Advances in Industrial Engineering, Winter 2020, 54(1): 75-98
DOI: 10.22059/jieng.2021.321634.1758

RESEARCH PAPER

# A Comparison of Solution Methods for the Multi-Objective Closed Loop Supply Chains

**Omid Abdolazimi[a], Mitra Salehi Esfandarani[b], Maryam Salehi[b], Davood Shishebori[c,*]**

[a] *Department of Economy, Kharazmi University, Tehran, Iran.*
[b] *Department of Civil Engineering, University of Memphis, Memphis, TN, USA.*
[c] *Department of Industrial Engineering, Yazd University, Yazd, Iran.*

Received: 07 April 2021, Revised: 29 April 2021, Accepted: 01 May 2021


## Abstract

Increased pressure on natural resources, rising production costs, and multiple disposal challenges resulted in a growing global demand for integrated closed sustainable supply chain networks. In this paper, a bi-objective mixed-integer linear programming model is developed to minimize the overall cost and maximize the use of eco-friendly materials and clean technology. The paper evaluates the exact, heuristic, and metaheuristic methods in solving the proposed model in both small and large sizes. The sensitivity analysis was conducted on the LP-metric method as it outperformed the other two exact methods in solving the small size problems. The evaluation of LP-metric, modified ε-constraint, and TH as the exact methods and Lagrange relaxation algorithm as the heuristic method in terms of solution value and CPU time revealed the inability of exact methods in solving the large size problems. The best combination of effective parameters for meta-heuristic algorithms was determined using the Taguchi method. The evaluation of MOPSO, NSGA-II, SPEA-II, and MOEA/D as the metaheuristic methods by means of Number of Pareto Solutions (NPS), Mean Ideal Distance (MID), The Spread of Non-dominance Solutions (SNS), and CPU Time revealed the performance of these methods in solving the proposed model in a large size. The implementation of the VIKOR technique identified the SPEA-II as the best method among the meta-heuristic methods. This study provides a holistic view regarding the importance of selecting an appropriate solution methodology based on the problem dimension to ensure obtaining the optimum and accurate solution within the reasonable processing time.

**Keywords:**
Closed-loop Supply Chain (CLSC);
Exact Methods;
Lagrange Relaxation Algorithm;
Heuristic and Meta-Heuristic Algorithms;
VIKOR Technique

## Introduction

During the recent decades, economic merits associated with improving the product quality along with environmental and regulatory considerations have prompted the industrial sectors focusing more on sustainable practices like collecting, recovering, and recycling the products at the end of their useful life [1,2]. Effective consumption of resources and minimizing the environmental impacts have been pursued by the development of closed-loop supply chain (CLSC) networks [3], which made a significant contribution to the industry's enhanced profitability [4]. The CLSC networks have received much attention from research scholars and industries to implement the economic and/or environmental considerations [5,6,7,8]. For

* Corresponding author: (D.Shishebori)
Email: shishebori@yazd.ac.ir

instance, the total profit in terms of construction and operational process costs were maximized in the CLSC networks [9,10,11,12,13]. The environmental considerations were applied mostly by minimizing the $CO_2$ emission during production and transportation processes [14,15,16]. Based on the best of our knowledge, most of the recent studies have been focused solely on mathematical model development and not on optimizing the solving methods despite, its critical importance. Thus, this study conducted a comprehensive analysis comparing three exact, one heuristic, and four meta-heuristic methods in solving a bi-objective mathematical model which minimizes the cost and environmental issues in both small and large sizes.

The main contribution of this study is conducting a comprehensive comparison between exact, heuristic, and meta-heuristic methods in solving a bi-objective problem in both small and large sizes. This paper presents an analysis comparing four meta-heuristic methods (MOPSO, NSGA-II, SPEA-II, MOEA/D), and three exact methods (LP-Metric, $\varepsilon$-Constraint, TH). Furthermore, a heuristic model is implemented to demonstrate the inability of exact methods in solving a large size problem. The algorithms and solutions are compared based on several criteria including NPS, MID, SNS, and CPU time for both small and large dimensions problems.

The rest of this paper is presented as follows. The literature review is presented in Section 2 and the model description is presented in Section 3. The mathematical model is described in Section 4 and the solving methods are introduced in Section 5. The model performance is evaluated using various numerical examples which are presented in Section 6. Finally, the results are summarized, and future works are presented in Section 7.

## Literature Review

Accounting for economic considerations is essential in designing the CLSC networks, but it's not sufficient to fulfill the sustainability requirements. The economic considerations in designing the CLSC problems mainly involve minimizing the total cost or maximizing the total profits [11]. The study conducted by Wang and Hsu [10] presented a CLSC design problem that minimized the production, transportation, and fixed costs. The model developed by Devika et al. [17] minimized the production, transportation, handling, total opening, and purchasing costs. Paksoy et al. [15] minimized the $CO_2$ cost for a multi-product CLSC network. Talaei et al. [18] also minimized the total $CO_2$ emission in a CLSC design problem during the facility's construction, production, transportation, and product disposal. Özceylan et al. [14] developed a mathematical model to minimize the total $CO_2$ emission for the automotive industry during transporting the finished and returned products. Fahminia et al. [19] and Kannan et al. [20] minimized the total cost associated with $CO_2$ emission during production and transportation processes. A large number of studies addressed both environmental and economic factors while optimizing the CLSC network [21,22,23,24,]. Papen and Amin [26] developed a closed-loop supply chain model to minimize the environmental impacts and maximize the total profits in a bottled water production process. They implemented a weighted sum method (WSM), distance, and $\varepsilon$-constraint to solve the problem and selected the best supplier in terms of carbon footprint, cost, and on time delivery.

Developing the multi-objective models for designing and optimizing the CLSC network is a powerful tool to address both economic and sustainability concerns. A variety of exact, heuristic, and meta-heuristic methods has been applied to solve the multi-objective models. A two-layer supply chain distribution model was presented by Validi et al. [27] to minimize the total $CO_2$ emission and total costs. The MOGA-II was applied to solve this model by providing a set of non-dominated solutions along the Pareto frontier. Furthermore, a TOPSIS method was used to rank the set of solutions. The obtained results outperform the previous results from the existing solution methods. Niu et al. [28] developed a multi-period, multi-product, and multi-

objective mathematical model to maximize customer satisfaction and minimize the total costs. To solve the model, a cooperative evolutionary method was implemented. The results demonstrated the great performance of the model in terms of generating a non-dominated solution. Khalifezade et al. [29] designed a four-echelon supply chain model with multiple suppliers, producers, distributors, and customers to minimize the total operating costs and maximize the reliability for all supply chain elements. Then they applied a novel heuristic approach called, comparative particle swarm optimization to solve the model in large size instances. The results demonstrated the efficiency of the applied heuristic solving method. Bottani et al. [30] developed a bi-objective mathematical model to maximize the total profit and minimize the lead time along the supply chain. Solving this model using the Ant Colony Optimization (ACO) technique revealed its performance in designing a resilient supply chain. Amin and Zhang [31] developed a three-objective model for a CLSC network. They maximized the profit and weight of suppliers and minimized the defect rates, although the environmental factors were not accounted.

Few CLSC studies have considered the cost and environmental factors simultaneously while applying a variety of solving methods. Amin and Zhang [32] developed a mixed-integer linear programming model to minimize the total cost and consider the environmental factors in a CLSC. They applied the weighted sum method (WSM) and ε-constraint methods to solve the model. Alshamsi and Diabat [33] formulated a mixed-integer linear programming model to identify the optimal capacities of remanufacturing facilities and inspection centers, optimal locations for inspection and remanufacturing centers, and transportation decisions in a reverse supply chain. An exact method was applied to solve the problem on a large scale. Due to an unreasonable computation time of the conventional Benders Decomposition (BD) method, several accelerating methods such as logistics constraints, trust-region, restructuring of the problem, and Pareto-optimal cuts were added. Santibanez-Gonzalez and Diabat [34] proposed an upgraded Benders Decomposition Scheme to solve the remanufacturing supply chain design problem. A set of authentic inequalities were introduced to accelerate the conventional Benders algorithm convergence and improve the lower bound quality. Quasi Pareto-optimal cuts were implemented to improve the convergence. Based on computational results, the improved Benders decomposition scheme showed a better performance than the conventional algorithm. Alshamsi and Diabat [35] solved the problem with a heuristic method (Genetic Algorithm) in a large dimension. Also, Min and Ko (2006) proposed a mixed-integer nonlinear programming model for a reverse supply chain network. Then they applied a genetic algorithm for a multi-level reverse supply chain network problem. They considered the temporal and spatial combination of return products. The study conducted by Diabat et al. [36] considered a closed-loop inventory-location problem. The problem was formulated to select the best distribution and remanufacturing centers in a form of mixed-integer nonlinear location allocation model and then solved by a two-phase Lagrange relaxation algorithm.

## Model Description

The logistic network provided in this paper is a multi-level, multi-period, and multi-objective closed-loop supply chain. This network supports a wide variety of industries that are looking for efficient recycling plants. This logistic network involves multiple decisions regarding the location of production, distribution, inspection, collection, and recycling centers. Furthermore, it considers the inventory control policies like supply shortages at the end of each period. As demonstrated in Fig. 1, in the forward network, suppliers provide the raw materials; the products are manufactured at the production centers, distributed through distribution centers, and delivered to the demand markets. Similar to Papen's study [26], we considered the combined production and distribution centers. In the reverse network, the returned products are

transported to the inspection centers, where they get divided into three groups of non-recyclables, recyclable, and those which need to be repaired. The non-recyclable items are transported to the disposal centers and the recyclable items are transported to the recycling centers. After recycling, the recovered materials are transported to plant and distribution centers, and finally, get delivered to the demand market.

