Solving a New Bi-Objective Multi-Echelon Supply Chain Problem with a Jackson Open-Network Issue Under Uncertainty

Sepideh Azadbakhsh

Kharazmi University

Ali Ghodratnama (✉ aghodratn@gmail.com)

Kharazmi University <https://orcid.org/0000-0003-0091-6337>

Reza Tavakkoli-Moghaddam

University of Tehran

Research Article

Keywords: Closed-loop supply chain network, Jackson network, Queueing system, Bi-objective probabilistic fuzzy model, Meta-heuristic algorithms.

Posted Date: June 8th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-532989/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Version of Record: A version of this preprint was published at Soft Computing on October 11th, 2021. See the published version at <https://doi.org/10.1007/s00500-021-06309-9>.

Solving a new bi-objective multi-echelon supply chain problem with a Jackson open-network issue under uncertainty

Sepideh Azadbakhsh^a, Ali Ghodratnama^{a,*}, Reza Tavakkoli-Moghaddam^b

^a *Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran*

^b *School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran*

Abstract

In today's world, changes in the economic and industrial sectors are taking place more quickly than in the past. Based on today's competitive market, considering the needs and desires of the customer is very important. Thus, designing the supply chain network to minimize the costs and improve the system efficiency, the service facility, and attention to the timely service delivery to the customer should be considered. Also, deployment costs to determine the number and position of facilities are regarded as another strategic issue. Therefore, to become closer to the real-world application, by considering the Jackson network model (queueing theory), the present study aims to design a closed-loop supply chain (CLSC) network to reduce the waiting time in the queue. Two objectives, namely minimizing the cost and waiting time incurred in queues formed in producers and distributors, are investigated. Queueing mathematical expressions are used both in dealing with objectives and constraints. To develop a model, which is close to real by considering uncertainty, the parameters, including demand rate and shipping cost of products, were considered as a trapezoidal fuzzy number. Also, we crisp this fuzzy uncertainty using both Jimenez and chance-constrained programming approaches. Due to the suggested model's complexity and nonlinearity, we linearized the mathematical model as much as possible, then two meta-heuristic algorithms, namely multi-objective particle swarm optimization (MOPSO) and non-dominated sorted genetic algorithm (NSGA-II), are used to solve the model. First, the structural parameters of these algorithms are tuned by the Taguchi method. Then, numerous numerical test problems ranging from small to large scales are used to compare these algorithms considering Pareto front measuring indexes and various hypothesis tests.

Keywords: Closed-loop supply chain network; Jackson network; Queueing system; Bi-objective probabilistic fuzzy model; Meta-heuristic algorithms.

1 Introduction

A supply chain is a collection of organizations linked to each other through materials, information, and financial flows. Depending on the decision-making variables within the time horizon, the decisions taken in the supply chain can be placed in one of three levels of decision-making, including strategic, tactical, and operational. Supply chain networks are divided into different categories depending on the organization and environment. The forward supply chain includes the flow of products and services from the supplier to the final consumer. The network consists of facilities, such as manufacturing, distribution, and retailing centers.

On the other hand, in the reverse supply chain, consumed products flow from the final consumer to manufacturing centers. The network consists of facilities, such as a collection center,

* Corresponding author. Telefax: +98 21 88830891.

E-mail address: ghodratnama@khu.ac.ir (A. Ghodratnama).

reproducer, producer, and disposal center. In this paper, distribution centers are also considered a collection center for defective products.

Loosely speaking, the integration of the forward and reverse supply chains leads to the closed-loop supply chain (CLSC) (Souza 2013; Deng et al. 2020; Yang et al. 2020; Zhang et al. 2020).

The aim of designing a supply chain network is to determine the number and location of facilities, materials flow, and the like to reduce the costs and increase network efficiency. Since the market is moving toward competitiveness, attention to the need of the customer is significant. The customer needs include quality, price, timely delivery of the product or service, etc. The queueing theory was integrated into the supply chain topics to meet customer needs. The queueing system can be divided into two discrete and continuous groups. A discrete system is a system in which state variables change only in a set of different points of time. The bank is an example of a discrete system because the state variable (the number of customers present in a bank) changes only when a customer enters or services to a customer.

A continuous system is a system, whose state variables change continuously over time (e.g., water leakage behind the dam). The features of the queueing process include the entering process rate of customers into the system, customer behavior, service time, and service sequences. The quantification criteria of queueing performance include waiting time in line or system, the number of clients in the queue or the system, and free time of servers.

Intuitively, supply chain management (SCM) is based on a customer-centric approach. Therefore, the timely and complete communication between all the chain elements to understand the customer needs and level of meeting needs are the necessities of the chain. The provision of reliable and fast service to customers requires proper planning for deploying the facility and its management. Considering today's competitive market, paying attention to customer demand is of great importance, and addressing it will increase market share. In this research, a novel mathematical model with the objectives includes minimizing the chain's costs and minimizing the average waiting time in product delivery to the customer.

To achieve the second objective, we have examined the queue problem in two positions:

- A queue formed in the production line in the producer, and the Jackson queueing theory (the main novelty) is used to constrain the queue length in producer facilities
- A queue formed in service lines in distribution centers and reducing waiting time in distribution centers has been used as a second objective in the mathematical model.
- The probability of queueing in distribution centers as a factor in calculating the waiting time of the product in distribution centers to deliver the product or service to the customer is important.
- Obtaining efficient solutions regarding the second objective function of the problem, Therefore, this factor plays a decisive role. Also, numerous decision variables are considered in this research.

Constraints of the queue at producing plants with a difference (i.e., the number of production lines as a variable) are considered. Besides, the present study aims to determine the number of optimal lines in producing centers using the Jackson network, regarding the constraint of the queue length and balancing the costs. This study also aims to develop the previous models and considers the real-world constraints in more detail. Due to the NP-hardness of these kinds of problems (i.e., supply chain mathematical models) to solve this developed mathematical model, two meta-heuristic algorithms, namely Multi-Objective Particle Swarm Optimization (MOPSO) and Non-Dominated Sorting Genetic Algorithm (NSGA-II), are employed. Other purposes of this research are summarized below:

- Determining the acceptable number of potential facilities and their location.

- Determining the adequate flow rate between the facilities.
- Providing an appropriate encoding to solve the model.
- Comparison of these meta-heuristic algorithms using comparative metrics.

In brief, this paper is organized as follows. Section 2 presents the motives for doing this research. Section 3 reviews the relevant literature. Section 4 illustrates the uncertain model of the problem with its assumptions, variables, and parameters. In Section 5, the solution method is developed. We solve numerical examples, including an extensive sensitivity analysis in Section 6. Finally, managerial insights and conclusions are drawn in sections 7 and 8, respectively.

2 Motives

In this paper, we consider a reverse supply chain problem, in which the cycle of flow is taken into account in the way that defective parts are first returned from the customer to the distributor, which is the collection center. The collection center transfers the products to the recycling center. Some of the goods are lost and discarded in the recycling center, and some of them are sent to the producer as semi-finished products and continue to the distributor, and the cycle continues. Also, one aspect of the queuing theory (i.e., Jackson) deals with the bi-objective supply chain problem. One objective function aims to minimize the total cost involving the establishment cost of facilities. The other one aims to minimize the total waiting time of materials or goods to be served in the distributor. However, we use mathematical, probabilistic expressions related to the queuing theory defined by Marianov and Serra (2013).

Due to the nonlinearity inheritance of the mathematical model, we linearize it as much as possible. To be close to reality, we consider uncertainty (i.e., fuzzy logic) related to some parameters (e.g., demands and transportation cost). We also crisp this uncertainty using a chance-constrained programming approach (Inuiguchi and Ramik, 2000). We use two multi-objective meta-heuristic algorithms because of the mathematical model (i.e., existing binary or integer variable and nonlinearity expression).

To generate a feasible solution, we exploit a method defined by Gen et al. (2008). Due to the existing two objectives, we use two multi-objective meta-heuristic algorithms, namely MOPSO and NSGA-II. To investigate the formed Pareto front by each algorithm, five indices are investigated. Numerous test problems ranging from small to large sizes with 10 replicates of each problem considering the defined lower and upper bounds for each parameter are solved. Numerous figures related to each index involving trends and boxplots are drawn to show the impact of each algorithm's size of test problems and habit. Finally, to take a decisive decision of superiority of each algorithm over the other, we use one of the multi-attribute decision-making methods, namely Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). Results show that the NSGA-II is more appropriate as a whole.

3 Literature review

In recent years, several papers have been published due to the growing importance of designing closed-loop and reverse supply chain networks. For example, Pishvae et al. (2011) developed a robust optimization approach to the closed-loop supply chain network design (SCND) under uncertainty.

Mahmoodi (2019) investigated a new multi-objective model of agile supply chain network design considering transportation limits. Mohebalizadehgashti (2019) addressed designing a green meat supply chain network. Resat and Unsal (2019) designed a novel multi-objective optimization approach for a sustainable supply chain. Singh and Goh (2019) devised a multi-objective mixed-

integer programming model and investigated an application in a pharmaceutical supply chain. Vafaeenezhad et al. (2019) considered a multi-objective mathematical model for sustainable SCM in the paper industry.

However, some researchers have devised and solved the supply chain mathematical model by efficient algorithms; for example, Robles et al. (2016) optimized a hydrogen supply chain network design by multi-objective genetic algorithms. Kayvanfar et al. (2017) investigated a bi-objective intelligent water drops algorithm to a practical multi-echelon supply chain optimization problem. Nobil et al. (2017) considered a multi-objective evolutionary approach in location and allocation decisions for a multi-echelon supply chain network.

Cheraghalipour et al. (2018) considered a bi-objective optimization model for the citrus CLSC using Pareto-based algorithms. Kusolpuchong et al. (2019) studied a genetic algorithm for multi-objective cross-dock scheduling in supply chains. Mardan (2019) planned an accelerated Bender's decomposition algorithm for a bi-objective green CLSC network design problem. Niu et al. (2019) devised a cooperative bacterial foraging optimization method for a multi-objective multi-echelon supply chain optimization problem.

Roghanian & Cheraghalipour (2019) addressed a set of meta-heuristics to solve a multi-objective model for a citrus CLSC considering CO₂ emissions. Mohammed and Duffuaa (2020) investigated a tabu search-based algorithm for the optimal design of multi-objective multi-product supply chain networks.

Some researchers investigated congestion related to the supply chain network. For example, Teimoury et al. (2012) used a queuing approach for making decisions about order penetration points in multi-echelon supply chains. Rioux et al. (2016) investigated the economic impacts of debottlenecking congestion in the Chinese coal supply chain. Marufuzzaman & Ekşioğlu (2017) managed congestion in supply chains via dynamic freight routing, including an application in the biomass supply chain. Fathian (2018) considered location and transportation planning in supply chains under uncertainty and congestion by using an improved electromagnetism-like algorithm. Hu et al. (2018) investigated a joint design of fleet size, hub locations, and hub capacities for third-party logistics networks with road congestion constraints. Siswanto (2018) took into account a simulation study of sea transport-based fertilizer products considering disruptive supply and congestion problems. Darestani and Hemmati (2019) optimized a robust bi-objective CLSC network for perishable goods considering a queue system. Poudel et al. (2019) managed congestion in a multi-modal transportation network under biomass supply uncertainty.

Jain and Raghavan (2019) used a queuing approach for inventory planning with batch ordering in multi-echelon supply chains. Nazari-Ghanbarloo and Ghodratinama (2019) optimized a robust tri-objective multi-period reliable supply chain network considering the queuing system and operational and disruption risks. Liu et al. (2020) coordinated a location-inventory problem with supply disruptions and optimized it using a two-phase queuing theory-optimization model approach. Mohtashami et al. (2020) designed a green CLSC model using a queuing system for reducing environmental impact and energy consumption. Fathi et al. (2021) integrated queuing-stochastic optimization with the genetic algorithm for a location-inventory supply chain network. Wu & Yang (2021) optimized a bi-objective queueing model with a two-phase heterogeneous service.

Salkuti & Kim (2019) managed congestion using a multi-objective glowworm swarm optimization algorithm. Li et al. (2019) took into account factors affecting bike-sharing behavior in Beijing, involving price, traffic congestion, and supply chain. Liu et al. (2020) addressed a time-dependent vehicle routing problem with time windows of city logistics with a congestion avoidance approach. Mohtashami et al. (2020) considered the green closed-loop SCND using a queuing system for reducing environmental impact and energy consumption.

Uncertainty is also investigated by many researchers recently. For example, Gholamian (2016) considered multi-objective multi-product multi-site aggregate production planning in a supply chain under a fuzzy environment. Dai (2016) designed a multi-objective fuzzy model of the closed-loop SCND considering risks and environmental impact. Kumar et al. (2017) designed multi-period SCND considering risk and emission. Yu et al. (2018) investigated multi-objective models and a real-case study for dual-channel SCND with fuzzy information.

Gholami et al. (2019) took into account the multi-objective robust SCND considering reliability. However, designing a multi-objective model under uncertainty was also investigated by many researchers. Amin and Zhang (2013) considered a multi-objective facility location model for the CLSC network under uncertain demand and return. Fakhrazad and Goodarzian (2019) accounted for a fuzzy multi-objective programming approach to develop a green closed-loop SCND problem under uncertainty and used an imperialist competitive algorithm. Goodarzian et al. (2020) developed a multi-objective pharmaceutical supply chain network based on a robust fuzzy model and compared the meta-heuristic algorithm used to solve it. Goodarzain and Hosseini-Nasab (2021) applied a fuzzy multi-objective model for a production-distribution problem by using a novel self-adaptive evolutionary algorithm.

After scrutinizing the research papers, specifically in recent years, to the best of our knowledge, there exists no research paper handling a supply chain issue regarding the Jackson queuing type and other features (e.g., uncertainty) and solving by multi-objective algorithms.

4 Mathematical model

In this section, the suggested model is described to design the multi-product closed-loop SCND problem with the constraint of the queuing length at the production center. This model includes raw materials supply centers, production centers, and distribution centers. Besides, customers in the backward supply chain network include centers for collecting return products, reproduction centers, and waste disposal centers in the reversed supply chain (i.e., downstream). For this purpose, the used assumptions, indices, parameters, and variables are first described before presenting the model. Fig. 1 shows the proposed model of the closed-loop SCND.

