ans	Best Time	First Ans	Last Ans	Best Time	First Ans	Last Ans	Best Time
1	29,898	29,848	5	29,958	29,848	14	29,848	29,848	7
2	52,416	51,601	21	51,975	51,601	30	53,413	52,713	32
3	90,513	90,253	260	96,217	90,253	371	93,433	92,478	343
4	156,013	153,053	321	164,563	153,053	522	157,853	157,023	258
5	158,315	156,228	524	174,900	156,228	862	166,368	166,271	844
6	169,084	164,118	862	189,183	164,118	892	170,678	164,524	1071
7	229,068	221,728	1080	251,038	221,728	1410	233,778	231,338	411
8	410,948	228,155	1438	29,848	250,450	1416	249,080	228,411	1477
9	244,077	277,183	1929	51,535	296,663	1989	297,373	275,923	1981
10	291,623	273,898	2242	91,663	292,270	2249	289,028	279,158	2323
11	295,554	315,003	3001	159,913	330,653	3099	326,705	315,643	3229
12	321,353	313,668	4201	161,651	335,928	4420	333,048	321,888	4335
13	343,001	363,004	6173	179,105	387,263	6189	385,317	368,551	6076
14	365,938	396,010	6766	232,058	412,676	6990	411,368	396,598	6992

meta-heuristic algorithms are analyzed for efficiency and effectiveness. The results obtained from the implementation of the algorithms on the given problems are pertaining to the best objective function value in the closed-loop network design problem. These costs include opening fixed costs and transportation costs within the network. Each algorithm was run on 14 problem samples of small, medium and large size, 10 times each. After each run, the obtained results were recorded corresponding to the best objective function (lowest cost) at 25% of the time and also total allocated time for the algorithm to stop and the first time to reach the best value of the objective function. For this purpose, each algorithm was encoded using *MATLAB7.11.0* (R2014$_a$) software. All problems were run on a computer with a dual-core processor 2.20 Gh, 2.2 GHz and 3.00 GB original memory using the Windows 7 operating system. After applying the meta-heuristic algorithms to the concerned problems, one-way ANOVA was performed using *MINITAB*^{16} software to perform detailed statistical analysis.

The result obtained from running three algorithms: genetic meta-heuristic, simulated annealing (SA), and cloud-based simulated annealing on 14 sample problems in small, medium, and large sizes are presented in Table 5. The results of the implementation of algorithms are analyzed from the following two points of view:

(1) The best value of the objective function, as a measure of effectiveness, the first time to reach the best value of the objective function

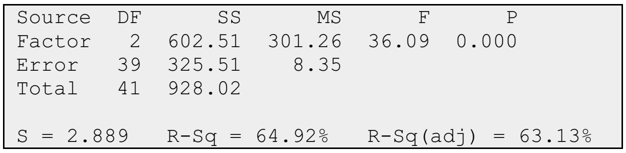
(2) The best value of the objective function in the first 25% of the total given period as a performance criterion.

In this study, we use the relative percentage deviation (RPD) to evaluate different meta-heuristic algorithms in three categories small, medium, and large. The method of calculation is given in Eq. (32), which represents the solution obtained by the developed algorithms and is the smallest value of each algorithm implemented in all three concerned samples of the problems. The RPD value indicates how far the obtained solutions are from the best solution in each algorithm. The longer the distance, the worse the generated solutions, and in turn the lower the distance, the better the generated solutions and thus the more suitable the algorithms.

$$PRD = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \times 100 \tag{32}$$

According to the equation, the results obtained from the implementation of different algorithms each time are transformed by the RPD criterion into an equal scale for all 14 concerned problem samples. The hypothesis tested regarding the mean equality of the obtained solutions in terms of the best value of the objective function in the given period is the first time to obtain the best solution by the four algorithms developed in different sizes.