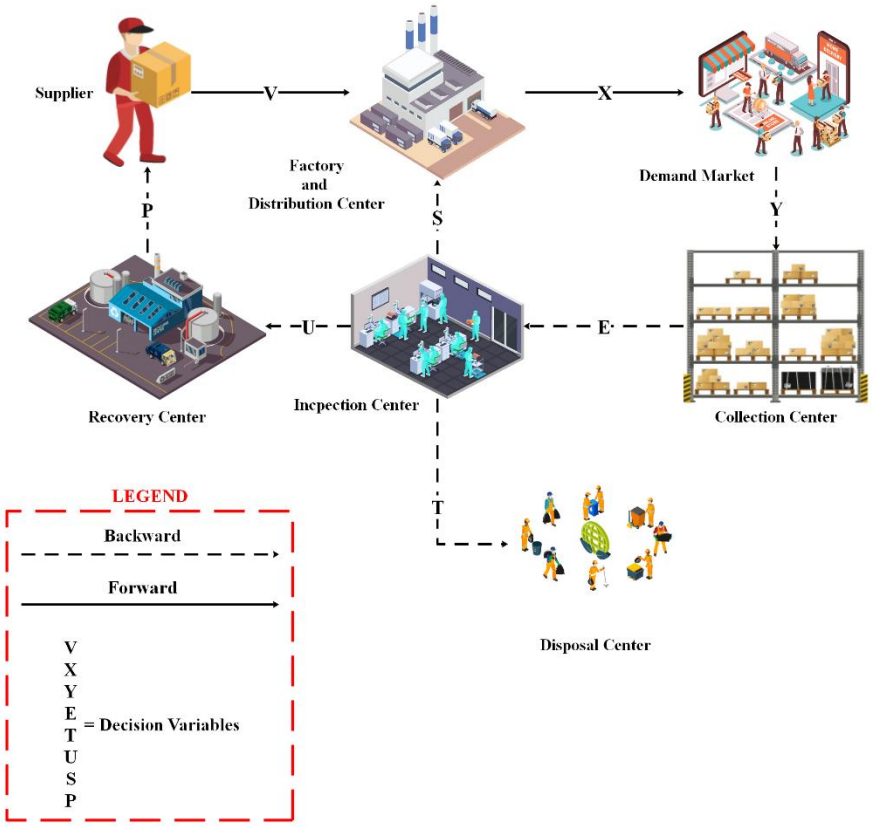



**Fig. 1.** The proposed CLSC supply chain network

In this paper, a multi-objective closed-loop logistics model for the returned products is presented to minimize the total cost and maximize the use of eco-friendly raw material and clean technology. This model identifies the potential locations for production/distribution, inspection, collection, and recycling centers with considering the following assumptions:

- The model is multi-level, multi-period, and multi-product;
- The locations of the demand markets, suppliers, and disposal centers are fixed;
- All returned products are collected from the demand market at the collection centers;
- The places of the potential production/distribution, inspection, and recycling centers are identified;
- The numbers of facilities to be built and their capacity are determined.

## Mathematical Model

The proposed model is formulated as a mixed-integer linear programming (MILP). The sets and parameters are shown below:

*Sets*

| | |
|---|---|
| *S* | Supplier's number, |
| *I* | Locations of potential manufacturing plants and distribution center, |
| *J* | Product's number, |

$K$ Locations of demand markets,
$L$ Locations of collection centers,
$A$ Locations of inspection centers,
$R$ locations of recovery centers,
$E$ Raw material's number,
$T$ Period's number.

*Parameters*

$A_{it}$ Production cost of product $j$ in period $t$,
$CS_e$ The cost of transporting raw material $e$ per km between plants and suppliers,
$B_j$ The cost of transporting product $j$ per km between markets and distribution centers,
$C_j$ The cost of transporting product $j$ per km between collection centers and demand markets,
$D_j$ The cost of transporting product $j$ per km between inspection centers and collection centers,
$O_j$ The cost of transporting product $j$ per km between disposal centers and inspection centers,
$CIP_j$ The cost of transporting product $j$ per km between plants and distribution centers and inspection centers,
$CIR_j$ The cost of transporting product $j$ per km between recovery centers and inspection centers,
$CPR_e$ The cost of transporting recovered raw material $e$ per km between plants and distribution centers and recovery centers,
$E_i$ Fixed opening cost for plant and distribution center $i$,
$F_l$ Fixed opening cost for collection center $l$,
$I_a$ Fixed opening cost for inspection center $a$,
$L_r$ Fixed opening cost for recovery center $r$,
$G_{jt}$ Saved cost of product $j$ in period $t$,
$SC_{est}$ Preparation cost of raw material $e$ from supplier $s$ in period $t$,
$CI_{jt}$ Cost of inspection for product $j$ in period $t$,
$H_{jt}$ Cost of disposal for product $j$ in period $t$,
$RC_{et}$ Cost of recovery for raw material $e$ in period $t$,
$Cb_{kjt}$ Cost of shortage for product $j$ at demand market $k$ in period $t$,
$CAS_{sjt}$ Capacity of supplier $s$ for product $j$ in period $t$,
$CAP_{ijt}$ Capacity of plant and distribution center $i$ for product $j$ in period $t$,
$Q_{ljt}$ Capacity of collection center $l$ for product $j$ in period $t$,
$CAD_{kjt}$ Capacity of demand market $k$ for product $j$ in period $t$,
$CAI_{ajt}$ Capacity of inspection center $a$ for product $j$ in period $t$,
$CAR_{rjt}$ Capacity of recovery center $r$ for product $j$ in period $t$,
$t_{si}$ The distance between plant $i$ and supplier $s$ based on the Euclidean method,
$t_{ik}$ The distance between plant and distribution center $i$ and demand market $k$ based on the Euclidean method,
$t_{kl}$ The distance between demand market $k$ and collection center $l$ based on the Euclidean method,
$t_{la}$ The distance between collection center $l$ and inspection center $a$ based on the Euclidean method,
$t_a$ The distance between inspection center $a$ and disposal center based on the Euclidean method,
$t_{ar}$ The distance between inspection center $a$ and recovery center $r$ based on the Euclidean method,
$t_{ai}$ The distance between inspection center $a$ and plant and distribution center $i$ based on the Euclidean method,
$t_{rs}$ The distance between recovery center $r$ and supplier $s$ based on the Euclidean method,
$D_{kjt}$ Demand of customer (demand market) $k$ for product $j$ in period $t$,
$\alpha_j$ Disposal rate of product $j$,

$\beta_j$ Recovery rate of product $j$,
$R_1$ Return rate of product $j$ from demand market to collection center,
$R_2$ Return rate of product $j$ from inspection center to plant,
$M_{ijet}$ Environmentally friendly raw material $e$ by plant and distribution center $i$ to produce product $j$ in period $t$,
$N_{ajt}$ Clean technology by inspection center $a$ to process product $j$ in period $t$.

### *Decision Variables*

$V_{esit}$ Quantity of raw material $e$ transported by supplier $s$ for plant and distribution center $i$ in period $t$,
$X_{ijkt}$ Quantity of product $j$ produced by plant and distribution center $i$ for demand market $k$ in period $t$,
$Y_{kljt}$ Quantity of returned product $j$ from demand market $k$ to collection center $l$ in period $t$,
$E_{jlat}$ Quantity of collected product $j$ from collection center $l$ to inspection center $a$ in period $t$,
$T_{jat}$ Quantity of returned product $j$ from inspection center $a$ to disposal center in period $t$,
$U_{jart}$ Quantity of returned product $j$ from inspection center $a$ to recovery center $r$ in period $t$,
$S_{jait}$ Quantity of returned product $j$ from inspection center $a$ to plant and distribution center $i$ in period $t$,
$P_{erst}$ Quantity of returned raw material $e$ from recovery center $r$ to supplier $s$ in period $t$,
$Lack_{kjt}$ The lack of product $j$ at demand market $k$ in period $t$,
$Z_i$ If a plant and distribution center are located and set up at potential site $i$, 1, otherwise, 0,
$W_l$ If a collection center is located and set up at potential site $l$, 1, otherwise, 0,
$K_a$ If an inspection center is located and set up at potential site $a$, 1, otherwise, 0,
$R_r$ If a recovery center is located and set up at potential site $r$, 1, otherwise, 0.

$$\begin{aligned}\text{Min } Z_1 = {} & \sum_i E_i Z_i + \sum_l F_l W_l + \sum_a I_a K_a + \sum_r L_r R_r \\ & + \sum_e \sum_s \sum_i \sum_t (SC_{est} + CS_e t_{si}) V_{esit} + \sum_i \sum_k \sum_j \sum_t (A_{jt} + B_j t_{ik}) X_{ijkt} + \sum_k \sum_l \sum_j \sum_t C_j t_{kl} Y_{kljt} + \\ & \sum_j \sum_l \sum_a \sum_t D_j t_{la} E_{jlat} + \sum_j \sum_a \sum_t (H_{jt} + O_j t_a) T_{ajt} + \sum_j \sum_a \sum_r \sum_t (CI_{jt} + CIR_j t_{ar}) U_{jart} + \\ & \sum_j \sum_a \sum_i \sum_t (-G_{jt} + CI_{jt} + CIP_j t_{ai}) S_{jait} + \sum_e \sum_r \sum_s \sum_t (RC_{et} + CPR_e t_{rs}) P_{erst} \\ & + \sum_k \sum_j \sum_t Cb_{kjt} Lack_{kjt}\end{aligned} \tag{1}$$

$$\text{Max } Z_2 = \sum_e \sum_i \sum_j \sum_t M_{ijet} \left( \sum_s V_{esit} + \sum_k X_{ijkt} + \sum_r \sum_s P_{erst} \right) + \sum_a \sum_j \sum_t N_{ajt} \left( \sum_l E_{jlat} + \sum_i S_{jait} + \sum_r U_{jart} \right) \tag{2}$$

The mathematical model including the two objectives of cost and eco-friendly use of raw material and clean technology are described as Eqs. 1 and 2, respectively. The first objective function minimizes the total cost. This cost includes fixed costs, operational costs, transportation costs, and shortage costs. The first part of Eq. 1 accounts for the 1 cost of constructing the production/distribution, collection, inspection, and recycling facilities. The second part of Eq. 1 demonstrates the transportation and operational costs for each facility [32]. The third part of Eq. 1 accounts for the shortage costs when the customer demand is not met. The second objective function accounts for using eco-friendly material and clean technology in production, transportation, and disposal phases. The first part of Eq. 2 considers the use of eco-friendly materials such as recyclable materials and the second part accounts for the use of recyclable and renewable energy such as solar power for transportation and disposal [32,37,38].

