Mathematical modelling of a decentralized multi-echelon supply chain network considering service level under uncertainty

Raheleh Nourifar a, Iraj Mahdavi a, Nezam Mahdavi-Amiri b, Mohammad Mahdi Paydar c

Abstract:

We study a multi-time, multi-product and multi-echelon supply chains aggregate procurement, production and distribution planning problem and discuss the implications of formulating a tri-level model to integrate procurement, production and distribution, maintaining the existing hierarchy in the decision process. In our model, there are three different decision makers controlling the procurement, production and the distribution processes in the absence of cooperation because of different optimization strategies. First, we present a hierarchical tri-level programming model to deal with decentralized supply chain problems. Then, an algorithm is presented to solve the proposed model. A numerical illustration is provided to show the applicability of the optimization model and the proposed algorithm. In order to evaluate the application of the model and the proposed algorithm, ten sets of small and large problems are randomly generated and tested. The experimental results show that our proposed fuzzy-stochastic simulation based hierarchical interactive particle swarm optimization (Sim-HIPSO) performs well in finding good approximate solutions within reasonable computation times.

Keywords: Decentralized decision making; supply chain network; uncertainty; tri-level programming; service level.

1. Introduction

Supply chain management plays a vital role in entrepreneurial activities and is an integral part of most businesses. Today’s fierce competition markets, introduction of products with new characteristics in short life cycles, rising customers’ expectations, and also the rapid growth of technology have caused business enterprises to seriously invest in their supply chains in order to retain and ameliorate their own positions in marketplace. A supply chain is a network of organizations working together to convert and move products from the raw material stage to the final customer. These organizations are interconnected through physical, informational and financial flows. Having an efficient and lean supply chain network with effective interconnections in between various levels of the chain can lead to a significant increase of profitability, reduction of costs, and improvement of customer satisfaction. Pursuing these
objectives, without taking into account certain strategic, tactical and operational decisions, could have a negative impact on performance of the supply chain.

Decision making in a supply chain network is characterized in two ways: centralized or decentralized. In a centralized supply chain, decisions are made by a single decision maker at a central location for the entire supply chain system. The typical objective in a centralized supply chain is to optimize the total cost/profit of the system. In a decentralized supply chain, each entity decides its own effective strategy without considering the impact on other entities of the supply chain system. This way, centralized decisions lead to global optimization, whereas decentralized decisions lead to local optimization. Almost all supply chain networks (SCNs), due to the existence of various enterprises with different objectives, often conflicting with each other in various levels of chain, are often not centrally handled by a single decision maker [1]. Hence, in today’s competitive markets, the study of SC management problems appears to be beneficial in nearly every business [2]. In this regard, the role of decentralized SC management is even more significant in industries such as cement, glass, etc. As pointed out by Ma et al. [23], poor procurement, production and distribution planning in these industries can lead to significant losses, specially in large scale supply chains. To the best our Knowledge, in most research works dealing with decentralized supply chain networks (SCNs), procurement policy about lot sizing decisions for the raw material is often examined separately from the production planning problem and logistics management, because of the complexity of the interplays among these areas. Therefore, the current research aims at determining simultaneously the procurement, production and distribution policies with decentralized decision-making structure. The main features of some recent works in the study of decentralized SCNs are summarized in Table 1.

An important issue in optimizing decentralized supply chain network is uncertainty. Although deterministic supply chain models are useful, they may not be realistic because these models use the average values of the system parameters, while various sources of uncertainty can be identified in these systems. On the other hand, incorporating inherent uncertainties helps companies to provide better levels of service to their customers. In supply chain, the service level is usually defined as the probability of being able to service the customers’ demand ever facing any backorder or lost sale. Thus, increasing the inventory level of distribution centers improves the service level, but also increases the risk of unsold products. However, probabilistic features of these costs and the associated complications of their considerations in the mathematical model may be the reasons for their absence in the operational costs considered for decentralized network design problems. Here, we attempt to incorporate these cost in the proposed model.

The principal focus of our work is to propose an appropriate model for the procurement-production-distribution system in a multi-echelon supply chain network with a decentralized decision structure and multiple uncertainty issues including demand-side, process-side and supply-side simultaneously with considering service level as well as presenting effective computational tools for determining solutions of the proposed model. To overcome the described problem, the main contributions of our work here as compared to the available approaches can be summarized as follows:
1) Development of a mathematical model for the decentralized procurement-production-distribution planning problem with a tri-level programming structure.
2) Handling multiple uncertainties including demand-side, process-side and supply-side, to more closely reflect reality.
3) Considering uncertainties with fuzzy and stochastic parameters simultaneously (available works with uncertain parameters deal with fuzzy random parameters or stochastic problems).
4) Incorporating the responsiveness level and its corresponding cost in the problem.
5) Presenting a novel hybrid PSO approach combined with a simulation-based method for solving large problems.

The remainder of our work is organized as follows. In the next section, a literature review is presented discussing different approaches to supply chain network design followed by an outline of our contributions. In Section 3, first some concepts of multi-level programming and fuzzy set theoretical applications are given. Then, a description of the procurement-production-distribution planning (PPDP) problem along with notation, assumptions and mathematical formulation of a decentralized supply chain (SC) model are presented is given in Section 4. In Section 5, a hybrid algorithm is proposed for solving the tri-level PPDP model. Numerical experimentation and some results obtained by the proposed algorithm are presented and discussed in section 6. Finally, conclusions and directions for future research are provided in Section 7.

2. Literature review

A supply chain network is commonly defined as a set of organizations directly or indirectly involved in satisfying the customers’ orders. These organizations include customers, wholesalers/distributors, manufacturers and suppliers. The network is comprised of all functions involved in fulfilling customers’ requirements and needs. These functions consist of procurement, production and distribution. Most supply chain models are merely concerned with the competitive market in the sales process (demand side), while the chains often face the competitive markets in the procurement process. They are provided with the raw materials needed from the competitive market through multiple suppliers and this upstream competition affects the chain’s costs. So in order to improve the income of the chain, this upstream competition (supply side) is very important [1]. In the literature, in most research works dealing with supply chain networks, procurement policy about lot sizing decisions for raw materials is often examined separately from the production planning problem and inventory and logistics management, because of the complexity of the interplays among these areas [3-8]. However, PPDP in a perfect supply chain network that includes several suppliers, producers, distributors and customers is one of the most important optimization problems in the design of supply chain networks and is rarely addressed in the literature.

A decentralized approach to supply chain network (SCN) is subject to various economic factors at different levels of the chain and the effectiveness of the decision is quite important. Decision makers at every level control a set of variables and the objective function of the level which may be contrasted with the optimal solution of the whole system. Every entity in the supply chain could also make its own decisions based on its goals and objectives,
potentially causing distributed and decentralized decision making in the supply chain. The important and fundamental challenge is appropriately modeling the relationships among the entities of the supply chain. The optimization problem needs to be formulated as a multi-level (bi-level or tri-level) mathematical programming problem when two or more decision makers in a hierarchical structure are being considered [9].