4.1 Assumptions

The assumptions of multi-product closed-loop SCND problem are as follows:

- The proposed model includes several different products.
- Customer demand is possible from any product, and the shortage is not allowed.
- The locations of the potential facility are predefined and distinct.
- The capacity of all potential facilities is limited and specific.
- Demand parameters, transportation costs are uncertain and are considered as trapezoidal fuzzy.
- Produced products at the production center from a queue with an unknown number of service providers.
- Products in the distribution centers are sent to the customer in a distribution line with an unknown number of service providers.
- The number of service providers in the queue of the production and distribution center is unspecified.
- Service rates in distribution and production centers are known.
 - According to the above-mentioned assumptions, the most crucial aim of this research is to choose the location of supplier centers, production centers, distribution centers

of products, reproduction centers, and disposal centers, as well as the determination of optimal flow rate among centers and the number of service providers in production and distribution centers.

Fig. 1 Proposed model of the closed-loop SCND

4.2 Indices

This section introduces indices used in our proposed mathematical model.

$i \in \{1, 2, \dots, IT\}$	Supplier
$j \in \{1, 2, \dots, JT\}$	Producer
j'	A server (channel) in the producer
$k \in \{1, 2, \dots, KT\}$	Distributor
k'	A server (channel) in the distributor
$l \in \{1, 2, \dots, LT\}$	Customers
$c \in \{1, 2, \dots, CT\}$	Reproducer
$d \in \{1, 2, \dots, DT\}$	Disposal center
$p \in \{1, 2, \dots, PT\}$	Semi-finished or finished products

4.3 Parameters

This section introduces the parameters used in our proposed mathematical model.

FI_i	Establishment cost of supplier i
FJ_j	Establishment cost of producer j
FC_c	Establishment cost of reproducer c
FK_k	Establishment cost of a distributor k

FD_d	Establishment cost of disposal center d
$F\mathcal{K}_k$	Establishment cost of a server at the distributor k
$F\mathcal{J}_j$	Establishment cost of a server at the producer j
$\tilde{T}_{IJp_{i,j,p}}$	Transportation cost of each raw or semi-finished product p from supplier i to producer j
$\tilde{T}_{JKP_{j,k,p}}$	Transportation cost of each product p from producer j to distributor k
$\tilde{T}_{KLP_{k,l,p}}$	Transportation cost of each product p from collection center k (distributor k) to remanufacture c
$\tilde{T}_{LKP_{l,k,p}}$	Transportation cost of each product p from customer l to collection center k (distributor k)
$\tilde{T}_{KCP_{k,c,p}}$	Transportation cost of each product p from collection center (distributor) k to reproducer c
$\tilde{T}_{CJP_{c,j,p}}$	Transportation cost of each product p from reproducer c to the manufacture j
$\tilde{T}_{CDP_{c,d,p}}$	Transportation cost of each product p from reproducer c to disposal center d
$\tilde{D}_{El,p}$	Demand amount of customer l for product p
$\beta_{CP_{c,p}}$	Percentage of defective returned product p from reproducer c
$CKP_{k,p}$	Distribution capacity of product p related to the distributor k
$CKP'_{k,p}$	Collecting capacity of product p in the collection center k (distributor k)
$CDP_{d,p}$	Disposal capacity of product p in disposal center d
$CCP_{c,p}$	Reproducing capacity of product p in the reproducer c
$CJP_{j,p}$	Producing capacity of product p related to the producer j
$CIP_{i,p}$	Supplying capacity of raw or semi-finished product p related to supplier i
T	Number of distributors (collection centers) that must be established
FC_c	Establishment cost of reproducer c
FJ_j	Establishment cost of producer j
FI_i	Establishment cost of supplier i
FD_d	Establishment cost of disposal center d
$F\mathcal{K}_k$	Establishment cost of a server at the distributor k
$F\mathcal{J}_j$	Establishment cost of a server at the producer j
$\tilde{D}_{El,p}$	The demand for customer l for product p
$CKP_{k,p}$	Distribution capacity of distributor k for product p
$C'KP_{k,p}$	Collecting capacity of distributor k (collection center k) for product p
$CCP_{c,p}$	Reproducing capacity of reproducer c for product p
$CJP_{j,p}$	Producing capacity of producer j for product p
$CIP_{i,p}$	Supplying capacity of supplier i for raw or semi-finished product p
$CDP_{d,p}$	Disposal capacity of disposal center d for product p
$\tilde{T}_{LKP_{l,k,p}}$	Transportation cost of each product from customer l to collection center k (distributor k)
$\tilde{T}_{KLP_{k,l,p}}$	Transportation cost of each product p from distributor k to customer l
$\tilde{T}_{KCP_{k,c,p}}$	Transportation cost of each product p from reproducer c to producer j

$\tilde{T}_{JKP}_{j,k,p}$	Transportation cost of each product p from producer j to distributor k
$\tilde{T}_{CDP}_{c,d,p}$	Transportation cost of each product p from reproducer c to disposal center d
$\alpha_{l,p}$	Percentage of returned products p from customer l
$\beta dp_{d,p}$	Percentage of defective returned products p from reproducer d
μK_k	Service rate of each server at distributor k (Exponential distribution)
μJ_j	Service rate of each server at producer j (Exponential distribution)
$\beta jp_{j,p}$	The upper limit of queue length for product p in producer j
$BQ_{j,p}$	The upper limit probability of excessive queue length of product p in producer j
M	A big value number

4.4 Decisions variables

This section introduces the decision variables used in our developed mathematical model.

4.4.1 Binary variables

y_i	1 if a supplier is established in candidate location i ; 0, otherwise
y_j	1 if the producer is established in candidate location j ; 0, otherwise
y_c	1 if reproducer is established in candidate location c ; 0, otherwise
y_k	1 if distributor is established in candidate location k ; 0, otherwise
y_d	1 if disposal center is established in candidate location d ; 0, otherwise

4.4.2 Positive variables

g_k	Number of the service provider in distributor k
g_j	Number of the service provider in producer j
$xijp_{i,j,p}$	Transferred amount of semi-finished product p between supplier i and producer j
$xjkp_{j,k,p}$	Transferred amount of finished product p between producer j and distribution center k
$xklp_{k,l,p}$	Transferred amount of finished product p between distribution center k and customer l
$xlkp_{l,k,p}$	Transferred amount of finished product p between customer l and collection center k
$xkcp_{k,c,p}$	Transferred amount of finished product p between distributor k (collection center k) and reproducer c
$xcjp_{c,j,p}$	Transferred amount of finished product p between reproducer c and producer j
$xcdp_{c,d,p}$	Transferred amount of finished product p between reproducer c and disposal center d
$\omega kp_{k,p}$	Amount of product p transferred from distributor k
$\omega jp_{j,p}$	Amount of product p transferred from producer j

$\pi_{0,k,p}$	Probability of forming queue back to distributor k for product p
$pr_{j',j,p}$	Probability of forming queue back to producer j server j' for product p
$w_{k,p}$	Waiting time of product p in distributor k

4.5 Objective functions

Two separate objective functions are taken into consideration in our proposed mathematical model as follows:

3.5.1 First objective function: minimizing the total cost

The first objective function is formulated as follows:

$$\begin{aligned}
z_1 = \min(& \sum_{i=1}^{IT} FI_i \cdot yi_i + \sum_{i=1}^{IT} FJ_j \cdot yj_j + \sum_{i=1}^{IT} FC_c \cdot yc_c + \sum_{i=1}^{IT} FK_k \cdot yk_k + \sum_{i=1}^{IT} FI_i \cdot yi_i + \\
& + \sum_{i=1}^{IT} FD_d \cdot yd_d + \sum_{k=1}^{KT} FgJ_k \cdot gk_k + \sum_{j=1}^{JT} FgJ_j \cdot gj_j + \sum_{i=1}^{IT} \sum_{j=1}^{JT} \sum_{p=1}^{PT} \hat{T}_{IJP} \cdot xijp_{i,j,p} + \\
& + \sum_{j=1}^{JT} \sum_{k=1}^{KT} \sum_{p=1}^{PT} \hat{T}_{JKP} \cdot xjkp_{j,k,p} + \sum_{k=1}^{KT} \sum_{l=1}^{LT} \sum_{p=1}^{PT} \hat{T}_{KLP} \cdot xklp_{k,l,p} + \\
& + \sum_{l=1}^{LT} \sum_{k=1}^{KT} \sum_{p=1}^{PT} \hat{T}_{LKP} \cdot xlkp_{l,k,p} + \sum_{k=1}^{KT} \sum_{c=1}^{CT} \sum_{p=1}^{PT} \hat{T}_{KCP} \cdot xkcp_{k,c,p} + \\
& + \sum_{c=1}^{CT} \sum_{j=1}^{JT} \sum_{p=1}^{PT} \hat{T}_{CJP} \cdot xcjp_{c,j,p} + \sum_{c=1}^{CT} \sum_{d=1}^{DT} \sum_{p=1}^{PT} \hat{T}_{CDP} \cdot xcdp_{c,d,p})
\end{aligned} \tag{1}$$

The first objective function minimizes the cost of the entire supply chain network, in which these objectives include fixed costs for the establishment of centers, fixed costs for utilizing server, and shipping (transportation) costs.

3.5.2 Second objective function: minimizing the total waiting time

The second objective function is given by:

$$z_2 = \min(\sum_{k=1}^{KT} \sum_{l=1}^{LT} \sum_{p=1}^{PT} xklp_{k,l,p} \cdot yk_k \cdot w_{k,p}) \tag{2}$$

The second objective function minimizes the total waiting time resulting from products in the queue of the distribution center.

4.6 Proposed bi-objective probabilistic fuzzy mixed-integer nonlinear mathematical model

Regarding the indices, parameters, and decision variables, the multi-product closed-loop SCND as a bi-objective probabilistic fuzzy mixed-integer nonlinear mathematical model is as follows:

$$\begin{aligned}
z_1 = & \min(\sum_{i=1}^{IT} FI_i \cdot yi_i + \sum_{i=1}^{IT} FJ_j \cdot yj_j + \sum_{i=1}^{IT} FC_c \cdot yc_c + \sum_{i=1}^{IT} FK_k \cdot yk_k + \sum_{i=1}^{IT} FI_i \cdot yi_i + \\
& + \sum_{i=1}^{IT} FD_d \cdot yd_d + \sum_{k=1}^{KT} F \mathcal{G} J_k \cdot \mathcal{G} k_k + \sum_{j=1}^{JT} F \mathcal{G} J_j \cdot \mathcal{G} j_j + \sum_{i=1}^{IT} \sum_{j=1}^{JT} \sum_{p=1}^{PT} \hat{T} IJP_{i,j,p} \cdot xijp_{i,j,p} + \\
& + \sum_{j=1}^{JT} \sum_{k=1}^{KT} \sum_{p=1}^{PT} \hat{T} JKP_{j,k,p} \cdot xj kp_{j,k,p} + \sum_{k=1}^{KT} \sum_{l=1}^{LT} \sum_{p=1}^{PT} \hat{T} KLP_{k,l,p} \cdot xklp_{k,l,p} + \\
& + \sum_{l=1}^{LT} \sum_{k=1}^{KT} \sum_{p=1}^{PT} \hat{T} LKP_{l,k,p} \cdot xlkp_{l,k,p} + \sum_{k=1}^{KT} \sum_{c=1}^{CT} \sum_{p=1}^{PT} \hat{T} KCP_{k,c,p} \cdot xkcp_{k,c,p} + \\
& + \sum_{c=1}^{CT} \sum_{j=1}^{JT} \sum_{p=1}^{PT} \hat{T} CJP_{c,j,p} \cdot xcjp_{c,j,p} + \sum_{c=1}^{CT} \sum_{d=1}^{DT} \sum_{p=1}^{PT} \hat{T} CDP_{c,d,p} \cdot xc dp_{c,d,p}) \\
z_2 = & \min(\sum_{k=1}^{KT} \sum_{l=1}^{LT} \sum_{p=1}^{PT} xklp_{k,l,p} \cdot yk_k \cdot w_{k,p})
\end{aligned}$$

s. t.