**Model Constraints**

The constraints placed on the input and output flows to the facility are described by Eqs. 3 to 11. Constraint (3) guarantees the total product inflow to the production/distribution center by all suppliers and inspection centers is equal to the total outflow from that center. Constraint (4) guarantees the demand of all customers in-demand markets is met. Constraint (5) guarantees the total inflow to all collection centers is equal to all returned products from the demand market. Constraint (6) guarantees that the total inflow of each product to the inspection center is equal to its total outflows. Constraint (7) guarantees the outflow from inspection centers to the disposal centers is equal to inflows from inspection centers multiplied by the rate of disposal. Constraint (8) ensures that product outflows from inspection centers to all recycling centers are equal to the inflows from all collection centers multiplied by the recycling rate. Constraint (9) guarantees the outflow from inspection centers to all production/distribution centers is equal to the inflow multiplied by the return rate from the inspection center to the production/distribution center. Constraint (10) guarantees the inspection center's inflow is equal to the total outflow to the recycling centers, production/distribution, and disposal centers. Constraint (11) guarantees the total inflow to each recycling center is equal to the total outflow from these recycling centers.

$$\sum_{e}\sum_{s} V_{esit} + \sum_{j}\sum_{a} S_{jait} = \sum_{j}\sum_{k} X_{ijkt}, \qquad \forall i,t, \tag{3}$$

$$\sum_{i} X_{ijkt} + Lack_{kjt} = D_{kjt} + Lack_{kjt-1}, \qquad \forall k,j,t, \tag{4}$$

$$\sum_{l} Y_{kljt} = D_{kjt}\mathrm{R}_1, \qquad \forall k,j,t, \tag{5}$$

$$\sum_{a} E_{jlat} = \sum_{k} Y_{kljt}, \qquad \forall l,j,t, \tag{6}$$

$$T_{jat} = \alpha_j \sum_{l} E_{jlat}, \qquad \forall a,j,t, \tag{7}$$

$$\sum_{r} U_{jart} = \beta_j \sum_{l} E_{jlat}, \qquad \forall a,j,t, \tag{8}$$

$$\sum_{i} S_{jait} = \mathrm{R}_2 \sum_{l} E_{jlat}, \qquad \forall a,j,t, \tag{9}$$

$$\sum_{l} E_{jlat} = \sum_{i} S_{jait} + \sum_{r} U_{jart} + T_{jat}, \qquad \forall a,j,t, \tag{10}$$

$$\sum_{j}\sum_{a} U_{jart} = \sum_{e}\sum_{s} P_{erst}, \qquad \forall r,t, \tag{11}$$

The capacity constraints are described by Eqs. 12 to 17. Constraint (12) ensures the total outflow from each supplier to all distribution/production centers does not surpass the supplier's capacity. Constraint (13) guarantees that total inflow from inspection centers and suppliers does not surpass the production and distribution centers' capacities. Constraint (14) guarantees the total inflow of each demand market does not surpass demand capacity. Constraint (15) guarantees the total inflow of collection centers from all demand markets does not surpass the collection centers' capacity. Constraint (16) guarantees the total inflow of each inspection center from all collection centers does not surpass the inspection centers capacity, and ultimately the constraint (17) ensures the total inflow to recycling centers by all inspection centers does not surpass the recycling centers capacity. Constraints (18) to (21) ensure that

distribution/production, collection, inspection, and recycling centers are built up with the specific capacity level, respectively. Constraint (22) shows that the decision variable is binary, and constraint (23) indicates the decision variables of the mathematical model are positive.

$$\sum_e \sum_i V_{esit} \le \sum_j CAS_{sjt}, \quad \forall s,t, \tag{12}$$

$$\sum_e \sum_s V_{esit} + \sum_j \sum_a S_{jait} \le Z_i \sum_j CAP_{ijt}, \quad \forall i,t, \tag{13}$$

$$\sum_i \sum_j X_{ijkt} \le \sum_j CAD_{kjt}, \quad \forall k,t, \tag{14}$$

$$\sum_k \sum_j Y_{kljt} \le W_l \sum_j Q_{ljt}, \quad \forall l,t, \tag{15}$$

$$\sum_j \sum_l E_{jlat} \le K_a \sum_j CAI_{ajt}, \quad \forall a,t, \tag{16}$$

$$\sum_j \sum_a U_{jart} \le R_r \sum_j CAR_{rjt}, \quad \forall r,t, \tag{17}$$

$$Z_i \le 1, \quad \forall i, \tag{18}$$

$$W_l \le 1, \quad \forall l, \tag{19}$$

$$K_a \le 1, \quad \forall a, \tag{20}$$

$$R_r \le 1, \quad \forall r, \tag{21}$$

$$Z_i, W_l, K_a, R_r \in \{0,1\}, \tag{22}$$

$$V_{esit}, X_{ijkt}, Y_{kljt}, E_{jlat}, T_{jat}, U_{jart}, S_{jait}, P_{esrt}, Lack_{kjt} \ge 0. \tag{23}$$

## Solution Methods

The proposed mathematical model is solved by exact, heuristic, and meta-heuristic methods. Exact methods guarantee to obtain an optimal solution while other methods just guarantee to obtain a solution in a reasonable time. The three types of solving methods implemented in this study are described in detail in the following sub-sections.

### Exact Methods

To solve the model in small size and ensure the optimum solution, three different exact methods including LP-Metric, modified ε-constraint, and TH are implemented. The LP-metric method is minimizing the relative distance between the preferred solution and the ideal solution $f^*$. The distance between an arbitrary vector and an ideal vector is expressed as LP-norm. So, the optimization problem will turn to Eq. 24. Where $p$ is a numerical value and is usually 1, 2, or □. To simplify the formulation, $p$ is considered as 2 for Euclidean norm and considered □ for Tchebycheff norm [39]. In this solving method, by changing the power in distance formulation and assigning various weights to the objective functions, different points on the Pareto frontier could be found.

In the common ε-constraint method**,** all objective functions, except one, are converted to a constraint and an upper bound is assigned to the other objective functions. This method produces inappropriate Pareto solutions occasionally; therefore, this issue was addressed by introducing the modified version [40]. The mathematical form of the modified ε-constraint

method is shown in Eq. 25. Where $S$ is the feasible region, $f_g(x)$ is the objective function value, $S_g$ is the slack variable of the objective function which is converted to the constraint. The $r_g$ is the objective function range which can be determined by $r_g = f_g^{\max} - f_g^{\min}$. The $W_g$ is the weighted factor of objective functions, in a way that $\sum_{g=1}^{p} W_g = 1$ and $e_g$ is an upper bound for the objective function and can be calculated by optimizing the objective functions individually.

$$Min\left[\sum_{j=1}^{k}\left(\frac{f_j(x)-f_j^*}{f_j^*}\right)^P\right]^{\frac{1}{P}}$$

Subject to: (24)

$$x \in S$$
$$l \le P \le \infty$$

$$Max\left\{f_i(x)+r_i\times\left(\sum_{\substack{g=1\\ g\neq i}}^{p}\frac{w_g}{w_i}\times\frac{S_g}{r_g}\right)\right\}$$

Subject to: (25)

$$f_g(x) = e_g + S_g \qquad \forall g = 1,2,...,p, g \neq i$$
$$x \in S, S_g \in R^+$$

The TH method enables the decision-maker to select a suitable solution based on objective function priority and the satisfaction degree. This method generates unbalanced and balanced solutions based on the decision-maker's preference. The steps of this approach are as follows [41]:

**Step 1.** Determine the positive ideal solution (PIS) and the negative ideal solution (NIS) for the objective functions. For this purpose, each objective function is solved separately as Eqs. 26 and 27:

$$Z_n^{PIS} = \min Z_n \qquad n = 1,2,.... \tag{26}$$
$$Z_n^{NIS} = \max Z_n \qquad n = 1,2,.... \tag{27}$$

**Step 2.** Calculate the membership function for the objective functions based on Eq. 28:

$$\mu_n(\upsilon) = \begin{cases} 1 & if \ \ Z_n \prec Z_n^{PIS} \\ \dfrac{Z_n^{NIS} - Z_n}{Z_n^{NIS} - Z_n^{PIS}} & if \ \ Z_n^{PIS} \le Z_n \le Z_n^{NIS} \qquad n = 1,2,..... \\ 0 & if \ \ Z_n \succ Z_n^{NIS} \end{cases} \tag{28}$$

**Step 3.** Change the multi-objective model to the TH model by using the TH integration function (Eq. 29):

$$\max \lambda(\upsilon) = \gamma\lambda_0 + (1-\gamma)\sum_h \theta_h \mu_h(\upsilon) \tag{29}$$

Subject to:

$$\lambda_0 \le \mu_h(\upsilon) \qquad h = 1,2,....,n$$
$$\upsilon \in F(\upsilon) \qquad \lambda_0 \text{ and } \gamma \in [0,1]$$

Where $\mu_h(v)$ indicates the satisfaction level of objective function *h* and $\lambda_0 = \min_h \{\mu_h(v)\}$ shows the minimum satisfaction of the objectives.