Study on decentralized SCN design has attracted much attention in the past decade [10-13, 40]. Most researchers so far have focused on modeling production-distribution (PD) problems for a decentralized SCN design. For instance, Calvete et al. [14] considered a PD problem in a three-level supply chain network. The authors presented a bi-level model for the problem wherein the first level decides the design of the routes serving the clients and the second level is involved with decision on production planning. They utilized an ant colony based approach to solve the problem at the first level using ants to create the routes and a direct optimization method to solve the second level problem. Camacho-Vallejo et al. [15] studied the problem of planning production and distribution in a supply chain. The problem is formulated as a bi-level mathematical problem where the upper level is to decide the amount of products sent from the distribution centers to the retailers with the aim to minimize the transportation costs of acquiring the products coming from the plants. However, the lower level controls the amount of produced products with the aim to minimize the plant’s operational costs. To solve the problem, they proposed a heuristic algorithm based on Scatter Search incorporating the Stackelberg’s game. It is considerable that their solution approach improved upon the obtained objective function value and the required time needed by the method of Calvete et al. [14]. Xu et al. [16] presented a tri-level model for the supply chain problem consisting of supplier, manufacturer and retailer based on the conditional value-at-risk (CVaR) measure of risk management. In this model, the supplier and the manufacturer, at the top and the middle levels, maximize their own profits while at the bottom level, the retailer maximizes her CVaR of the expected profit. To solve the problem, the proposed model was transformed into a bi-level programming one. Calvete et al. [2] investigated a decentralized supply chain network composed of manufacturing plants, warehouses and customers. The authors adopted a bi-level model for the PDP problem. In the bi-level model, the distribution company controls the opening of depots and how to dispatch products from depots to customers, and the manufacturer company controls the manufacturing process. A metaheuristic approach based on an evolutionary optimization algorithm was presented to solve the problem. An integrated production and distribution problem considering a parallel-machine production environment and a product batch-based delivery was investigated by Guo et al. [9]. A mixed integer nonlinear bi-level programming problem was formulated in this model and solved by a bi-level evolutionary optimization procedure based on a memetic algorithm and an evolutionary strategy.

Another important challenge in designing a supply chain network is how to incorporate different issues of disruption and uncertainty in different supply chain entities of suppliers to customers due to the complex interactions among them. In the optimization model, ignoring uncertainty results in infeasible or non-optimal solutions in real cases. Therefore, the need to appropriately consider uncertainty and variations in the supply chain is essential for the design and modelling of supply chain networks from strategic point of view.
In order to cope with uncertainty, different approaches have been used in the literature: first is the distribution-based approach, where the normal distribution with specified mean and standard deviation is widely invoked for modeling the uncertain parameters; second is fuzzy programming, where the forecasting parameters are considered to be fuzzy numbers with accompanied membership functions; third is stochastic programming, where uncertainty is directly based on scenarios, in which several discrete scenarios with associated probability levels are used to describe expected occurrences of particular outcomes, and fourth is chance constraint-based approach in which each uncertain parameter is treated as a random variable with a given probability distribution.

Some studies incorporated uncertainty in supply chain models [17-22,32]. However, inclusion of uncertainties in decentralized supply chain networks has rarely been considered in the literature. Wang and Lee [11] addressed a location-allocation problem at two-level and three-level supply chains considering risky demands. A bi-level stochastic programming model was developed to maximize the profit. The problem was solved by an improved algorithm based on ant colony optimization. Ma et al. [23] developed a bi-level programming model for the integrated PD planning problem in supply chain networks considering contradiction and coordination under uncertainty. In this model, the core firm in the upper level of the hierarchy controls the opening of plants and warehouses for serving the customers, minimizing the total cost. In the lower level, the production sub-problem is aimed to minimize manufacturing and transportation costs and the distribution sub-problem is aimed to minimize the inventory and distribution costs, making decisions based on the firm’s core decisions. In their model, customer demands and transportation costs were considered to be fuzzy variables. They utilized an improved genetic algorithm (GA) with a fuzzy logic controller in order to control the GA parameters during the genetic search process. Saranwong and Likasiri [36, 37] modeled a distribution network problem using a bi-level programming approach and applied it to a case study in Thailand. The authors proposed five heuristic algorithms and an exact approach using the CPLEX software package. Yue and you [38] presented a supply chain design and operation problem using a bi-level programming framework. They developed a reformulation and decomposition algorithm for decoding the problem. Fard and Hajaghaei-Keshteli [39] studied the location-allocation problem in a supply chain considering the forward and reverse network simultaneously. They formulated the problem as a tri-level programming problem and proposed five metaheuristic algorithms to solve it.

In accordance with the above considerations, here we propose a multi-period, multi-product procurement-production-distribution planning (PPDP) model in a decentralized supply chain with multiple uncertainty parameters considering service level and responsiveness. A forward supply chain network encompasses companies involved with various stages of the chain consisting of suppliers, manufacturers, distribution centers and consumer zones. There are uncertainties involved in all the three levels of procurement, production, and distribution planning of the SC network. The uncertainties corresponding to imprecise parameters in the demand-side, process-side and supply-side are handled with stochastic and fuzzy numbers. In order to elevate the responsiveness level of the chain, two major factors consisting of shortage and delay costs are considered in the model.
3. Preliminaries

3.1. Multi-level optimization

Multi-level linear programming (MLLP) models have been used to address a variety of real world problems, in which a hierarchical and competitive structure of decision making prevails. In a hierarchical decision making, the first level is known as the leader, whereas at the second level is called the follower [25]. Control variables divide decisions between the different levels of decision-making, and the individual objective functions are optimized [26]. Bi-level linear programming (BLLP) has been applied to multi-level decision problems when there are only two levels of decision. A general BLLP problem can be stated as follows [27]:

$$\min_{x \in X} F_1(x, y)$$

s.t.  \( G_1(x, y) \leq 0 \)

$$\min_{y \in Y} F_2(x, y)$$

s.t.  \( G_2(x, y) \leq 0 \)

\( x, y \geq 0, \)

where \( x \in R^p \) and \( y \in R^q \) are the decision variables of the upper-level and lower-level, respectively. \( F_i : R^p \times R^q \rightarrow R \), \( i = 1, 2 \), are the objective functions of the upper-level and lower-level, respectively, and \( G_i : R^p \times R^q \rightarrow R \), \( i = 1, 2 \), are the constraints of the upper-level and lower-level, respectively.

By extending bi-level programming, multi-level programming such as tri-level programming is introduced, where the second level is itself a bi-level program. To describe a tri-level decision problem, a basic Linear Tri-level Programming (LTLP) model can be stated as follows:

$$\min_{x \in X} F_1(x, y, z)$$

s.t.  \( G_1(x, y, z) \leq 0 \)

$$\min_{y \in Y} F_2(x, y, z)$$

s.t.  \( G_2(x, y, z) \leq 0 \)

$$\min_{z \in Z} F_3(x, y, z)$$

s.t.  \( G_3(x, y, z) \leq 0 \)

\( x, y, z \geq 0, \)

where \( x \in R^p \), \( y \in R^q \) and \( z \in R^r \) are the decision variables of the top-level, middle-level and bottom-level, respectively. \( F_i : R^p \times R^q \times R^r \rightarrow R \), \( i = 1, 2, 3 \), are the objective functions of the
According to the conditions of the tri-level problem, the problem is non-convex due to the involvement of a constraint zone specified by another optimization problem and have been proved to be strongly NP-hard even when all the functions are linear. Several heuristic methods have been proposed for solving multi-level programming problems [25, 28, 34].

### 3.2. Fuzzy set theory

Here, we state some necessary results of fuzzy set theory. We refer to [31, 33] for details.

**Definition 1.** A fuzzy set $\tilde{a}$ of a universe $X$ is a set of ordered pairs $(x, \mu_{\tilde{a}}(x))$, $\mu_{\tilde{a}}(x)$ called the membership functions which is a real number in the interval $[0, 1]$ and associates with each element $x$ in $X$. The classical membership degrees are represented by 1 (is a member) and 0 (not a member).

**Definition 2.** A fuzzy variable $\tilde{x}$ is said to be convex if $\mu_{\tilde{a}}(\lambda x_1 + (1-\lambda)x_2) \geq \min\{\mu_{\tilde{a}}(x_1), \mu_{\tilde{a}}(x_2)\}$, $\forall x_1, x_2 \in X, \lambda \in [0,1]$.