$$\sum_{k=1}^{KT} xklp_{k,l,p} \geq \hat{D}E_{l,p} \quad \forall l, p \quad (3)$$

$$\sum_{k=1}^{KT} xlkp_{l,k,p} \geq \alpha_{ll,p} \cdot \hat{D}E_{l,p} \quad \forall l, p \quad (4)$$

$$\sum_{l=1}^{LT} xlkp_{l,k,p} = \sum_{c=1}^{CT} xkcp_{k,c,p} \quad \forall k, p \quad (5)$$

$$\beta CP_{c,p} \cdot \sum_{k=1}^{KT} xkcp_{k,c,p} = \sum_{d=1}^{DT} xc dp_{c,d,p} \quad \forall c, p \quad (6)$$

$$(1 - \beta CP_{c,p}) \cdot \sum_{k=1}^{KT} xkcp_{k,c,p} = \sum_{j=1}^{JT} xcjp_{c,j,p} \quad \forall c, p \quad (7)$$

$$\sum_{j=1}^{JT} xj kp_{j,k,p} = \sum_{l=1}^{LT} xklp_{k,l,p} \quad \forall k, p \quad (8)$$

$$\sum_{l=1}^{LT} xijp_{i,j,p} + \sum_{c=1}^{CT} xcjp_{c,j,p} = \sum_{k=1}^{KT} xj kp_{j,k,p} \quad \forall j, p \quad (9)$$

$$\sum_{l=1}^{LT} xlkp_{l,k,p} \leq CKP'_{k,p} \cdot yk_k \quad \forall k, p \quad (10)$$

$$\sum_{l=1}^{LT} xklp_{k,l,p} \leq CKP_{k,p} \cdot yk_k \quad \forall k, p \quad (11)$$

$$\sum_{c=1}^{CT} xc dp_{c,d,p} \leq CDP_{d,p} \cdot yd_d \quad \forall d, p \quad (12)$$

$$\sum_{j=1}^{JT} xcjp_{c,j,p} \leq CCP_{c,p} \cdot yc_c \quad \forall c, p \quad (13)$$

$$\sum_{k=1}^{KT} xjkp_{j,k,p} \leq CJP_{j,p} \cdot yj_j \quad \forall j, p \quad (14)$$

$$\sum_{j=1}^{JT} xjkp_{j,k,p} \leq CIP_{i,p} \cdot yi_i \quad \forall i, p \quad (15)$$

$$\sum_{k=1}^{KT} yk_k = T \quad (16)$$

$$\sum_{l=1}^{LT} xklp_{k,l,p} = \omega kp_{k,p} \quad \forall k, p \quad (17)$$

$$\pi_{0,k,p} = \frac{1}{\sum_{k'=0}^{gk_k-1} \frac{\left(\frac{\omega kp_{k,p}}{\mu_k}\right)^{k'}}{k'} + \frac{\left(\frac{\omega kp_{k,p}}{\mu_k}\right)^{gk_k}}{gk_k!} + \frac{1}{1 - \frac{\omega kp_{k,p}}{\mu_k \cdot gk_k}}} \quad \forall k, p \quad (18)$$

$$w_{k,p} = \left[\frac{\pi_{0,k,p}}{gk_k!} \cdot \left(\frac{\omega kp_{k,p}}{\mu_k}\right)^{gk_k!} \cdot \frac{gk_k \cdot \mu_k}{(gk_k \cdot \mu_k - \omega kp_{k,p})^2} + \frac{1}{\mu_k} \right] \quad \forall k, p \quad (19)$$

$$P \left[\left(\begin{array}{c} \text{Queue lenght for product } p \text{ at} \\ \text{the manufacture} \end{array} \right) \geq \beta JP_{j,p} \right] \leq BQ_{j,p} \quad \forall j, p \quad (20)$$

$$xijp_{i,j,p}, xjkp_{j,k,p}, xklp_{k,l,p}, xlp_{l,k,p}, xkcp_{k,c,p}, \quad \forall i, j, k, l, c, d \quad (21)$$

$$xcjp_{c,j,p}, xcdp_{c,d,p} \geq 0$$

$$yi_i, yj_j, yk_k, yc_c, yd_d \in \{0,1\} \quad \forall i, j, k, c, d \quad (22)$$

Constraint (3) ensures the demand of customers from each product. Constraint (4) shows the fraction of the consumed and returned products from customers to the collection center. Constraint (5) ensures the equality of returned products from customers to the collection center with transferred products to the reproduction center. Constraint (6) indicates the fraction of returned products from a customer that can be reproduced. Constraint (7) shows the fraction of defective returned products from the customer, which are transferred from reproduction to disposal center. Constraint (8) shows the balance in the distribution center and ensures the transmission of new production and reproduction products to the customer. Constraint (9) indicates the constraint in balance in the producing center and shows that customer demand is provided by a total of reproduction products and products received from the supplier. Constraints (10) to (15) show the constraint in the capacity of potential facilities and include collection, disposal, reproduction, manufacturing, and supplying centers, respectively.

It should be noted that, in all of the above constraints, if any facility is chosen, only the capacity of that center can be used. Constraint (16) shows the number of required dual centers of

distribution/collection. Constraint (17) shows the product's input rate to the queueing system of the distribution system. Constraints (18) and (19) are related to queue length constraints in the distribution center and show the occurrence possibility and waiting time of each product in the constructed distribution center. For further information about how these equations were obtained, please refer to (Shortle et al., 2018), for example. Constraint (20) indicates the queue length from manufacturing and reproduction products in the producing center. Constraints (21) and (22) show the type of decision variables.

4.7 Jackson network (Queuing theory)

The studied Jackson network in the present research is illustrated in Fig. 2.

Fig. 2 Jackson network studied in the proposed model

The equations of the proposed network are as follows:

$$\pi_1' \cdot \lambda_1' + \pi_2 \cdot \lambda_2 = \lambda_1 \quad (23)$$

Throughout our investigation, π_1' value (i.e., probability of product entering the producer from the supplier) and π_2 value (i.e., probability of product entering the producer from the reproducer) are considered equal to one, respectively. This means that the amount of product specified to be sent from a specific supplier or reproducer to a specific producer will enter the production center with a probability of 1. In other words, we assume that there exists no waste or difficulty (e.g., disaster) in receiving the raw materials or semi-finished product from the supplier and the reproducer (i.e., reworking process) to the producer at all. Therefore, according to the mentioned values, the input rate to the producing center (λ_1) is calculated by:

$$\lambda_1' + \lambda_2 = \lambda_1 \quad (24)$$

Variables λ_1' and λ_2 are the input rate of the product from the supplier and the reproducer to the producer, respectively.

Consequently, the input rate corresponds to the proposed CLSC network model is as follows:

$$\omega j p_{j,p} = \sum_{i=1}^{IT} x i p_{i,j,p} + \sum_{c=1}^{CT} x c j p_{c,j,p} \quad \forall j, p \quad (25)$$

Based on Constraint (20), we use a queueing model to turn this constraint into deterministic constraints. In this queueing $M/M/C/K$ model, we have C channels with system capacity K involving service rate μ for each channel, entrance rate λ into the production, or distribution centers (note that the time between each occurrence follows the exponential distribution and as a result number of occurrence follow Poisson distribution). With the notes mentioned above, therefore, the Constraint (20) changes into Constraint (26) as follows (Marianov and Serra 2013):

$$\sum_{j'=\mathcal{G}j_j+\beta J P_{j,p}+1}^C pr_{j',j,p} \leq BQ_{j,p} \text{ or } 1 - \sum_{j'=0}^{\mathcal{G}j_j+\beta J P_{j,p}} pr_{j',j,p} \leq BQ_{j,p} \quad \forall j, p \quad (26)$$

Constraint (26) assures that the probability of occurrence in which the queue length for product p in the production center j to be greater than $\beta J P_{j,p}$ with $\mathcal{G}j_j$ the server must be lower than $BQ_{j,p}$. Note that $pr_{j',j,p}$ is related to this amount of probability corresponding j' server.

Note that the service rate in producer j is logically computed by:

$$\mu j_j = \begin{cases} n \cdot \mu J_j & n \leq \mathcal{G}j_j \\ \mathcal{G}j_j \cdot \mu J_j & \mathcal{G}j_j < n < c \end{cases} \quad \forall j, p \quad (27)$$

By combining Constraint (27), Constraint (18) changes into Constraint (28) as follows:

$$\pi_{0,j,p} = \left[\sum_{j'=0}^{\mathcal{G}j_j-1} \frac{1}{j'!} \cdot \left(\frac{\omega j p_{j,p}}{\mu j_j} \right)^{j'} + \frac{1}{\mathcal{G}j_j!} \cdot \left(\frac{\omega j p_{j,p}}{\mu j_j} \right)^{\mathcal{G}j_j} \cdot \left(\frac{\mathcal{G}j_j \cdot \mu j_j}{\mathcal{G}j_j \cdot \mu j_j - \omega j p_{j,p}} \right) \right]^{-1} \quad \forall j, p \quad (28)$$

As a result, Constraint (26) changes into Constraint (29) as below:

$$\left(\sum_{j'=0}^{\mathcal{G}j_j} \frac{\omega j p_{j,p}^{j'}}{j'! \cdot \mu j_j^{j'}} + \sum_{j'=\mathcal{G}j_j+1}^{\mathcal{G}j_j+\beta J P_{j,p}} \frac{\omega j p_{j,p}^{j'} \cdot \mathcal{G}j_j^{j'-\mathcal{G}j_j}}{j'! \cdot \mu j_j^{j'}} \right) \cdot \pi_{0,j,p} \geq 1 - BQ_{j,p} \quad \forall j, p \quad (29)$$

where the j & p are the indices of production center and product, respectively.

4.8 Linearization of the model

Due to the nonlinearity of the model, we linearize the mathematical model as much as possible (second objective function).

$$xyw_{k,l,p} = xklp_{k,l,p} \cdot yk_k \cdot w_{k,p} \quad (30)$$

$$xyw_{k,l,p} \leq M \cdot yk_k \quad (31)$$

$$xyw_{k,l,p} \leq xklp_{k,l,p} \cdot w_{k,p} \quad (32)$$

$$xklp_{k,l,p} \cdot w_{k,p} \leq xyw_{k,l,p} + M \cdot (1 - yk_k) \quad (33)$$

$$xyw_{k,l,p} \geq 0 \quad (34)$$

4.9 Crisp auxiliary model

In this section, after determining the probabilistic model of the queue, we aim to determine the model derived from fuzzy parameters. In this mode, the parameters, including transportation costs and customer demand, are considered as fuzzy parameters with a trapezoidal membership function. The following steps represent the model of changing an uncertain model to a crisp mode.

Given the uncertain nature of some important parameters (including demands, the cost of each unit of the manufacturing product, the cost of transportation and the transfer of each unit of the product and the purchase part of each unit of the supplier) and the unavailability of the required historical data at the design stage, these parameters are estimated based on expert opinions. Therefore, the above ambiguous parameters are formulated as uncertain ones in the form of trapezoidal fuzzy numbers as follows:

$$\begin{aligned} \tilde{DE}_{l,p} &= (DE_{l,p}^1, DE_{l,p}^2, DE_{l,p}^3, DE_{l,p}^4) \\ \tilde{TJIP}_{i,j,p} &= (TJIP_{i,j,p}^1, TJIP_{i,j,p}^2, TJIP_{i,j,p}^3, TJIP_{i,j,p}^4) \\ \tilde{TJKP}_{j,k,p} &= (TJKP_{j,k,p}^1, TJKP_{j,k,p}^2, TJKP_{j,k,p}^3, TJKP_{j,k,p}^4) \\ \tilde{TJKP}_{j,k,p} &= (TJKP_{j,k,p}^1, TJKP_{j,k,p}^2, TJKP_{j,k,p}^3, TJKP_{j,k,p}^4) \\ \tilde{TKLP}_{k,l,p} &= (TKLP_{k,l,p}^1, TKLP_{k,l,p}^2, TKLP_{k,l,p}^3, TKLP_{k,l,p}^4) \\ \tilde{TLKP}_{l,k,p} &= (TLKP_{l,k,p}^1, TLKP_{l,k,p}^2, TLKP_{l,k,p}^3, TLKP_{l,k,p}^4) \\ \tilde{TKCP}_{k,c,p} &= (TKCP_{k,c,p}^1, TKCP_{k,c,p}^2, TKCP_{k,c,p}^3, TKCP_{k,c,p}^4) \\ \tilde{TCJP}_{c,j,p} &= (TCJP_{c,j,p}^1, TCJP_{c,j,p}^2, TCJP_{c,j,p}^3, TCJP_{c,j,p}^4) \\ \tilde{TCDP}_{c,d,p} &= (TCDP_{c,d,p}^1, TCDP_{c,d,p}^2, TCDP_{c,d,p}^3, TCDP_{c,d,p}^4) \end{aligned} \quad (35)$$

It is worth noting that evaluating a deterministic demand is hard. It is sometimes impossible for long-term decisions (strategic decisions) because of a lack of past experiences and previous data. The demand for each product, along with the cost of producing and transporting a unit of product that changes in short-term planning, is considered a fuzzy one. Moreover, due to the uncertainty of the purchase price of the parts from the supplier, it is assumed as a vague (fuzzy) value as well.

Also, possibilistic chance-constrained programming (PCCP) is often used to deal with uncertain constraints, having non-deterministic data on the left or right side. If this method is used, to control the level of assurance of creating these uncertain constraints, the concept of the decision can achieve the minimum level of certainty as a safe margin for each of these constraints (Pishvaei et al., 2012). In this case, two standard fuzzy methods, including possibility optimistic standard (POS) and necessity (NEC), are usually used.

It should be noted that POS indicates the level of optimistic possibility of an uncertain event involving uncertain parameters, while NEC presents pessimistic decision-making about an

uncertain event. Consequently, it is more conservative to use NEC (Inuiguchi and Ramik, 2000) (i.e., we assume that decision-making has a pessimistic (conservative) attitude to establish uncertain constraints).

Therefore, NEC is used to ensure the establishment of uncertain constraints. Based on the fuzzy parameters and the use of the expected value for the objective function and the NEC measure for uncertain constraints, the obvious transformation of the uncertain model can be formulated. To do this issue, first, the abbreviation for the proposed model is considered as follows:

$$\begin{aligned}
& \text{Min } f_1 = D + C.y + F.x \\
& \text{Min } f_2 = x \\
& \text{s.t.} \\
& A.x \geq d \\
& B.x \leq y \\
& y \in \{0,1\}, x \geq 0
\end{aligned} \tag{36}$$

where the C , F , and d are the fixed costs, variable costs, and demand, respectively. Moreover, A and B are a matrix of coefficients, and finally, x and y are continuous, zero, or one variable, respectively. Now, it is assumed that F and d vectors are presented as uncertain parameters in the above model. According to the PCCP, the objective function and NEC's expected value are considered to deal with the objective function and uncertain constraint. The abbreviation form of the basic model of PCCP is shown below:

$$\begin{aligned}
& \text{Min } E[f_1] = D + C.y + E[\tilde{F}].x \\
& \text{Min } f_2 = x \\
& \text{s.t.} \\
& NEC\{A.x \geq \tilde{d}\} \geq \alpha \\
& B.x \leq y \\
& y \in \{0,1\}, x \geq 0
\end{aligned} \tag{37}$$

where α controls the minimum degree of assurance to establish an uncertain constraint by the decision-making (NEC) approach. Regarding the trapezoidal distribution function for vague (fuzzy) parameters, the general form of Model (37) is as follows:

$$\begin{aligned}
& \text{Min } E[f_1] = D + C.y + \left(\frac{F^1 + F^2 + F^3 + F^4}{4} \right).x \\
& \text{Min } f_2 = x \\
& \text{s.t.} \\
& A.x \geq (1 - \alpha).d^3 + \alpha.d^4 \\
& B.x \leq y \\
& y \in \{0,1\}, x \geq 0
\end{aligned} \tag{38}$$