**Step 4.** The single-objective models should be solved based on $\theta_h$ and $\gamma$ values. The algorithm is completed when the decision-maker is satisfied with the solution. Otherwise, the values of $\gamma$ will change and return to step 3.

**Heuristic Method**

Heuristic methods are practical approaches that guarantee short-term and immediate solutions but not optimum values. These methods are generally applied when finding the optimal solution is impractical or infeasible. The Lagrange Relaxation Algorithm as one of the heuristic methods utilizes Lagrange's theorem to simplify the complex mathematical models to achieve an approximate solution at a logical time as demonstrated in Eq. 30. The Lagrange relaxation algorithm is based on relaxing the complicated constraints and adding them to the objective function by using the Lagrangian multiplier. Initially, the complex constraints of the problem are selected and relaxed and then if the relaxed constraint is violated, a penalty will be added to the objective function. The relaxed form of the model is shown as Eq. 31, where $\mu^T$ is the Lagrange coefficient. Finally, the Lagrange function is considered as $L(\mu) = c^t x + \mu^t (Ax - b)$. In most cases, the obtained solution of the relaxed mathematical model is infeasible for the original mathematical model. Its infeasible solution is considered for the original mathematical model as a lower or an upper bound. Since the optimal solution is between these bounds, the algorithm seeks to reduce these bounds and ultimately achieve a solution that is close to the optimal.

Selecting the correct constraint for relaxation is critical as it has a direct impact on its performance. In this paper, constraint (4) of the mathematical model is selected for relaxing. Therefore, the relaxed model of the proposed supply chain is converted as Eqs. 32 and 33. The constraints (3) to (23), except the constraint (4) should be added to the constraint (33). Where $u_1$ is the Lagrange coefficient and is non-negative. The Lagrange relaxation algorithm starts with constant values of this Lagrange coefficient, which should be updated in each iteration of the algorithm. This paper applies the sub-gradient which to solve the Lagrange relaxation problems. Using the sub-gradient approach, the Lagrange coefficient in iteration *C+1* is calculated as Eq. 34, where $\pi_1^c$ is the algorithm step size and is calculated as Eq. 35. The *BUB* and $UB^c$ are the best calculated lower and upper bounds in the iteration of *C* and $v^c$ is a coefficient between 0 and 2.

$$\min\ c^T x$$
Subject to:
$$Ax \le b \qquad (30)$$
$$x \in X$$

$$\min c^T x + \mu^T (Ax - b) \qquad (31)$$

Subject to:

$$x \in X$$

$$Min\ Z_1 + u_1 \times \left[ \sum_i \sum_j \sum_k \sum_t X_{ijkt} + \sum_k \sum_j \sum_t Lack_{kjt} - \sum_k \sum_j \sum_t D_{kjt} - \sum_k \sum_j \sum_t Lack_{kjt-1} \right] \quad (32)$$

Subject to:

$$LB_2 \le Z_2 \le UB_2 \quad (33)$$

$$u_1^{c+1} = \max\left[ 0, \left\{ u_1^c + \pi_1^c . (\sum_i \sum_j \sum_k \sum_t X_{ijkt} + \sum_k \sum_j \sum_t Lack_{kjt} - \sum_k \sum_j \sum_t D_{kjt} - \sum_k \sum_j \sum_t Lack_{kjt-1}) \right\} \right] \quad (34)$$

$$\pi_1^c = \frac{v^c . (UB^c - BUB)}{(\sum_i \sum_j \sum_k \sum_t X_{ijkt} + \sum_k \sum_j \sum_t Lack_{kjt} - \sum_k \sum_j \sum_t D_{kjt} - \sum_k \sum_j \sum_t Lack_{kjt-1})^2} \quad (35)$$

**Meta-Heuristic Methods**

One of the recent evolutions in approximate search methods is the meta-heuristic methods [42]. To evaluate the model performance for a large size problem, three meta-heuristic methods are used including multi-objective particle swarm optimization (MOPSO), Non-dominated sorting genetic algorithm (NSGA), and Strength Pareto evolutionary algorithm (SPEA-II). The MOPSO algorithm as an extension of the PSO method is introduced by Coello and Lechuga [48] for the multi-objective model. The MOPSO algorithm is described in the following steps (Figure SI-1):

**Step 1.** Generate an initial population.
**Step 2.** Separate non-dominated members and keep them to archive
**Step 3.** Determine the feasible region
**Step 4.** Choose a leader based on cell probability which is $prob_j = \frac{e^{\beta n_i}}{\sum_j e^{-\beta n_j}}$, where $n_i$ is the number of *i*th area, $\beta$ is a selection pressure
**Step 5.** Apply the MOPSO operators. Eq. 36 is used to determine the particle velocity. Where $C_1$ and $C_2$ are constant numbers, $pbest_{ij}$ is the best solutions, $gbest_{ij}$ is the best leader, *w* is the inertial constant, $r_1$ and $r_2$ are the random numbers between 0 and 1, $V_{ij}$ is particle velocity and $x_{ij}$ is particle position. The new particle position is determined based on Eq. 37.
**Step 6.** Update the best personal experience of each particle
**Step 7.** Add non-dominated members of the current population to archive
**Step 8.** Remove non-dominated archive members.
**Step 9.** Remove members over archive capacity
**Step 10.** If the stop condition is satisfied, stop, otherwise go back to step 3.

$$V_{ij}(t+1) = wV_{ij} + c_1 r_1 (pbest_{ij}(t) - x_{ij}(t)) + c_2 r_2 (gbest_{ij}(t) - x_{ij}(t)) \quad (36)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad (37)$$

The NSGA is an extension of the genetic algorithm for solving multi-objective problems [43]. This algorithm is useful for solving multi-objective problems, however, it has two drawbacks including high computational complexity and selecting the dominant particles. To address these drawbacks, the modified NSGA-II was formulated by Deb and Meyarivan [44]. The NSGA-II algorithm is described in the following steps (Figure SI-2):

**Step 1.** Generate the initial population $P_0$

**Step 2.** Apply the combination and mutations operators on $P_0$ and generate a new population of $Q_0$.

**Step 3.** If the stop condition is met, stop. Otherwise, go to the next step.

**Step 4.** Set $R_t = P_t \cup Q_t$.

**Step 5.** Use the fast-non-dominated sorting algorithm to identify the dangling fronts of $F_1$, ..., $F_k$

**Step 6.** Calculate the crowding distance for $i$=1, ..., $k$, on the front $F_i$ and create the population $P_{t+1}$ based on following modes:

**Mode I.** If $|P_{t+1}|+|F_i| \le N$, set $|P_{t+1}| = |P_{t+1}| \cup F_i$.

**Mode II.** If $|P_{t+1}|+|F_i| \succ N$, add $N - |P_{t+1}|$ of solution with the lowest crowding distances to $P_{t+1}$.

**Step 7.** Select the parent from $P_{t+1}$ based on crowding distance and apply combination and mutations operator $P_{t+1}$ to generate a population of $Q_{t+1}$ with size $N$.

**Step 8.** Set $t = t+1$ and go to step 3.

The SPEA and SPEA-II algorithm both use an external archive to keep the non-dominated solutions during the searching process. The SPEA algorithm has some shortcomings in calculating the fitness and strength values which prompted the scholars to introduce a new version of SPEA-II [45]. Its steps are described as follows (Figure SI-3):

$N_E$ : The largest archive of non-dominated solutions $E$.

$N_P$ : Population size.

$K$: The parameter of computing density ($K = \sqrt{N_E + N_P}$).

**Step 1.** Generate an initial population $P_0$ and set $E_0 = \varnothing$.

**Step 2.** Calculate the fitness value for each solution $i$ in set of $P_t \cup E_t$ based on $R(i) = \sum_{j \in P_i \,\&\, j \succ i} s(i)$

**Step 2.1.** Calculate the raw fitness by using $R(i) = \sum_{j \in P_i \cup E_i \,\&\, j \succ i} s(i)$, where $s(i)$ shows the strength of obtained solution from $s(i) = \left| \{ j \,|\, j \in P_t \cup E_t \,\&\, i \succ j \} \right|$.

**Step 2.2.** Calculate the solution density by using $D(i) = \frac{1}{\sigma_i^k + 2}$ ,where $\sigma_i^k$ is the distance between the solution $i$ and the nearest neighbor $k$.

**Step 2.3.** Obtain the fitness value by using $F(i) = R(i) + D(i)$.

**Step 3.** Copy all non-dominated solutions from set $P_t \cup E_t$ to set $E_{t+1}$.

**Step 4.** If stop condition is satisfied, stop and report the $E_{t+1}$ solution, otherwise go to next step.

**Step 5.** Use the dual competition methods to select the parents from $E_{t+1}$ set.

**Step 6.** Use the combination and mutation operators to produce $N_P$ (Population size) and copy the offspring to $P_{t+1}$ and go to step 2.

The Multi-objective evolutionary algorithm by decomposition (MOEA/D) algorithm was developed by Zhang and Li [45]. In this algorithm, the multi-objective problem divides into several sub-optimization problems and optimized each of them. In this method, a weighted vector is defined for each sub-problem and then combined with the objective functions. Theoretically, each of the sub-problems depicts a solution from the Pareto front at the end of the search. Its steps are described as follows (Figure SI-4):

**Step 1.** Initialization

**Step 1.1.** Set $E_P = \varnothing$, where $E_P$ is an archive of approximated solutions

**Step 1.2.** Generate $w^1,...,w^n$ uniformly and calculate the Euclidian distance between them. Set $A(i) = \{i_1,...,i_Q\}$ for $i = 1,...,n$ and define $w^{i_1},...,w^{i_Q}$

**Step 1.3.** Create an initial population of solutions $x^1,...,x^n$ and calculate their objective functions.

**Step 1.4.** Set ideal vector *z*.

**Step 2.** Update (main loop): for $i = 1,...,n$:

**Step 2.1.** Create a new solution *y* in conjunction with neighborhood solutions in $\{x^{i_1},...,x^{i_Q}\}$

**Step 2.1.1.** Select two vectors $y_1$ and $y_2$ randomly from neighborhood solutions of $\{x^1,x^2,...,x^i,...,x^Q\}$.