**Definition 3.** A fuzzy number $\tilde{a}$ is a convex subset of the real line satisfying the following conditions:

(a) $\mu_{\tilde{a}}(x)$ is piecewise continuous;

(b) $\mu_{\tilde{a}}(x)$ is normalized, that is, there exists $m \in \mathbb{R}$ with $\mu_{\tilde{a}}(m)=1$, where $m$ is called the mean value of $\tilde{a}$.

**Definition 4.** The $\alpha-$cut $\tilde{A}_\alpha$, and strong $\alpha-$cut $\tilde{A}^+_{\alpha}$, of the fuzzy set $\tilde{A}$ in the universe of discourse $X$ are respectively defined by

\[
\tilde{A}_\alpha = \{ x \mid \mu_{\tilde{A}}(x) \geq \alpha, \ x \in X \}, \quad \text{where } \alpha \in [0,1]
\]

\[(2)\]

\[
\tilde{A}^+_{\alpha} = \{ x \mid \mu_{\tilde{A}}(x) > \alpha, \ x \in X \}, \quad \text{where } \alpha \in [0,1].
\]

\[(3)\]

The lower and upper points of any $\alpha-$cut $\tilde{A}_\alpha$, are represented by $\inf \tilde{A}_\alpha$ and $\sup \tilde{A}_\alpha$, respectively, and we assume that both are finite. For convenience, we denote $\inf \tilde{A}_\alpha$ by $A^-_{\alpha}$ and $\sup \tilde{A}_\alpha$ by $A^+_{\alpha}$ (see Figure. 1).
Definition 5. Let \( \tilde{a} \) be a fuzzy number with membership function \( \mu_{\tilde{a}}(x) \). Then, for \( \xi > r \), the possibility, necessity, and credibility measures are defined as follows:

**Possibility measure:**

\[
\text{Poss} \left( \xi \geq r \right) = \text{Sup} \left( \mu(x) \right).
\]

**Necessity measure:**

\[
\text{Nec} \left( \xi \geq r \right) = 1 - \text{Pos} \left( \xi < r \right) = 1 - \text{Sup} \left( \mu(x) \right).
\]

**Credibility measure:**

\[
\text{Cr} \left( \xi \geq r \right) = \frac{1}{2} \left( \text{Pos} \left( \xi \geq r \right) + \text{Nec} \left( \xi \geq r \right) \right).
\]

4. Problem description

Here, we consider the development of a model for a decentralized supply chain network in markets. The chain is composed of several customer groups, distributors, manufacturers and suppliers (see Fig. 2). Customers are at the first level, while at the second level there are storage centers which transport final products to the end customers and at the third level, there are manufacturers (producers) which direct the end products to storage centers. The unsatisfied demand of each customer in each time period is assumed to be backordered; however, the entire unsatisfied customers’ demands must eventually be satisfied. Finally, at the lowest level, there are suppliers providing raw materials to the factories for production. The main assumptions for this problem are:

1. The decision making approach within the hierarchical or the tri-level structure is decentralized.
2. The network corresponds to a multi-product, multi-echelon, and multi-time SC problem involving several suppliers, manufacturers, potential distribution centers (DCs) and end customers.
3. The precise demand and material supply at the DC and supplier levels are not known, and thus demand and supply parameters are considered to be stochastic parameter with known distribution functions.
4. There are several products in the chain and each customer zone can be supplied by any DC.
5. The numbers and locations of suppliers, manufacturers and customers are known.

To formulate the mathematical model, the following notations are used.

**Sets of indices:**

- \( D \) Number of suppliers \( (d = 1, \ldots, D) \)
- \( M \) Number of manufacturing plants \( (m = 1, \ldots, M) \)
- \( J \) Number of potential warehouse distribution centers \( (j = 1, \ldots, J) \)
- \( I \) Number of customers \( (i = 1, \ldots, I) \)
- \( K \) Number of end products \( (k = 1, \ldots, K) \)
- \( L \) Number of raw materials \( (l = 1, \ldots, L) \)
\( T \) Number of time periods \( (t = 1, \ldots, T) \).

\textbf{Parameters:}

- \( p^t_{mk} \) Cost of production per unit of end product \( k \) by plant \( m \) at time period \( t \)
- \( pcr^t_{dl} \) Cost of manufacturing per unit of raw material by supplier \( d \) at time period \( t \)
- \( h^t_{mk} \) Cost of holding per unit of end product \( k \) in plant \( m \) at period \( t \)
- \( hd^t_{jk} \) Holding cost per unit of end product \( k \) in distribution center \( j \) at period \( t \)
- \( hs^t_{ld} \) Holding cost per unit of raw material \( l \) in supplier \( d \) at period \( t \)
- \( \eta^t_{mj} \) Transportation cost per unit of end product \( k \) from plant \( m \) to distribution center \( j \) at period \( t \)
- \( td^t_{ji} \) Transportation cost per unit of end product \( k \) from distribution center \( j \) to customer \( i \) at period \( t \)
- \( ts^t_{dmt} \) Transportation cost per unit of raw material \( l \) from suppliers \( d \) to plant \( m \) at period \( t \)
- \( scp^t_{mt} \) Set up cost per unit of end product \( k \) by plant \( m \) at period \( t \)
- \( scs^t_{jd} \) Set up cost to produce per unit raw material by supplier \( d \) at period \( t \)
- \( pp^t_{jmk} \) Price per unit of product \( k \) supplied by plant \( m \) to distribution center \( j \) at period \( t \)
- \( pr^t_{ldmt} \) Price per unit of raw material \( l \) supplied by supplier \( d \) to plant \( m \) at time period \( t \), which is a fuzzy number
- \( bc^t_{ki} \) Cost per unit of backorder of product \( k \) to supply the demand of customer \( i \) at time period \( t \)
- \( dc^t_{ni} \) Delay cost due to the shipping per unit of product \( k \) to the demand of customer \( i \) at time period \( t \)
- \( S^t_{dl} \) Capacity of production of supplier \( d \) for raw material \( l \) at time period \( t \), which is a stochastic number
- \( A^t_{mt} \) Capacity of production of plant \( m \) for end product \( k \) at time period \( t \), which is a stochastic number
- \( D^t_{ni} \) Demand value of customer \( i \) for end product \( k \) at time period \( t \), which is a stochastic number
- \( f^t_{j} \) Fixed cost for establishing distribution center \( j \) at time period \( t \)
- \( pt^t_{mt} \) Processing time required for production plant \( m \) to produce one unit of product \( k \) at time period \( t \), which is stochastic number
- \( st^t_{mt} \) Set up time of producing end product \( k \) by manufacturing plant \( m \) at time period \( t \), which is a stochastic number

Total time for production plant \( m \) to produce products at time period \( t \), which is a stochastic number \( tt^t_{mt} \)
Delay time of the shipping product $k$ to the demand of customer $i$ at time period $t$, which is a fuzzy number.

$\text{WP}_m$ Total capacity for manufacturing plant $m$ to store products ($m^3$)

$\text{WR}_m$ Total capacity for manufacturing plant $m$ to store raw materials ($m^3$)

$w_j$ Total capacity for distribution center $j$ to store products ($m^3$)

$R^k_m$ Total transportation capacity of manufacturing plant $m$ to deliver product $k$ at period $t$

$v_k$ Volume of a unit of end product $k$ ($m^3$)

$\beta_l$ Usage rate of raw material $l$ to manufacture end product $k$

$M$ A positive large number.

