RAIRO-Oper. Res. 52 (2018) 1147–1173
https://doi.org/10.1051/ro/2018012

RAIRO Operations Research
www.rairo-ro.org

A HYBRID MARKOV PROCESS-MATHEMATICAL PROGRAMMING APPROACH FOR JOINT LOCATION-INVENTORY PROBLEM UNDER SUPPLY DISRUPTIONS

Ehsan Dehghani, Mir Saman Pishvaee* and Mohammad Saeed Jabalameli

Abstract. This paper introduces a joint location-inventory problem, in which facilities become temporarily unavailable. A hybrid approach based on the Markov process and mathematical programming techniques is presented to design the distribution network of a supply chain in an integrated manner. In the first phase, the Markov process derives some performance features of inventory policy. In the second phase, using outputs of the Markov process, the location-inventory problem is formulated as a mixed-integer nonlinear programming model. Moreover, a robust possibilistic programming approach is utilized, which is able to provide a more stable supply chain structure under almost all possible values of imprecise parameters. Since the proposed problem is complicated to solve by means of exact methods, we develop a simulated annealing algorithm in order to find near-optimal solutions in reasonable computational times. The obtained computational results reveal the efficiency and effectiveness of the proposed solution approach. Finally, some insights are provided and the performance of the proposed robust optimization approach is compared to traditional possibilistic chance constrained method.

Mathematics Subject Classification. 90B80, 90B05, 90C40, 90C70

Received 31 May 2017. Accepted 31 January 2018.

1. Introduction

Logistics is the process of planning, executing, and controlling procedures for the efficient and effective transportation and storage of goods between origin and consumption points in order to meet the demands of customers [1]. Historically, companies have administered the distribution and storage of products in a disparate way within different functional departments. However, decisions of supply chain are interrelated to each other and managers are well informed nowadays that optimizing the logistics system as a whole is an urgent need [2]. For instance, inventory management needs the efficient location of distribution centers (DCs) and the optimal amount of storage at these centers. An efficient transportation plan also depends on the location of DCs. Generally speaking, integrating decisions across the supply chain makes a significant cost savings and considerably affects the customer's consent that can provide a great benefit to the company in today's increasingly competitive markets [2]. One of the important integration issues in supply chain is the location-inventory problem

Keywords and phrases: Facility location, inventory control, disruption, Markov process, robust possibilistic programming.

School of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran.

* Corresponding author: pishvaee@iust.ac.ir

that incorporates decisions about stocks into facility location and determines location, allocation, and inventory decisions simultaneously. This problem can be implemented in many businesses, where inventory management plays a significant role in the distribution and production systems [3, 4].

The dynamic and complex nature of supply chain imposes a wide range of uncertainties, which considerably influences the overall performance of supply chain network [5]. In this regard, a common event that exists in the real environment is the disruption possibility of a facility. Traditional studies in the location-inventory problem generally consider that the constructed facilities are everlasting and do not fail under any circumstances. However, in practice, facilities are subject to potential operational disruptions due to reasons such as maintenance, equipment breakdown, repair, and inclement weather. The failure of a facility will cause customers either leave service and entail a penalty or travel longer distances to obtain service from another facility. Thus, system's operational cost increases and customer satisfaction deteriorates [6]. In some real situations, especially for strategic level decisions, uncertainty is also associated with lack of knowledge about the exact value of some ill-defined and imprecise data [7]. In such situations, using stochastic programming methods to deal with the uncertainty may be impossible since they need reliable historical data. Moreover, applying stochastic programming models significantly increase the computational complexity of the problem. In this sense, fuzzy mathematical programming methods that can use both objective and subjective data are the most suitable tools to handle the ambiguity of parameters [8, 9]. Furthermore, in order to immunize the optimal solution for almost all possible realizations of uncertain parameters and make a more stable supply chain structure, considering possible variations existing in parameters over a long-term horizon is critical [10, 11]. This leads to the need to find a robust solution, which considers feasibility robustness and optimality robustness, simultaneously. Indeed, robust optimization approach aims to provide the preferred risk aversion of the decision makers and find a solution which is less sensitive *versus* changing uncertain parameters [5, 12].

Motivated by the above discussions, the current paper proposes a joint location-inventory problem to design the distribution network of a supply chain under random supply disruptions. It is presumed that facilities change intermittently between available and unavailable states. The time period length of each available and unavailable state is uncertain. An integrated hybrid approach based on the Markov process and mathematical programming techniques is presented to determine decisions across the supply chain simultaneously. In the first phase, some performance features of inventory policy, *i.e.*, the number of reorders, the number of shortages, and the mean inventory level are derived using the Markov process. Then, based on the outputs of the Markov process, the location-inventory model is constructed to determine the following decisions: (4.1) the number of facilities to be located; (4.2) the location of facilities; (4.3) the assignment of retailers to opened facilities; and (4.4) the optimal inventory policy at each established facility. To benefit from both fuzzy and robust programming approaches, a robust possibilistic programming approach is applied, which is able to make supply chain configuration and total costs stable when facing uncertainties caused by imprecise data. The proposed problem belongs to the class of NP-hard problems, since it is an extension of the capacitated facility location problem (CFLP), which is a well known NP-hard problem [13]. Thus, in order to find high-quality solutions in reasonable times, a simulated annealing (SA) based meta-heuristic algorithm is developed.

The remainder of this paper is structured as follows. In Section 2, the related literatures towards the location-inventory, disruption, and queuing-inventory models are reviewed. Section 3 states the assumptions and objectives of the concerned problem. In Section 4, the hybrid approach is presented to formulate the problem. A robust possibilistic programming approach is implemented in Section 5. Section 6 provides an efficient solution procedure for solving the proposed location-inventory model and the computational results and insights are reported in Section 7. Finally, the concluding remarks and possible future research directions are given in Section 9.

2. Literature review

In this section, we first review the literature of location-inventory, facility disruption, and queuing system with inventory and then introduce the literature gaps, which this study addresses.

2.1. Location-inventory

Traditionally, the decisions at different levels of supply chain have been taken into account separately that yield in sub-optimal design [14]. Hence, most of the efforts have tended to incorporate them across the supply chain [3]. Likewise, given the complex nature of supply-chain integration problems, a number of scholars have attempted to develop solution methods such as Lagrangian relaxation and meta-heuristic algorithms for solving them. An early contribution to the joint location-inventory problem was provided by Baumol and Wolfe [15], who introduced the idea of integrating inventory costs into location models. They developed the uncapacitated facility location problem and proposed a method that could obtain a local optimum. Barahona and Jensen [16] proposed a joint location-inventory model and formulated it as an integer programing model. They used a Dantzig–Wolfe decomposition to solve the model and utilized a sub gradient optimization method to accelerate its convergence. Daskin *et al.* [17] presented a facility location problem and incorporated safety stock inventory and working inventory costs in their model. A stochastic three-level supply was presented by Javid and Azad [18], where a heuristic method based on a hybrid Tabu Search and SA algorithm was applied to tackle the large instances of the problem. Jin [19] presented a location-inventory problem and developed a solution procedure based on the Lagrangian relaxation method for solving it in an efficient way. Mousavi *et al.* [20] extended a seasonal multiple-product location-inventory problem and proposed two meta-heuristic algorithms (*i.e.*, particle swarm optimization and SA) to solve the problem. Nekooghadirli *et al.* [21] devised a novel bi-objective location-routing-inventory problem and developed four efficient meta-heuristic algorithms for their model. Diabat and Theodorou [14] investigated a two-echelon inventory management problem and applied a piecewise linearization to efficiently solve the problem. A mixed integer programming model was devised by Rabbani *et al.* [22] to investigate the effect of the lease contract on inventory and pricing decisions. Sadjadi *et al.* [23] examined a stochastic location-inventory problem and considered an $(S-1, S)$ inventory policy for opened DCs. They applied a queuing approach to obtain some characteristics of inventory policy and then formulated the problem using the acquired results. Jindal and Solanki [24] developed a vendor-buyer inventory model considering inflation and time value of money. The objective of their model was determining the order quantity, number of lots, and safety factor in a way to minimize the total costs. Liao *et al.* [25] devised a multi-objective dual-channel supply chain network model and proposed a heuristic solution approach to solve it. Implementing neural network and evolutionary algorithms, AmalNick and Qorbanian [26] optimized different pricing policies under demand uncertainty. Díaz-Mateus *et al.* [27] presented a non-linear optimization model for a two-echelon supply chain and developed a meta-heuristic algorithm based on particle swarm optimization to solve their model.

2.2. Facility disruption

Classic facility location researches often suppose that a facility, once built, will remain functioning forever during its lifetime. However, facilities may be subject to operational disruptions from time to time in many real-world problems [28]. Meanwhile, most of studies have addressed this issue in the basic facility location problems and the body of location-inventory literature is very thin in this part. One of the first studies in this problem was presented by Drezner [29]. He assumed that facilities become unavailable with a known probability in the P-median and P center problems. Furthermore, he provided a heuristic procedure for the problem. Lee [30] presented the formal P-median location problems by taking disruption of facilities into account and minimized the transportation costs between the customers and available facilities. Babazadeh *et al.* [31] addressed facility disruptions in the P-median and uncapacitated fixed-charge location problems. They assumed that if the primary facility interrupts, the customer is assigned to backup facilities. Lee and Chang [32] provided discrete location problems, where facilities were subject to failure.

A stochastic supply chain design problem in presence of random disruptions was presented by Aryanezhad *et al.* [16, 33]. They supposed that safety stocks are held in each opened DC to provide appropriate service levels for customers. As such, an effective solution procedure based on genetic algorithm was applied to solve the model. Chen *et al.* [28] presented a location-inventory problem regarding disruptions and assumed that facilities may fail independently with an equal probability. Last but not the least, Zhang *et al.* [34] devised a discrete competitive

Table 1. Classification of location-inventory papers.

Reference	Year	Disruption	Inventory policy			Type of shortage		Uncertainty				Solution approach		
			(R,T)	$(S.Q)$	$(S-1,S)$	Backlogged	Lost sale	Demand	Lead-Time	Cost	Capacity	Lagrangian relaxation	Commercial software	Meta-heuristic algorithm
[17]	2002			*				*				*		
[41]	2003			*		*								
[42]	2005			*		*		*				*		
[43]	2007			*				*				*		
[2]	2008			*		*		*				*		
[18]	2010			*		*		*						*
[28]	2011	*		*		*						*		
[44]	2012		*			*		*				*		
[21]	2014			*				*						*
[3]	2015			*		*						*		
[45]	2015	*		*		*		*						*
[23]	2016				*	*		*	*				*	
Our work	-	*		*			*	*	*	*	*			*

facility location problem under facility failure patterns and developed a variable neighborhood decomposition search heuristic to solve their problem.

2.3. Queuing system with inventory

In many systems such as production or inventory systems, satisfying demands requires on-hand inventory and a service or process that takes some times. The important purpose in these systems is the reaction of inventory management to queuing of demands. The queuing approach is a powerful tool for describing these behaviors, which contributes to more realistic models [35]. Several papers can be found in literature, which address this subject. In this regard, Parlar [36] proposed a continuous-review inventory policy considering supplier disruptions and formulated the problem as a semi-Markov process. Mohebbi [37] developed an analytical model for a continuous inventory policy, where the demand and lead-time followed Poisson and Erlang distributions, respectively, and stockouts were lost. Schwarz *et al.* [38] obtained stationary distributions of joint queue length and inventory processes in explicit product form for different M/M/1-systems. Teimoury *et al.* [39] applied a queuing approach for production-inventory planning and presumed that lead-time is uncertain and follows an exponential distribution. Using an M/G/1 queuing system, Garg *et al.* [40] developed a multi-item single stage production-inventory. They obtained optimal production frequencies and proposed an approximate convex program for calculating the costs.