Finally, the crisp mathematical model of the CLSC network is modeled by considering the length of the queue in the production and distribution system as follows:

$$\begin{aligned}
z_1 = & \min \left(\sum_{i=1}^{IT} FI_i \cdot yi_i + \sum_{i=1}^{IT} FJ_j \cdot yj_j + \sum_{i=1}^{IT} FC_c \cdot yc_c + \sum_{i=1}^{IT} FK_k \cdot yk_k + \sum_{i=1}^{IT} FI_i \cdot yi_i + \right. \\
& + \sum_{i=1}^{IT} FD_d \cdot yd_d + \sum_{k=1}^{KT} F \mathcal{G} J_k \cdot \mathcal{G} k_k + \sum_{j=1}^{JT} F \mathcal{G} J_j \cdot \mathcal{G} j_j + \\
& \sum_{i=1}^{IT} \sum_{j=1}^{JT} \sum_{p=1}^{PT} \left(\frac{TIJP_{i,j,p}^1 + TIJP_{i,j,p}^2 + TIJP_{i,j,p}^3 + TIJP_{i,j,p}^4}{4} \right) \cdot xijp_{i,j,p} + \\
& \sum_{j=1}^{JT} \sum_{k=1}^{KT} \sum_{p=1}^{PT} \left(\frac{TJKP_{j,k,p}^1 + TJKP_{j,k,p}^2 + TJKP_{j,k,p}^3 + TJKP_{j,k,p}^4}{4} \right) \cdot xjpk_{j,k,p} + \\
& \sum_{k=1}^{KT} \sum_{l=1}^{LT} \sum_{p=1}^{PT} \left(\frac{TKLP_{k,l,p}^1 + TKLP_{k,l,p}^2 + TKLP_{k,l,p}^3 + TKLP_{k,l,p}^4}{4} \right) \cdot xklp_{k,l,p} + \\
& \sum_{l=1}^{LT} \sum_{k=1}^{KT} \sum_{p=1}^{PT} \left(\frac{TLKP_{l,k,p}^1 + TLKP_{l,k,p}^2 + TLKP_{l,k,p}^3 + TLKP_{l,k,p}^4}{4} \right) \cdot xlkp_{l,k,p} + \\
& \sum_{k=1}^{KT} \sum_{c=1}^{CT} \sum_{p=1}^{PT} \left(\frac{TKCP_{k,c,p}^1 + TKCP_{k,c,p}^2 + TKCP_{k,c,p}^3 + TKCP_{k,c,p}^4}{4} \right) \cdot xkcp_{k,c,p} + \\
& \sum_{c=1}^{CT} \sum_{j=1}^{JT} \sum_{p=1}^{PT} \left(\frac{TCJP_{c,j,p}^1 + TCJP_{c,j,p}^2 + TCJP_{c,j,p}^3 + TCJP_{c,j,p}^4}{4} \right) \cdot xcjp_{c,j,p} + \\
& \sum_{c=1}^{CT} \sum_{d=1}^{DT} \sum_{p=1}^{PT} \left(\frac{TCDP_{c,d,p}^1 + TCDP_{c,d,p}^2 + TCDP_{c,d,p}^3 + TCDP_{c,d,p}^4}{4} \right) \cdot xc dp_{c,d,p} \Big)
\end{aligned} \tag{39}$$

$$z_2 = \min \left(\sum_{k=1}^{KT} \sum_{l=1}^{LT} \sum_{p=1}^{PT} xyw_{k,l,p} \right) \tag{40}$$

s.t.

$$\sum_{k=1}^{KT} xklp_{k,l,p} \geq (1 - \alpha_2) \cdot DE_{l,p}^3 + \alpha_2 \cdot DE_{l,p}^4 \quad \forall l, p \tag{41}$$

$$\sum_{k=1}^{KT} xlkp_{l,k,p} \geq \alpha_{1l,p} \cdot \left((1 - \alpha_2) \cdot DE_{l,p}^3 + \alpha_2 \cdot DE_{l,p}^4 \right) \quad \forall l, p \tag{42}$$

Constraints (5)-(19), (21)-(22), (25)-(34).

where α_2 is cut level considered to crisp our fuzzified related model.

5 Solution approaches

Given that the problem is the NP-hard type, the meta-heuristic algorithms are used to solve our developed mathematical model. These algorithms are NSGA-II and MOPSO. Following the mechanism of generating feasible solutions and structures of the above-mentioned algorithms are described.

5.1 Generating feasible solution

Considering the below mathematical model being the brief structure of our primary proposed mathematical model, we explain by a simple numerical example how we generate a feasible solution and navigate through feasible solution space.

$$Z = \min(\sum_i \sum_j C_{ij} \cdot x_{ij} + \sum_i F_i \cdot y_i) \quad (43)$$

s.t.

$$\sum_i x_{ij} \geq Dem_j \quad \forall j \quad (44)$$

$$\sum_j x_{ij} \leq Cap_i \cdot y_i \quad \forall i \quad (45)$$

In the above model, the aim is to minimize total cost involving shipment and establishment cost that demands customers to be met, and the amount of shipments is less than considered capacity. Note that C_{ij} is shipment cost between supplier i and customer j , Dem_j is the demand of customer j , Cap_i is capacity of supplier i , and x_{ij} is shipment amount between supplier i and customer j and y_i is one if supplier i is established otherwise zero.

In this example, the priority-based decoding introduced by Gen et al. (2008) is used. This encoding considers a permutation of real numbers, the length of the number of nodes in each level. The encoding method is illustrated in Fig. 3. At one of the supply chain network levels with four suppliers and three customers, encoding is based on a permutation of the number of nodes, as shown in this figure as (2-5-3-7-4-1-6). It has been shown that the priorities (4-1-6) are customer-related and (2-5-3-7) supplier-related. The following two steps must be taken to encoding.

Step 1: First, the largest priority among the selected suppliers (priority 7 to the fourth supplier), and if this supplier can meet all customer demand, the priority of the rest of the donation centers will be reduced to zero.

In the example, the fourth supplier capacity is 850, while the total customer demand is 1050. In that case, the next supplier will be selected with the next largest priority (priority 5 to the second supplier). Now the capacity of two suppliers (1600 amount) is greater than the total customer demand (1050 amount). In this case, the priorities of the first and third suppliers will be reduced to zero.

Step 2: After determining the number and location of suppliers, the optimal allocation is made between the selected suppliers and the customers. At this stage, the highest priority (priority 7 to the fourth supplier) is selected, and the lowest shipping cost associated with this supplier is identified (the first customer cost is 11.42) and the minimum amount of capacity selected, and customer determined as optimal allocation (minimum value is 300).

Priority is reduced to zero after updating residual capacity or unmet demand. Repeat the second step until all values of all priorities are reduced to zero.

The feasible space of our main model is generated in this way. Note that for some of the constraints penalty function approach, which is a deviation from the right-hand side value of related constraints, is used. These deviations are included in objective functions, and whenever this amount is equal to zero, feasible solutions are obtained. Fig. 3 illustrates the described mechanism.

Fig. 3 Modified priority-based encoding

5.2 NSGA-II algorithm

The NSGA-II is one of the most widely used and powerful algorithms for solving multi-objective optimization problems and has been proven to solve various problems. Srinivas and Deb (1994) introduced the NSGA optimization method to solve multi-objective optimization problems. In the NSGA-II, among each generation's solutions, a number of them are selected using the binary tournament selection method. In the binary selection method, two answers are randomly selected from the population, and then a comparison is made between the two solutions, whichever is the better, ultimately. Selection criteria in the NSGA-II are primarily based on the rank and secondly the crowding distance.

The lower the rank and the higher the crowding distance, the more desirable it is. By repeating the binary selection operator on the population of each generation, a set of individuals of that generation are selected to participate in the crossover and the mutation. On the part of the selected set of members, the crossover is performed, and on the others, the mutation is performed and a population of children and mutants is created. This population then merges with the original population. Newly formed population members are first ranked in ascending order. Members of the population with the same rank are arranged in descending order of density. Now the population is sorted primarily by rank and secondarily by the crowding distance. Equal to the number of people in the main population, members are selected from the top of the sorted list and the rest of the population is discarded. Selected members make up the next generation of the population, and the cycle will be repeated in this section until termination occurs. The unsuccessful solutions to the multi-objective optimization problem are often referred to as the Pareto front. None of the Pareto Front's answers is preferable to the other, and depending on the circumstances, each can be considered an optimal decision.

The steps of the NSGA-II are summarized as follows:

Step 1: Generate the initial population in this method as usual on the scale and constraint of the problem.

Step2: Evaluate the produced population from the defined objective functions.

Step 3: Apply a non-dominated sorting method. Population members are grouped in such a way that members in the first category are a completely non-dominated set by other members of the current population. The members in the second category are thus only dominated by the members

of the first category, and the process continued in the same way as all other members in each category were assigned a rank based on the category number.

Step 4: Calculate the control parameter, called the crowding distance.

This parameter is calculated for each member in each group and represents how close the sample is to other members of that population and group. A large amount of this parameter will lead to better divergence and range in the population set.

Step 5: Select the parent and population for reproduction. One of the selection mechanisms based on the binary tournament is between two randomly selected members from the population.

Step 6: Perform the mutation and crossover.

Fig. 4 illustrates the steps, as mentioned earlier, schematically.

Fig. 4 Mechanism of the NSGA-II

- *Crossover operator:* To perform the crossover, the desired crossover rate is multiplied by the number of the population then the result is divided by the number two, then the output is rounded, and the obtained result is multiplied by two finally. In this way, the number of crossovers is calculated. In the following equations, the crossover mechanism is illustrated, where x_1 and x_2 are parents α belongs to the $[0, 1]$ interval, and y_1 and y_2 are obtained, children.

$$y_1 = \alpha.x_1 + (1 - \alpha).x_2 \quad (46)$$

$$y_2 = \alpha.x_2 + (1 - \alpha).x_1 \quad (47)$$

- *Mutation operator:* To make a mutation, the desired mutation rate is multiplied by the number of population, and then the result is rounded. Now, the number of mutations that must be executed is obtained. Note that the members that mutation must be executed on them are also randomly selected from the total population. In the following equation, the mutation mechanism is shown. x is a considered member of the main population, σ belongs to the $[0,1]$ interval and r is a random variable that follows uniform distribution, whose boundaries are 0,1, respectively. The resulted mutated member is y .

$$y = x + (\sigma.r) \quad (48)$$

5.3 MOPSO algorithm

Sierra and Coello Coello (2006) introduced a multi-objective particle swarm optimization (MOPSO) algorithm. Research shows that the algorithm performs well in solving multi-objective optimization problems. This algorithm, like other mass algorithms, starts with a random population. Each member is a piece that makes up the collection. This set moves to the optimal point according to the velocities of each particle and the set in the decision space. Thus, the movement vector of each particle in each iteration is affected by the best position the particle has ever reached (Pbest) and the best position that the best member of the set has ever achieved (Gbest).

In this method, for each purpose, a generated set and its associated Gbest are extracted at each iteration. To move to the next step, each Gbest suite will move to the next suite. However, Pbest is available for each set individually. Finally, after a certain number of iterations of the last set, they are introduced as the final set and its dominated points as the set of solutions or Pareto. The members of the Pareto set have no advantage over each other, and by decreasing the value of the target function, the value of the other objective function increases, and vice versa.

As a whole, the steps of this algorithm are summarized as follows:

Step 1: Create an initial population.

Step 2: Separate undesirable members of the population and store them in the repository.

Step 3: Tabulation of the explored objective space.

Step 4: Each particle of Repository members chooses a leader and makes its move.

Step 5: The best personal memory of each particle is updated.

Step 6: Repeated members of the current population are added to the repository.

Step 7: Delete dominated members from a repository.

Step 8: If the repository member numbers exceed the specified capacity, remove the extra members and tabulate once again.

Step 9: Return to Step 3 if the termination is not fulfilled and otherwise terminate.

Fig. 5 demonstrates the steps above schematically.

Fig. 5 Mechanism of MOPSO

6. Computational results

6.1. Generating test problems

To prove that our model is valid, five problems have been tested. All parameters are the same for all problems (following from the specified normal distribution function). The problems we are considering are different in size. To be closer to reality, all parameters follow a normal distribution. The characteristics of the problems that we are considering are given in Table 1.

The values of the parameters are specified in Table 2. The model parameters were generated using the MATLAB software in terms of the normal distribution presented in Table 2.