**Decision variables:**

$C^k_m$ Binary variable which is 1, if product $k$ is produced by plant $m$ at period $t$, and 0 otherwise

$Y_j$ Binary variable which is 1, if distribution center $j$ is opening, and 0 otherwise

$X^k_d$ Binary variable which is 1, if supplier $d$ produces raw material, and 0 otherwise

$c^k_{jit}$ Binary variable which is 1, if distribution center $j$ shipped product $k$ to customer $i$, and 0 otherwise

$Q^k_m$ Quantity of product $k$ produced by plant $m$ at period $t$

$QR^l_d$ Quantity of raw material $l$ produced by suppliers $d$ at period $t$

$U^k_{jm}$ Quantity of end product $k$ transported by manufacturing plant $m$ to warehouse $j$ at period $t$

$N^k_j$ Quantity of end product $k$ dispatched by distribution center $j$ to customer $i$ at period $t$

$P^l_d$ Quantity of raw material $l$ shipped from supplier $d$ to plant $m$ at period $t$

$B^k_i$ Backorder quantity of product $k$ for demand of customer $i$ at period $t$

$P^k_m$ Inventory of end product $k$ in plant $m$ at the end of period $t$

$IR^l_m$ Inventory of raw material $l$ in plant $m$ at the end of period $t$

$Id^k_j$ Inventory of product $k$ in distribution center $j$ at the end of period $t$

$IS^l_d$ Inventory of raw material $l$ in supplier $d$ at the end of period $t$.

### 4.1. Decentralized PPDP model

For modeling the procurement-production-distribution planning (PPDP) problem with a decentralized structure, we will introduce a tri-level programming model considering the optimal decisions of the supplier, the manufacturer and the distributor in a four-echelon supply chain. The top level (distribution problem) as leader in the hierarchical structure controls which DCs...
should be applied and specifies the amounts of customer orders provided by the DCs aiming to minimize the total of the fixed costs and operating costs of the opened DCs associated with the inventory and transportation costs. The unmet demand of each customer of any product in each period will be considered as backordered; although the demands of all customers must be met eventually.

Sufficient supplies of product must be available at the DCs in order to meet the demands of the customers. After decisions of the leader (top level), the middle level (production problem), as follower, aims for cost reduction by minimizing production cost, transportation cost and inventory cost from the plant to the DCs. Ultimately, at the bottom level, procurement problem is to decide which suppliers should provide raw materials to the manufacturers. Feedback regarding the production, transportation, and inventory decisions by the production problem and procurement problem are sent to the leader and the leader may then modify the decision with regard to cost minimization. Until an optimal solution is found, this process continues. Therefore, through the interactivity of all involved SC members, an optimal decision is found.

Tri-level programming (TLP) can be utilized to demonstrate the interaction among the leader and the follower as shown by the tri-level structure of the complex real-life decentralized PPDP model in Fig. 3. The tri-level optimization PPDP problem is formulated as follows:

\[
\begin{align*}
\text{Min } F_{\text{dis}} &= \sum_{j=1}^{J} f_{j} Y_{j} + \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{j=1}^{J} pp_{jmt}^{k} U_{jmt}^{k} + \sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{k=1}^{K} hd_{jmt}^{k} Id_{jmt}^{k} + \sum_{j=1}^{J} \sum_{i=1}^{I} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{l=1}^{L} td_{jmt}^{k} N_{jmt}^{k} \\
&+ \sum_{i=1}^{I} \sum_{t=1}^{T} \sum_{m=1}^{M} b_{mt}^{k} B_{mt}^{k} + \sum_{i=1}^{I} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{j=1}^{J} \max(0, l_{jt}) dc_{jmt}^{k} c_{jmt}^{k} 
\end{align*}
\]

(1)

\[
\begin{align*}
\sum_{k=1}^{K} v_{k} Id_{jmt}^{k} &\leq W_{j} Y_{j} , \quad \forall j, t \\
\sum_{j=1}^{J} U_{jmt}^{k} &\leq R_{m} G_{mt}^{k} , \quad \forall k, t, m
\end{align*}
\]

(2) \hspace{1cm} (3)

\[
Id_{jmt}^{k} = Id_{j-1}^{k} + \sum_{m=1}^{M} U_{jmt}^{k} - \sum_{i=1}^{J} N_{jmt}^{k} , \quad \forall j, k, t
\]

(4)

\[
B_{jmt}^{k} = B_{j-1}^{k} + D_{a} - \sum_{i=1}^{I} N_{jmt}^{k} , \quad \forall i, k, t
\]

(5)

\[
\begin{align*}
\text{Min } F_{\text{man}} &= \sum_{d=1}^{D} \sum_{m=1}^{M} \sum_{l=1}^{L} pr_{dml}^{l} P_{dml}^{l} + \sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{k=1}^{K} scp_{mt}^{k} G_{mt}^{k} + \sum_{j=1}^{J} \sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{k=1}^{K} pc_{mt}^{k} Q_{mt}^{k} \\
&+ \sum_{t=1}^{T} \sum_{m=1}^{M} \sum_{k=1}^{K} tp_{jmt}^{k} U_{jmt}^{k} + \sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{k=1}^{K} hp_{mt}^{k} I_{mt}^{k}
\end{align*}
\]

(6)

s.t.
\[
\sum_{k=1}^{K} \sum_{m=1}^{M} \sum_{j=1}^{J} v_{k,j,m,t} Y_{j,m,t} \leq W_{j,m,t}, \quad \forall j, t
\] (7)

\[
\sum_{k=1}^{K} \sum_{m=1}^{M} p_{k,m,t} P_{k,m,t} + \sum_{k=1}^{K} \sum_{m=1}^{M} s_{k,m} G_{k,m,t} \leq t_{m,t}, \quad \forall m, t
\] (8)

\[
\sum_{k=1}^{K} v_{k,m,t} P_{k,m,t} \leq W_{m,t}, \quad \forall m, t
\] (9)

\[
\sum_{k=1}^{K} P_{k,m,t} \leq A_{m,t}, \quad \forall d, t
\] (10)

\[
\sum_{k=1}^{K} \sum_{j=1}^{J} U_{j,m,t} \leq \sum_{k=1}^{K} N_{i,j,t}, \quad \forall j, m, t
\] (12)

\[
IP_{m,t}^k = IP_{m,t}^k + \sum_{j=1}^{J} U_{j,m,t}^k, \quad \forall m, k, t
\] (13)

\[
\text{Min } F = \sum_{d=1}^{D} \sum_{l=1}^{L} \sum_{t=1}^{T} (mcr_{dt} OR_{dt}^l + scs_{dt} X_{dt}^l + hs_{dt} IS_{dt}^l) + \sum_{d=1}^{D} \sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{l=1}^{L} t_{m,t}^l P_{m,t}^l
\] (14)

s.t.

\[
\sum_{l=1}^{L} Q_{l,dt}^l \leq S_{dt}^l, \quad \forall d, t
\] (15)

\[
\sum_{m=1}^{M} \sum_{l=1}^{L} P_{l,m,t}^l \leq Q_{l,dt}^l, \quad \forall d, t
\] (16)

\[
\sum_{l=1}^{L} Q_{l,dt}^l \leq M X_{dt}^l, \quad \forall d, t
\] (17)

\[
IS_{d,j,t}^l + Q_{d,j,t}^l = \sum_{m=1}^{M} P_{d,m,t}^l + IS_{d,j,t}^l, \quad \forall d, l, t
\] (18)

\[
IR_{m,j,t}^l + \sum_{d=1}^{D} P_{d,m,t} - \sum_{k=1}^{K} P_{k,m,t} \cdot \beta_{l,k} = IR_{m,t}^l, \quad \forall m, l, t
\] (19)

\[
X_{d,t}^l, Y_{j,t}^l, Z_{k,m,t}^l \in \{0,1\}, \quad \forall d, m, j, k, t
\] (20)

\[
Q_{k,m,t}^l, IP_{k,m,t}^l, A_{k,m,t}^l, S_{k,m,t}^l, N_{i,j,t}^l, U_{j,m,t}^l, I_{k,j,t}^l, IS_{d,t}^l \geq 0, \quad \forall d, m, j, k, t.
\] (21)

It should be noted that parameters with a tilde (\(~\)) and a hat (\(^\)\)) at the top are respectively signified as fuzzy and stochastic parameters. The first objective function shown in (1) correspond to the minimization of the sum of the DCs’ and customers’ costs, including fixed cost of establishing DCs, purchasing products, holding costs of products, transportation costs from DCs
to customers, inventory shortage costs at customers and costs of delay for shipping orders to customers.