Table 1 summarizes the main characteristics of the some location-inventory papers and compares them with the model developed in this study. According to the literature review and Table 1, modeling efforts that incorporate both inventory decisions and random supply disruptions in context of facility location problem are very scarce. In addition, these studies mainly assume that the disruptions occur with a known probability. Also, nearly all location-inventory studies consider backorder for unsatisfied demands, while lost sales shortages are also applicable in many industries such as consumable products and spare parts (see, Schwarz *et al.* [38] and Teimoury *et al.* [39]). They typically do not consider the lead-time or assume that it is deterministic and overlook the imprecise nature of parameters such as costs and capacities. These restricted assumptions are not efficient to tackle real world problems and can result in inaccurate outputs.

To overcome the literature gaps, this paper contributes to the area of joint location-inventory problem in the following ways. This study is able to incorporate random supply disruptions in the problem. That is, opened facilities change intermittently between available and unavailable states that period of each state is uncertain. Considering lost sale shortages is the other issue that distinguishes this study from the ones existed in the literature. The demands of retailers and lead-time are also considered to be uncertain to make the problem more realistic. Meanwhile, to the best of our knowledge, this is the first time in the location-inventory problems that a robust possibilistic program is implemented to cope with the lack of knowledge about the real value of input parameters. Moreover, to solve the model in a reasonable time, an efficient SA-based meta-heuristic algorithm is proposed.

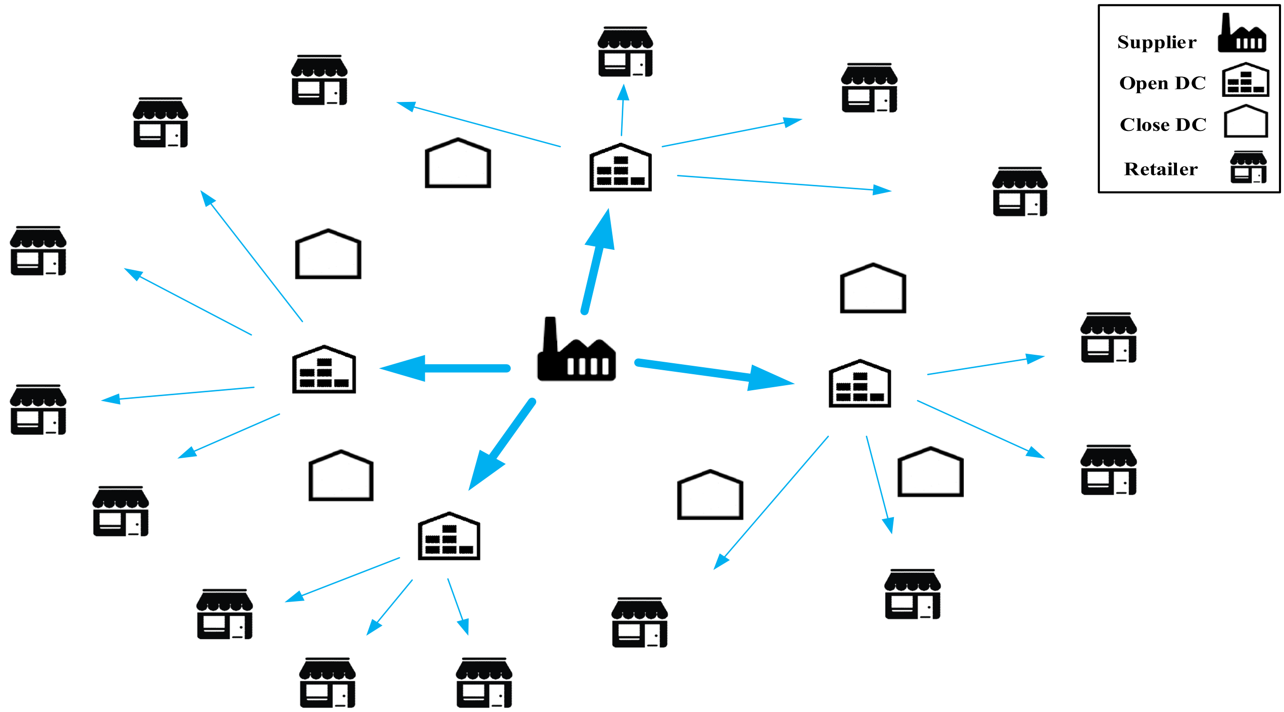

Figure 1. Structure of the concerned supply chain.

3. Problem description

The problem under investigation is to design a three-level supply chain that comprises a supplier, a set of potential DCs, and multiple retailers. The general structure of the concerned supply chain is shown in Figure 1. The decisions to be made are: (4.1) the number of DCs to be located; (4.2) the location of DCs; (4.3) the assignment of retailers to opened DCs; and (4.4) the optimal inventory policy at each established DC. Additionally, the objective function aims at minimizing the total costs of location, transportation, and inventory. It should be pointed out that inventory costs include holding, shortage, ordering, and purchase costs.

Specifically, the problem adopts the following assumptions:

- Each retailer is allocated to an opened DC.
- The demands of retailers independently arrive to DC according to a Poisson distribution and each demand decreases the inventory level for one unit. Hence, by considering the previous assumption, the demands of each opened DC follow a Poisson distribution with intensity λ that is obtained from the sum of its allocated retailers demand rates.
- Each opened DC has a version of continuous-review inventory policy, which is called (S, Q). In this inventory policy, an order with size Q is triggered as soon as the inventory level becomes equal or less than the reorder level S. It is supposed that $S < Q$ to prevent the degeneration of inventory cycles, in which no order is triggered [35, 39]. Besides, inventory management policy follows first come, first served (FCFS) strategy.
- Each opened DC intermittently switches between two states, *i.e.*, available and unavailable states, based on an alternating stochastic process. That is, the available and unavailable times are considered to be uncertain, which are respectively exponentially distributed with parameter α and β (following Parlar

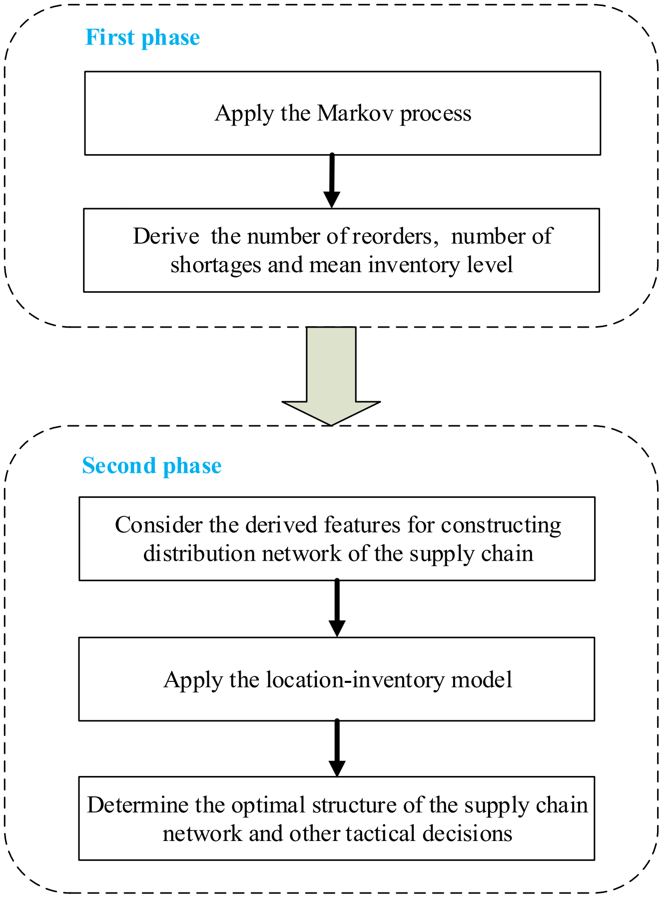

Figure 2. Framework of the proposed hybrid approach.

and Perry [46] and Mohebbi and Hao [47]; with respect to the memoryless property of the exponential distribution and since constant rates of available and unavailable periods, this assumption may be acceptable).
- Events of facility failures are independent.
- The opened DCs work as the direct intermediary facilities between the supplier and retailers. In other words, products are ordered from the opened DCs to the supplier and eventually delivered to the retailers.
- When inventory is depleted or the opened DC becomes unavailable, the arriving demands from its assigned retailers are lost.
- When each opened DC places a replenishment order to the supplier, it arrives after a random time that is exponentially distributed with parameter $\mu > 0$.
- The costs and capacities of the established DCs are tainted with epistemic uncertainty.
- The supplier is not subject to any capacity restrictions, but the storage spaces of DCs are finite.

4. Problem formulation

In this section, a hybrid approach based on the Markov process and mathematical programming techniques is presented to design the distribution network of the concerned supply chain. The framework of the proposed hybrid approach is visualized in Figure 2.

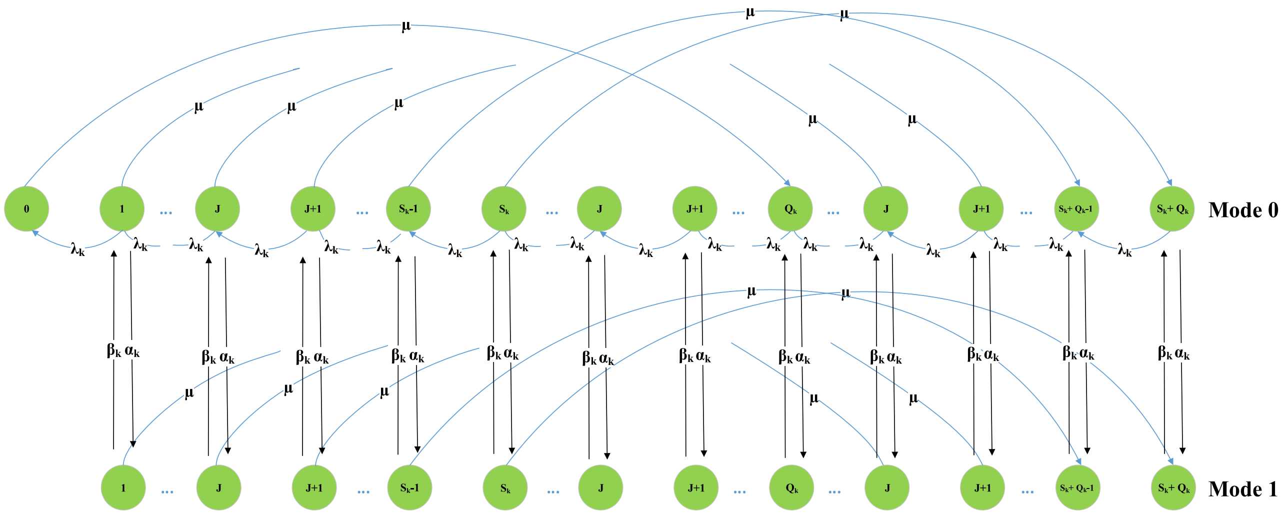

FIGURE 3. Rate diagram of the Markov process.