Table 1
Test problems

Index	Description	Problems				
		1	2	3	4	5
<i>IT</i>	No. of suppliers	3	5	8	12	18
<i>JT</i>	No. of manufacturers	3	5	8	12	18
<i>CT</i>	No. of remanufacturers	2	4	6	8	12
<i>KT</i>	No. of distribution & collection centers	2	4	6	8	10
<i>LT</i>	No. of customers	5	10	15	20	30
<i>DT</i>	No. of disposal centers	2	4	6	8	10
<i>PT</i>	No. of semi-finished or finished products	2	3	4	5	6
<i>TT</i>	No. of dual centers	3	5	8	12	18

Table 2
Parameters in test problems

Parameters	Values	Parameters	Values
$DE_{l,p}^1$	$\sim$ Normal (200, 215)	FJ_j	$\sim$ Normal (100000, 120000)
$DE_{l,p}^2$	$\sim$ Normal (215, 225)	FI_i	$\sim$ Normal (100000, 120000)
$DE_{l,p}^3$	$\sim$ Normal (225, 235)	FD_d	$\sim$ Normal (100000, 120000)
$DE_{l,p}^4$	$\sim$ Normal (235, 250)	$F9K_k$	$\sim$ Normal (1000, 1200)
$TIJP_{i,j,p}^1$	$\sim$ Normal (5, 6)	$F9J_j$	$\sim$ Normal (1000, 1200)
$TIJP_{i,j,p}^2$	$\sim$ Normal (6, 7)	$\alpha_{l,p}$	$\sim$ Normal (0.1, 0.2)
$TIJP_{i,j,p}^3$	$\sim$ Normal (7, 8)	$\beta DP_{d,p}$	$\sim$ Normal (0.1, 0.2)
$TIJP_{i,j,p}^4$	$\sim$ Normal (8, 10)	$\beta JP_{j,p}$	10
$TJKP_{j,k,p}^1$	$\sim$ Normal (5, 6)	μJ_j	$\sim$ Normal (100, 200)
$TJKP_{j,k,p}^2$	$\sim$ Normal (6, 7)	μK_k	$\sim$ Normal (100, 200)
$TJKP_{j,k,p}^3$	$\sim$ Normal (7, 8)	$BQ_{j,p}$	$\sim$ Normal (0.05, 0.1)
$TJKP_{j,k,p}^4$	$\sim$ Normal (8, 10)	$CKP_{k,p}$	$\sim$ Normal (800, 1000)
$TKLP_{k,l,p}^1$	$\sim$ Normal (5, 6)	$CKP'_{k,p}$	$\sim$ Normal (150, 250)
$TKLP_{k,l,p}^2$	$\sim$ Normal (6, 7)	$CCP_{c,p}$	$\sim$ Normal (150, 250)
$TKLP_{k,l,p}^3$	$\sim$ Normal (7, 8)	$CJP_{j,p}$	$\sim$ Normal (800, 1000)
$TKLP_{k,l,p}^4$	$\sim$ Normal (8, 10)	$CIP_{i,p}$	$\sim$ Normal (800, 1000)

$TLKP_{l,k,p}^1$	$\sim \text{Normal (5, 6)}$	$CDP_{d,p}$	$\sim \text{Normal (100, 180)}$
$TLKP_{l,k,p}^2$	$\sim \text{Normal (6, 7)}$	$TCJP_{c,j,p}^1$	$\sim \text{Normal (5, 6)}$
$TLKP_{l,k,p}^3$	$\sim \text{Normal (7, 8)}$	$TCJP_{c,j,p}^2$	$\sim \text{Normal (6, 7)}$
$TLKP_{l,k,p}^4$	$\sim \text{Normal (8, 10)}$	$TCJP_{c,j,p}^3$	$\sim \text{Normal (7, 8)}$
$TKCP_{k,c,p}^1$	$\sim \text{Normal (5, 6)}$	$TCJP_{c,j,p}^4$	$\sim \text{Normal (8, 10)}$
$TKCP_{k,c,p}^2$	$\sim \text{Normal (6, 7)}$	$TCDP_{c,d,p}^1$	$\sim \text{Normal (5, 6)}$
$TKCP_{k,c,p}^3$	$\sim \text{Normal (7, 8)}$	$TCDP_{c,d,p}^2$	$\sim \text{Normal (6, 7)}$
$TKCP_{k,c,p}^4$	$\sim \text{Normal (8, 10)}$	$TCDP_{c,d,p}^3$	$\sim \text{Normal (7, 8)}$
FK_k	$\sim \text{Normal (100000, 120000)}$	$TCDP_{c,d,p}^4$	$\sim \text{Normal (8, 10)}$
FC_c	$\sim \text{Normal (100000, 120000)}$		

6.2. Comparison criteria

To compare the two aforementioned algorithms, we consider four measures, in which except one of them (i.e., computational time), the other investigate the formed Pareto front involving non-dominated solutions related to each algorithm.

6.2.1. Central processing unit (CPU) time

Logically, big computational time with respect to the maximum iteration considered for each algorithm is not adequate. Thus, in the comparison between two considered algorithms, the one that yields a smaller amount is more acceptable.

6.2.2. Number of Pareto solutions in the first frontier

In comparison between the two above mentioned algorithms, the more non-dominated solutions formed in the first frontier is more acceptable, because the user has more alternative to choose.

6.2.3. Maximum spreading index (MSI)

Zitzler et al. (2007) presented this criterion the length of the formed frontier by endpoints of the non-dominated Pareto front. The bigger this index, the better it would be.

$$D = \sqrt[p]{\sum_{k=1}^{nobj} |f_k^{\max} - f_k^{\min}|^p} \quad (49)$$

6.2.4. Spacing metric index (SI)

The criterion presented by Schott (1995) is the relative distance between successive non-dominated solutions. The measured interval is the minimum of the sum of the absolute magnitude of the difference in the values of the objective functions between the i -th solution and other solutions located in the final frontier. This index measures the standard deviation of different values. When the solutions are evenly scattered, then the value of s will also be small, so an algorithm with these non-dominated solutions with small space is better.

$$s_i = \min_j \left(\sum_{k=1}^{nobj} |f_k^i - f_k^j| \right) \quad \forall i, j, i \neq j \quad (50)$$

$$\bar{s} = \frac{1}{n} \sum_{i=1}^n s_i \quad (51)$$

$$s = \left[\frac{1}{n-1} \sum_{i=1}^n (s_i - \bar{s})^2 \right]^{\frac{1}{2}} \quad (52)$$

6.2.5. Surface metric index (SM)

The most important criterion is the covering surface. The covering surface indicates the ratio of the number of non-dominated Pareto solutions belonging to the related algorithm is an ideal common frontier formed by considered algorithms. The higher the criterion and closer to one value in comparison with other considered algorithms will be more acceptable

6.3. Tuning structural parameters of NSGA-II & MOPSO algorithms

First of all, the software used to run these two algorithms is Matlab R2016a, and the characteristics of the computer machine used to deal with the algorithms are Intel(R), Core(TM) i5-4200U, CPU @ 1.6GHz 2.3 GHz, 3.89 GB of RAM. The performance of the NSGA-II will be at its highest if the structural parameters of the NSGA-II, including the number of population per iteration, maximum number of iterations, mutation, and crossover rates are set to 80, 120, 0.6, and 0.5, respectively. Also, for the MOPSO algorithm, these values for the parameters, such as the number of particles per each iteration, the maximum number of iterations, initial velocity coefficient, and secondary velocity coefficient must be set on the numbers 80, 120, 2, and 1.5, respectively. Tables 3 and 4 show the considered levels and related values of the aforementioned parameters and their optimal values obtained, respectively.

As a whole, we categorize the obtained results as follows. In the first step, a small instance problem is solved by two algorithms (i.e., NSGA-II and MOPSO). Then, decision variables are extracted and analyzed. In the second step, five different sizes of the proposed problem are generated, and ten instance problems are designed for each of them (at the same size). The simple problems of any size are solved by the proposed algorithms and compared using the comparison indices or metrics (i.e., NPF, CPU time, MSI, SI, and SM) of the multi-objective meta-heuristics. The *t*-test is used to measure the significance of the difference between the obtained indices in each size. Finally, the best algorithm is identified and selected using the TOPSIS method.

Table 3

Structural parameter values of two algorithms considered during parameter tuning mechanism

Algorithm	Structural parameter	Level -1	Level -2	Level -3
NSGA-II	Population size per each iteration	80	120	200
	Maximum number of iterations	80	120	200
	Mutation rate	0.5	0.7	0.9
	crossover rate	0.2	0.4	0.6
MOPSO	Number of particles per iteration	80	120	200

Maximum number of iterations	80	120	200
Initial velocity coefficient	1	1.5	2
Secondary velocity coefficient	1	1.5	2

Table 4

Optimal structural parameter values of the NSGA-II and MOPSO obtained through tuning mechanism

NSGA-II		MOPSO	
Structural parameter	Optimum value	Structural parameter	Optimum value
Population size per each iteration	80	Number of particles per iteration	80
Maximum number of iterations	120	Maximum number of iterations	120
Mutation rate	0.6	Initial speed coefficient	2
crossover rate	0.5	Secondary speed coefficient	1.5

6.4. Analysis based on test problem no. 1 in a probabilistic-fuzzy mode

In this section, in the first step to evaluate the results and comparisons of the proposed model in a probabilistic-fuzzy mode, a simple problem is provided and solved by NSGA-II and MOPSO algorithms. Then, decision variables are extracted and analyzed. Given that the bi-objective model contains a set of efficient solutions, therefore, there is no optimal value for this model. Thus, in this study, we aim to express the values of decision variables of one of the effective solutions obtained from the NSGA-II and MOPSO algorithms. Table 5 shows the number of obtained effective solutions, including first and second objectives. Fig. 6 shows the Pareto frontier obtained in 120 repetitions for both algorithms. The convex form of the Parent frontiers confirms the conflict between two objective functions.

Table 5

Efficient solutions obtained using the NSGA-II and MOPSO in relation to the considered test problem in 120 iterations

The n -th efficient solution	NSGA-II		MOPSO	
	First objective function	Second objective function	First objective function	Second objective function
1	923150.74	1188.43	922808.8	1244.55
2	923709.86	1184.9	924457.48	1228.37
3	924222.98	1138.69	925380.54	1166.19
4	925812.73	1100.15	926024.43	1069.28
5	925888.57	1099.28	929165.69	998.56
6	927814.72	1094.34	933647.11	996.39
7	928722.09	151.95	933960.82	991.68
8	99320.87	979.47	934097.54	883.35
9	931384.31	967.43	942561.79	874.64
10	934015.23	929.38	945270.42	873.53
11	936631.88	928.93	945840.81	865.98

12	938283.62	891.88	949054.01	834.64
13	942454.11	855.14	950008.77	829.72
14	942221.72	836.15		
15	945618.73	804.76

Fig. 6. Pareto front formed using NSGA-II and MOPSO (Test problem size 1)

Tables 6 to 12 show the number of optimal selected locations on one of the efficient solutions to the above-mentioned problem by two algorithms, in which the cut-off alpha α_2 in the trapezoidal fuzzy model is 0.5. Note that the results for both algorithms are related to the first solution from the left-hand side of the Pareto front. All of the following tables exhibit the amount of transported goods between levels of the related supply chain and also specify which facilities involving (producer and distributor) are located and being active. Besides, how many servers to be established in producers and distributors are also specified. For example, in Table 7, the transferred materials flow is shown between the selected facilities by the two meta-heuristic algorithms. The amount of transferred flow between suppliers and producers is shown.

Table 6

Number and location of the optimal potential facilities in relation to the test problem (size no. 1)

Algorithm	NSGA-II		MOPSO	
	Optimum number	Selected location	Optimum number	Selected location
Supplier	2	1, 3	2	1, 2
Manufacturer	2	2, 3	2	2, 3
Dual Center	2	1, 3	2	2, 3
Remanufacturer	1	1	2	1, 2
Disposal Center	1	1	1	1
Number of servers in the distribution center	9	(5,0,4)	10	(5,0,5)
Number of servers in the manufacturing center	8	(6,2,0)	8	(4,0,4)

Table 7Transferred amount of semi-finished product p between supplier i and producer j

Algorithms			
NSGA-II		MOPSO	
$xijp_{1,1,1}$	905.87	$xijp_{1,3,1}$	297.21
$xijp_{1,3,1}$	54.83	$xijp_{2,2,1}$	252.03
$xijp_{3,3,1}$	222.83	$xijp_{2,3,1}$	634.3
$xijp_{1,2,2}$	834.71	$xijp_{1,3,2}$	340.17
$xijp_{1,3,2}$	40.19	$xijp_{2,2,2}$	308.21
$xijp_{3,3,2}$	295.99	$xijp_{2,3,2}$	522.51

Table 8Transferred amount of finished product p between producer j and distribution center k

Algorithms			
NSGA-II		MOPSO	
$xj kp_{2,3,1}$	905.87	$xj kp_{2,2,1}$	252.03
$xj kp_{3,1,1}$	276.36	$xj kp_{3,2,1}$	597.37
$xj kp_{3,3,1}$	1.29	$xj kp_{3,3,1}$	334.14
$xj kp_{2,1,2}$	285.89	$xj kp_{2,2,2}$	33.11
$xj kp_{2,3,2}$	548.82	$xj kp_{2,3,2}$	275.1
$xj kp_{3,3,2}$	336.18	$xj kp_{3,2,2}$	862.68

Table 9Transferred amount of finished product p between distribution center k and customer l

Algorithms			
NSGA-II		MOPSO	
$xklp_{1,1,1}$	37.6	$xklp_{2,1,1}$	234.66
$xklp_{1,2,1}$	238.76	$xklp_{2,2,1}$	238.76
$xklp_{3,1,1}$	197.06	$xklp_{3,3,1}$	234.64
$xklp_{3,3,1}$	234.64	$xklp_{2,4,1}$	239.23
$xklp_{3,4,1}$	239.23	$xklp_{2,5,1}$	136.75
$xklp_{3,5,1}$	236.25	$xklp_{3,5,1}$	99.5
$xklp_{1,3,2}$	232.92	$xklp_{2,1,2}$	235.79
$xklp_{3,1,2}$	235.79	$xklp_{2,2,2}$	193.88
$xklp_{3,2,2}$	236.18	$xklp_{2,3,2}$	232.92
$xklp_{1,5,2}$	52.97	$xklp_{3,2,2}$	42.3
$xklp_{3,4,2}$	232.79	$xklp_{2,5,2}$	233.21
$xklp_{3,5,2}$	180.23	$xklp_{3,4,2}$	232.79

Table 10Transferred amount of finished product p between customer l and collection center k

	NSGA-II		MOPSO
$xlkp_{1,3,1}$	30.78	$xlkp_{1,2,1}$	30.78
$xlkp_{2,3,1}$	41.07	$xlkp_{2,2,1}$	41.07
$xlkp_{3,3,1}$	40.37	$xlkp_{3,2,1}$	40.37
$xlkp_{4,3,1}$	34.42	$xlkp_{4,2,1}$	34.42
$xlkp_{5,3,1}$	25.57	$xlkp_{5,2,1}$	25.57
$xlkp_{1,3,2}$	24.69	$xlkp_{1,2,2}$	24.69
$xlkp_{2,3,2}$	36.01	$xlkp_{2,2,2}$	36.01
$xlkp_{3,3,2}$	30.41	$xlkp_{3,2,2}$	30.41
$xlkp_{4,3,2}$	24.69	$xlkp_{4,2,2}$	24.69
$xlkp_{5,3,2}$	38.16	$xlkp_{5,2,2}$	38.16

Table 11Transferred amount of finished product p between collection center k and remanufacturer c

	NSGA-II		MOPSO
$xkcp_{3,2,1}$	172.21	$xkcp_{2,1,1}$	165.15
$xkcp_{3,2,2}$	153.96	$xkcp_{2,2,1}$	7.06
		$xkcp_{1,1,2}$	153.96

Table 12Transferred amount of finished product p between remanufacturer c and disposal center d

	NSGA-II		MOPSO
$xcdp_{2,1,1}$	24.59	$xcdp_{1,1,1}$	22.74
$xcdp_{2,1,2}$	21.1	$xcdp_{2,1,1}$	1.01
		$xcdp_{1,1,2}$	19.41

In the second step, five different sizes of the proposed problem are generated, and ten instance problems are designed for each of them at the same size. After evaluating algorithms and providing an output from an efficient solution of the instance problem, a comparison between NSGA-II and MOPSO is conducted in the larger dimension. For this purpose, ten instance problems of any size are created according to Table 13, and the metrics obtained by each algorithm (i.e., NPF, CPU, MSI, SI, and SM) are compared with the t -test for each size. Finally, to solve the proposed model of the closed-loop supply chain, the best algorithm is chosen using the TOPSIS method.