The control of potential warehouse capacities is considered in the constraints (2). The total quantity shipped from each plant is controlled by constraint (3). The constraints (4) correspond to the end inventories balance of potential DCs. Similarly, the shortages of the customers’ demands balance equations are considered by the constraints (5).

The second objective function (6) is to minimize the producers’ costs, including procurement (buying) of raw materials, setup cost of the production, manufacturing of the products, transportation costs from plants to DCs and inventory holding costs of products. The constraints (7) show that the volumes of the products transported to potential DCs correspond to their total storage capacities. The constraints (8) ensure that the sum of required time to manufacture the products is not more than the total time for production plant. According to the constraints (9), the production volume is less than the total storage capacity of the plants. The constraint (10) impose that the volume of finished product produced by plants should be less than the total production capacity of the plants. The final product inventory according to the total storage capacity of the plants is controlled by constraints (11). Based on constraints (12), the total quantity shipped from any plant is not more than its capacity. The remaining inventory in each plant at the end of each period is controlled by constraints (13). The third objective function (14) is to minimize the total operational costs of suppliers, including production, setup and transportation costs from suppliers to the plants. The constraints (15) and (16) respectively specify that the raw material volume produced and shipped by suppliers are less than the total production capacity of the suppliers. The constraints (17) give the supplier manufacturing raw materials at each time period. The constraints (18) and (19) respectively give the inventory balances of the raw materials for suppliers and plants. Types of the variables are defined by (20) and (21).

4.2. Hybrid uncertainty of model

The parameters in the deterministic models of SC problems are known and deterministic. In these models, determining the features of activities extending from the suppliers to the customers precisely is very difficult, if not impossible. In general, deterministic assumptions for SC models are unrealistic. In reality, both tangible and intangible information are usually available to decision makers related to decision criteria and constraints. Model parameters and environmental coefficients have hybrid uncertainties due to some inherent is variability and/or unavailability of information over the planning horizon and are frequently imprecise. With the SC containing several layers and several parts in each layer, uncertainty may occur in different parts as procurement, production and distribution because of the dynamic and turbulent nature of the supply chain. Table 2 defines different parameters that are considered to be uncertain according to the three identified uncertainty types in an SC tactical planning problem.

Price and delay time uncertainties are different from demand uncertainty in supply chain, as the price of material or product and delay time cannot be predicted more accurately than the demand uncertainty due to usual fluctuations in a large range. Thereupon, they are inherently fuzzy data
rather than crisp values. Hence, for modeling uncertainty in the price of material, a possibility theory rather than a probability theory is more suited.

Most business enterprises for forecasting demands focus on collecting a large data set [24]. Therefore, demand uncertainties are often modeled by probabilistic distributions. Furthermore, processing time, production capacity and supplier capacity are also uncertain. So, in order to engage these hybrid uncertainties of ill-known parameters in the mathematical model, we develop an integrated tri-level model to accommodate for both fuzzy and stochastic data, which we entitle a fuzzy-stochastic tri-level programming (FSTLP) model. In this regard, fuzzy set theory is used to handle fuzzy data and utility theory is employed to treat stochastic data.

5. Solution methodology

The intrinsic complexity involved in a multi-level optimization model due to the mutual relations among various layers have turned the problem into yet one of the most challenging and most difficult problems to solve in real cases. There are many approaches to tackle complicated multi-level optimization problems (MOPs) [34]. They can be categorized into four main approaches: (1) nested sequential approach, 2) single-level transformation approach, (3) multi-objective approach, and (4) co-evolutionary approach.

In the first approach, the optimization problem in lower level is solved by a sequential and nested manner assessing the solutions produced at the upper-levels of MOP. The second approach reformulates MOP into a single-level structure by a penalty function, a marginal function and KKT conditions. The single-level problem can be solved by available methods. In the third approach, after MOP is converted into a multi-objective optimization problem, a multi-objective metaheuristic approach is applied to solve the problem. In the fourth approach, which is a well-known methodology to solve MOPs, at each level, an evolutionary-based optimization approach (e.g., genetic algorithm, particle swarm, simulated annealing, etc.) is directly employed to find an (approximate) optimal solution. Each level tries to maintain and improve its own solutions separately. In general, the levels can evolve in parallel and cooperate by exchanging information. A multi period PPD planning problem in a decentralized supply chain formulated as a MOP is naturally complex. Only small problems can be solved to optimality in a reasonable computing time and medium and large problems are difficult to be solved optimally. This fact lead us to develop a heuristic algorithm for finding approximate optimal solutions. Among the methods described above, a co-evolutionary approach, due to its capability and its efficiency in providing satisfactory solutions in reasonable times, is employed to develop a novel algorithm entitled Hierarchical Interactive Particle Swarm Optimization (HIPSO). In the sequel, the technical steps needed to solve the model are described.

5.1. Review of particle swarm optimization (PSO)

Particle Swarm Optimization (PSO) is a biologically inspired computational search developed in 1995 by Kennedy and Eberhart [29]. PSO simulates the behavior of bird flocking or fish
schooling. A number of basic variations have been developed to improve the speed of convergence and quality of solution found by PSO.

PSO is initialized with an initial population of solutions called initial population generated randomly. Each single solution is a bird in the search space, called particle, which represents a feasible solution for the problem. A set of particles that are produced in every step of the algorithm is called a swarm. Every particle has a fitness value which is obtained by fitness function to be optimized, and has a velocity to reach a new position. The particles fly through the problem space by following the current optimal particles. Then, PSO searches for better solutions by updating the generation. Each particle moving in the search space to reach new positions (solutions) is composed of the following three components:

**Cognitive component:** The best solution that a particle acquires alone \((X_{pb}^i)\) with coefficient \(c_1\).

**Social component:** The best solution recognized by the entire group \((X_{gb})\) with coefficient \(c_2\).

**Inertia component:** The impact velocity of particle in the step before the current speed is determined by weight \(w\).

After finding the two best values \((X_{pb}, X_{gb})\), the particle updates its velocity and position as follows:

\[
V_{k+1}^i = c_1 \cdot r_1 (X_{pb}^i - X_{gb}^i) + c_2 \cdot r_2 (X_{gb}^i - X_k^i) + w \cdot V_k^i, \tag{23}
\]

\[
X_{k+1}^i = V_{k+1}^i + X_k^i, \tag{24}
\]

where \(V_k^i\) is the velocity vector of particle \(i\) in iteration \(k\), \(w\) is the inertia weight coefficient, \(c_1\) and \(c_2\) respectively represent the relative influence of the social and cognitive components, \(X_k^i\) is position of particle \(i\) in iteration \(k\), \(X_{pb}^i\) is personal best position and \(X_{gb}\) is general best position, \(r_1\) and \(r_2\) are random numbers uniformly distributed between \(0\) and \(1\).