**Step 2.1.2.** Create a new solution vector $y = (y_1,...,y_m)$ by using crossover operator

**Step 2.1.3.** If random number is less than the mutation probability $P_m$, modify the new solution by using mutation operator

**Step 2.2.** Update the ideal vector by setting $z(j) = f_j(y)$ when $z(j) \prec f_j(y)$

**Step 2.3.** Update the current solution *by setting* $x^j = y$ and $F(x^j) = y$, when $g^{te}(y \mid w^j, z) \prec g^{te}(x^j \mid w^j, z)$.

**Step 2.4.** Update $E_P$ set by adding *y* to $E_P$ and removes the dominant vectors

**Step3.** If stop condition is satisfied, stop, otherwise, go to step 2.

## Computational Results

Fifteen different numerical examples are utilized in both small and large sizes to evaluate the exact and heuristic methods in terms of solution value and CPU Time. The meta-heuristic methods are evaluated by means of Number of Pareto Solutions (NPS), Mean Ideal Distance (MID), The Spread of Non-dominance Solutions (SNS), and CPU Time. Similar to the study conducted by Kannan et al. [20] small, medium, and large sizes are determined based on the model index. If the index of the problem is less than 50 or 60, the problem is considered as small size, for the indexes greater than 80 or 90, the problem is considered as a large size problem, and indexes between 60-80 demonstrated the medium size problem.

**Evaluation of Exact Methods**

Initially, 15 numerical examples are presented in small size and solved by LP-metric, TH, and $\varepsilon$-constraint methods (as shown in Table SI-1). The input data for the TH method are as $\gamma = 0.25$, $\theta_1 = 0.55$, and $\theta_2 = 0.45$. To solve the mathematical model in small dimensions, the GAMS software version 24.1.3, CPLEX solver, and a system with specifications CPU=Cori7 6700 HQ and RAM=16 GIG DDR4 are utilized. Each objective function is solved individually to obtain an exclusive solution as shown in Fig. 2.

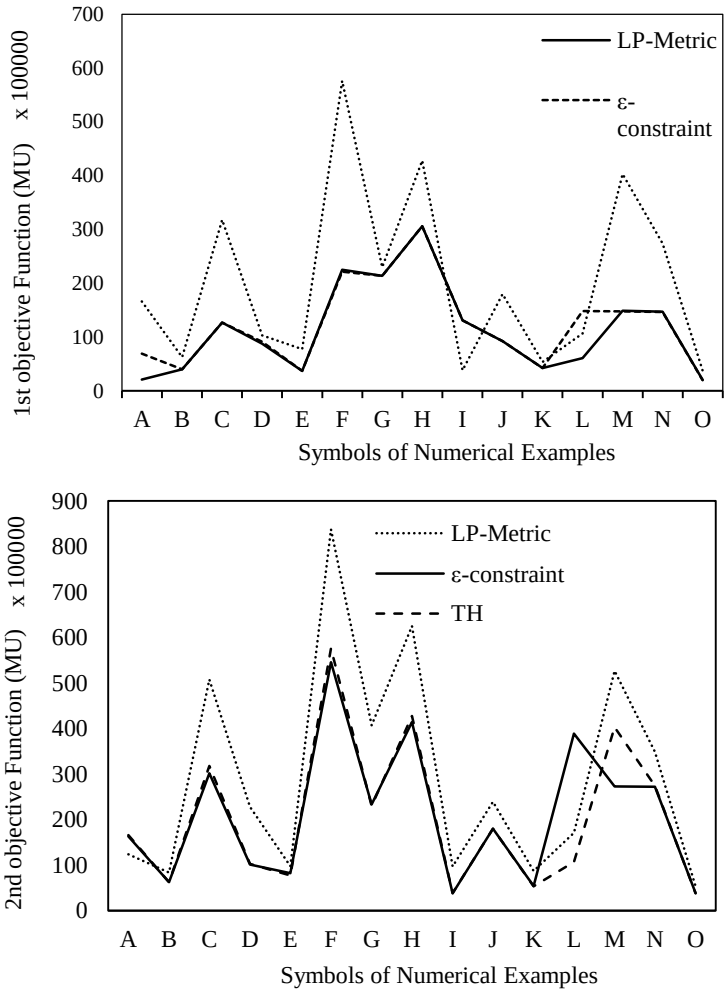


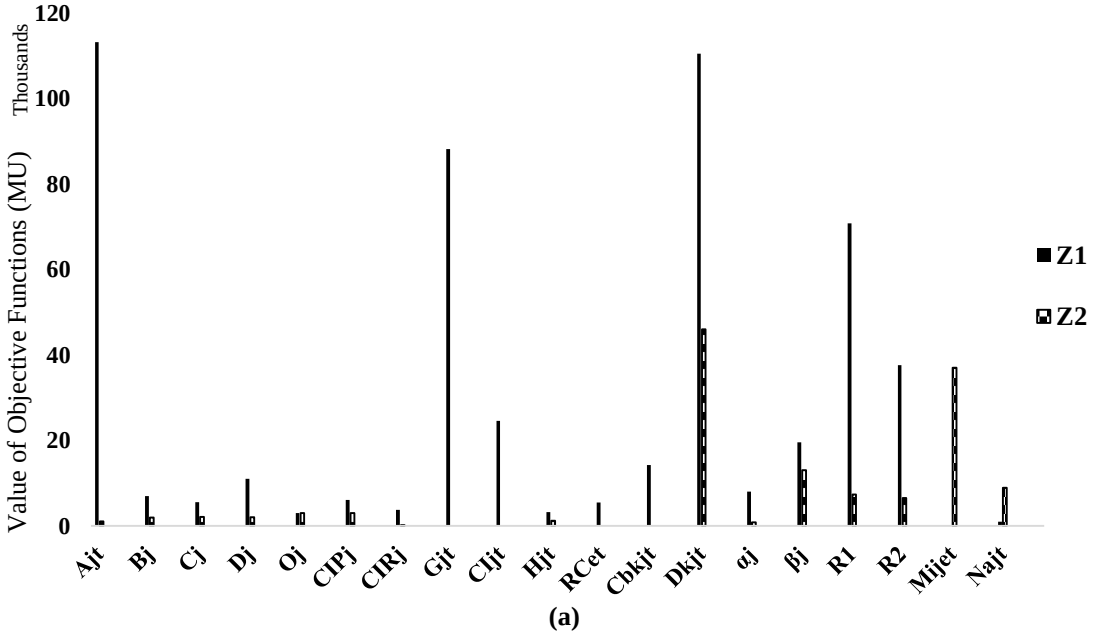

**Fig. 2.** Objective function values in 15 numerical examples

As demonstrated in Fig. 2 the first and second objective functions have a trivial difference in minimizing the total costs and maximizing the eco-friendly usage of raw material. It is also clear that the LP-metric method has better performance rather than two other methods due to its lower values for the first objective function (minimization objective) and higher values for the second objective function (maximization objective). The difference between the worst and best values for the first and second objective functions is calculated for all exact solving methods using Eq. 38 (Table SI-2 to SI-3). The superiority of the LP-metric method compared to the other two exact methods is proved as it has resulted in the least deviation from the best value. The LP-Metric method has the minimum total deviation (1.9) and TH has the maximum total deviation (27.4) among these three methods, so the LP-metric is selected as the best method.

The sensitivity analysis is performed to investigate the impact of fluctuations of the model's parameters on objective functions' values. This analysis is conducted only on the LP-Metric method as it was selected as the best exact method following the least deviation evaluation. For this purpose, three scenarios are defined by 70%, 100%, and 130% variations of nominal parameters' values [55]. As demonstrated in Fig. 3, in the scenario of 70% fluctuation, the production cost (**(**$A_{jt}$ ) saved cost ($G_{jt}$ ), customer demand ($D_{kjt}$ ), and the products return rate ( $R_1$ ) have the highest impacts on the first objective function. However, the environmentally friendly raw materials ( $M_{ijet}$ )and the costumers' demand ( $D_{kjt}$ ) have the highest impacts on the second objective function under the 70% fluctuation scenario. For the second scenario (130%), $A_{jt}$ , $R_1$ and $D_{kjt}$ have the highest impacts on the first objective function and similar to the first scenario (70%), $M_{ijet}$ and $D_{kjt}$ have the highest impacts on the second objective function.

$$S_i = \frac{Z_i - Z^{\min}}{Z^{\max} - Z^{\min}} \quad \text{For Minimization Problems}$$

$$S_i = \frac{Z^{\max} - Z_i}{Z^{\max} - Z^{\min}} \quad \text{For Maximization Problems} \tag{38}$$