The Markov process is first applied to derive performance features of inventory policy comprising the number of reorders, the number of shortages, and the mean inventory level. Notably, queuing-inventory systems are more practical and general in contrast to traditional inventory models [35]. For more information about the advantages of queuing-inventory systems, the interested readers can refer to Frizelle [48]. Let us introduce the following notations:

$M_k(t)$	The level of inventory in opened DCs k at time t,
$Y_k(t)$	The state of opened DC k at time t,

where,

$$Y_k(t) = \begin{cases} 1 & \text{If the opened DC } k \text{ at time } t \text{ is unavailable,} \\ 0 & \text{If the opened DC } k \text{ at time } t \text{ is available.} \end{cases}$$

Accordingly, $\{Y_k(t), M_k(t); t \geq 0\}$ is a continuous-time Markov process with state space $E_k = \{(0,n) \mid n = 0, 1, \ldots, S_k + Q_k\} \bigcup \{(1,n) \mid n = 1, 2, \ldots, S_k + Q_k\}$. Moreover, the steady state probabilities of the system are as follows:

$$p_{0k}(j) = \lim_{t\to\infty} \{Y_k(t) = 0, M_k(t) = j\} \qquad j = 0, 1, \ldots, S_k + Q_k, \tag{4.1}$$

$$p_{1k}(j) = \lim_{t\to\infty} \{Y_k(t) = 1, M_k(t) = j\} \qquad j = 1, 2, \ldots, S_k + Q_k. \tag{4.2}$$

The rate diagram of this Markov process is depicted in Figure 3. As it can be seen from Figure 3, when an opened DC is unavailable (*i.e.*, mode 1), the arriving demands cannot be met. In both available and unavailable states, when the level of inventory is less than $S+1$ and an order arrives from the supplier, the level of inventory increases for Q units. In addition, opened DC intermittently switches between available and unavailable states (*i.e.*, mode 0 and mode 1) with parameters α and β. Noteworthy, when inventory is depleted and the opened DC does not meet the arriving demands, the failure cannot be occurred.

The equilibrium equations for this Markov process will be:

$$\mu P_{0k}(0) = \lambda_k P_{0k}(1) \tag{4.3}$$
$$(\lambda_k + \alpha_k + \mu) P_{0k}(J) = \beta_k P_{1k}(J) + \lambda_k P_{0k}(J+1) \quad 1 \le J \le S_k \tag{4.4}$$
$$(\lambda_k + \alpha_k) P_{0k}(J) = \beta_k P_{1k}(J) + \lambda_k P_{0k}(J+1)\; S_k + 1 \le J \le Q_k - 1 \tag{4.5}$$
$$(\lambda_k + \alpha_k) P_{0k}(J) = \beta_k P_{1k}(J) + \lambda_k P_{0k}(J+1) + \mu P_{0k}(J - Q_k)\; Q_k \le J \le Q_k + S_k - 1 \tag{4.6}$$
$$(\lambda_k + \alpha_k) P_{0k}(Q_k + S_k) = \beta_k P_{1k}(Q_k + S_k) + \mu P_{0k}(S_k) \tag{4.7}$$
$$(\beta_k + \mu) P_{1k}(J) = \alpha_k P_{ok}(J)\; 1 \le J \le S_k \tag{4.8}$$
$$\beta_k P_{1k}(J) = \alpha_k P_{0k}(J)\; S_k + 1 \le J \le Q_k \tag{4.9}$$
$$\beta_k P_{1k}(J) = \alpha_k P_{0k}(J) + \mu P_{1k}(J - Q_k) \quad Q_k + 1 \le J \le Q_k + S_k \tag{4.10}$$
$$\beta_k P_{1k}(Q_k + S_k) = \alpha_k P_{0k}(Q_k + S_k) + \mu P_{1k}(S_k), \tag{4.11}$$

where, we have:

$$\sum_{j=0}^{Q_k+S_k} P_{0k}(j) + \sum_{j=1}^{Q_k+S_k} P_{1k}(j) = 1. \tag{4.12}$$

The required system characteristics of inventory policy are acquired as follows:

- *The number of reorders* (RO_k) : each opened DC orders to the supplier when its inventory level is less than S+1. Thus, the number of reorders is obtained by equation (4.13).

$$RO_k = \lambda_k \left[P_{0k}(S_k + 1) + P_{1k}(S_k + 1)\right]. \tag{4.13}$$

- *The number of shortages* (SO_k) : when inventory is depleted or the DC becomes unavailable, the demands arrived from retailers are lost. Therefore, the number of shortages is calculated as below.

$$SO_k = \lambda_k \left[P_{0k}(0) + \sum_{j=1}^{Q_k+S_k} P_{1k}(j)\right]. \tag{4.14}$$

- *The mean inventory level* (MI_k) : the mean inventory level is determined by equation (4.15).

$$MI_k = \sum_{j=1}^{j=Q_k+S_k} j\left[P_{0k}(j) + P_{1k}(j)\right]. \tag{4.15}$$

Now, we introduce the notations used in the proposed mathematical programming model. Note that parameters with tilde on show coefficients tainted with epistemic uncertainty.

Indices:

K Set of potential DCs, indexed by k,

I Set of retailers, indexed by i.

Parameters:

$\tilde{C}_k$ Unit purchase cost of DC k from the supplier($\forall k \in K$),
$\tilde{\pi}_k$ Unit shortage cost at DC $k(\forall k \in K)$,
$\tilde{F}_k$ Fixed (per unit time) cost of locating DC $k(\forall k \in K)$,
$\tilde{T}_{ki}$ Unit transportation cost from DC k to retailer i $(\forall\ k\ \in\ K)(\forall\ i\ \in\ I)$,
$\tilde{A}_k$ Unit ordering cost at DC k $(\forall k \in K)$,
$\tilde{h}_k$ Unit holding cost at DC $k(\forall k \in K)$,
$\tilde{U}_k$ Storage capacity at DC $k(\forall k \in K)$,
$\tilde{\lambda}_k$ Demand rate (Poisson) of retailer i $(\forall i \in I)$,
ϑ Weight factor associated with transportation costs,
θ Weight factor associated with inventory costs.

Decision variables:

z_k 1 if DC k is opened, 0 otherwise $(\forall k \in K)$,
y_k 1 if retailer i is assigned to DC k, 0 otherwise $(\forall\ k\ \in\ K)(\forall\ i\ \in\ I)$,
S_k Reorder level at DC$k(\forall k \in K)$,
Q_k Reorder quantity at DC k $(\forall k \in K)$,
λ_k Demand rate (Poisson) of DC $k(\forall k \in K)$.

The location-inventory problem can be formulated as follows:

$$\text{Min } \tilde{w} = \sum_{k\in \mathrm{K}} \tilde{F}_k z_k + \vartheta \sum_{k\ \in\ K} \sum_{i\ \in\ I} \tilde{T}_{ki} y_{ki} \lambda_i^{'} \left(1 - P_{0k}\left(0\right) - \sum_{j=1}^{Q_k+S_k} P_{1k}\left(j\right) \right) + \theta \sum_{k\in K} z_k \left(\tilde{h}_k MI_k + \tilde{\pi}_k SO_k + \tilde{A}_k RO_k + \tilde{C}_k RO_k Q_k \right). \tag{4.16}$$

Subject to

$$\sum_{k\in K} y_{ki} = 1, \quad \forall i \in I \tag{4.17}$$

$$y_{ki} \le z_k, \quad \forall i \in I, \forall k \in K \tag{4.18}$$

$$\sum_{i\in I} \lambda_i^{'} y_{ki} = \lambda_k\ , \quad \forall k \in K \tag{4.19}$$

$$Q_k + S_k \le \tilde{U}_k\ z_k, \quad \forall k \in K \tag{4.20}$$

$$Q_k \ge S_k + 1, \quad \forall k \in K \tag{4.21}$$

$$y_{ki} \in \{0,1\}\ , \quad \forall i \in I, \forall k \in K \tag{4.22}$$

$$z_k \in \{0,1\}\ , \quad \forall k \in K \tag{4.23}$$

$$S_k \ge 0 \quad Q_k \ge 0 \qquad Integer, \quad \forall k \in K. \tag{4.24}$$

Equations (4.3)–(4.15).

The objective function (4.16) minimizes the total annual costs of supply chain. The first term shows the costs of locating facilities. The second term indicates the transportation costs from opened DCs to retailers. Note that since the shortages are lost, the transportation costs are only computed for the satisfied demands. The third term shows the inventory costs that include the holding, shortage, ordering, and purchase costs.

Constraints (4.17) impose that each retailer is allocated to exactly one DC. Constraints (4.18) ensure that a retailer can only be allocated to DCs that are opened. Constraints (4.19) guarantee that the demand rate of

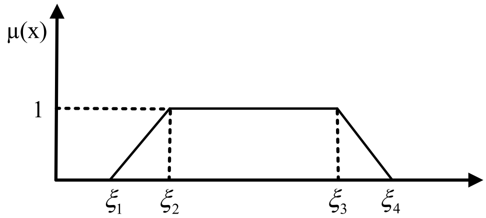

FIGURE 4. Trapezoidal possibility distribution of fuzzy parameter ξ.

each opened DC is equal to the sum of demand rates of its assigned retailers. Constraints (4.20) demonstrate the storage capacities of opened DCs. Constraints (4.21) ensure that if the level of inventory reaches zero, the inventory system does not remain in shortage incessantly. Constraints (4.22) to (4.24) define the decision variables.

5. Robust optimization

In some real situations, especially for strategic-level decisions, there is no sufficient objective/historical data and it is difficult or even impossible/meaningless to approximate probabilistic distributions for them. In such situations, the parameters are tainted with epistemic uncertainty and fuzzy mathematical programming methods are used to deal with the uncertainty.

Overall, fuzzy mathematical programming methods can be categorized into the two main classes comprising the flexible programming and possibilistic programming methods. The methods of the first class typically handle the elasticity in goal values of the objective functions and/or the flexibility of constraints (see, Pishvaee *et al.* [49], Inuiguchi and Ramlk [50]). The methods of the second class cope with ambiguous/imprecise parameters in the objective functions and constraints, which are mainly formulated by possibilistic distributions based on the available objective data and subjective information of the decision maker. Our model can be handled by methods of the second class due to it only includes imprecise parameters [51].

Considering possible variations existing in parameters over a long-term horizon is necessary in order to make the configuration of supply chain stable and provide solution that is less sensitive to the variations in the noisy and uncertain data [11, 52]. This leads to the need to find a robust solution, which under almost all possible realizations of uncertain parameters, it remains feasible (feasibility robustness) and its objective function is close to optimal value (optimality robustness).

For the goal of using the advantages of both fuzzy programming and robust programming, Pishvaee *et al.* [49] introduced a new possibilistic programming approach named robust possibilistic programming (RPP), which has been constructed based on the possibilistic chance constrained programming (PCCP). In their approach, trapezoidal possibility distribution, as a more general form compared to triangular form [9], has been utilized to show the uncertain parameters (see, Fig. 4). Notably, the proposed formulation optimizes feasibility robustness and optimality robustness beside the expected value of possibilistic objective function.