5.2. **Hierarchical interactive particle swarm optimization (HIPSO)**

Here, we develop a PSO-based method (denoted as HIPSO) to solve the fuzzy stochastic trilevel linear programming (FSTLP) problem. We use a tri-level decision procedure. Our model consists of three hierarchical levels of decision making, termed as leader in the top-level (distribution center), as follower in the mid-level (manufacturer), and as sub-follower in bottom-level (supplier). The successive decision making take place from the top-level to the mid-level and then to the bottom-level by decision entities individually with the aim of optimizing the respective objectives. In particular, the leader has the top priority to decision making; this decision is implicitly specified by the followers’ actions. Then, the mid-level follower optimizes
its own objective function and reacts to the decision made by the leader to take into view the implicit reactions of the follower of the bottom-level. Ultimately, the decision process is repeatedly executed until the Stackelberg's equilibrium is obtained in the three-level vertical structure. In the following, details of the procedures for solving the problem at hand are described.

(1) Generating initial population
The initial population of particles of size $N$ representing candidate solutions are randomly generated. Each particle is shown as an $M$ dimensional real valued vector, where $M$ is the number of optimized parameters. Initially, we produce the random numbers for the decision variables at the top-level. Then, the mid-level and bottom-level problems are solved to obtain the corresponding solutions by a branch and bound approach.

(2) Evaluating fitness value
The fitness value of the generated particles are obtained by using the top-level objective function. Each particle’s fitness value is compared with its local best solution. If current value is better than local best, then lbest is reset to be the current value. The global best solution, gbest, of the group at every iteration is also identified to be the best value obtained so far, and its index is saved in the variable g.

(3) Updating particles
In every iteration $k$, each particle $i$ tends to move from current position toward a new position in the problem space with velocity $V_{i,k+1}$. The position and velocity vectors of each particle are updated according to (23) and (24).

(4) Termination criterion
HIPSO algorithm is terminated after a criterion is met, typically after reaching the maximum number of iterations or when sufficiently good fitness values are at hand.

(5) Optimization process of the HIPSO algorithm
Based on the above discussions, a flowchart of the proposed HIPSO algorithm is given in Fig.4.

5.3. Simulation-based trilevel optimization

According to the formulation of the problem, each of the imprecise parameters in the constraints as demand, supply, processing time or capacity is represented by a stochastic random number, and in the objective functions, costs and price are represented by fuzzy random numbers. So, the proposed model is a fuzzy and stochastic programming one. Chance-constrained programming technique (CCPT) by Liu [30] is a stochastic programming approach. Based on Liu’s approach,
the main idea is to optimize the critical value of a stochastic or fuzzy constraint, in a predetermined confidence level. Here, we use this idea to tackle uncertainty in uncertain functions. Our chance-constraint programming model is defined as follows:

\[
\begin{align*}
\text{Min} & \quad \bar{f} \\
\text{s.t.} & \quad \text{Cr} \left\{ C(x, \xi) \leq f \right\} \geq \beta_i, \\
& \quad \text{Pr} \left\{ g_k(x, \partial) \leq 0 \right\} \geq \beta_i, \quad i = 2,3,...,n, \quad k = 1,2,...,P,
\end{align*}
\]

where \( \beta_i \in [0,1], i=1,...,n, \) are the acceptable confidence levels, \( x \) denotes the decision variables of the model, \( \xi \) denotes fuzzy random variable, \( \partial \) denotes the stochastic variable, \( \text{Cr} \left\{ \right\} \) is the credibility measure of a fuzzy event and \( \text{Pr} \left\{ \right\} \) denotes the probability of a stochastic event.

In general, solving a mixed fuzzy and stochastic programming tri-level problem directly is difficult. Monte Carlo simulation can be applied as an effective tool for these problems, whose analytical solutions do not exist or are too complex to obtain. We adopted a Monte Carlo simulation approach to search for an optimal solution within the range of random numbers, in order to handle the uncertain parameters of the proposed model. In the followings, we present two simulation-based methods using fuzzy random and stochastic programming, called fuzzy random simulation and stochastic simulation. The fuzzy random simulation and stochastic simulation programs for the CCPT model are shown in details as Algorithm 1 and Algorithm 2, respectively.

**Algorithm 1: Fuzzy random simulation for the objective functions.**

**Step 1.** Sample \( \omega \) according to the probability measure \( \text{Pr} \) from the sample space \( \Omega \).

**Step 2.** Generate \( \omega_1, \omega_2, ..., \omega_n \) randomly from \( \mathcal{E} \) -level set of fuzzy vector \( \xi \), where \( \mathcal{E} \) is a sufficiently small number so that \( \mu(\omega_j) \geq \mathcal{E} \).

**Step 3.** Generate \( \beta \in [0,1] \) uniformly.

**Step 4.** Compute upper bound \( \bar{f}_{\text{min}} = \max \left\{ C(x, \omega_j) \right\} \) and lower bound \( f_{\text{min}} = \min \left\{ C(x, \omega_j) \right\} \).

**Step 5.** Compute \( f_{\text{min}} = \frac{(\bar{f}_{\text{min}} + f_{\text{min}})}{2} \).

**Step 6.** Compute \( \text{Cr}(C(x, \xi) \leq f) \).

**Step 7.** If \( \text{Cr}(C(x, \xi) \leq f) \geq \beta \) then \( f_{\text{min}} = \bar{f}_{\text{min}} \), else \( f_{\text{min}} = f_{\text{min}} \).
**Step 8.** Repeat the two to seven steps until $(\overline{f_{\text{min}}} - f_{\text{min}}) > \phi$ (with $\phi$ a sufficiently small number).

**Step 9.** Return $f_{\text{min}}$.

---

**Algorithm 2: Stochastic simulation for the constraints.**

**Step 1.** Set $p = 0$ and $p' = 0$.

**Step 2.** Generate $\beta \in [0, 1]$ uniformly.

**Step 3.** Generate sample $\omega = (\omega_1, \omega_2, ..., \omega_n)^T$ uniformly according to the probability measure $Pr$ from the sample space $\Omega$.

**Step 4.** If $g_i(x, \partial) \leq 0, \quad i = 1, 2, ..., n$, then $p = p + 1$, else $p' = p' + 1$.

**Step 5.** Let $Q$ be the number of times, we have $g_i(x, \partial) \leq 0, \quad i = 1, 2, ..., n$, and let $\beta = \frac{Q}{N}$.

**Step 6.** If $\frac{Q}{N} \leq \beta$ then the constraints $g_i(x, \partial) \leq 0, \quad i = 1, 2, ..., n$, are feasible else they are not feasible.

Next, we explain the steps of our algorithm in more details in order to clarify the proposed algorithm. The steps of simulation based HIPSO are defined as follows:

**Step 1.** \{Set parameters\}

- a) Set population size $N$.
- b) Set the maximum velocity $V_{\text{max}}$.
- c) Set the inertial weight coefficient $w$.
- d) Set the learning factors $C_1$ and $C_2$.
- e) Set the random numbers between 0 and 1, $\text{rand()}$.
- f) Set mutation rate $m_{\text{rate}}$.

**Step 2.** \{Initialization\}

- a) Create the initial population of particles (solutions) and velocities randomly with initial position $X_{id}$ and speed $V_{id}$.
- b) Create the local best solution as $l_{\text{best}}$.

**Step 3.** Compute the individual fitness value of each particle and store the particle with better fitness value.

**Step 4.** Update the local position best solution and global best position.

**Step 5.** Set $i = 1$

**Step 5.1.** Fix the top-level variables and solve the mid-level problem and obtain the solution.
Step 5.2. Fix the top-level and mid-level variables and solve the bottom-level problem and obtain the solution.

Step 5.3. If the obtained solution is feasible then update the particle else, go to the next step.

Step 5.4. Update local best solution. If $i < N$ then set $i = i + 1$ and go to Step 5.1, else go to step 6.

Step 6. Update the global best solution.