As stated in the problem definition, the purpose of this study is to minimize considered costs. As a result, the first



```
Source   DF       SS       MS       F       P
Factor    2   602.51   301.26   36.09   0.000
Error    39   325.51     8.35
Total    41   928.02

S = 2.889   R-Sq = 64.92%   R-Sq(adj) = 63.13%
```

Fig. 10 Output from the variance analysis for the best value of the objective function in the first 25% of the running time

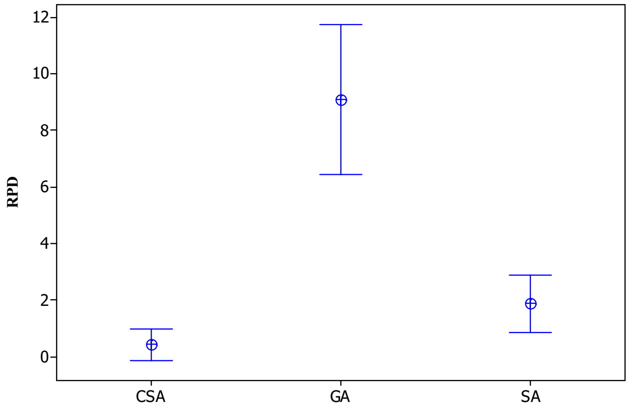
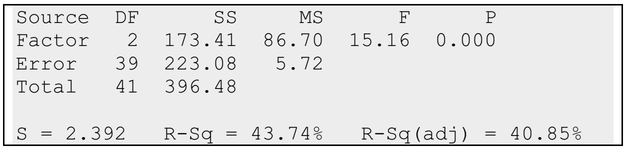


Fig. 11 Diagram of the mean and 95% confidence interval for the best value of the objective function in the first 25% of the running time



```
Source   DF       SS      MS       F       P
Factor    2   173.41   86.70   15.16   0.000
Error    39   223.08    5.72
Total    41   396.48

S = 2.392   R-Sq = 43.74%   R-Sq(adj) = 40.85%
```

Fig. 12 The Output obtained from variance analysis for the best value of the objective function in the total running time

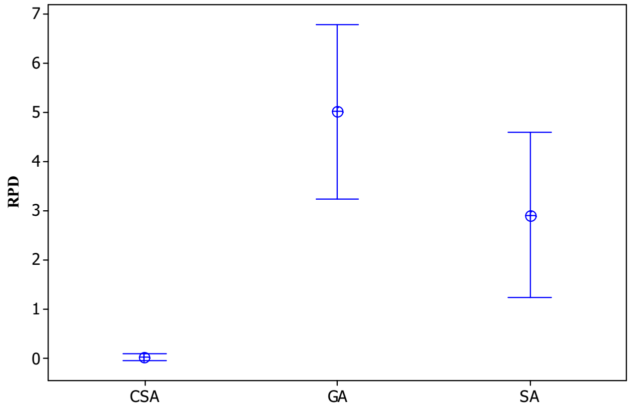


Fig. 13 Diagram of the mean in 95% confidence interval for the best value of the objective function in the total running time

hypothesis that pertains to the mean equality of the best value of the objective function is defined by the three algorithms developed in small, medium and large sizes as stated below:

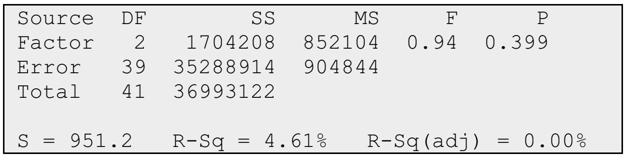
$$\begin{cases} H_0 \ : \ \mu_{GA(cost)} = \mu_{SA(cost)} = \mu_{CSA(cost)} \\ H_1 \ : \ \text{At least one of the algorithms has a different mean than the other algorithms} \end{cases} \tag{33}$$

The first hypothesis, H_0 states that the means obtained from the developed algorithms are not significantly different from each other, but H_1 states that at least one of the algorithms has a different mean than the other algorithms. One-way ANOVA was used to test this hypothesis and the results are shown in Fig. 10.