In the following, we describe how the robust method is implemented to create the crisp counterpart of our mathematical programming model. In order to work more convenient, we adopt the proposed model (4.16)–(4.24) with a compact form as follows:

$$\text{Min } w = \tilde{f}z + \tilde{c}x \tag{5.1}$$

$$Ax \le \tilde{U}z \tag{5.2}$$

$$Bx \le 0 \tag{5.3}$$

$$Nz = 1 \tag{5.4}$$

$$z \in \{0,1\}, \quad x \ge 0 \quad \text{Integer,} \tag{5.5}$$

where, the vector $\tilde{f}$ relates to the fixed opening and transportation costs. Besides, $\tilde{c}$ attributes to the inventory costs (*i.e.*, holding, shortage, ordering, and purchase costs) and $\tilde{U}$ denotes storage capacities. Additionally, the matrices A, B, N are coefficient matrices and the vectors z and x represent binary and integer variables, respectively. Here, it is presumed that the vectors $\tilde{f}$, $\tilde{c}$, and $\tilde{U}$ are tainted with epistemic uncertainty. Noteworthy, the necessity measure (Nec), as the most conservative fuzzy measure and at the same time the closest fuzzy measure to certainty [49], is utilized to model the possibilistic chance constraints with imprecise parameters. The fuzzy expected value operator (*i.e.*, E [.]) is also used for adapting the possibilistic objective function into the crisp counterpart. Regarding to the above-mentioned descriptions, the PCCP formulation can be formulated as below:

$$\text{Min } \; E\,[w] = E\,[\tilde{f}]z + E\,[\tilde{c}]x \tag{5.6}$$

$$Bx \le 0 \tag{5.7}$$

$$Nec\left\{Ax \le \tilde{U}z\right\} \ge \alpha \tag{5.8}$$

$$Nz = 1 \tag{5.9}$$

$$z \in \{0,1\}\,, \quad x \ge 0 \quad \text{Integer}, \tag{5.10}$$

where α denotes the minimum confidence level of chance constraint. The crisp counterpart formulation of the aforementioned model (*i.e.*, PCCP model) can be also presented as follows (see, Dubois and Prade [53], Heilpern [54], and Inuiguchi and Ramlk [50]):

$$\text{Min } E\,[w] = \frac{f_1 + f_2 + f_3 + f_4}{4}z + \frac{c_1 + c_2 + c_3 + c_4}{4}x \tag{5.11}$$

$$Bx \le 0 \tag{5.12}$$

$$Ax \le [\alpha U_1 + (1-\alpha)\,U_2]\,z \tag{5.13}$$

$$Nz = 1 \tag{5.14}$$

$$z \in \{0,1\}\,, \quad x \ge 0 \quad \text{Integer}. \tag{5.15}$$

Based on the PCCP model, the RPP model can be formulated as follows:

$$\text{Min } \; obj = E\,[w] + \gamma(w_{\max} - w_{\min}) + \delta\,[\alpha U_1 + (1-\alpha)\,U_2 - U_1]\,z \tag{5.16}$$

$$Bx \le 0 \tag{5.17}$$

$$Ax \le [\alpha U_1 + (1-\alpha)\,U_2]z \tag{5.18}$$

$$Nz = 1 \tag{5.19}$$

$$z \in \{0,1\}\,, \quad x \ge 0 \quad \text{Integer} \quad .5 < \alpha \le 1\,. \tag{5.20}$$

Similar to the PCCP model, the first term of the objective function shows the expected value of w, which measures average total performance of the related system. The second term, $i.e.,\gamma(w_{\max} - w_{\min})$, represents the difference between two extreme possible values of w. In other words, $w_{\max}$ and $w_{\min}$ are obtained as follows:

$$w_{\max} = f_4 z + c_4 x, \tag{5.21}$$

$$w_{\min} = f_1 z + c_1 x. \tag{5.22}$$

Besides, γ indicates the importance of this term against the two other terms in objective function. Indeed, this term aims to measure the optimality robustness of the solution. The third term of the objective function, *i.e.*, $\delta[\alpha U_1 + (1-\alpha)\,U_2 - U_1]$, shows feasibility penalty function, which is applied to penalize violation of the control constraint. In other words, $[\alpha U_1 + (1-\alpha)\,U_2 - U_1]$ illustrates the difference between the value used in chance constraint and the worst case value of imprecise parameter and δ is also the weight of this term in objective

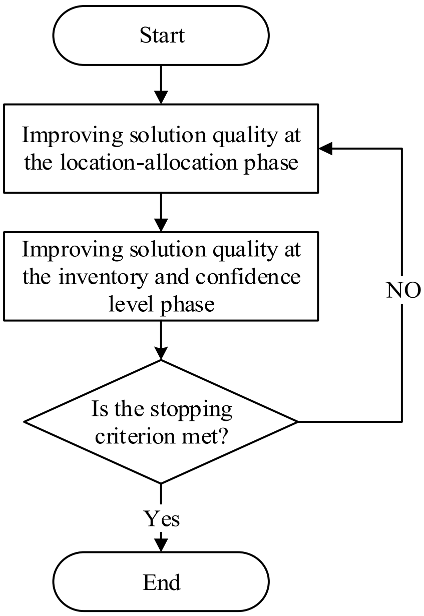

FIGURE 5. Flowchart of the proposed algorithm.

DC	3	2	3	1	…	3	…	3	3	4	2
Retailer	1	2	3	4		i		n-3	n-2	n-1	n

FIGURE 6. Example of location-allocation representation.

function. In contrast to the PCCP model, here, the minimum confidence level of chance constraint (*i.e.*, α) is a decision variable and its value is specified by optimizing the mentioned model. Therefore, RPP model avoids subjective judgment about the best value of α and determines the global optimum value for it. With respect to the aforementioned explanations, the objective function of RPP model aims to identify the trade-off solutions between three terms: (4.1) average performance, (4.2) optimality robustness, and (4.3) feasibility robustness [49].

6. Solution method

The presented problem obviously is an extension of the CFLP that is a well known NP-hard problem [13]. Therefore, we develop an SA algorithm to solve the problem in a reasonable time. This algorithm is one of the powerful meta-heuristic techniques, which have been successfully used to tackle complex models in the literature of joint location-inventory problems (see e.g., Ahmadi Javid and Azad [55] and Nekooghadirli *et al.* [56], etc.). The problem is decomposed into two sub-problems during the algorithm run. First, the location-allocation decisions are determined. Then, using the results of pervious stage, the inventory decisions and the confidence levels are specified. This process continues until the algorithm terminates. The flowchart of this process is illustrated in Figure 5. In this section, we first explain these two sub-problems and then describe the proposed meta-heuristic algorithm in details.

6.1. Location-allocation phase

At this stage, an array $1 \times n$ is created, where n is the number of retailers. Each cell of the array shows which facility supplies the demands of the corresponding retailer. A sample array is depicted in Figure 6, in which the retailers 1, 2, and 3 are allocated to the DCs 3, 2, and 3, respectively.

6.2. Inventory and confidence level phase

Using the array introduced in Section 6.1, the opened DCs and their allocations can be determined. In the second stage, the optimal inventory policy and confidence levels for opened DCs are to be specified. Here, the following model should be optimized for each opened DC (k is index of opened DCs).

$$\text{Min}\, obj_k\left(\alpha_k, S_k, Q_k\right) = \; E\left[w_k^{'}\right] + \gamma\left(w_{k_{\max}}^{'} - w_{k_{\min}}^{'}\right) + \delta\;\left[\alpha_k U_{k_1} + \left(1-\alpha_k\right) U_{k_2} - U_{k_1}\right] \tag{6.1}$$

$$Q_K \geq S_k + 1 \tag{6.2}$$

$$S_k + Q_k \leq \alpha_k U_{k_1} + \left(1-\alpha_k\right) U_{k_2} \tag{6.3}$$

$$S_k, Q_k \geq 0 \quad \text{integer}, \quad .5 < \alpha_k \leq 1, \tag{6.4}$$

where,

$$\tilde{w}_k^{'} = \; \theta\left[\tilde{h}_k MI_k + \tilde{\pi}_k SO_k + \tilde{A}_k RO_k + \tilde{C}_k RO_k Q_k\right] + \vartheta \tilde{g}_k \left(1 - P_{0k}\left(0\right) - \sum_{j=1}^{Q_k+S_k} P_{1k}\left(j\right)\right). \tag{6.5}$$

And,

$$\tilde{g}_k = \sum_{i \in I} \tilde{T}_{ki} y_{ki} \lambda_i. \tag{6.6}$$

For solving the above model, we suggest the following algorithm.

Step 1: Choose the distinguish ability constant ε such that $2 \times \mathcal{E} > 0$, and the allowable final length of uncertainty such that $l > 0$. Let $[\alpha_{k_1}, \alpha_{k_2}]$ be the initial interval of uncertainty (we take it [0.5,1]).
Step 2: Obtain $\Omega_k = \frac{\alpha_{k_1}+\alpha_{k_2}}{2} - \varepsilon$ and $\phi_k = \frac{\alpha_{k_1}+\alpha_{k_2}}{2} + \varepsilon$.
Step 3: Let $S_k = 0$ and $Obj_k^{*1} = \inf$, where *inf* is a big number.
Step 4: Let $Q_k = S_k + 1$.
Step 5: Solve the equilibrium relations and obtain the steady-state probabilities.
Step 6: Obtain the required performance measures.
Step 7: Compute $obj_k(\Omega_k, S_k, Q_k)$.
Step 8: If $Obj_k^{*1} > obj_k(\Omega_k, S_k, Q_k)$, let $Obj_k^{*1} = obj_k(\Omega_k, S_k, Q_k)$.
Step 9: If $Q_k > \Omega_k * U_{k_1} + (1-\Omega_k) * U_{k_2} - S_k$ go to step 10. Otherwise, let $Q_k = Q_k + 1$ and go to step 5.
Step 10: If $S_k > \left[\frac{\Omega_k * U_{k_1} + (1-\Omega_k) * U_{k_2}}{2}\right]$ go to step 11. Otherwise, let $S_k = S_k + 1$ and go to step 4.
Step 11: Let $S_k = 0$ and $Obj_k^{*2} = \inf$, where *inf* is a big number.
Step 12: Let $Q_k = S_k + 1$.
Step 13: Solve the equilibrium relations and obtain the steady-state probabilities.
Step 14: Obtain the required performance measures.
Step 15: Compute $obj_k(\phi_k, S_k, Q_k)$.
Step 16: If $Obj_k^{*2} > obj_k(\phi_k, S_k, Q_k)$ let $Obj_k^{*2} = obj_k(\phi_k, S_k, Q_k)$.
Step 17: If $Q_k > \phi_k * U_{k_1} + (1-\phi_k) * U_{k_2} - S_k$ then go to step 18. Otherwise, let $Q_k = Q_k + 1$ and go to step 13.
Step 18: If $S_k > \left[\frac{\phi_k * U_{k_1} + (1-\phi_k) * U_{k_2}}{2}\right]$, go to step 19. Otherwise, let $S_k = S_k + 1$ and go to step 12.
Step 19: If $Obj_k^{*1} < obj_k^{*2}$, then let $\alpha_{k_2} = \phi_k$. Otherwise let $\alpha_{k_1} = \Omega_k$
Step 20: If $\alpha_{k_2} - \alpha_{k_1} < l$, then stop and let $OW_k^* = \min(Obj_k^{*1},\ Obj_k^{*2})$. Otherwise, go to step 2.

The pseudo code of the algorithm is also illustrated in Figure 7.