Step 7. If the local and global best solutions change then go to step 9, else go to 8.

Step 8. Apply the mutation operator of GA and update the particle as follows:

$$X_{k+1}^{new} = X_k^i + rand 	imes 3 	imes N[0,1].$$

Step 9. Update all the particles $X_{k+1}^{new}$ using (23) and (24).

Step 10. Compute fitness value for every particle.

Step 11. Update the global best solution.

Step 12. {Check the termination criterion}. If the number of generations reaches the maximum satisfied value then stop, else go to step 2.

6. Computational results

The presented model here is a novel tri-level linear programming one and there is no available test problem for its numerical assessment. So, here we consider generating two different sets of test problems for evaluating the performance of the proposed algorithm. First, 10 sets of small problems are generated to assess the algorithm’s capability in obtaining the global optimum. The problem is solved by Lingo 14 optimization software package for the global optimal values. Second, 10 sets of large problems are randomly generated in order to consider a widespread range of problems. We implement our algorithms in MATLAB 7.12 (R2014a) software environment. All computations were run on a Pentium IV 2.8 GHz processor personal computer with 4 GB memory. The parameter ranges were generated using uniform distributions as shown in Table 3. The population size is 100. The maximum number of iteration is set to be 150. Other parameters including $C_1$, $C_2$, $r$, $V_{max}$, $NPop$ and $NGen$, are respectively set to 2.1, 2.15, 10 and 0.9. These values have been set after experimentation with a simple tuning procedure. Next, we provide the details of the generated test problems and the parameter tunings of the algorithm.

6.1. Description of test problems

6.1.1. Small test problems
We apply the proposed algorithm to solve a set of tri-level problems having three units of every agent, that is, there will be three suppliers, three manufacturers, three distribution centers and three customer groups, for three products and three raw materials in three periods, where there are 658 constraints and 1216 decision variables including 33 binary variables. To solve small problems, the parameters involved in the proposed algorithm are chosen as in Table 3. With these parameters, the proposed method is executed 30 independent times on each of the 10 test problems, and the average of results are noted.

6.1.2. Large test problems
Considering that the resulting model is a tri-level mixed integer linear programming one, direct optimization algorithm can only solve small instances in a reasonable computing time. All the 10 kinds of small instances can be solved by the LINGO software package in a short time, but direct optimization methods are not efficient in terms of computing time needed for large problems. Hence, in order to further assess the proposed HIPSO algorithm in terms of performance and capability, 10 different large instances were generated as shown in Table 4 and the average of results are noted. We applied the proposed algorithms to solve the instances. Using maximum of 100 iterations.

6.2. Comparison
In this section, we are to show experimental results and compare the proposed algorithm with other algorithms. Since the proposed problem is an NP-hard, a simulated based hierarchical interactive particle swarm optimization (Sim-HIPSO) algorithm was proposed. We used optimality difference percent (ODP) as a performance measure for comparing the solutions obtained by the proposed algorithm and the optimal solutions of the tri-level problem:

\[
\text{ODP} = \left( \frac{Z_{\text{alg}} - Z_{\text{op}}}{Z_{\text{op}}} \right) \times 100, \tag{26}
\]

where \( Z_{\text{alg}} \) is the solution value obtained by the proposed algorithm, and \( Z_{\text{op}} \) is the optimal solution of the tri-level problem found by the LINGO software package. Furthermore, a relative difference percent (RDP) is computed to compare the solutions obtained by Sim-HIPSO and GPSO methods:

\[
\text{RDP} = \left( \frac{Z_{\text{alg}} - Z_{\text{GPSO}}}{Z_{\text{GPSO}}} \right) \times 100, \tag{27}
\]

where \( Z_{\text{GPSO}} \) is the solution found by GPSO and \( Z_{\text{alg}} \) is the solution obtained by the proposed algorithm.

Given the results of the proposed algorithm on small test problems, the proposed algorithm is able to find the near optimal solutions in a reasonable computational time, but on small problems LINGO computes optimal solutions in a shorter time. However, the differences between the
global optima obtained by LINGO and the ones obtained by our proposed algorithm are small. Based on the results obtained on small problems, there is only an average of 12% difference with the global optima, which indicates an acceptable performance of the proposed algorithm. The comparative results obtained from 30 runs on small problems for the simulated based HIPSO, the general PSO (GPSO), and LINGO, including the sample maximal, mean, and standard deviations (SD) for the fitness values, and optimality difference percent (ODP) are shown in Table 5. The fitness values of LINGO, GPSO, and simulated based HIPSO shown in Table 5 reveal that there are very small deviations among the results obtained by the two algorithms and the ones obtained using LINGO. This is an indication of the effectiveness of the proposed algorithm. On large test problems, Table 6 presents the results obtained on 30 runs (i.e., sample size) for the simulated based HIPSO, and the general PSO, showing the fitness values, the number of iterations for convergence, the computation times, and RDP (%) values. It can be seen that the simulated based HIPSO was able to search for the global optima better than the general PSO, but was somewhat slower. As the interactive evolutionary mechanism in the simulated based HIPSO assists the algorithm to determine better solutions for the tri-level model, more time is needed for convergence.

In order to evaluate and compare the performance of the simulated based HIPSO algorithm with the ones obtained by GPSO algorithm, a graphical approach has been used. Fig. 5 displays the convergence process of the solution. It is obvious that the proposed algorithm converges to the optimal solution quickly and the fitness values of the solutions obtained after around 200 iterations is approximately equal to that of the optimal solutions. According to these graphs, it can be concluded that simulated based HIPSO algorithm has a better performance than GPSO algorithm in terms of the obtained fitness values.

The ODP and RDP values obtained for both small and large problems are provided in Fig. 6. For small problems, Fig. 6a displays that the ODP values of LINGO are the best, followed by the values of Sim-HIPSO in most cases. From Fig. 6b, in addition, we observe that Sim-HIPSO performs better than other algorithms on large problems in most cases.

We give the details of the solution found on a small problem. This problem involved 4 suppliers, 3 production sites, 6 candidate distribution sites, and 5 customers. Each entity in the decentralized supply chain can gain a suitable profit. The results including the facilities location and allocation are shown in Fig. 7. In this case, distributors 1, 2, 3 and 5 were selected to be established making the profit of 14.25 units, manufacturers 2 and 4 were chosen to build 2 products with 235 and 182 units per period, making a profit 10.34 units. The profit of the raw material supplier is 6.94 units. A total supply chain profit of 20.66 units is gained. Note that the distributer accounts for approximately 69% of the profit. In fact, the distributer gains a considerably higher share of the profit. Result is likely due to the competitive advantage the distributer has of being able to act first. Thus, this entity can make the best decision based on all of the available data. Downstream entities should wait on the distributer to set the economic landscape before they make decisions. The proposed model being able to efficiently solve small and large problems comprising inherent uncertain parameters.
7. Concluding remarks and suggestions for future studies

We studied the decentralized supply chain network as a multi-period multi-product procurement-production-distribution planning problem considering service level in presence of multiple uncertainties in procurement, production and distribution layers. The proposed model incorporated strategic decisions as well as tactical and operational decisions such as production planning, logistic and inventory decisions. The problem was formulated as a fuzzy-stochastic tri-level programming model. The model being an uncertain mixed-integer tri-level problem, a hybrid particle swarm optimization algorithm along with a fuzzy simulation technique and a stochastic simulation was designed to solve the proposed model. The effectiveness of the developed fuzzy stochastic optimization model as well as the usefulness of the proposed solution approach were investigated by solving both small and large problems. The model was solved to optimality on small problems. In order to evaluate the application of the model and the proposed algorithm, ten sets of small and large problems were randomly generated. The experimental results revealed that our proposed fuzzy-stochastic simulation based hierarchical interactive particle swarm optimization algorithm performed better in finding the best solutions within reasonable computing times and was effective on all the test problems. Our study, by presenting a comprehensive fuzzy and stochastic programming model, makes new contributing to the decentralized supply chain (SC) network literature. The provided model and solution methodology offer useful guidelines for the design of complex decentralized SC networks.