According to Fig. 10, the p-value obtained from one-way ANOVA results is zero. The zero represents the number of spins that the hypothesis has not received any confirmation from the sample, thus rejecting the hypothesis. As is evident, at the 95% confidence level the assumption is rejected and the assumption is confirmed. Meaning that there is a significant difference between the algorithms in terms of the best value of the objective function. Now that it is clear that there is a significant difference between the algorithms, they need to be evaluated against one another to determine which algorithms have a significant difference and to what extent. The algorithms also need to be ranked in terms of effectiveness. As a result, Tukey's test was used to further analyze and find significant differences between the algorithms. Tukey's test, by binary grouping the algorithms, compares them in terms of significant differences as well as their values.

In Figs. 10 and 11, we examined the output obtained from the variance analysis, diagram of the mean, and 95% confidence interval for the best value of the objective function in the first 25% of the running time. This is carried out to evaluate the convergence speed of the algorithms at the beginning of each task. According to these two analyses performed on this criterion, the algorithms of SA and CSA are superior to the GA algorithm and converge at a higher speed at the beginning of the compared period. But there is no significant difference between the SA and CSA algorithms.

In Figs. 12 and 13, the output obtained from the variance analysis, the diagram of the mean, and 95% confidence interval for the best value of the objective function after reaching the stopping conditions for evaluation of the effectiveness of the algorithms are assessed. Based on these two analyses, the hypothesis is rejected and according to this criterion, the CSA algorithm is superior to the SA algorithm and the SA



```
Source   DF         SS       MS      F       P
Factor    2    1704208   852104   0.94   0.399
Error    39   35288914   904844
Total    41   36993122

S = 951.2   R-Sq = 4.61%   R-Sq(adj) = 0.00%
```

Fig. 14 Output from the variance analysis for the first time slice to reach the best solution

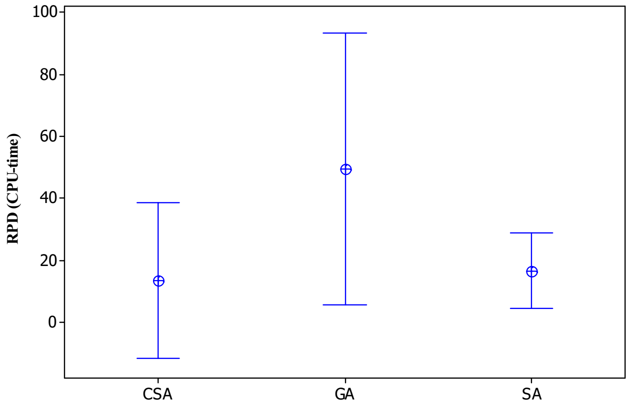


Fig. 15 Diagram of the mean and 95% confidence interval for the first time slice to □reach the best solution□□□□

algorithm is superior to the GA algorithm and converges at a higher speed at the beginning of the compared period. But there is no significant difference between the SA and CSA algorithms.

In Figs. 14 and 15, the output obtained from the variance analysis, the diagram of the mean, and 95% confidence interval are examined for the first time to reach the best solution for the evaluation of the algorithm's performance. According to both performed analyses and based on this criterion, no significant difference between SA, CSA, and GA algorithms is noted.

The results of Tukey’s test showed that the CSA algorithm is significantly different from other SA and GA algorithms in terms of the best objective function value, however, GA and SA algorithms do not have significant differences against each other. On the one hand, this test was performed on the best value of the objective function on the first 25% of the running time, which showed that the GA algorithm is significantly different from the other two algorithms and on the other hand, the SA and CSA algorithms do not significantly differ from each other. As a result, it can be concluded that the CSA and SA algorithms are in a better position than the GA in terms of convergence speed at the beginning of the execution of the algorithms.

7 Conclusion

In this paper, after explaining the closed-loop supply chain design model and discussing the necessity and importance of the problem, we review the literature and studies in this field and identify the research needs derived from this study. By considering a variety of characteristics and real-world conditions a new algorithm in this field was presented using the complex spanning tree representation method to improve the running time and provide more accurate solutions to the problems of different sizes. To evaluate the quality of the designed algorithm, experimental problems in different dimensions and specifications were designed to simulate real-world conditions. These problems were solved by the following three algorithms:

(1) Genetic algorithms (derived from the literature),
(2) Simulated annealing (derived from the literature) and;
(3) Cloud-based simulated annealing algorithms.