6.5. Analysis based on test problem no. 1.

Table 14 shows the results of the 10-instance problem in size 1. In this table, the average results from all the efficient solutions obtained by each sample problem are described. The comparison of meta-heuristic algorithms shows different indices (i.e., NPF, CPU, MSI, SI, and SM) as depicted in Figs. 7 to 16 in detail.

Table 14

Pareto front measurement indexes in relation to the test problem size no.1

Prob.	NSGA-II							MOPSO						
	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2
1.00	15	165.31	32698.13	2150.52	0.85	941293.34	814.78	13	161.35	18305.50	874.88	0.60	955064.09	905.72
2.00	14	168.74	22128.86	864.37	0.69	988071.61	643.56	14	164.55	16839.78	612.76	0.46	987535.21	923.92
3.00	10.	163.19	15193.66	735.29	0.48	930948.37	850.13	12	166.37	13826.16	691.22	0.65	931370.87	945.18
4.00	13	161.31	20114.11	979.64	0.58	937797.65	794.44	12	162.18	18683.14	937.78	0.35	939067.62	929.01
5.00	11	169.18	27151.24	6881.05	0.80	1045376.63	652.98	10	159.83	20815.61	670.25	0.73	1046320.57	862.31
6.00	10	167.55	17127.44	1159.79	0.49	944013.38	1000.92	9	162.04	9261.79	522.43	0.67	942563.34	979.72
7.00	10	173.21	29532.15	1013.52	0.37	1051336.34	997.74	11	166.23	24994.73	639.47	0.18	1049213.68	969.61
8.00	12	174.39	28759.46	6522.53	0.73	913076.98	629.27	14	160.90	37207.34	799.39	0.78	914318.41	963.98
9.00	7	168.33	13315.90	939.51	0.42	1032328.49	901.71	8	168.30	11774.17	996.62	0.63	1030540.29	881.62
10.00	9	162.27	11979.72	979.38	0.63	934566.68	714.98	6	163.18	9053.81	526.53	0.61	934965.67	688.53
Mean	11.10	167.35	21800.07	2222.56	0.60	971880.95	800.05	10.90	163.49	18076.20	727.13	0.57	973095.98	904.96

NPF index: Related to the test problem no. 1. The average NPF measure for ten instances is relatively equal. However, the solution scattering for MOPSO is relatively larger than the NSGA-II. Concerning, the descending trend is reported for both algorithms, but in some points, the NSGA-II and MOPSO outperform each other.

Fig. 7. NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 1)

Fig. 8. NPF measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 1)

CPU time index: Related to the test problem no. 1. The average CPU time index for ten instances in the NSGA-II is bigger than MOPSO. Also, solution scattering for the NSGA-II is larger than MOPSO as well. As a whole, both algorithms have a relatively equal habit. Descending and ascending pasterns are also reported for both algorithms, and in some points, MOPSO & NSGA-II outperform each other as well.

Fig. 9. CPU time measure obtained using NSGA-II and MOPSO (Box Plot). (Test problem size 1)

Fig. 10. CPU time incurred using NSGA-II and MOPSO vs. problem no. (Test problem size. 1)

MSI index: Associated with, the NSGA-II has a bigger MSI index average than MOPSO. Also, the scattering of solutions is bigger for the NSGA-II as well. All in all, both algorithms have virtually equal habits. Descending and ascending patterns are also reported for both algorithms the same. In some points, the NSGA-II and MOPSO have superiority over each other as well.

Fig. 11. MSI measure obtained using NSGA-II and MOPSO (Box Plot). (Test problem size 1)

Fig. 12. MSI measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 1)

SI index: Associated with, the NSGA-II has a bigger SI index average than MOPSO. Also, the scattering of solutions is more significant for the NSGA-II tremendously as well. Generally, MOPSO for all instances outperforms the NSGA-II and for points 5 and 6 tremendously as well.

Fig. 13. SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size1)

Fig. 14. SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 1)

SM index: Associated with the NSGA-II and MOPSO has the same averages. Also, the scattering of solutions is the same for both, relatively. All in all, both algorithms have virtually equal habits. Descending and ascending patterns are also reported for both algorithms the same. In some points, the NSGA-II and MOPSO have superiority over each other as well.

Fig. 15. SM measure obtained using NSGA-II and MOPSO (Box Plot). (Test problem size 1)

Fig. 16. SM measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 1)

6.6. Analysis based on test problem no. 2.

Table 15 illustrates the numerical results obtained by the algorithms mentioned above vs. the specified indices in detail. The comparison of meta-heuristic algorithms shows different indices (i.e., NPF, CPU, MSI, SI, and SM) as depicted in Figs. 17 to 26 in detail.

Table 15

Pareto front measurement indexes in relation to the test problem size no.2

Prob.	NSGA-II							MOPSO						
	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2
1	9	243.7	31567.91	6963.62	0.75	1505604.1	1219.54	10	233.27	59794.37	6720.39	0.62	1519426.2	1229.58
2	6	241	45582.9	14060.11	0.7	1455603.4	1295.62	9	219.19	17357.94	1266.88	0.62	1462656.3	1387.85
3	9	233.77	64181.72	12924.29	0.69	1461989.9	1285.62	12	225.6	17221.4	1533.28	0.89	1451862	1314.12
4	9	231.42	56928.94	11299.89	0.62	1489684.7	1475.05	14	222.92	29240.52	2144.71	0.97	1483461.6	1389.41
5	8	229.9	22638.85	3449.2	0.68	1460154.5	1328.44	8	226.6	29990.24	5306.41	0.52	1468755.1	1222.72
6	5	234.02	46993.94	12166.94	0.99	1478096.2	1204.49	12	233.2	23571.03	2213.24	0.77	1465613.8	1438.01
7	8	240.91	49242.83	9883.34	0.84	1485182.9	1418.45	11	225.7	21381.86	868.53	0.64	1486647.7	1353.48
8	13	241.54	184256.05	41617.59	0.89	1502014.8	1671.56	13	230.03	22871.23	1430.24	0.95	1490505.4	1476.29
9	10	237.02	134863.11	30035.95	0.82	1477529.4	1518.44	8	224.81	32938.88	4271.5	0.84	1464273.4	1381.67
10	5	240.18	23138.16	4933.3	0.47	1446520.6	1539.23	11	224.31	64852.32	9964.29	0.83	1455901.1	1471.73
Mean	8.2	237.346	65939.441	14733.423	0.745	1476238.1	1395.644	10.8	226.563	31921.979	3571.947	0.765	1474910.3	1366.486

NPF index: Related to test Problem no. 2, the average NPF index for 10 instances is reported bigger than for MOPSO. However, the solution scattering for MOPSO & NSGA-II is somewhat equal. Concerning, in all points except point 9, the NPF measure for MOPSO is more significant.

Fig. 17. NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Fig. 18. NPF measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 2)

CPU time index: It is related to test problem no. 2. The average CPU time index for ten instances in the NSGA-II is bigger than MOPSO. Also, the solution scattering for the NSGA-II and MOPSO is relatively the same. As a whole, CPU time for MOPSO for all points is better than the NSGA-II.

Fig. 19. CPU-time measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Fig. 20. CPU time incurred using the NSGA-II and MOPSO vs. problem no. (Test problem size. 2)

MSI index: Associated with, the NSGA-II has a bigger MSI index average than MOPSO. Also, the scattering of solutions is bigger for the NSGA-II as well. All in all, for the majority of points except points 1, 5, and 10, the NSGA-II outperforms the MOPSO algorithm.

Fig. 21. MSI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Fig. 22. MSI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 2)

SI index: Associated with, the NSGA-II has a bigger SI index average than MOPSO. Also, the scattering of solutions is bigger for the NSGA-II tremendously as well. Generally, MOPSO for all instances outperforms the NSGA-II except for points 5 and 10 as well.

Fig. 23. SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Fig. 24. SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 2)

SM index: Associated with the NSGA-II and MOPSO has the same averages. Also, the scattering of solutions for MOPSO is reported bigger. All in all, descending and ascending patterns are also reported for both algorithms. However, in some points, the NSGA-II and MOPSO have superiority over each other as well.

Fig. 25. SM measure obtained using NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Fig. 26. SM measure obtained using the NSGA-II and MOPSO vs problem no. (Test problem size 2)

6.7 Analysis based on test problem no. 3.

Table 16 clarifies numerical results obtained by the aforementioned algorithms vs. specified indices in detail. The comparison of meta-heuristic algorithms shows different indices (i.e., NPF, CPU, MSI, SI, and SM) as depicted in Figs. 27 to 36 in detail.

Table 16

Pareto front measurement indexes in relation to the test problem size no.3

Prob.	NSGA-II							MOPSO						
	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2
1	5	380.07	21481.37	2667.93	0.54	2182105.6	2306.91	15	401.067	47645.33	2287.12	0.94	2192900.1	2344.38
2	14	367.24	59884.81	3219.11	0.57	2082218.5	2110.1	12	394.51	136609.14	27804.26	0.73	2080733.9	2228.27
3	9	364.43	67569.22	9065.58	0.53	2065368.8	2182.47	6	389.7	14596.54	2615.43	0.5	2047308.6	2274.48
4	14	376.1	179157.43	33579.53	0.78	2197980.5	2351.58	9	389.6	181905	54141.02	0.92	2204837.7	2444.29
5	9	381.56	33704.91	1896.83	0.61	2099942.2	2241.48	16	405.91	136067.38	21151.55	0.69	2117370.4	2193.44
6	10	399.72	317253.51	40681.94	0.95	2235192.5	2229.61	10	408.02	42406.75	3048.57	0.68	2176931.2	2148.53
7	11	368.62	178658.08	10395.81	0.9	2135965.3	2113.08	10	405.93	265015.4	70757.05	0.84	2141948.9	2451.15
8	9	374.85	42659.66	1826.71	0.45	2061008.8	2279.9	8	397.93	142593.52	2465.97	0.71	2107589.5	2011.31
9	10	370.96	59487.81	4612.04	0.58	2137851.5	2063.35	7	406.9	266288.87	55768.77	1.21	2177254.5	2537.45
10	10	369.59	277499.62	77008.83	0.88	2137827.6	2244.61	20	386.08	33752.76	2154.5	0.9	2108916.7	2287.84
Mean	10.1	375.314	123735.64	18495.431	0.679	2133546.1	2212.31	11.3	398.5647	126688.07	24219.424	0.812	2135579.2	2292.11

NPF index: Related to test problem no. 3, the average of the NPF measure for ten instances is reported bigger than for MOPSO. Solution scattering for MOPSO is tremendously bigger.

Fig. 27. NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Concerning, as a whole, the same trend is reported for both algorithms. However, in some points, NSGA-II and MOPSO outperform each other.

Fig. 28. NPF measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 3)

CPU time index: It is related to the test Problem no. 3. The average CPU time index for ten instances in MOPSO is bigger than the NSGA-I. Also, the solution scattering for the NSGA-II and MOPSO is relatively the same. As a whole, the CPU time for the NSGA-II for all points is better than MOPSO.

Fig. 29. CPU time measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Fig. 30. CPU time incurred using the NSGA-II and MOPSO vs. problem no. (Test problem size. 3)

MSI index: Associated with the NSGA-II and MOPSO have the same MSI index average. Also, the scattering of solutions is relatively the same for both algorithms. Concerning, as a whole, the same trend is reported for both algorithms. However, in some points, the NSGA-II and MOPSO outperform each other.

Fig. 31. MSI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Fig. 32. MSI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 3)

SI index: Associated with, MOPSO has a bigger SI index average than the NSGA-II. Also, the scattering of solutions is reported bigger for MOPSO as well. Generally, NSGA-II and MOPSO have the same habits. However, in some points, the NSGA-II and MOPSO outperform each other.

SI

Fig. 33. SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Fig. 34. SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 3)

SM index: Associated with, the NSGA-II has a smaller average than MOPSO. Also, the scattering of solutions for NSGA-II is reported bigger. Generally, the NSGA-II and MOPSO have the same habits. However, in some points, the NSGA-II and MOPSO outperform each other.

Fig. 35. SM measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Fig. 36. SM measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 3)

5.8. Analysis based on test problem no. 4.

Table 17 illuminates the numerical results obtained by the aforementioned algorithms vs. the specified indices in detail. The comparison of meta-heuristic algorithms shows different indices (i.e., NPF, CPU, MSI, SI, and SM) as depicted in Figs. 37 to 46 in detail.

Table 17

Pareto front measurement indices in relation to the test problem size no.4

Prob.	NSGA-II							MOPSO						
	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2
1	9	570.79	53101.2	6041.62	0.64	3040177.7	3162.92	10	768.51	37512.45	2122.54	0.67	3073091	2919.43
2	12	597.39	46584.62	4305.91	0.72	3122290.9	3023.28	11	790.19	90060.98	12385.98	0.61	3123108.4	2992.21
3	3	593.46	32839.27	7215.83	0.29	3036578.4	2853.88	6	779.68	37746.34	5234.46	0.72	3035431.2	3121.88
4	12	593.92	80965.95	6266.66	0.54	3102393.4	2143.41	6	776.3	15432.35	998.27	0.36	3091370.7	3203.88
5	12	583.85	50538.69	5321.64	0.71	3101425.7	3109.81	9	775.34	49832.37	7085.73	0.61	3136983.7	3110.44
6	6	591.57	30584.08	2568.44	0.68	3027753.9	3235.29	11	769.94	89195.83	15934.93	0.74	3056390.9	2840.17
7	9	590.63	19826.39	1668.44	0.81	3062813.3	3142.47	7	787.38	192823.7	4981.17	0.72	3122467.9	2983.17
8	10	603.47	39403.82	3106.53	0.61	3073655.5	3203.54	10	770.23	115025.54	9107.45	0.96	3113829.4	3070.44
9	14	605.86	197781.22	29660.82	0.73	3108338.4	3062.1	10	797.07	54623.41	2685.1	0.56	3107517.4	2935.98
10	12	604.43	79066.46	14503.4	0.64	3075211.1	3025	8	756.9	61354.59	13031.94	0.62	3092616.2	3172.7
Mean	9.9	593.537	63069.17	8065.929	0.637	3075063.8	2996.17	8.8	777.154	74360.756	7356.757	0.657	3095280.7	3035.03

NPF index: It is related to test problem no. 4. The average of NPF measure for ten instances is reported bigger for NSGA-II. Solution scattering for the NSGA-II is also relatively bigger. Generally, the NSGA-II and MOPSO have the same habits. However, in some points, the NSGA-II and MOPSO outperform each other.