**(a)**

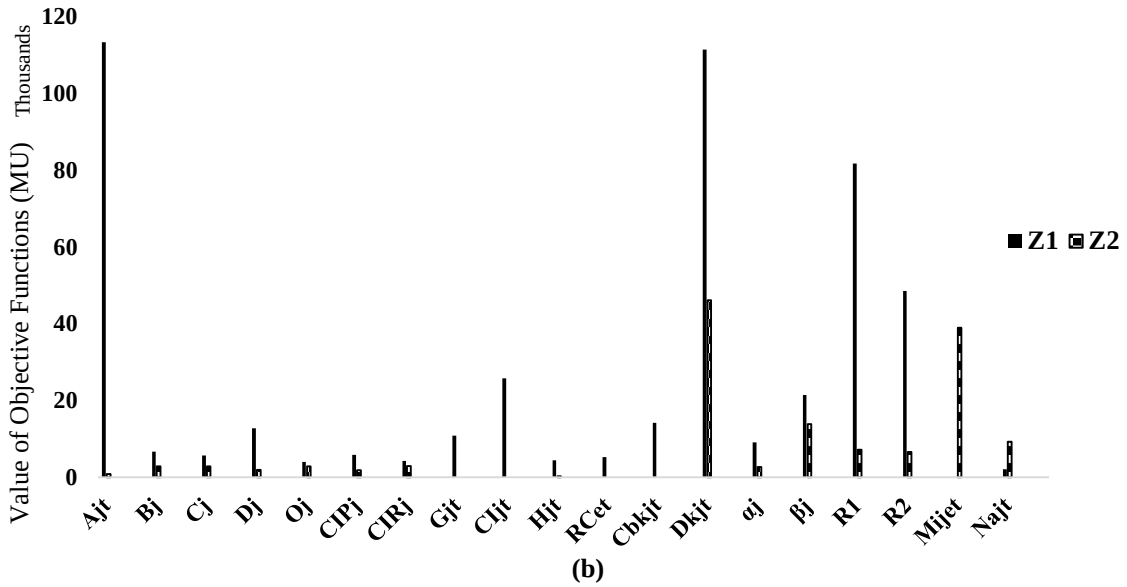



**Fig. 3**. Sensitivity analysis for two objective functions at: a)70% and b)130% variations

## Evaluation of a Heuristic Method

The heuristic algorithms are generally utilized to address the increased time consumption in solving large-size problems. The CPU Time could be reduced by modification of the problem structure using the Lagrange relaxation algorithm. To evaluate this proposed algorithm, the mathematical model is solved in small and large sizes and compared with a reference value which is the exact solution obtained by GAMS software for a small-size example. However, using the GAMS software is very challenging and time-consuming to solve the large-size examples. The input values for the numerical examples are shown in Table SI-4, all of which are based on uniform distribution. To determine the values for the required parameters in the Lagrange relaxation algorithm, the maximum iterations is considered as 20, other parameters are as follows $\nu^0 = 1.5$, $\pi^0 = 2$, $u_1 = 0.2$, and $\varepsilon = 0.1$. If $\nu^0$ does not improve to the lower bound of the algorithm after 5 iterations, it will be reduced to half of its previous value, and this process will continue until the value of $\gamma$ reaches 0.01.

The exact results obtained by GAMS software are compared with the Lagrange relaxation algorithm's results. For this purpose, 10 different numerical examples are defined in small size as shown in Table SI-5. The CPU Time and difference between exact and heuristic solution are considered for the comparison. As shown in Table 1, the mean difference between the exact and heuristic method is 30.5. The largest difference between the exact and heuristic method is 52.4, and the least difference is 24.2. The small difference between the results obtained from exact and heuristic methods (first criteria) indicates the effectiveness of the Lagrange algorithm. Furthermore, the average solving time for the exact method is 2.7 seconds; while for the proposed heuristic method is only 0.1 seconds. Our analysis revealed that the Lagrange algorithm outperforms the exact method in terms of both CPU Time and solution value.

**Table 1.** The results of Lagrange algorithm and exact methods in solving the small numerical examples

| Numerical examples | Value of objective function (MU) | | | CPU Time (s) | |
|---|---|---|---|---|---|
| | GAMS | Lagrange | Percentage of difference | GAMS | Lagrange |
| 1 | 7254531.27 | 5495703.37 | 24.24 | 0.14 | 0.01 |
| 2 | 11615230.00 | 8737460.21 | 24.77 | 0.43 | 0.03 |
| 3 | 13789120.00 | 10213930.00 | 25.92 | 0.81 | 0.04 |
| 4 | 11029330.00 | 8113580.21 | 26.43 | 1.06 | 0.04 |
| 5 | 27072010.00 | 19733300.00 | 27.10 | 2.46 | 0.11 |
| 6 | 23049280.00 | 16652010.00 | 27.75 | 4.90 | 0.17 |
| 7 | 48211330.00 | 35208950.00 | 26.97 | 4.59 | 0.15 |
| 8 | 12238380.00 | 7963636.44 | 34.92 | 3.39 | 0.14 |
| 9 | 23902220.00 | 11380060.00 | 52.38 | 5.40 | 0.23 |
| 10 | 28281110.00 | 18515190.00 | 34.53 | 3.78 | 0.12 |
| **Average** | | | **30.50** | **2.70** | **0.10** |

The performance of the LP-metric method is compared with the Lagrange relaxation algorithm in solving the large size problems. Ten numerical examples are implemented for this comparison as shown in Table SI-6. The difference between the lower and upper bounds of the algorithm is used as a criterion to evaluate the algorithm's performance. As shown in Table 2, the average difference between the upper and lower bounds of the Lagrange algorithm is 2.015. Although the exact method is not able to obtain the optimal solution during a predetermined time of 3000 seconds, the Lagrange relaxation algorithm obtains the optimal value in a reasonable time with an average of 1700.99 seconds.

**Table 2.** Results of Lagrange algorithm and exact algorithm on large numerical examples

| Numerical examples | Value of objective function (MU) | | | CPU Time (s) |
|---|---|---|---|---|
| | Lower bond | Upper bond | Difference% | |
| 1 | 84124521 | 87524522 | 3.885 | 1155.013 |
| 2 | 78125642 | 78625642 | 0.636 | 1279.191 |
| 3 | 95438652 | 95732543 | 0.307 | 1504.924 |
| 4 | 101253254 | 103453254 | 2.127 | 1156.164 |
| 5 | 132575623 | 136552821 | 2.913 | 2004.687 |
| 6 | 105585632 | 106558582 | 0.913 | 1677.318 |
| 7 | 112585756 | 114582565 | 1.743 | 649.720 |
| 8 | 138556252 | 145622623 | 4.853 | 2102.065 |
| 9 | 123055362 | 124478532 | 1.143 | 2545.151 |
| 10 | 127104771 | 129214859 | 1.633 | 2935.664 |
| **Average** | | | **2.015** | **1700.99** |

## Evaluation of Meta-heuristic algorithms in solving the large size problems

The performance of four meta-heuristic algorithms including MOPSO, NSGA-II, SPEA-II, and MOEA/D in solving 15 different large-size numerical examples is evaluated. The evaluation is conducted through the determination of the Number of Pareto Solutions (NPS), Mean Ideal Distance (MID), The Spread of Non-dominance Solutions (SNS), and CPU Time.

(A) Number of Pareto Solutions (NPS): This criterion calculates the number of non-dominated solutions which are obtained each time by applying the algorithm. According to this criterion, the greater number of non-dominated solutions shows that the algorithm works better.

(B) Mean Ideal Distance (MID): The shows the distance between Pareto points and the ideal point for each algorithm. The $f_{i,total}^{max}$ and $f_{i,total}^{min}$ are the highest and lowest values of the objective functions and $n$ is the number of Pareto points. Eq. 39 shows the coordinates of the ideal point ($f_1^{best}$, $f_2^{best}$). The lower value of this index indicates the superiority of the algorithm.

$$c_i = \sqrt{\left(\frac{f_{1i} - f_1^{best}}{f_{1,total}^{max} - f_{1,total}^{min}}\right)^2 + \left(\frac{f_{2i} - f_2^{best}}{f_{2,total}^{max} - f_{2,total}^{min}}\right)^2}$$
$$MID = \frac{\sum_{i=1}^{n} c_i}{n} \quad (39)$$

(C) The Spread of Non-dominance Solutions (SNS): This criterion calculates the dispersion between the set of non-dominated solutions which are obtained by the algorithm, and is calculated by Eq. 40. The dispersion of the solution is higher and more desirable for the greater SNS values.

$$SNS = \sqrt{\frac{\sum_{i=1}^{n} (MID - c_i)^2}{n-1}} \quad (40)$$

(D) CPU Time: The computational time of the algorithm is one of the most crucial indicators in the efficiency of each meta-heuristic algorithm.

*Setting the algorithm parameters using Taguchi method*

Converging to the global optima and avoid of being trapped in local optima is highly influenced by setting the initial model parameters and selecting the operator' types [49]. The best combination of effective parameters for Meta-heuristic algorithms is determined by implementing the Taguchi method [50,51,52,53]. The Taguchi method is applied to identify and minimize the effective parameters for MOPSO, NSGA-II, SPEA-II, and MOEA/D algorithms. Porkar et al. [54] conducted the Taguchi method based on the S/N indicator and determine the parameter's value for NSGA-II and MOPSO methods. The main advantage of the Taguchi method compared to the other experimental methods is the reduction of both cost and processing time (Tortum et al., 2007). In this method, Eq. 41 is used to reduce the perturbation around the especial objective and determine the effective parameters.

$$\frac{S}{N_t} = 10\log\left(\frac{\overline{y^2}}{S^2}\right) \quad (41)$$

In order to determine the level of input parameters (low, medium, and high) for meta-heuristic algorithms, the Taguchi method in MINITAB 19 based on average ratio is implemented and results are summarized in Table 3. After comparing these values, the parameters' optimum values are determined (Table SI-7 and Figure SI-5), and the model is solved.