These algorithms were then compared to one another in terms of the quality of the obtained solutions and in terms of small, medium, and large running times. The results of these comparisons at 95% confidence level are:

- In the first 25% of stop time:

1CSA
1SA
2GA

In fact, the quality of the solutions of the proposed algorithm was better than the GA algorithm in literature and compared to the SA algorithm in the literature it was not significantly different.

- From the point of view of the first reaching time that the algorithms achieve optimal solutions:

1GA
1SA
1CSA

As a result, there was no significant difference between SA and CSA algorithms in terms of convergence speed of the algorithms, but both algorithms were faster than GA.

• From the point of view of the best value of the objective function over the total given period:

1CSA
2SA
3GA

As a result, there was a significant difference among the three algorithms in terms of algorithm effectiveness, where CSA performed better than the other algorithms. It can also be concluded that by incorporating the cloud theory, the random conversion of annealing temperature can increase the diversity of the searched space and avoid being trapped by a local minimum more efficiently than an original SA algorithm. Also, Spanning tree representation incorporates the minimum alleles for the solution representation which is one of the most advantages of this method. However, we should note that this solution representation may encounter three kinds of infeasible solutions without respecting the constraint. These infeasible solutions need some repair mechanism which would lead to a more time-consuming algorithm. As a result, applying this kind of Solution representation need a trade-off analysis to distinguish if the final function of the algorithm is acceptable or not.

To expand future research in this field, it is suggested that the spanning tree representation method be utilized given that it uses fewer arrays to represent the meta-heuristic algorithms solutions in the field of network design. Also, new meta-heuristic approaches that have shown their effectiveness in combinatorial optimization models should be implemented in the design of closed-loop networks where the usage of these algorithms has been less favored than in other areas.