Fig. 37. NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Fig. 38. NPF measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 4)

CPU time index: It is related to test problem no. 4. The average CPU time index for ten instances in MOPSO is bigger than the NSGA-I. Also, solution scattering for the NSGA-II and MOPSO is fairly the same. As a whole, CPU time for the NSGA-II for all points is better than MOPSO.

Fig. 39. CPU time measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Fig. 40. CPU time incurred using the NSGA-II and MOPSO vs. problem no. (Test problem size. 4)

MSI index: Associated with, MOPSO has a relatively bigger MSI index average than the NSGA-II. Also, the scattering of solutions is somewhat smaller for the NSGA-II. In some points, the NSGA-II and MOPSO outperform each other.

Fig. 41. MSI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Fig. 42. MSI measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size. 4)

SI index: Associated with, the NSGA-II has a bigger SI index average than MOPSO. Also, the scattering of solutions is reported bigger for MOPSO as well. In some points, the NSGA-II and MOPSO outperform each other.

Fig. 43. SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Fig. 44. SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 4)

SM index: Associated with the NSGA-II and MOPSO has relatively the same SI index average. Also, the scattering of solutions is reported bigger for MOPSO as well. In some points, the NSGA-II and MOPSO outperform each other.

Fig. 45. SM measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Fig. 46. SM measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 4)

6.9. Analysis based on test problem no. 5.

Table 18 illuminates the numerical results obtained by the aforementioned algorithms vs. specified indexes in detail. The comparison of meta-heuristic algorithms shows different indices (i.e., NPF, CPU, MSI, SI, and SM) as depicted in Figs. 47 to 56 in detail.

Table 18

Pareto front measurement indexes in relation to the test problem size no.5

Prob.	NSGA-II							MOPSO						
	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2	NPF	CPU	MSI	SI	SM	Mean of Z_1	Mean of Z_2
1	9	1094.03	218331.78	48180.8	0.88	4522081.2	3532.35	17	1757.82	401041.41	20466.17	0.88	4610238.6	3490.6
2	8	1108.39	386094.1	28065.05	0.53	4672951.9	3223.67	5	1837.9	277505.98	43499.76	0.66	4669059.9	3206.58
3	5	1205.17	51109.93	3086.89	0.38	4507684.1	3496.72	10	1841.7	171928.22	29169.53	0.58	458394.04	3333.34
4	10	1209.31	405223.45	115158.64	0.92	4613856	3791.72	13	1862.4	344931.36	81694.43	0.89	4732501.4	3618.66
5	13	1225.45	239039.69	24389.49	1	5693263.6	3372.81	6	1875.1	364532.63	54889.89	0.6	4732419.4	3239.79
6	11	1202.87	297845.63	33975.76	1.09	4565663.3	3806.58	8	1843.55	294124.75	37693.37	0.62	4647910.9	3437.5
7	8	1183.88	186368.78	52831.64	0.85	4538024.5	3806.82	7	1838	48725.97	2582.67	0.64	4616815.7	3586.16
8	6	1223.3	65476.54	7010.25	0.8	4556242.1	3613.13	8	1848.9	287455.73	33356.16	0.94	4717874.5	3334.26
9	13	1220.57	391839.44	36773.35	1.01	4565396	3557.78	9	1823.89	156216.08	30554.14	0.69	4587727.4	3272.38
10	7	1221.91	29880.37	1872.8	0.43	4493571.5	3450.85	8	1821.5	255278.19	32746.8	1.1	4615945	3751.91
Mean	9	1189.48	227120.97	35134.46	0.789	4672873.4	3565.24	9.1	1835.076	260174.03	36665.29	0.76	4238888.7	3427.118

NPF index: Related to test problem no. 5, the average of the NPF measure for ten instances is reported relatively the same for both algorithms. Solution scattering for the NSGA-II is also the same. Generally, the NSGA-II and MOPSO have the same habits. However, in some points, the NSGA-II and MOPSO outperform each other.

NPF

Fig. 47. NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Fig. 48. NPF measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 5)

CPU time index: It is related to test problem no. 5. The average CPU time index for ten instances in MOPSO is bigger than the NSGA-II. Also, solution scattering for the NSGA-II and MOPSO is fairly the same. As a whole, CPU time for the NSGA-II for all points is better than MOPSO.

Fig. 49. CPU time measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Fig. 50. CPU time incurred using the NSGA-II and MOPSO vs. problem no. (Test problem size. 5)

MSI index: Associated with, MOPSO has a relatively bigger MSI index average than the NSGA-II. Also, the scattering of solutions is tremendously more significant for the NSGA-II. Concerning, in some points, the NSGA-II and MOPSO outperform each other.

Fig. 51. MSI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Fig. 52. MSI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 5)

SI index: Associated with averages of the NSGA-II and MOPSO are fairly the same. Also, the scattering of solutions is reported bigger for the NSGA-II as well. Generally, the NSGA-II and MOPSO have the same habits. However, in some points, the NSGA-II and MOPSO outperform each other.

Fig. 53. SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Fig. 54. SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 5)

SM index: Associated with the NSGA-II and MOPSO has fairly the same SM index average. Also, the scattering of solutions is reported bigger for the NSGA-II as well. Generally, the NSGA-II and MOPSO have the same habits. However, in some points, the NSGA-II and MOPSO outperform each other.

Fig. 55. SM measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Fig. 56. SM measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 5)

6.10. Comparison between two algorithms

The *t*-test is used to measure the significance of the difference between the obtained indices in sample problem no. 1. Finally, the best algorithm is identified and selected using the TOPSIS method.

6.10.1. Performing the *t*-test hypothesis

For the overall evaluation of these algorithms, the mean differences between these indices should be significantly determined by the *t*-test. Tables 19 to 22 examine the significance of the (i.e., NPF,

CPU, MSI, and SI) indexes by using the *t*-test, respectively. Suppose the *p*-value is less than 0.05 (note that the considered confidence level is 0.95). In that case, there is a significant difference in the averages of the proposed algorithms in that particular index.

NPF index: Because the related *p*-value is bigger than 0.05, the related null hypothesis is accepted. There is no superiority of each algorithm on another related to the NPF index. Table 19 exhibits the results in detail.

Table 19

Survey on the meaningful mean of the number of Pareto front solution index related to the test problem no. 1

Algorithm	Average	Standard deviation	Average standard error	T-value	P-value
NSGA- II	11.1	2.42	0.77	0.18	0.862
MOPSO	10.9	2.64	0.84		

CPU time index: Simply, because the related *p*-value is less than 0.05, the related null hypothesis is rejected. As a result, the NSGA-II has a longer computational time. Table 20 demonstrates the results in detail.

Table 20

Survey on the meaningful mean of computational time index related to the test problem no. 1

Algorithm	Average	Standard deviation	Average standard error	T-value	P-value
NSGA-II	167.35	4.4	1.4	2.35	0.033
MOPSO	163.49	2.76	0.87		

MSI index: Principally, because the related *p*-value is bigger than 0.05, the related null hypothesis is accepted, and there exists no advantage of each algorithm on another related to the MSI index. Table 21 exhibits the results in detail.

Table 21

Survey on the meaningful mean of the maximum spreading index in relation to the test problem no. 1

Algorithm	Average	Standard deviation	Average standard error	T-value	P-value
NSGA-II	21800	7397	2339	1.05	0.309
MOPSO	18076	8434	2667		

SI index: Primarily, since the related *p*-value is bigger than 0.05, the related null hypothesis is accepted, and there exists no benefit of each algorithm on another related to the MSI index. Table 22 displays the results in detail.

Table 22

Survey on the meaningful mean of the spacing metric index regarding test problem no. 1

Algorithm	Average	Standard deviation	Average standard error	T-value	P-value
NSGA-II	0.604	0.163	0.052	0.49	0.632
MOPSO	0.566	0.184	0.058		

As a whole, according to Tables 19 to 22, there is a significant difference among the proposed meta-heuristic algorithms by considering the *p*-value only in the computational time averages.

6.10.2. Using the TOPSIS method

Briefly, the TOPSIS method considers some criteria and alternatives. In our survey, alternatives are the NSGA-II and MOPSO algorithm, and criteria are CPU time, NPF, MSI, and SI. The mechanism of this approach is that for each criterion, the positive and negative ideal alternatives are considered. Ideally, the main aim is to be closer as much as possible to the positive ideal and farther as much as possible to the negative ideal alternative. The keynote is that the interaction of other criteria and alternatives generally does not allow the ideal occurrence to take place for each alternative. Fig. 57 shows this mechanism.

Fig. 57. Comparing alternatives in the TOPSIS method

However, as a whole, this approach presents acceptable choices. For further information about the detailed mechanism of this approach, please refer to the (Munda, 2005). In our survey, after comparing each instance problem in different sizes, it was observed that the MOPSO algorithm in small sizes (1 and 2) and the NSGA-II in medium and large sizes (3, 4, and 5) had better performance for solving the model. Therefore, the TOPSIS method is used to determine the best and most efficient algorithm by considering all sample issues. Table 23 shows the results of comparing the indices of the NSGA-II and MOPSO. It is concluded that the NSGA-II is chosen as the most efficient algorithm for this proposed model.

Table 23

Comparison of meta-heuristic Algorithms using TOPSIS approach

Index	CPU-time	NPI	MSI	SMI
Weight(entropy)	0.9522	0.0331	0.0041	0.0106
	NSGA-II 0.9935	>		MOPSO 0.0065

7 Managerial and decision-support insights

In this research, one aspect of queuing theory Jackson model is investigated for the first time regarding the supply chain problem. Because of complexity, two multi-objective meta-heuristic algorithms are used. Computation results affirm that two objective functions conflict (i.e., upward convexity of Pareto front). For the first test problem (smallest size), computational results assure that all mathematical constraints are satisfied.

Due to the numerous indices for the other test problems, only non-dominated Pareto front-defined indices are observed for the decision-makers. Five test problems are investigated, and then 10 replicates for each are solved. Two algorithms report reasonable solutions at the appropriate time. For each Pareto front index, trend figures and box plots (statistical analysis) are drawn and interpreted separately. Finally, to take a decisive decision, the TOPSIS method depicts that the NSGA-II outperforms MOPSO regarding five indices. This method can help managers effectively utilize resources. Also, the decision-makers and managers can use the hypothesis test to find out that the MOPSO algorithm is more efficient in small-sized problems; however, the NSGA-II algorithm is more efficient in medium- and large-sized problems. They can use the actual data taken from a case study for further investigation, which assures our contribution.

8 Conclusion and further research remarks

The study was a follow-up of the conducted research in the closed-loop supply chain (CLSC) and facilities location. By considering the Jackson queueing models in manufacturing centers and a waiting queue in distribution centers, it aimed to design a multi-product, multi-level CLSC network with two objectives including supplier, producer, distributor/collector, reproducer, and disposal center as well as uncertain parameters (e.g., shipping cost and amount of demand). First, a mixed-integer nonlinear model of the CLSC network design was designed, and then Jackson's queuing models were added. Finally, after completing the model, the trapezoidal fuzzy membership function was considered for the mentioned fuzzy parameter.

In the analyzing and evaluation section, the proposed model was analyzed using the proposed meta-heuristic algorithms, namely NSGA-II and MOPSO. First, a numerical example was developed and analyzed using two proposed algorithms. Then, the results of the two solutions obtained by these algorithms were reported (i.e., the first solution of the left-hand side Pareto front) in the tables in detail. Next, the model was solved in larger sizes. For this purpose, five different sizes were designed, and from each size, the 10-instance problem was generated and solved using the NSGA-II and MOPSO algorithms. Five evaluating Pareto front indices were used to compare these aforementioned algorithms. These indices were first analyzed in general. We draw numerous figures representing trends and statistical boxplots regarding each index to show the impact of the size of problems on the solutions. Using the hypothesis test, the MOPSO algorithm was efficient in small sizes, and the NSGA-II algorithm was more efficient in medium and large sizes. Finally, to select the best algorithm out of these two algorithms, the TOPSIS method was used, which resulted

in selecting the NSGA-II algorithm as better than the MOPSO algorithm for solving a bi-objective CLSC. After conducting this research, some suggestions are suggested to improve the proposed model and solving methods in future research as follows:

- Providing other priority-based encodings.
- Remodeling so that boundaries of all indices are predefined and all of the relation to be linear to solve by commercial software (e.g., GAMS and CPLEX) in a reasonable time.
- Using other algorithms involving heuristic, meta-heuristic, and exact ones in solving the related model.
- Using other queuing systems in modeling.
- Adding the green objective function to the considered model because of the importance of this issue in recent years.
- Integrating the two-objective model into a single-objective model and solving it by multi-objective decision-making methods (e.g., Lp-metric, goal attainment, and goal programming).

Compliance with ethical standards

- **Conflict of interest.** The authors declare that there is no potential conflict of interest regarding the publication of this paper.
- **Ethical approval.** The authors certify that they have no affiliation with or involvement with human participants or animals performed by any of the authors in any organization or entity with any financial or non-financial interest in the subject matter or materials discussed in this paper.