Input $\alpha_{k_1} = .5$, $\alpha_{k_2} = 1$, $\mathcal{E}, l$
While α_{k_2}-$\alpha_{k_1} > l$ ***DO***
$\Omega_k = \frac{\alpha_{k_1}+\alpha_{k_2}}{2} - \mathcal{E}$.
$\phi_k = \frac{\alpha_{k_1}+\alpha_{k_2}}{2} + \mathcal{E}$
$S_k = 0$
${Obj^{*1}}_k$=inf (inf means a big number.)
While $S_k \leq \left[\frac{\Omega_k * U_{k_1} + (1-\Omega_k) * U_{k_2}}{2}\right]$ ***DO***
$Q_k = S_k + 1$
While $(Q_k \leq \Omega_k * U_{k_1} + (1-\Omega_k) * U_{k_2}$
Solve the equilibrium relations and obtain the steady-state probabilities
Obtain the required performance measures
Compute $obj_k(\Omega_k, S_k, Q_k)$
If $obj_k(\Omega_k, S_k, Q_k) - {Obj^{*1}}_k < 0$ ***Then***
${Obj^{*1}}_k = obj_k(\Omega_k, S_k, Q_k)$
End if
$Q_k = Q_k + 1$
End while
$S_k = S_k + 1$
End while
$S_k = 0$
${Obj^{*2}}_k$=inf (inf means a big number.)
While $S_k \leq \left[\frac{\phi_k * U_{k_1} + (1-\phi_k) * U_{k_2}}{2}\right]$ ***DO***
$Q_k = S_k + 1$
While $(Q_k \leq \phi_k * U_{k_1} + (1-\phi_k) * U_{k_2} - S_k)$ ***DO***
Solve the equilibrium relations and obtain the steady-state probabilities
Obtain the required performance measures
Compute $obj_k(\phi_k, S_k, Q_k)$
If $obj_k(\phi_k, S_k, Q_k) - {Obj^{*2}}_k < 0$***Then***
${Obj^{*2}}_k = obj_k(\phi_k, S_k, Q_k)$
End if
$Q_k = Q_k + 1$
End while
$S_k = S_k + 1$
End while
If ${Obj^{*1}}_k < {Obj^{*2}}_k$ **Then**
$\alpha_{k_2} = \phi$
Else
$\alpha_{k_1} = \Omega$
End if
$OW_k^* = \min({Obj^{*1}}_k, {Obj^{*2}}_k)$
End while

FIGURE 7. Pseudo code of the proposed algorithm for determining the inventory decision and confidence level.

After we obtain the value of OW_k^* for each opened DC, the value of $TC\,(SO)$ for corresponding array can be calculated as follows:

$$TC\,(SO) = \sum_{k \in A} E\,[\tilde{F}_k] + \gamma\,(F_{k_4} - F_{k_1}) + OW_k^*, \tag{6.7}$$

where A is the set of opened DCs in solution SO.

6.3. The proposed SA algorithm

SA was first introduced by Metropolis *et al.* [57] and later popularized by Kirkpatrick *et al.* [58]. The SA is an appropriate solution method for solving complex problems with large solution spaces and produces results close to the global optimum value in a short period of time [59]. The basic idea behind SA is to permit moves resulting in solutions with worse quality than the current solution (uphill moves) in order to escape from local optimum.

The algorithm typically starts from a randomly created initial solution, and randomly transforms to neighbor solution. If there is an improvement in the objective function ($-\Delta E$), transformation to a new solution is accepted. In addition, the algorithm escapes from a local optimum through accepting not improved solutions with probability $e^{\frac{-\Delta E}{T}}$. Temperature plays an important role in acceptance of not improved solutions, where the probability of acceptance will decrease by decreasing of temperature and vice versa. In addition, by decreasing temperature with low rate at each state, the solution space is searched better.

Now, we describe the implemented SA algorithm. First, we define the parameters of the algorithm as follows:

T	Current temperature;
T_f	final temperature;
T_0	Initial temperature;
$I_{\text{SA}-\text{main}}$	Maximum number of not improved solutions in the outside loop;
$I_{\text{SA}-\text{inner}}$	Maximum number of not improved solutions in the inside loop;
NT_{main}	Counter of not improved solutions in the outside loop;
NT_{inner}	Counter of not improved solutions in the inside loop;
SO	Current solution;
SO_0	Initial solution;
$SO^{'}$	Solution, which is selected in neighborhood SO;
SO_{best}	Best found solution;
$TC(SO)$	Objective function of solution SO;
CR	Cooling rate.

The steps of proposed SA algorithm are as follows:

Step 1: Create the initial solution SO_0 randomly and let $SO = SO_0$, $SO_{\text{best}} = SO_0$, $T = T_0$, $NT_{\text{main}} = 0$, $NT_{\text{inner}} = 0$, and Y=False.
Step 2: Create the solution $SO^{'}$ in neighborhood SO.
Step 3: Obtain ΔE. If $\Delta E \leq 0$, let $SO = SO^{'}$. Otherwise, produce a random number falling in the (0, 1) interval and let it r. If $r < \mathrm{e}^{\frac{-\Delta \mathrm{E}}{\mathrm{T}}}$ then let $SO = SO^{'}$.
Step 4: If $TC(SO) < TC(SO_{\text{best}})$, then let $SO_{\text{best}} = SO$, $NT_{\text{inner}} = 0$, and Y=True. Otherwise, let $NT_{\text{inner}} = NT_{\text{inner}} + 1$.
Step 5: If $NT_{\text{inner}} > I_{\text{SA}-\text{inner}}$, go to step 7. Otherwise, go to step 2.
Step 6: Let $T = CR * T$
Step 7: If Y=True, then $NT_{\text{main}} = 0$. Otherwise, let $NT_{\text{main}} = NT_{\text{main}} + 1$.
Step 8: Let Y=False.
Step 9: If $NT_{\text{main}} > I_{\text{main}}$ or $T < T_f$, then stop. Otherwise, go to step 2.

In Figure 8, the pseudo code of the SA algorithm is given.

In the following, the aspects of the proposed algorithm will be described.

```
Input: T_0, T_f, I_{SA-inner}, I_{SA-main}, CR.
Generate SO_0
S_best = SO_0, SO = SO_0, T = T_0, NT_main = 0
While (NT_main ≤ I_{SA-main} and T ≥ T_f) DO
    NT_inner = 0
    Y = False
        While (NT_inner ≤ I_{SA-inner}) DO
            Generate SO' in the neighborhood of SO
            ΔE = TC(SO') − TC(SO) ;
              If ΔE ≤ 0 Then
                SO = SO'
                  If TC(SO) < TC(SO_best) Then
                      SO_best = SO ; NT_inner = 0;  Y = True
                Else
                    NT_inner = NT_inner + 1
              End if
            Else
          Generate r → uniform (0,1) randomly
              If r < e^(−ΔE/T) Then
                SO = SO'
             End if
   End if
End While
    T = CR * T
      If Y = True then
       NT_main = 0
      Else
     NT_main = NT_main + 1
    End If
End While
Output Best Solution, Objective function, CPU Time.
```

Figure 8. Pseudo code of the proposed SA.

TABLE 2. Distribution of the generated parameters.

h_k	α_k	β_k	A_k	C_k	π_k	F_k	$\lambda_i^{'}$	U_k	μ	T_{ki}	Θ	ϑ
U[25,35]	U[2,3]	U[4,5]	U[5,10]	U[5,10]	U[70,80]	U[5000,6500]	U[75,110]	U[15,20]	U[150,350]	N[4,10]	1	1

6.3.1. Move

Two procedures are utilized to generate a neighbor for solution SO that one of them is randomly selected with the same probability.

1. Two retailers that are supplied by different DCs are selected randomly and then their DCs are exchanged.
2. A retailer is selected randomly and then a new random DC is considered for it.

6.3.2. Temperature reduction function

The temperature is reduced based on the following geometric function.

$$T_{it} = CR \times T_{it-1}. \tag{6.8}$$

6.3.3. Equilibrium condition

At each temperature, when there is no improvement after predetermined number of iterations ($I_{\text{SA-inner}}$), the equilibrium condition is applied.

6.3.4. Stopping criterion

The algorithm is terminated when one of two conditions is satisfied; the current temperature is less than the final temperature T_f or the algorithm is not improved after predetermined number of iterations ($I_{\text{SA-main}}$).

7. Computational results

In this section, we perform wide experiments to evaluate the effectiveness and efficiency of the proposed algorithm. Likewise, we conduct sensitivity analysis and draw some important insights. The algorithm is coded in C++ language and run on a PC with Intel (R) Core (TM) i5-3210M CPU @ 2.50Hz with 4.00GB of RAM. For each problem instance, we run the algorithm five times and the average of results is reported. The parameters of the model are uniformly generated that their ranges are given in Table 2. For each fuzzy parameter, the random generated value is considered for the value of ξ_3 and other prominent values, *i.e.*, ξ_1, ξ_2, and ξ_4, are respectively obtained by multiplying the numbers 0.33, 0.5, and 1.33 by the value of ξ_3.

7.1. Parameter tuning

In this part, we aim to tune the values of algorithm parameters regarding two objectives, which include quality of solution (cost) and CPU time of algorithm. To do so, we apply a procedure, first introduced by Shishebori *et al.* [60], which has been built based on 2^k factorial design. In this procedure, each objective is considered as a response and expressed by a regression model shown by equation (7.1).

$$RE\,(x_1, x_2, \ldots, x_K) = \beta E_0 + \sum_{i=1}^{K} \beta E_i x_i + \sum \sum_{i<j} \beta E_{ij} x_i x_j, \tag{7.1}$$

TABLE 3. Considered factors and their levels for the proposed algorithm.

NO	Factors	Low level	High level
1	T_0	1000	1200
2	T_f	0.01	0.03
3	CR	0.9	0.95
4	$I_{\text{SA}-\text{inner}}$	60	75
5	$I_{\text{SA}-\text{main}}$	15	25

TABLE 4. Regression model coefficients related to solution quality.

β_o	β_1	β_2	β_3	β_4	β_5	β_{12}	β_{13}	β_{14}	β_{15}	β_{23}	β_{24}	β_{25}	β_{34}	β_{35}	β_{45}
104090	−196	587	960	290	181	−408	−407	265	45	446	−91	−54	88	−16	−307
β_{123}	β_{124}	β_{125}	β_{134}	β_{135}	β_{145}	β_{234}	β_{235}	β_{245}	β_{345}	β_{1234}	β_{1235}	β_{1245}	β_{1345}	β_{2345}	β_{12345}
−512	−234	140	188	17	379	−151	−276	−409	−535	−409	−101	98	208	−602	−152

TABLE 5. Regression model coefficients related to CPU Time.