**Table 3.** Setting the proposed algorithm parameters based on three levels

| Algorithm | Parameters | Low | Medium | High | Ideal |
|---|---|---|---|---|---|
| **MOPSO** | *Max iteration* | 60 | 80 | 100 | **80** |
| | *Population Size* | 50 | 70 | 90 | **90** |
| | *Repository Size* | 80 | 90 | 100 | **100** |
| | *Inertia Weight* | 0.1 | 0.3 | 0.4 | **0.1** |
| | *Number of Grids per Dimension* | 3 | 4 | 5 | **3** |
| | *Inflation Rate* | 0.1 | 0.3 | 0.5 | **0.1** |
| | *Leader Selection Pressure* | 0.5 | 1 | 2 | **0.5** |
| | *Deletion Selection Pressure* | 0.5 | 1 | 2 | **2** |
| | *Inertia Weight Damping Rate* | 0.75 | 0.85 | 0.95 | **0.75** |
| | *Personal Learning Coefficient* | 0.5 | 1 | 2 | **2** |
| | *Global Learning Coefficient* | 0.5 | 1 | 2 | **2** |
| | *Mutation Rate* | 0.1 | 0.3 | 0.5 | **0.1** |
| **NSGA-II** | *Max iteration* | 60 | 80 | 100 | **100** |
| | *Population Size* | 50 | 70 | 90 | **90** |
| | *Crossover Percentage* | 0.5 | 0.7 | 0.9 | **0.5** |
| | *Mutation Percentage* | 0.2 | 0.4 | 0.6 | **0.2** |
| | *Mutation Rate* | 0.01 | 0.02 | 0.03 | **0.01** |
| **SPEA-II** | *Max iteration* | 60 | 80 | 100 | **100** |
| | *Population Size* | 50 | 70 | 90 | **90** |
| | *Archive Size* | 80 | 90 | 100 | **90** |
| | *Crossover Percentage* | 0.5 | 0.7 | 0.9 | **0.9** |
| | *Mutation Percentage* | 0.1 | 0.3 | 0.5 | **0.5** |
| **MOEA/D** | *Max iteration* | 60 | 80 | 100 | **80** |
| | *Population Size* | 50 | 70 | 90 | **90** |
| | *Archive Size* | 80 | 90 | 100 | **80** |
| | *Number of Neighbors* | 10 | 20 | 30 | **10** |
| | *Crossover Percentage* | 0.1 | 0.3 | 0.5 | **0.1** |

The result obtained through the implication of described meta-heuristic algorithms (Table 4) was statistically analyzed in terms of NPS, MID, SNS, and CPU time at a 95% confidence interval. This statistical analysis demonstrated the MOEA/D algorithm as the best method among applied meta-heuristic algorithms in terms of NPS and SNS ($p$-value<0.05). However, the SPEA-II algorithm performed better in terms of MID and CPU Time ($p$-value<0.05). Furthermore, a VIKOR technique is implemented to select the best algorithm. This method was developed for the multi-criteria optimization of complex systems and is used in decision-making with non-consistent criteria [46,47]. Since the results of this ranking depend highly on criteria weight, the VIKOR algorithm is implemented with different weights. The VIKOR analysis demonstrated the SPEA-II as the best algorithm for a majority of numerical examples (Table 5).

**Table 4.** The results of implementing the four algorithms

| Numerical Example | MOPSO | | | | NSGA-II | | | | SPEA-II | | | | MOEA/D | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NPS | MID | SNS | Time(s) | NPS | MID | SNS | Time (s) | NPS | MID | SNS | Time (s) | NPS | MID | SNS | Time (s) |
| 1 | 7 | 0.57 | 0.24 | 140.13 | 17 | 0.64 | 0.91 | 198.32 | 17 | 0.67 | 0.85 | 162.57 | 32 | 0.84 | 1.86 | 356.11 |
| 2 | 5 | 0.81 | 0.33 | 191.35 | 14 | 0.57 | 0.93 | 210.87 | 19 | 0.67 | 1.05 | 163.91 | 32 | 0.92 | 1.90 | 244.52 |
| 3 | 14 | 0.65 | 1.85 | 472.21 | 7 | 0.57 | 0.82 | 181.85 | 11 | 0.73 | 0.76 | 136.53 | 29 | 0.74 | 2.47 | 191.03 |
| 4 | 9 | 0.79 | 0.66 | 80.64 | 16 | 0.65 | 1.06 | 115.35 | 18 | 0.54 | 0.90 | 95.62 | 33 | 0.81 | 1.79 | 142.13 |
| 5 | 8 | 1.10 | 0.28 | 150.75 | 7 | 0.68 | 0.70 | 186.41 | 16 | 0.65 | 0.96 | 136.50 | 27 | 0.89 | 1.09 | 197.22 |
| 6 | 4 | 0.96 | 0.09 | 856.31 | 11 | 0.67 | 0.67 | 508.04 | 7 | 0.65 | 0.69 | 411.84 | 27 | 0.80 | 1.71 | 601.90 |
| 7 | 8 | 0.91 | 0.27 | 876.31 | 14 | 0.67 | 0.68 | 278.22 | 16 | 0.61 | 0.77 | 191.11 | 26 | 0.53 | 2.02 | 296.94 |
| 8 | 7 | 1.0· | 0.01 | 786.31 | 16 | 0.57 | 0.77 | 99.91 | 16 | 0.64 | 0.85 | 112.07 | 23 | 1.18 | 0.82 | 112.93 |
| 9 | 4 | 0.80 | 0.65 | 658.31 | 21 | 0.54 | 0.96 | 111.57 | 17 | 0.65 | 0.82 | 91.27 | 24 | 0.71 | 1.84 | 130.50 |
| 10 | 9 | 0.96 | 0.33 | 440.26 | 14 | 0.70 | 0.69 | 235.91 | 18 | 0.66 | 0.87 | 151.52 | 28 | 0.68 | 1.78 | 358.37 |
| 11 | 9 | 0.76 | 0.62 | 580.12 | 18 | 0.62 | 0.90 | 84.81 | 14 | 0.64 | 0.85 | 69.94 | 29 | 0.92 | 2.03 | 119.55 |
| 12 | 10 | 0.74 | 0.77 | 161.91 | 13 | 0.63 | 0.82 | 122.25 | 15 | 0.68 | 0.76 | 71.81 | 34 | 0.87 | 2.09 | 99.10 |
| 13 | 9 | 0.82 | 0.49 | 152.87 | 13 | 1.06 | 0.30 | 68.87 | 19 | 0.66 | 1.08 | 69.81 | 28 | 0.82 | 2.18 | 62.36 |
| 14 | 5 | 0.80 | 0.49 | 147.90 | 16 | 1.08 | 0.30 | 97.68 | 12 | 0.57 | 0.96 | 66.39 | 27 | 0.75 | 1.10 | 97.26 |
| 15 | 7 | 0.86 | 0.74 | 362.83 | 9 | 0.63 | 0.76 | 192.36 | 13 | 0.52 | 0.99 | 161.60 | 31 | 0.85 | 1.88 | 227.42 |
| **Average** | **7.667** | **0.84** | **0.52** | **478.24** | **13.733** | **0.69** | **0.75** | **179.49** | **15.2** | **0.64** | **0.88** | **139.50** | **28.667** | **0.82** | **1.77** | **215.82** |

**Table 5.** Results of VIKOR algorithm

| Numerical Examples | $W_1$ | $W_2$ | $W_3$ | $W_4$ | Best algorithm |
|---|---|---|---|---|---|
| 1 | 0.40 | 0.20 | 0.20 | 0.20 | SPEA-II |
| 2 | 0.30 | 0.20 | 0.20 | 0.30 | SPEA-II |
| 3 | 0.30 | 0.20 | 0.20 | 0.30 | SPEA-II |
| 4 | 0.25 | 0.25 | 0.25 | 0.25 | MOEA/D |
| 5 | 0.25 | 0.50 | 0.25 | 0.75 | MOEA/D |
| 6 | 0.20 | 0.40 | 0.10 | 0.30 | SPEA-II |

## Conclusion

The growing industry demand for sustainable manufacturing processes while maintaining profitability has prompted the development of closed-loop supply chain (CLSC) networks. In this study, a multi-objective, multi-product, multi-level, and multi-period closed-loop supply chain is developed. This bi-objective model is formulated as a mixed-integer linear programming to minimize the costs and maximize the eco-friendly usage of raw material and clean technology. A comprehensive comparison between exact, heuristic and meta-heuristic methods in solving the small and large-size problems is conducted. The implemented exact methods including the LP-metric, modified ε-constraint, and TH demonstrated an inability in

solving the large size problems, unlike the small size problems. The Lagrange relaxation algorithm as the heuristic method outperformed the exact methods in terms of solution value and CPU time in solving the large-size problems. The meta-heuristic algorithms including MOPSO, NSGA-II, SPEA-II, and MOEA/D well performance in solving the model in a large size problem, demonstrating the best performance through the SPEA-II method. The results indicate the capability of meta-heuristic methods in an efficient solving of large-size CLSC problems. Future suggestions that will consider for this paper include I) Using fuzzy decision methods and robust optimization methods to deal with uncertainty; II) Implementing the proposed model in a real case study and examining its flexibility; III) Considering other environmental criteria as well as the social assessment indicators in making the objective functions; IV) Considering the reliability as well as the quality index of final and return items in the supply chain of this paper; and, V) implementing the Robust methods for overcoming the innate uncertainty in some parameters.

## References

[1] Meade, L., Sarkis, J., and Presley, A. (2007). The theory and practice of reverse logistics. *International Journal of Logistics Systems and Management*, *3*(1), 56-84.

[2] Üster, H., Easwaran, G., Akçali, E., and 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.

[3] Toma, P., Massari, S., and Miglietta, P. P. (2016). Natural resource use efficiency and economic productivity. *Life Cycle Approaches to Sustainable Regional Development; Massari, S., Sonnemann, G., Balkau, F., Eds*, 143-148.

[4] Guide Jr, V. D. R., Teunter, R. H., and Van Wassenhove, L. N. (2003). Matching demand and supply to maximize profits from remanufacturing. *Manufacturing and Service Operations Management*, *5*(4), 303-316.