References

- Amin SH, Zhang G (2013) A multi-objective facility location model for closed-loop supply chain network under uncertain demand and return. *Appl Math Model* 37:4165–4176.
- Balzary J, Mohais A (2018) Consideration for multi-objective metaheuristic optimisation of large Iron ore and coal supply chains, from resource to market, in: *Advances in Applied Strategic Mine Planning*. Springer International Publishing, pp. 297–318.
- Beheshtinia MA, Ghasemi A (2018) A multi-objective and integrated model for supply chain scheduling optimization in a multi-site manufacturing system. *Eng. Optim.* 50, 1415–1433.
- Cheraghalipour A, Paydar MM, Hajiaghaei-Keshteli M (2018) A bi-objective optimization for citrus closed-loop supply chain using Pareto-based algorithms. *Appl Soft Comput J* 69:33–59.
- Dai Z (2016) Multi-objective fuzzy design of closed-loop supply chain network considering risks and environmental impact. *Hum Ecol Risk Assess* 22:845–873.
- Darestani SA, Hemmati M (2019) Robust optimization of a bi-objective closed-loop supply chain network for perishable goods considering queue system. *Comput Ind Eng* 136:277–292.
- Deng Y, Jiang M, Ling C (2020) An improved diffusion model for supply chain emergency in uncertain environment. *Soft Comput* 24:6385–6394.
- Fakhrzad MB, Goodarzian F (2019) A Fuzzy Multi-Objective Programming Approach to Develop a Green Closed-Loop Supply Chain Network Design Problem under Uncertainty: Modifications of Imperialist Competitive Algorithm. *RAIRO - Oper Res* 53:963–990.
- Fathian M, Jouzdani J, Heydari M, Makui A (2018) Location and transportation planning in supply chains under uncertainty and congestion by using an improved electromagnetism-like algorithm. *J Intell Manuf* 29:1447–1464.

- Fathi M, Khakifirooz M, Diabat A, Chen H (2021) An integrated queuing-stochastic optimization hybrid Genetic Algorithm for a location-inventory supply chain network. *Int J Prod Econ* 237:108139.
- Gen M, Altıparmak F, Lin L (2006) A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spectr* 28:337–354.
- Gholami F, Paydar MM, Hajiaghaei-Keshteli M, Cheraghalipour A (2019) A multi-objective robust supply chain design considering reliability. *J Ind Prod Eng* 36:385–400.
- Gholamian N, Mahdavi I, Tavakkoli-Moghaddam R (2016) Multi-objective multi-product multi-site aggregate production planning in a supply chain under uncertainty: Fuzzy multi-objective optimisation. *Int J Comput Integr Manuf* 29:149–165.
- Goodarzian F, Hosseini-Nasab H (2021) Applying a fuzzy multi-objective model for a production–distribution network design problem by using a novel self-adoptive evolutionary algorithm. *Int J Syst Sci Oper Logist* 8:1–22.
- Goodarzian F, Hosseini-Nasab H, Muñuzuri J, Fakhrazad MB (2020) A multi-objective pharmaceutical supply chain network based on a robust fuzzy model: A comparison of meta-heuristics. *Appl Soft Comput J* 92:106331.
- Hu L, Zhu JX, Wang Y, Lee LH (2018) Joint design of fleet size, hub locations, and hub capacities for third-party logistics networks with road congestion constraints. *Transp Res Part E Logist Transp Rev* 118:568–588.
- Inuiguchi M, Ramík J (2000) Possibilistic linear programming: A brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem. *Fuzzy Sets Syst* 111:3–28.
- Jain S, Raghavan NRS (2009) A queuing approach for inventory planning with batch ordering in multi-echelon supply chains. *Cent Eur J Oper Res* 17:95–110.
- Kayvanfar V, Hussein SMM, Karimi B, Sajadieh MS (2017) Bi-objective intelligent water drops algorithm to a practical multi-echelon supply chain optimization problem. *J Manuf Syst* 44:93–114.
- Kumar RS, Choudhary A, Babu SAKI, et al (2017) Designing multi-period supply chain network considering risk and emission: a multi-objective approach. *Ann Oper Res* 250:427–461.
- Kusolpuchong S, Chusap K, Alhawari O, Suer G (2019) A Genetic Algorithm Approach for Multi Objective Cross Dock Scheduling in Supply Chains. *Procedia Manuf* 39:1139–1148.
- Li L, Liu Y, Song Y (2019) Factors affecting bike-sharing behaviour in Beijing: price, traffic congestion, and supply chain. *Ann Oper Res*.
- Liu C, Kou G, Zhou X, et al (2020) Time-dependent vehicle routing problem with time windows of city logistics with a congestion avoidance approach. *Knowledge-Based Syst* 188:.
- Liu Y, Dehghani E, Jabalameli MS, et al (2020) A coordinated location-inventory problem with supply disruptions: A two-phase queuing theory–optimization model approach. *Comput Ind Eng* 142:.
- Mahmoodi M (2019) A new multi-objective model of agile supply chain network design considering transportation limits. *Prod Manuf Res* 7:1–22.
- Mardan E, Govindan K, Mina H, Gholami-Zanjani SM (2019) An accelerated benders decomposition algorithm for a bi-objective green closed loop supply chain network design problem. *J Clean Prod* 235:1499–1514.
- Marianov V, Serra D (2003) Location models for airline hubs behaving as M/D/c queues. *Comput Oper Res* 30:983–1003.
- Marufuzzaman M, Eksioğlu SD (2017) Managing congestion in supply chains via dynamic freight routing: An application in the biomass supply chain. *Transp Res Part E Logist Transp Rev* 99:54–76.

- Mohammed AM, Duffuaa SO (2020) A tabu search based algorithm for the optimal design of multi-objective multi-product supply chain networks. *Expert Syst Appl* 140:.
- Mohebalizadehgashti F, Zolfagharinia H, Amin SH (2020) Designing a green meat supply chain network: A multi-objective approach. *Int J Prod Econ* 219:312–327.
- Mohtashami Z, Aghsami A, Jolai F (2020) A green closed loop supply chain design using queuing system for reducing environmental impact and energy consumption. *J Clean Prod* 242:.
- Munda G (2005) Multiple criteria decision analysis and sustainable development. In: *International Series in Operations Research and Management Science*. Springer New York LLC, pp 953–986
- Nazari-Ghanbarloo V, Ghodratinama A (2019) Optimizing a robust tri-objective multi-period reliable supply chain network considering queuing system and operational and disruption risks. *Oper Res*.
- Niu B, Tan L, Liu J, et al (2019) Cooperative bacterial foraging optimization method for multi-objective multi-echelon supply chain optimization problem. *Swarm Evol Comput* 49:87–101.
- Nobil AH, Niaki STA, Cárdenas-Barrón LE (2017) Some Observations on “Location and Allocation Decisions for Multi-echelon Supply Chain Network: A Multi-objective Evolutionary Approach.” *Int J Appl Comput Math* 3:1561–1563.
- Pishvae MS, Razmi J, Torabi SA (2012) Robust possibilistic programming for socially responsible supply chain network design: A new approach. *Fuzzy Sets Syst* 206:1–20.
- Poudel SR, Quddus MA, Marufuzzaman M, et al (2019) Managing congestion in a multi-modal transportation network under biomass supply uncertainty. *Ann Oper Res* 273:739–781.
- Resat HG, Unsal B (2019) A novel multi-objective optimization approach for sustainable supply chain: A case study in packaging industry. *Sustain Prod Consum* 20:29–39.
- Reyes-Sierra M, Coello Coello CA (2006) Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art
- Rioux B, Galkin P, Murphy F, Pierru A (2016) Economic impacts of debottlenecking congestion in the Chinese coal supply chain. *Energy Econ* 60:387–399.
- Robles JO, Almaraz SDL, Azzaro-Pantel C (2016) Optimization of a Hydrogen Supply Chain Network Design by Multi-Objective Genetic Algorithms. In: *Computer Aided Chemical Engineering*. Elsevier B.V., pp 805–810
- Roghianian E, Cheraghalipour A (2019) Addressing a set of meta-heuristics to solve a multi-objective model for closed-loop citrus supply chain considering CO2 emissions. *J Clean Prod* 239:.
- Salkuti SR, Kim SC (2019) Congestion Management Using Multi-Objective Glowworm Swarm Optimization Algorithm. *J Electr Eng Technol*.
- Schott JR (Jason R (1995) Fault tolerant design using single and multicriteria genetic algorithm optimization
- Shortle JF, Thompson JM, Gross D, Harris CM *Fundamentals of queueing theory*
- Singh SK, Goh M (2019) Multi-objective mixed integer programming and an application in a pharmaceutical supply chain. *Int J Prod Res* 57:1214–1237.
- Siswanto N, Kurniawati U, Latiffianti E, et al (2018) A Simulation study of sea transport based fertilizer product considering disruptive supply and congestion problems. *Asian J Shipp Logist* 34:269–278.
- Souza GC (2013) Closed-Loop Supply Chains: A Critical Review, and Future Research*. *Decis Sci* 44:7–38.
- Srinivas N, Deb K (1994) Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evol Comput* 2:221–248.
- Teimoury E, Modarres M, Khondabi IG, Fathi M (2012) A queuing approach for making decisions about order penetration point in multiechelon supply chains. *Int. J. Adv. Manuf. Technol*.

- Vafaeenezhad T, Tavakkoli-Moghaddam R, Cheikhrouhou N (2019) Multi-objective mathematical modeling for sustainable supply chain management in the paper industry. *Comput Ind Eng* 135:1092–1102.
- Wu CH, Yang DY (2021) Bi-objective optimization of a queueing model with two-phase heterogeneous service. *Comput Oper Res* 130: 105230.
- Yang H, Chen W, Hao Y fei (2020) Supply chain partnership, inter-organizational knowledge trading and enterprise innovation performance: the theoretical and empirical research in project-based supply chain. *Soft Comput* 24:6433–6444.
- Yu J, Gan M, Ni S, Chen D (2018) Multi-objective models and real case study for dual-channel FAP supply chain network design with fuzzy information. *J Intell Manuf* 29:389–403.
- Zhang C, Fan LW, Tian YX (2020) Correction to: Optimal operational strategies of capital-constrained supply chain with logistics service and price dependent demand under 3PL financing service (*Soft Computing*, (2020), 24, 4, (2793-2806), 10.1007/s00500-019-04500-7). *Soft Comput.* 24:2807
- Zitzler E, Brockhoff D, Thiele L (2007) The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In: *Lecture Notes in Computer Science* (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*). Springer, Berlin, Heidelberg, pp 862–876

Figures

Figure 1

Proposed model of the closed-loop SCND

Figure 2

Jackson network studied in the proposed model

Supplier	Fixed cost of establishment	Capacity	Customer	Demand	Transportation cost			
					Supplier	Customer		
						1	2	3
1	9500	800	1	300	1	19.57	14.22	16.56
2	10000	750	2	400	2	14.85	19.16	10.34
3	9800	900	3	350	3	18.00	17.92	18.49
4	9600	850			4	11.42	19.59	19.34

$$v(K+J)$$

	Supplier				Customer		
Node	1	2	3	4	1	2	3
Priority	2	5	3	7	4	1	6

$$v(K+J)$$

	Supplier				Customer		
Node	1	2	3	4	1	2	3
Priority	0	5	0	7	4	1	6

Figure 3

Modified priority-based encoding

Figure 4

Mechanism of the NSGA-II

Figure 5

Mechanism of MOPSO

Figure 6

Pareto front formed using NSGA-II and MOPSO (Test problem size 1)

Figure 7

NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 1)

Figure 8

NPF measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 1)

CPU-Time

Figure 9

CPU time measure obtained using NSGA-II and MOPSO (Box Plot). (Test problem size 1)

Figure 10

CPU time incurred using NSGA-II and MOPSO vs. problem no. (Test problem size. 1)

MSI

Figure 11

MSI measure obtained using NSGA-II and MOPSO (Box Plot). (Test problem size 1)

Figure 12

MSI measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 1)

SI

Figure 13

SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size1)

Figure 14

SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 1)

SM

Figure 15

SM measure obtained using NSGA-II and MOPSO (Box Plot). (Test problem size 1)

Figure 16

SM measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 1)

NPF

Figure 17

NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Figure 18

NPF measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 2)

CPU time

Figure 19

CPU-time measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Figure 20

CPU time incurred using the NSGA-II and MOPSO vs. problem no. (Test problem size. 2)

MSI

Figure 21

MSI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Figure 22

MSI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 2)

SI

Figure 23

SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Figure 24

SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 2)

SM

Figure 25

SM measure obtained using NSGA-II and MOPSO (Box Plot). (Test problem size 2)

Figure 26

SM measure obtained using the NSGA-II and MOPSO vs problem no. (Test problem size 2)

NPF

Figure 27

NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Figure 28

NPF measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 3)

Figure 29

CPU time measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Figure 30

CPU time incurred using the NSGA-II and MOPSO vs. problem no. (Test problem size. 3)

MSI

Figure 31

MSI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Figure 32

MSI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 3)

SI

Figure 33

SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Figure 34

SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 3)

SM

Figure 35

SM measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 3)

Figure 36

SM measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 3)

NPF

Figure 37

NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Figure 38

NPF measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size 4)

Figure 39

CPU time measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Figure 40

CPU time incurred using the NSGA-II and MOPSO vs. problem no. (Test problem size. 4)

Figure 41

MSI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Figure 42

MSI measure obtained using NSGA-II and MOPSO vs. problem no. (Test problem size. 4)

SI

Figure 43

SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Figure 44

SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 4)

SM

Figure 45

SM measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 4)

Figure 46

SM measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 4)

NPF

Figure 47

NPF measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Figure 48

NPF measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 5)

CPU time

Figure 49

CPU time measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Figure 50

CPU time incurred using the NSGA-II and MOPSO vs. problem no. (Test problem size. 5)

MSI

Figure 51

MSI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Figure 52

MSI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 5)

SI

Figure 53

SI measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Figure 54

SI measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size. 5)

SM

Figure 55

SM measure obtained using the NSGA-II and MOPSO (Box Plot). (Test problem size 5)

Figure 56

SM measure obtained using the NSGA-II and MOPSO vs. problem no. (Test problem size 5)

Figure 57

Comparing alternatives in the TOPSIS method