β_o	β_1	β_2	β_3	β_4	β_5	β_{12}	β_{13}	β_{14}	β_{15}	β_{23}	β_{24}	β_{25}	β_{34}	β_{35}	β_{45}
67.14	−0.83	−0.03	9.84	7.42	6.12	−0.85	−0.16	−1.06	0.97	−0.37	−3.06	−1.39	0.6	3.45	1.4
β_{123}	β_{124}	β_{125}	β_{134}	β_{135}	β_{145}	β_{234}	β_{235}	β_{245}	β_{345}	β_{1234}	β_{1235}	β_{1245}	β_{1345}	β_{2345}	β_{12345}
−0.36	0.05	0.72	0.42	−0.74	2.77	−2.29	0.15	−1.56	1.12	−2.55	−0.14	0.59	1.88	−1.77	0.63

where the βEs are regression coefficients and x_i is coded variable of factor i, *i.e.*, if the factor i is at high level, x_i is equal to +1 and if it is at low level, then x_i is equal to −1. Coded variables are often easy to interpret and preferred for comparing engineering units [61]. As shown in Table 3, five factors are taken into account to investigate their impacts on the performances of the algorithm. Each factor is also regarded at two levels and the design is replicated twice. Therefore, 64 experiments are performed while their orders are randomly generated. The coefficients of each response regression model are reported in Tables 4 and 5. Also, Figures 9 and 10 exhibit the normal probability plots of residuals. It is worth noting that residuals are calculated as the estimates of experimental errors obtained by subtracting the observed responses from the predicted responses. From these figures, it seems that the normality assumptions of residuals are agreeable, so we have no reason to be doubtful that there are any problem with the accuracy of the results.

Each objective is independently optimized using general algebraic modeling system (GAMS) -version 24.1- software, where CONOPT solver is implemented to solve it. The results are summarized in Table 6. According to this table, there are some conflicts in the results that we cannot simultaneously adjust a specified constant value for each factor. Therefore, we utilize a multi-objective optimization method to determine the best values of the factors. To this aim, we employ the weighted sum method, as shown in equation (7.2).

$$Z = \sum_{m=1}^{2} WO_m \left| \frac{RE^m - RE^{*m}}{RE^{*m}} \right|, \tag{7.2}$$

where WO_m and RE^{*m} are the weight and the optimal value of the response m, respectively. The multi-objective model is also optimized using GAMS software that the results (*i.e.*, the optimal values of factors) are given in Table 7. It should be pointed out that the values of WO_1 and WO_2 are equal to 0.6 and 0.4, respectively.

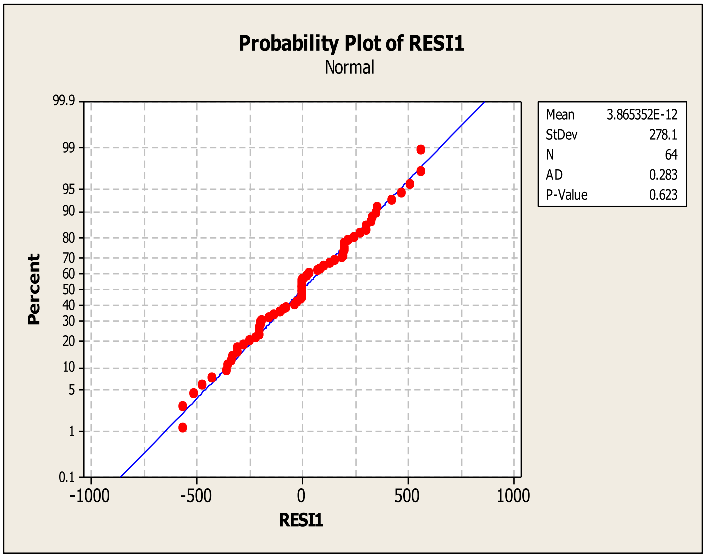

FIGURE 9. Normal probability plot of residuals related to solution quality.

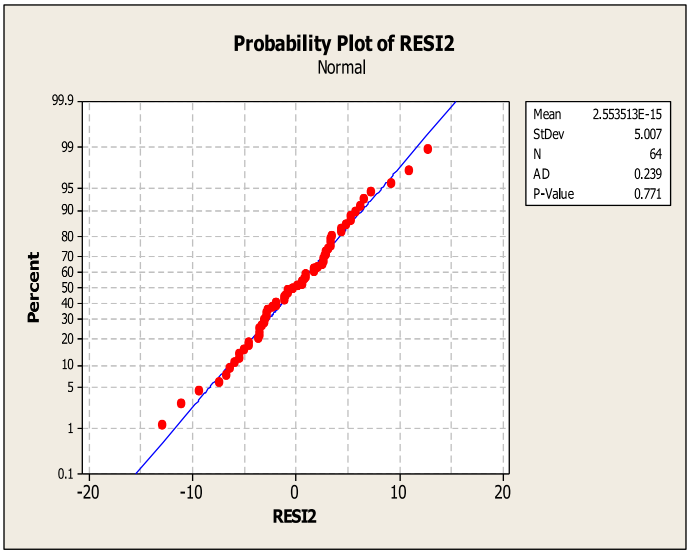

FIGURE 10. Normal probability plot of residuals related to CPU time.

TABLE 6. Best value of the considered factors.

Objectivefunction	X_1^*	X_2^*	X_3^*	X_4^*	X_5^*
Quality	-1	-1	-1	-1	-1
CPU time	-1	-1	$+1$	-1	-1

TABLE 7. Best value of the considered factors.

Objective function	X_1^*	X_2^*	X_3^*	X_4^*	X_5^*
Quality and CPU time	+1	−1	+1	−1	+1

TABLE 8. Comparison of FEM and GAMS software with the proposed algorithm.

No	# Potential DCs	# Retailers	GAMS		FEM		Proposed algorithm			
			Cost ($)	CPU time (sec)	Cost ($)	CPU time (sec)	Cost ($)	CPU time (sec)	GAP1 (%)	GAP2 (%)
1	2	4	50 690.23	6 307.2	50 690.23	0.29	50 690.23	9.71	0	0
2	3	7	78 599.52	9 602.64	78 599.52	48.64	78 599.52	13.42	0	0
3	4	9	105 115.01	10800 Limit	103 388.42	6 252.26	103 388.42	16.97	0	1.64
4	5	10	117 112.52	10800 Limit	114 293.48	209 764.37	114 323.04	48.83	−0.02	2.38
5	6	15	181 420.17	10800 Limit	–	–	175 948.1802	98.003	–	3.01

$\text{GAP}_1(\%) = 100 \times (\text{FEM. Cost} - \text{SA. Cost})\,/\text{FEM.Cost}.$

$\text{GAP}_2(\%) = 100 \times (\text{GAMS. Cost} - \text{SA. Cost})\,/\text{Gams.Cost}.$

7.2. Validating the proposed algorithm

In order to validate the applicability and effectiveness of the proposed algorithm, its performance is first compared with full enumeration method (FEM) and GAMS software. FEM tests all possible solutions for array introduced in Section 6.1 and chooses the best solution. In other words, if the problem includes n retailers and m potential DCs, then all m^n alternative solutions are checked. From Table 8, one can see that the solutions of the proposed algorithm are optimal for instances 1, 2, and 3 and the gap is 0.02% for instance 4. Thus, it is evident that the solutions of the proposed algorithm are optimal or near-optimal. For instance 5, FEM is unable to report solution within 72 hours, while the maximum CPU time of the proposed algorithm is 98.003 seconds. Therefore, with increase in size of the problem, the CPU time of FEM exponentially grows, while it has not significant effect on the CPU time of the proposed solution approach. The results also corroborate that the proposed algorithm has better performances than GAMS software in both terms of the solution quality and CPU time.

The proposed algorithm is also compared with the genetic algorithm in larger instances. The initial solutions and stopping criterion for both algorithms have been accounted to be identical. The results are given in Table 9. What is observed from the results is that in term of CPU time, the proposed algorithm outperforms the genetic algorithm. As such, in term of solution quality, the proposed algorithm mainly has better performances. Overall, it can be resulted that the proposed algorithm performs efficiently.

7.3. Sensitivity analysis

In this section, sensitivity analysis is carried out to investigate the impacts of parameters on the number of opened DCs and the objective function.

TABLE 9. Comparison of the genetic algorithm with the proposed algorithm in larger size instances.

				Genetic algorithm		Proposed algorithm		GAP(%)
Data set	No	# Potential depots	# Retailers	Cost ($)	CPU time (sec)	Cost ($)	CPU time (sec)	
Medium	1	6	15	175 922.3	114.45	175 948.18	98.003	−0.0147
	2	8	20	242 408.1	182.545	242 510.13	152.997	−0.04280
	3	10	25	297 247.5	232.241	297 069.84	194.617	0.0597
	4	12	35	413 632.8	395.776	410 854.63	319.184	0.671
	5	15	45	539 702.7	572.375	533 092.37	414.767	1.2248
Large	6	17	55	647 988.4	786.22	636 255.89	669.877	1.8106
	7	19	70	813 654.9	1 166.143	790 903.81	959.478	2.7961
	8	21	80	922 279.9	1 323.44	901 226.33	1 203.768	2.2827
	9	28	95	1 111 873	1 536.85	1 074 769.8	1 455.63	3.337
	10	35	100	1 154 040	16 451.52	1 123 858.60	1 596.881	2.6152

$$\text{GAP}(\%) = 100 \times (\text{GA.Cost} - \text{SA.Cost})\,/\text{GA.Cost}.$$

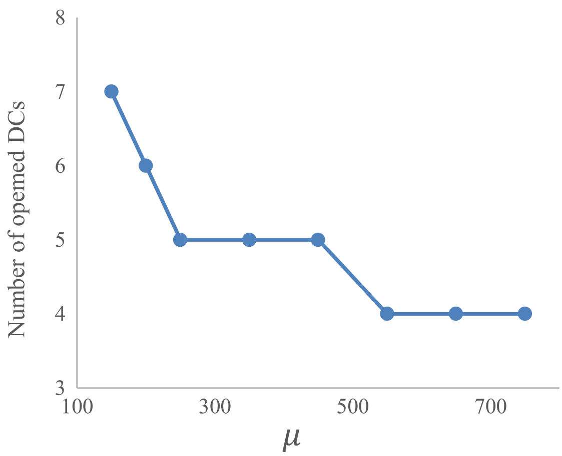

FIGURE 11. Sensitivity analysis of the number of opened DCs with respect to the value of μ.

7.3.1. Number of opened DCs

This sub-section aims to examine the effects of parameters on the number of established DCs. Figure 11 shows as the value of μ gets to be larger, the number of opened DCs decreases. The rationale behind this is that with increase in the value of μ, the orders of opened DCs are supplied sooner and less retailer demands are lost. Therefore, an opened DC can cover more retailers and the supply chain opens less opened DCs to decrease its setup costs. We multiply different integer coefficients by the repair and break rates of potential DCs and then study their impacts on the number of opened DCs. The results are shown in Figure 12. It can be seen that when the repair rates of potential DCs increase, more DCs are opened. On the other hand, as the break rates of potential DCs increase, the number of opened DCs decreases. It is manifested that the supply chain tends to open more strength DCs. Figure 13 compares the number of opened DCs for different instances between the PCCP and RPP models. What is seen form the results is that the number of opened DCs in the RPP model is

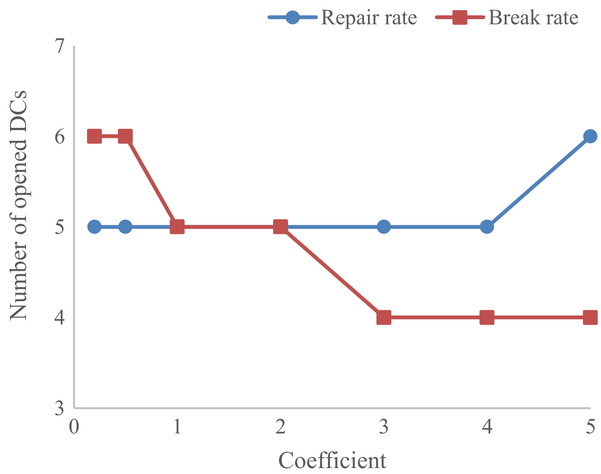

FIGURE 12. Sensitivity analysis of the number of opened DCs with respect to the coefficients multiplied by repair rates and break rates of potential DCs.