[5] Zhang, K., and Feng, S. (2014, October). Research on revenue sharing coordination contract in automobile closed-loop supply chain. In *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics* (pp. 298-302). IEEE.

[6] Vahdani, B., Tavakkoli-Moghaddam, R., Modarres, M., and Baboli, A. (2012). Reliable design of a forward/reverse logistics network under uncertainty: a robust-M/M/c queuing model. *Transportation Research Part E: Logistics and Transportation Review*, *48*(6), 1152-1168.

[7] Yun, Y., Chuluunsukh, A., and Gen, M. (2020). Sustainable Closed-Loop Supply Chain Design Problem: A Hybrid Genetic Algorithm Approach. Mathematics, 8(1), 84.

[8] Ghasemzadeh, Z., Sadeghieh, A., and Shishebori, D. (2021). A stochastic multi-objective closed-loop global supply chain concerning waste management: a case study of the tire industry. *Environment, Development and Sustainability*, *23*(4), 5794-5821.

[9] Min, H., Ko, C. S., and Ko, H. J. (2006). The spatial and temporal consolidation of returned products in a closed-loop supply chain network. *Computers and Industrial Engineering*, *51*(2), 309-320.

[10] Wang, H. F., and Hsu, H. W. (2010). A closed-loop logistic model with a spanning-tree based genetic algorithm. *Computers and operations research*, *37*(2), 376-389.

[11] Chen, Y. T., Chan, F. T. S., and Chung, S. H. (2015). An integrated closed-loop supply chain model with location allocation problem and product recycling decisions. *International Journal of Production Research*, *53*(10), 3120-3140.

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

[13] Shishebori, D., Yousefi Babadi, A., and Noormohammadzadeh, Z. (2018). A Lagrangian relaxation approach to fuzzy robust multi-objective facility location network design problem. *Scientia Iranica*, *25*(3), 1750-1767.

[14] Özceylan, E., Demirel, N., Çetinkaya, C., and Demirel, E. (2017). A closed-loop supply chain network design for automotive industry in Turkey. *Computers and industrial engineering*, *113*, 727-745.

[15] Paksoy, T., Bektaş, T., and Özceylan, E. (2011). Operational and environmental performance measures in a multi-product closed-loop supply chain. *Transportation Research Part E: Logistics and Transportation Review*, *47*(4), 532-546.

[16] Shishebori, D., Karimi-Nasab, M., and Snyder, L. V. (2017). A two-phase heuristic algorithm for designing reliable capacitated logistics networks under disruptions. *European Journal of Industrial Engineering*, *11*(4), 425-468.

[17] Devika, K., Jafarian, A., and Nourbakhsh, V. (2014). Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques. *European Journal of Operational Research*, *235*(3), 594-615.

[18] Talaei, M., Moghaddam, B. F., Pishvaee, M. S., Bozorgi-Amiri, A., and 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.

[19] Fahimnia, B., Sarkis, J., and Eshragh, A. (2015). A tradeoff model for green supply chain planning: A leanness-versus-greenness analysis. *Omega*, *54*, 173-190.

[20] Kannan, D., Diabat, A., Alrefaei, M., Govindan, K., and Yong, G. (2012). A carbon footprint based reverse logistics network design model. *Resources, conservation and recycling*, *67*, 75-79.

[21] Ramudhin, A., Chaabane, A., Kharoune, M., and Paquet, M. (2008, December). Carbon market sensitive green supply chain network design. In *2008 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 1093-1097). IEEE.

[22] Garg, K., Kannan, D., Diabat, A., and Jha, P. C. (2015). A multi-criteria optimization approach to manage environmental issues in closed loop supply chain network design. *Journal of Cleaner Production*, *100*, 297-314.

[23] Abdolazimi, O., Esfandarani, M. S., Salehi, M., and Shishebori, D. (2020a). Robust design of a multi-objective closed-loop supply chain by integrating on-time delivery, cost, and environmental aspects, case study of a Tire Factory. *Journal of Cleaner Production*, 121566.

[24] Liao, Y., Kaviyani-Charati, M., Hajiaghaei-Keshteli, M., and Diabat, A. (2020). Designing a closed-loop supply chain network for citrus fruits crates considering environmental and economic issues. *Journal of Manufacturing Systems*, *55*, 199-220.

[25] Papen, P., and Amin, S. H. (2019). Network configuration of a bottled water closed-loop supply chain with green supplier selection. *Journal of Remanufacturing*, *9*(2), 109-127.

[26] Validi, S., Bhattacharya, A., and Byrne, P. J. (2015). A solution method for a two-layer sustainable supply chain distribution model. *Computers and Operations Research*, *54*, 204-217.

[27] Niu, B., Tan, L., Liu, J., Liu, J., Yi, W., and Wang, H. (2019). Cooperative bacterial foraging optimization method for multi-objective multi-echelon supply chain optimization problem. *Swarm and Evolutionary Computation*, *49*, 87-101.

[28] Khalifehzadeh, S., Seifbarghy, M., and Naderi, B. (2015). A four-echelon supply chain network design with shortage: Mathematical modeling and solution methods. *Journal of Manufacturing Systems*, *35*, 164-175.

[29] Bottani, E., Murino, T., Schiavo, M., and Akkerman, R. (2019). Resilient food supply chain design: Modelling framework and metaheuristic solution approach. *Computers and Industrial Engineering*, *135*, 177-198.

[30] Amin, S. H., and Zhang, G. (2012). An integrated model for closed-loop supply chain configuration and supplier selection: Multi-objective approach. *Expert Systems with Applications*, *39*(8), 6782-6791.

[31] Amin, S. H., & Zhang, G. (2013). A multi-objective facility location model for closed-loop supply chain network under uncertain demand and return. *Applied Mathematical Modelling*, *37*(6), 4165-4176.

[32] Alshamsi, A., & Diabat, A. (2018). Large-scale reverse supply chain network design: An accelerated Benders decomposition algorithm. *Computers & Industrial Engineering*, *124*, 545-559.

[33] Santibanez-Gonzalez, E. D., & Diabat, A. (2013). Solving a reverse supply chain design problem by improved Benders decomposition schemes. *Computers & Industrial Engineering*, *66*(4), 889-898.

[34] Alshamsi, A., & Diabat, A. (2017). A Genetic Algorithm for Reverse Logistics network design: A case study from the GCC. *Journal of Cleaner Production*, *151*, 652-669.

[35] Diabat, A., Battaïa, O., & Nazzal, D. (2015). An improved Lagrangian relaxation-based heuristic for a joint location-inventory problem. *Computers & Operations Research*, *61*, 170-178.

[36] Ruan, J., & Xu, Z. (2011). Environmental friendly automated line for recovering the cabinet of waste refrigerator. *Waste management*, *31*(11), 2319-2326.

[37] Kemp, R., & Volpi, M. (2008). The diffusion of clean technologies: a review with suggestions for future diffusion analysis. *Journal of Cleaner Production*, *16*(1), S14-S21.

[38] Taylor, D. H. (Ed.). (1997). *Global cases in logistics and supply chain management*. Cengage Learning EMEA.

[39] Esmaili, M., Amjady, N., & Shayanfar, H. A. (2011). Multi-objective congestion management by modified augmented ε-constraint method. *Applied Energy*, *88*(3), 755-766.

[40] Torabi, S. A., & Hassini, E. (2008). An interactive possibilistic programming approach for multiple objective supply chain master planning. *Fuzzy sets and systems*, *159*(2), 193-214.

[41] Cuevas, E., Gálvez, J., & Avalos, O. (2020). Introduction to Optimization and Metaheuristic Methods. In *Recent Metaheuristics Algorithms for Parameter Identification* (pp. 1-8). Springer, Cham.

[42] Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, *2*(3), 221-248.

[43] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182-197.

[44] Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, *11*(6), 712-731.

[45] Abdolazimi, O., Esfandarani, M. S., & Shishebori, D. (2020b). Design of a supply chain network for determining the optimal number of items at the inventory groups based on ABC analysis: a comparison of exact and meta-heuristic methods. *Neural Computing and Applications*, 1-16.

[46] Opricovic, S., & Tzeng, G. H. (2007). Extended VIKOR method in comparison with outranking methods. *European journal of operational research*, *178*(2), 514-529.

[47] Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, *8*(3), 256-279.

[48] Ortiz-Rodriguez, J.M., Martinez-Blanco, M.R., VegaCarrillo, H.R., "ARTIFICIAL NEURALNETWORKS APPLICATION", Chapter 24, Intech, 2011.

[49] Peterson, G. E., Clair, D. S., Aylward, S. R., & Bond, W. E. (1995). Using Taguchi's method of experimental design to control errors in layered perceptrons. *IEEE transactions on neural networks*, *6*(4), 949-961.

[50] Yang, S. M., & Lee, G. S. (1999). Neural network design by using Taguchi method.

[51] Sukthomya, W., & Tannock, J. (2005). The optimisation of neural network parameters using Taguchi's design of experiments approach: an application in manufacturing process modelling. *Neural Computing & Applications*, *14*(4), 337-344.

[52] Tortum, A., Yayla, N., Çelik, C., & Gökdağ, M. (2007). The investigation of model selection criteria in artificial neural networks by the Taguchi method. *Physica A: Statistical Mechanics and its Applications*, *386*(1), 446-468.
[53] Porkar, S., Mahdavi, I., Vishkaei, B. M., & Hematian, M. (2018). Green supply chain flow analysis with multi-attribute demand in a multi-period product development environment. *Operational Research*, 1-31.
[54] Shakhsi-Niaei, M., & Esfandarani, M. S. (2019). Multi-objective deterministic and robust models for selecting optimal pipe materials in water distribution system planning under cost, health, and environmental perspectives. *Journal of cleaner production*, *207*, 951-960.

*Supplementary material is available at the journal's website (jieng.ut.ac.ir)