References

1. Tavana M et al (2022) A comprehensive framework for sustainable closed-loop supply chain network design. J Clean Prod 332:129777
2. Papadimitrakis M et al (2021) Metaheuristic search in smart grid: a review with emphasis on planning, scheduling and power flow optimization applications. Renew Sustain Energy Rev 145:111072
3. Fleischmann M et al (2001) The impact of product recovery on logistics network design. Prod Oper Manag 10(2):156–173
4. Pishvaee MS, Razmi J (2012) Environmental supply chain network design using multi-objective fuzzy mathematical programming. Appl Math Model 36(8):3433–3446
5. Govindan K, Fattahi M, Keyvanshokooh E (2017) Supply chain network design under uncertainty: A comprehensive review and future research directions. Eur J Operat Res 263:108–141
6. Ko HJ, Evans GW (2007) A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs. Comput Oper Res 34(2):346–366
7. Min H, Ko H-J (2008) The dynamic design of a reverse logistics network from the perspective of third-party logistics service providers. Int J Prod Econ 113(1):176–192
8. Lee D-H, Dong M (2009) Dynamic network design for reverse logistics operations under uncertainty. Trans Res Part E: Logist Trans Rev 45(1):61–71
9. Wang H-F, Hsu H-W (2010) A closed-loop logistic model with a spanning-tree based genetic algorithm. Comput Oper Res 37(2):376–389
10. Govindan K, Fattahi M (2015) Investigating risk and robustness measures for supply chain network design under demand uncertainty: a case study of glass supply chain. Int J Production Econ 183:680–99
11. Syarif I, Prugel-Bennett A, Wills G (2012) *Unsupervised clustering approach for network anomaly detection*. In: International conference on networked digital technologies. Springer
12. Wang Z et al (2021) A new configuration of autonomous CHP system based on improved version of marine predators algorithm: a case study. Int Trans Electr Energy Syst 31(4):e12806
13. Ramezani M, Bahmanyar D, Razmjooy N (2020) A new optimal energy management strategy based on improved multi-objective antlion optimization algorithm: applications in smart home. SN Appl Sci 2(12):1–17
14. Yang Z et al (2020) Model parameter estimation of the PEMFCs using improved barnacles mating optimization algorithm. Energy 212:118738
15. Yuan Z et al (2020) A new technique for optimal estimation of the circuit-based PEMFCs using developed sunflower optimization algorithm. Energy Rep 6:662–671
16. Jayaraman V, Pirkul H (2001) Planning and coordination of production and distribution facilities for multiple commodities. Eur J Oper Res 133(2):394–408
17. Jayaraman V, Gupta R, Pirkul H (2003) Selecting hierarchical facilities in a service-operations environment. Eur J Oper Res 147(3):613–628
18. Li J, Chen J, Wang S (2011) Introduction. Risk management of supply and cash flows in supply chains. Springer, pp 1–48
19. Tsiakis P, Papageorgiou LG (2008) Optimal production allocation and distribution supply chain networks. Int J Prod Econ 111(2):468–483
20. Syarif A, Yun Y, Gen M (2002) Study on multi-stage logistic chain network: a spanning tree-based genetic algorithm approach. Comput Ind Eng 43(1):299–314
21. Elhedhli S, Merrick R (2012) Green supply chain network design to reduce carbon emissions. Transp Res Part D: Transp Environ 17(5):370–379
22. Krikke H, van Harten A, Schuur P (1999) Business case Oce: reverse logistic network re-design for copiers. OR-Spektrum 21(3):381–409
23. Aras G, Crowther D (2008) Governance and sustainability: an investigation into the relationship between corporate governance and corporate sustainability. Manag Decis 46(3):433–448
24. Gírio FM et al (2010) Hemicelluloses for fuel ethanol: a review. Biores Technol 101(13):4775–4800
25. Govindan K, Khodaverdi R, Jafarian A (2013) A fuzzy multi criteria approach for measuring sustainability performance of a supplier based on triple bottom line approach. J Clean Prod 47:345–354
26. Lu Z, Bostel N (2007) A facility location model for logistics systems including reverse flows: The case of remanufacturing activities. Comput Oper Res 34(2):299–323
27. Salema MIG, Póvoa APB, Novais AQ (2009) A strategic and tactical model for closed-loop supply chains. OR Spectrum 31(3):573–599
28. Devika K, Jafarian A, Nourbakhsh V (2014) Designing a sustainable closed-loop supply chain network based on triple bottom line approach: a comparison of metaheuristics hybridization techniques. Eur J Oper Res 235(3):594–615
29. Yadegari E et al (2014) An artificial immune algorithm for a closed-loop supply chain network design problem with different delivery paths. Int J Strategic Decision Sci (IJSDS) 5(3):27–46
30. Yadegari E et al (2015) A flexible integrated Forward/Reverse logistics model with random path-based memetic algorithm. Iran J Manag Studies 8(2):287
31. Yadegari E, Zandieh M, Najmi H (2015) A hybrid spanning tree-based genetic/simulated annealing algorithm for a closed-loop logistics network design problem. Int J Appl Decision Sci 8(4):400–426

32. Ghayebloo S et al (2015) Developing a bi-objective model of the closed-loop supply chain network with green supplier selection and disassembly of products: the impact of parts reliability and product greenness on the recovery network. J Manuf Syst 36:76–86
33. Kaya O, Urek B (2016) A mixed integer nonlinear programming model and heuristic solutions for location, inventory and pricing decisions in a closed loop supply chain. Comput Oper Res 65:93–103
34. Yi P et al (2016) A retailer oriented closed-loop supply chain network design for end of life construction machinery remanufacturing. J Clean Prod 124:191–203
35. Gen M, Cheng R (2000) Genetic algorithms and engineering optimization. John Wiley & Sons, UK
36. Gottlieb J, Paulmann L (1988) *Genetic algorithms for the fixed charge transportation problem*. In: *Evolutionary computation proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on* IEEE
37. Lv P, Yuan L, Zhang J (2009) Cloud theory-based simulated annealing algorithm and application. Eng Appl Artif Intell 22(4–5):742–749
38. Deyi L, Haijun M, Xuemei S (1995) Membership clouds and membership cloud generators [J]. J Comput Res Develop 32(6):15–20

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.