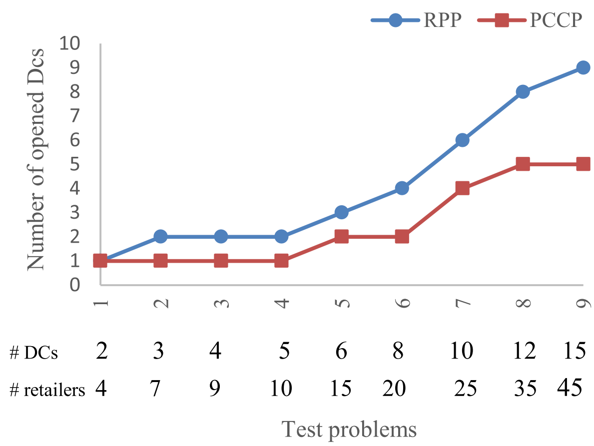

FIGURE 13. Comparing the number of opened DCs between the PCCP and RPP models in different instances.

greater than or equal to the one in the PCCP model. This highlights that the RPP model provides a risk-averse method to propose solution that is less sensitive to the variations in the noisy and uncertain data. Figure 14 displays with increase in optimality robustness importance, the number of opened DCs increases in order to decrease the difference between the two extreme possible values of the objective function and keep it close to optimal value under almost all possible realizations.

7.3.2. Objective function

We now investigate the behavior of the objective function with respect to the value of μ, the repair rates and the break rates of potential DCs. As shown in Figure 15, with increase in the value of μ, the objective function decreases. The intriguing point is that the rate of decrease in the objective function becomes more stabilized towards the larger values of μ. Practically, it can be inferred that if the supplier can decrease its lead-time in small ranges of μ, significant cost saving may be realized, while much improvement cannot be brought in larger values of μ. Figure 16 displays when the strength of potential DCs increases (*i.e.*, the break rates/repair rates

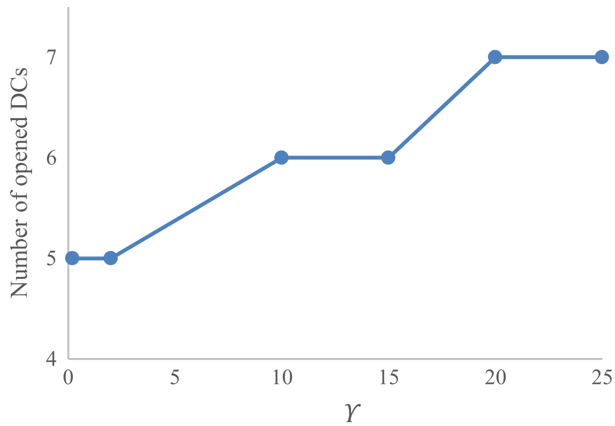

FIGURE 14. Sensitivity analysis of the number of opened DCs with respect to the value of γ.

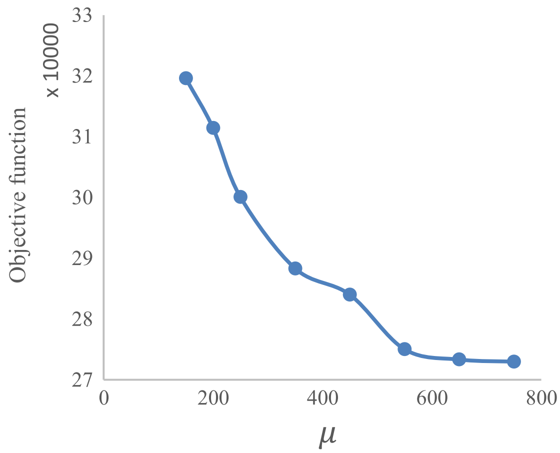

FIGURE 15. Sensitivity analysis of the objective function with respect to the value of μ.

of DCs decrease/increase), the costs of supply chain dramatically decrease. Indeed, as the strength of potential DCs gets to be larger, DCs are more capable to satisfy the demands and fewer demands are lost.

8. Performance analysis of the RPP model

In order to evaluate the effectiveness and desirability of the RPP model, we compare it with the PCCP model. For this end, 100 realizations of uncertain parameters are uniformly generated. In fact, if $\xi = (\xi_1, \xi_2, \xi_3, \xi_4)$ is an imprecise parameter with trapezoidal possibility distribution, the realization is created by generating a random number uniformly between the two extreme points of the related possibility distribution function (*i.e.*, $\xi_{\text{real}} \sim [\xi_1, \xi_4]$). Then, the solutions of the RPP and PCCP models are achieved (*i.e.*, (x^*, z^*)) and substituted in a linear programming model that its compact form is as follows:

$$\min w = f_{\text{real}} z^* + c_{\text{real}} x^* + \omega R \tag{8.1}$$

$$Ax^* - R \leq U_{\text{real}}\, z^* \tag{8.2}$$

$$Bx^* \leq 0 \tag{8.3}$$

$$Nx^* = 1 \tag{8.4}$$

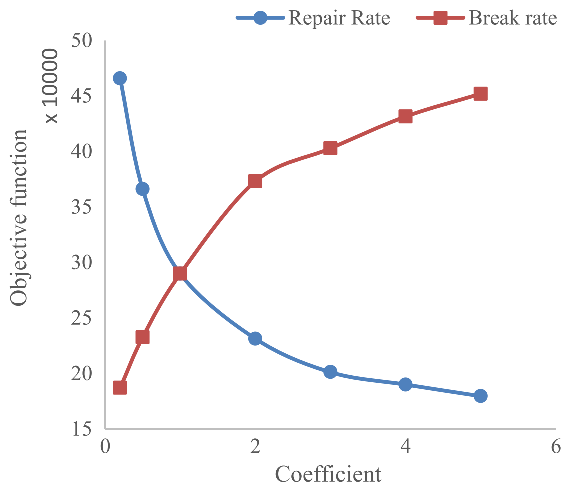

Figure 16. Sensitivity analysis of the objective function with respect to the coefficient multiplied by repair rates and break rates of potential DCs.

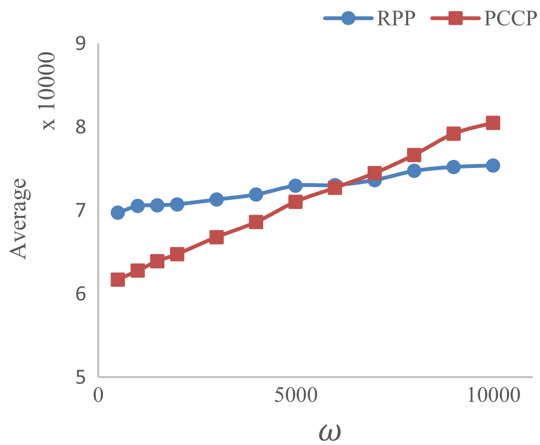

Figure 17. Comparison between the RPP and PCCP models with respect to the average of the objective functions under random realizations.

$$R \geq 0. \tag{8.5}$$

In above linear programming model, R is a decision variable that denotes the violation of chance constraint under random realization. In other words, if a constraint violates the corresponding crisped capacity, the value of R is greater than zero and otherwise, it equals zero. In addition, ω represents violation penalty. The average and standard deviation of the objective functions under random realizations are utilized to compare the performance of the models. The results of this experiment are demonstrated in Figures 17 and 18.

From Figure 17, it can be observed that with respect to the average of the objective functions, the performance of the RPP model is better when the value of ω is greater than about 6000. Additionally, Figure 18 illustrates that in term of standard deviation, the RPP model has a better performance as the value of ω is greater than about 4000. All in all, it can be said that under higher values of ω, the RPP model is more applicable *versus* the PCCP model.

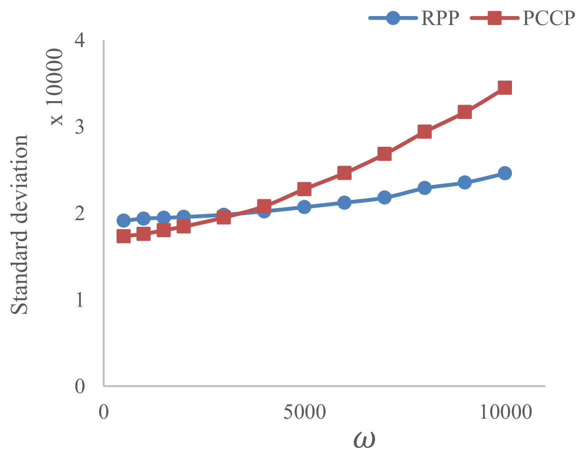

FIGURE 18. Comparison between the RPP and PCCP models with respect to the standard deviation of the objective functions under random realizations.

9. CONCLUSIONS

This paper proposes a joint location-inventory problem to design the distribution network of a supply chain under random supply disruptions. Motivated by shortcomings in the literature, this study develops a hybrid framework based on the Markov process and mathematical programming techniques to determine decisions across the supply chain simultaneously. In the first phase, some performance features of inventory policy, *i.e.*, the number of reorders, the number of shortages, and the mean inventory level are derived using the Markov process. In the second phase, based on the outputs of the Markov process, the location-inventory model is formulated to determine four decisions: (4.1) the number of facilities to be located; (4.2) the location of facilities; (4.3) the assignment of retailers to opened facilities; and (4.4) the optimal inventory policy at each established facility.

This paper contributes to the literature of joint location-inventory problem through the following avenues. The proposed location-inventory model is able to incorporate random supply disruptions in the problem. That is, opened facilities change intermittently between available and unavailable states that period of each state is considered under uncertainty. Considering lost sale shortages is the other issue that distinguishes this study from the ones existed in the literature. The demands of retailers and lead-time are also uncertain to make the problem more realistic. In addition, to the best of our knowledge, this is the first time in the location-inventory problems that a robust possibilistic program is implemented to tackle the lack of knowledge about the real value of input parameters. Finally, since the proposed problem belongs to the class of NP-hard problems, an SA-based meta-heuristic algorithm is developed to solve the problem in an efficient way.

Wide experiments are conducted to analyze the performances of the proposed hybrid approach and the algorithm. The numerical results propose a number of insights. Specially, we show that (4.1) the proposed solution approach performs efficiently for different sizes of the problem; (4.2) RPP provides solution that is less sensitive to the variations in the noisy and uncertain data; (4.3) the supply chain tends to open more strength DCs as well as the costs of supply chain dramatically decrease when the strength of DCs gets to be larger; (4.4) significant cost savings may be realized, if the supplier can improve its lead-time in small ranges of μ; and (4.5) under higher values of ω, the RPP model is more applicable *versus* the PCCP model.

The current study can be extended in a number of promising ways. Addressing routing decisions in order to investigate the joint location, inventory, and routing problem is an attractive subject for future research. The proposed problem can also be extended by accounting multi-product models, multi-echelon inventory control, and lateral transshipments between facilities. Finally, regarding computational complexity of the problem, future research may be aimed to develop other meta-heuristic methods for solving it.

References

[1] J.T. Mentzer, W. DeWitt, J.S. Keebler, S. Min, N.W. Nix, C.D. Smith and Z.G. Zacharia, Defining supply chain management. *J. Bus. Logist.* **22** (2001) 1–25.

[2] L. Ozsen, C.R. Coullard and M.S. Daskin, Capacitated warehouse location model with risk pooling. *Naval Res. Logist. (NRL)* **55** (2008) 295–312.

[3] A. Ahmadi-Javid and P. Hoseinpour, A location-inventory-pricing model in a supply chain distribution network with price-sensitive demands and inventory-capacity constraints. *Transp. Res. Part E: Logist. Transp. Rev.* **82** (2015) 238–255.

[4] A. Ahmadi-Javid and A.H. Seddighi, A location-routing-inventory model for designing multisource distribution networks. *Eng. Optim.* **44** (2012) 637–656.

[5] W. Klibi, A. Martel and A. Guitouni, The design of robust value-creating supply chain networks: a critical review. *Eur. J. Oper. Res.* **203** (2010) 283–293.

[6] S.-H. Huang and P.-C. Lin, A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transp. Res. Part E: Logist. Transp. Rev.* **46** (2010) 598–611.

[7] M.J. Naderi, M.S. Pishvaee and S.A. Torabi, Applications of fuzzy mathematical programming approaches in supply chain planning problems, in Fuzzy Logic in Its 50th Year. Springer (2016) 369–402.

[8] R. Babazadeh, J. Razmi, M.S. Pishvaee and M. Rabbani, A sustainable second-generation biodiesel supply chain network design problem under risk. *Omega* **66** (2017) 258–277.

[9] M. Pishvaee, J. Razmi and S. Torabi, An accelerated Benders decomposition algorithm for sustainable supply chain network design under uncertainty: a case study of medical needle and syringe supply chain. *Transp. Res. Part E: Logist. Transp. Rev.* **67** (2014) 14–38.

[10] M.S. Pishvaee, M. Rabbani and S.A. Torabi, A robust optimization approach to closed-loop supply chain network design under uncertainty. *Appl. Math. Model.* **35** (2011) 637-49.

[11] S.C. Leung, S.O. Tsang, W.L. Ng and Y. Wu, A robust optimization model for multi-site production planning problem in an uncertain environment. *Eur. J. Oper. Res.* **181** (2007) 224–238.

[12] A. Hasani, S.H. Zegordi and E. Nikbakhsh, Robust closed-loop supply chain network design for perishable goods in agile manufacturing under uncertainty. *Int. J. Prod. Res.* **50** (2012) 4649–4669.

[13] P.B. Mirchandani and R.L. Francis, Discrete Location Theory. Wiley, New York (1990).

[14] A. Diabat and E. Theodorou, A location–inventory supply chain problem: reformulation and piecewise linearization. *Comput. Ind. Eng.* **90** (2015) 381–389.

[15] W.J. Baumol and P. Wolfe, A warehouse-location problem. *Oper. Res.* **6** (1958) 252–263.

[16] F. Barahona and D. Jensen, Plant location with minimum inventory. *Math. Program.* **83** (1998) 101–111.

[17] M.S. Daskin, C.R. Coullard and Z.-J.M. Shen, An inventory-location model: formulation, solution algorithm and computational results. *Ann. Oper. Res.* **110** (2002) 83–106.

[18] A.A. Javid and N. Azad, Incorporating location, routing and inventory decisions in supply chain network design. *Transp. Res. Part E: Logist. Transp. Rev.* **46** (2010) 582–597.

[19] H.-W. Jin, A study on the budget constrained facility location model considering inventory management cost. *RAIRO: OR* **46** (2012) 107–123.

[20] S.M. Mousavi, N. Alikar and S.T.A. Niaki, A. Bahreininejad, Optimizing a location allocation-inventory problem in a two-echelon supply chain network: a modified fruit fly optimization algorithm. *Comput. Ind. Eng.* **87** (2015) 543–560.

[21] N. Nekooghadirli, R. Tavakkoli-Moghaddam, V. Ghezavati and S. Javanmard, Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics. *Comput. Ind. Eng.* **76** (2014) 204–221.

[22] M. Rabbani, S. Keyhanian, M. Hasannia, M. Eskandari and M. Jalali, Impact of end of lease contracts' option on joint pricing and inventory decisions of remanufacturable leased products. *Int. J. Ind. Eng. Comput.* **7** (2016) 191–204.

[23] S.J. Sadjadi, A. Makui, E. Dehghani and M. Pourmohammad, Applying queuing approach for a stochastic location-inventory problem with two different mean inventory considerations. *Appl. Math. Model.* **40** (2016) 578–596.

[24] P. Jindal and A. Solanki, Integrated vendor-buyer inventory models with inflation and time value of money in controllable lead time. *Decis. Sci. Lett.* **5** (2016) 81–94.

[25] S.-H. Liao, C.-L. Hsieh and W.-C. Ho, Multi-objective evolutionary approach for supply chain network design problem within online customer consideration. *RAIRO: OR* **51** (2017) 135-55.

[26] M. AmalNick and R. Qorbanian, Dynamic pricing using wavelet neural network under uncertain demands. *Decis. Sci. Lett.* **6** (2017) 251-60.

[27] Y. Díaz-Mateus, B. Forero, H. López-Ospina and G. Zambrano-Rey, Pricing and lot sizing optimization in a two-echelon supply chain with a constrained Logit demand function. *Int. J. Ind. Eng. Comput.* **9** (2018) 205–220.

[28] Q. Chen, X. Li and Y. Ouyang, Joint inventory-location problem under the risk of probabilistic facility disruptions. *Transp. Res. Part B: Methodol.* **45** (2011) 991–1003.

[29] Z. Drezner, Heuristic solution methods for two location problems with unreliable facilities. *J. Oper. Res. Soc.* **38** (1987) 509–514.

[30] S.-D. Lee, On solving unreliable planar location problems. *Comput. Oper. Res.* **28** (2001) 329–344.

[31] R. Babazadeh, J. Razmi, M. Rabbani and M.S. Pishvaee, An integrated data envelopment analysis–mathematical programming approach to strategic biodiesel supply chain network design problem. *J. Clean. Prod.* **147** (2015) 694–707.

[32] S.-D. Lee and W.-T. Chang, On solving the discrete location problems when the facilities are prone to failure. *Appl. Math. Model.* **31** (2007) 817–831.

[33] M.-B. Aryanezhad, S.G. Jalali and A. Jabbarzadeh, An integrated supply chain design model with random disruptions consideration. *Afr. J. Bus. Manag.* **4** (2010) 2393–2401.
[34] Y. Zhang, L.V. Snyder, T.K. Ralphs and Z. Xue, The competitive facility location problem under disruption risks. *Transp. Res. Part E: Logist. Transp. Rev.* **93** (2016) 453–473.
[35] M. Saffari, S. Asmussen and R. Haji, The M/M/1 queue with inventory, lost sale, and general lead times. *Queueing Syst.* **75** (2013) 65–77.
[36] M. Parlar, Continuous-review inventory problem with random supply interruptions. *Eur. J. Oper. Res.* **99** (1997) 366–385.
[37] E. Mohebbi, Supply interruptions in a lost-sales inventory system with random lead time. *Comput. Oper. Res.* **30** (2003) 411–426.
[38] M. Schwarz, C. Sauer, H. Daduna, R. Kulik and R. Szekli, M/M/1 queueing systems with inventory. *Queueing Syst.* **54** (2006) 55–78.
[39] E. Teimoury, M. Modarres, F. Ghasemzadeh and M. Fathi, A queueing approach to production-inventory planning for supply chain with uncertain demands: case study of PAKSHOO Chemicals Company. *J. Manuf. Syst.* **29** (2010) 55–62.
[40] S. Garg, D.K. Sundar and K. Ravikumar, A periodic tabular policy for scheduling of a single stage production-inventory system. *Comput. Ind. Eng.* **62** (2012) 21–28.
[41] Z.-J.M. Shen, C. Coullard and M.S. Daskin, A joint location-inventory model. *Transp. Sci.* **37** (2003) 40–55.
[42] J. Shu, C.-P. Teo and Z.-J.M. Shen, Stochastic transportation-inventory network design problem. *Oper. Res.* **53** (2005) 48–60.
[43] Z.-J.M. Shen and L. Qi, Incorporating inventory and routing costs in strategic location models. *Eur. J. Oper. Res.* **179** (2007) 372–389.
[44] O. Berman, D. Krass and M.M. Tajbakhsh, A coordinated location-inventory model. *Eur. J. Oper. Res.* **217** (2012) 500–508.
[45] J. Asl-Najafi, B. Zahiri, A. Bozorgi-Amiri and A. Taheri-Moghaddam, A dynamic closed-loop location-inventory problem under disruption risk. *Comput. Ind. Eng.* **90** (2015) 414–428.
[46] M. Parlar and D. Perry, Inventory models of future supply uncertainty with single and multiple suppliers. *Naval Res. Logist. (NRL)* **43** (1996) 191–210.
[47] E. Mohebbi and D. Hao, When supplier's availability affects the replenishment lead time – An extension of the supply-interruption problem. *Eur. J. Oper. Res.* **175** (2006) 992–1008.
[48] G. Frizelle, Seeing inventory as a queue, in Inventory Management: Non-Classical Views. CRC Press, Boca Raton Florida (2009) 152–17.
[49] M. Pishvaee, J. Razmi and S.A. Torabi, Robust possibilistic programming for socially responsible supply chain network design: a new approach. *Fuzzy Sets Syst.* **206** (2012) 1–20.
[50] M. Inuiguchi and J. Ramlk Possibilistic linear programming: a brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem. *Fuzzy Sets Syst.* **111** (2000) 3–28.
[51] B. Zahiri, S. Torabi, M. Mousazadeh and S. Mansouri, Blood collection management: methodology and application. *Appl. Math. Model.* **39** (2015) 7680–7696.
[52] M. Bashiri, H. Badri and J. Talebi, A new approach to tactical and strategic planning in production–distribution networks. *Appl. Math. Model.* **36** (2012) 1703–1717.
[53] D. Dubois and H. Prade, The mean value of a fuzzy number. *Fuzzy Sets Syst.* **24** (1987) 279–300.
[54] S. Heilpern, The expected value of a fuzzy number. *Fuzzy Sets Syst.* **47** (1992) 81–86.
[55] A. Ahmadi Javid and N. Azad, Incorporating location, routing and inventory decisions in supply chain network design. *Transp. Res. Part E: Logist. Transp. Rev.* **46** (2010) 582–597.
[56] N. Nekooghadirli, R. Tavakkoli-Moghaddam, V. Ghezavati and S. Javanmard, Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics. *Comput. Ind. Eng.* **76** (2014) 204–221.
[57] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller, Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21** (1953) 1087–1092.
[58] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, Optimization by simulated annealing. *Science* **220** (1983) 671–680.
[59] R.G. Askin, I. Baffo and M. Xia, Multi-commodity warehouse location and distribution planning with inventory consideration. *Int. J. Prod. Res.* **52** (2014) 1897–1910.
[60] D. Shishebori, M.J. Akhgari, R. Noorossana and G.H. Khaleghi, An efficient integrated approach to reduce scraps of industrial manufacturing processes: a case study from gauge measurement tool production firm. *Int. J. Adv. Manuf. Technol.* **76** (2014) 831–855.
[61] D.C. Montgomery, Design and Analysis of Experiments. John Wiley & Sons, New York (2008).