For future studies, incorporation of discounts on customer’ orders or inflation in system costs can be considered. Also, a more complete explanation of the features of the network can be raised by considering distance parameters between the layers or considering trip times for transportation of the products to the customer nodes. Consideration of man-made or natural disruptions such as terrorist attacks, floods, earthquakes, and economic crises and application of the robust optimization to the decentralized supply chain network design problem under uncertainty are interesting to be investigated. The proposed model may be considered by practical case studies, such as cement, brick and iron industries.

Acknowledgments

The first two authors thank Mazandaran University of Science and Technology, the third author thanks Sharif University of Technology and the fourth author thanks Babol University of Technology for supporting this work.

References


Tables:

Table 1. Review of decentralized SCN problem
Table 2. Uncertain parameters of the model
Table 3. Range of parameters in the numerical study
Table 4. Description of large test problems
Table 5. Obtained results by LINGO, GPSO and simulated-based HIPSO on small test problems
Table 6. Obtained results by GPSO and simulated-based HIPSO on large test problems

Figures:

Figure 1. An example of an $\alpha$–cut .
Figure 2. An architecture of a four-echelon supply chain network.
Figure 3. Decentralized PPDP model based on a tri-level structure.
Figure 4. A flow chart for the proposed simulated-based HIPSO algorithm.
Figure 5. Comparative convergence results between the simulated HIPSO and GPSO algorithms.
Figure 6. Comparative ODP% and RDP% results for different problems
Figure 7. Results for the small-scale case study considered in this example.
Table 1. Review of decentralized SCN problem

<table>
<thead>
<tr>
<th>References</th>
<th>Chain levels</th>
<th>Uncertain parameter</th>
<th>Type of problem</th>
<th>Type of modelling</th>
<th>Solution approach</th>
</tr>
</thead>
</table>


<table>
<thead>
<tr>
<th></th>
<th>LA: Location-Allocation</th>
<th>PD: Production-Distribution</th>
<th>PDP: Procurement-Production-Distribution</th>
<th>Bi-level</th>
<th>Tri-level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vallejo et al. [24]</td>
<td>3</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>2</td>
<td>Ma et al. [23]</td>
<td>3</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>3</td>
<td>Calvete et al. [2]</td>
<td>3</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>4</td>
<td>Calvete et al. [14]</td>
<td>3</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>5</td>
<td>Wang and Lee [11]</td>
<td>3</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>6</td>
<td>Guo et al. [9]</td>
<td>2</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>7</td>
<td>Xu et al. [16]</td>
<td>4</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>8</td>
<td>Jia et al. [35]</td>
<td>3</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>9</td>
<td>Saranwang and Likasiri [36,37]</td>
<td>3</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>10</td>
<td>Yue and You [38]</td>
<td>2</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>11</td>
<td>Fard and Hajiaghaee [39]</td>
<td>2</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>12</td>
<td>Present work</td>
<td>4</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

* LA: Location-Allocation  
PD: Production-Distribution  
PDP: Procurement-Production-Distribution
### Table 2. Uncertain parameters of the model

<table>
<thead>
<tr>
<th>Source of uncertainty in supply chains</th>
<th>Fuzzy and stochastic coefficient</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand</td>
<td>Product demand</td>
<td>$D^k_{it}$</td>
</tr>
<tr>
<td></td>
<td>Delay time</td>
<td>$lt_{jit}$</td>
</tr>
<tr>
<td></td>
<td>Price of product</td>
<td>$pp^k_{it}$</td>
</tr>
<tr>
<td>Process</td>
<td>Processing time</td>
<td>$pt^k_{it}$, $st^k_{it}$, $R^k_{it}$</td>
</tr>
<tr>
<td></td>
<td>Production capacity</td>
<td>$A^k_{it}$</td>
</tr>
<tr>
<td>Supply</td>
<td>Price of raw material</td>
<td>$p^k_{it}$</td>
</tr>
<tr>
<td></td>
<td>supplier capacity</td>
<td>$s^l_{dt}$</td>
</tr>
</tbody>
</table>

### Table 3. Range of parameters in the numerical study

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Range</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pc^k_{mt}$</td>
<td>Uniform ~ [3,8] to Uniform ~ [45,80]</td>
<td>Dollar ($) per unit</td>
<td>Production cost</td>
</tr>
<tr>
<td>$pcr^l_{it}$</td>
<td>Uniform ~ [0.2,2] to Uniform ~ [15,30]</td>
<td>Dollar ($) per unit</td>
<td>Backorder cost</td>
</tr>
<tr>
<td>$bc^k_{it}$</td>
<td>Uniform ~ [10,50] to Uniform ~ [10,50]</td>
<td>Dollar ($)</td>
<td>Inventory cost</td>
</tr>
<tr>
<td>$scp^k_{mt}$</td>
<td>Uniform ~ [0.2,2] to Uniform ~ [200,600]</td>
<td>Dollar ($) per unit</td>
<td>Setup cost</td>
</tr>
<tr>
<td>$scs^l_{it}$</td>
<td>Uniform ~ [8,20] to Uniform ~ [70,200]</td>
<td>Dollar ($) per unit</td>
<td>Transportation cost</td>
</tr>
<tr>
<td>$hp^k_{mt}$</td>
<td>Uniform ~ [5,15] to Uniform ~ [4,10]</td>
<td>Dollar ($) per unit</td>
<td>Price</td>
</tr>
<tr>
<td>$hd^l_{it}$</td>
<td>Uniform ~ [1,4] to Uniform ~ [10,20]</td>
<td>Dollar ($) per unit</td>
<td>Demand</td>
</tr>
<tr>
<td>$hs^k_{it}$</td>
<td>Uniform ~ [4,10] to Uniform ~ [5,15]</td>
<td>Dollar ($) per unit</td>
<td>Processing time</td>
</tr>
<tr>
<td>$tp^l_{mt}$</td>
<td>Uniform ~ [10,25] to Uniform ~ [90,130]</td>
<td>Dollar ($) per Ton</td>
<td>Capacity</td>
</tr>
<tr>
<td>$td^k_{it}$</td>
<td>Uniform ~ [2,10] to Uniform ~ [50,85]</td>
<td>Dollar ($) per unit</td>
<td>Capacity</td>
</tr>
<tr>
<td>$ts^l_{it}$</td>
<td>Uniform ~ [5,12] to Uniform ~ [65,100]</td>
<td>Dollar ($) per unit</td>
<td>Capacity</td>
</tr>
<tr>
<td>$pp^k_{mt}$</td>
<td>Uniform ~ [35,100] to Uniform ~ [260,450]</td>
<td>Dollar ($) per unit</td>
<td>Capacity</td>
</tr>
<tr>
<td>$pr^l_{mt}$</td>
<td>Uniform ~ [10,20] to Uniform ~ [15,30]</td>
<td>Dollar ($) per unit</td>
<td>Capacity</td>
</tr>
<tr>
<td>$d^k_{it}$</td>
<td>Uniform ~ [25,80] to Uniform ~ [100,500]</td>
<td>Ton</td>
<td>Demand</td>
</tr>
<tr>
<td>$pt^l_{mt}$</td>
<td>Uniform ~ [0.1,0.95] to Uniform ~ [10,15]</td>
<td>Second</td>
<td>Processing time</td>
</tr>
<tr>
<td>$st^k_{mt}$</td>
<td>Uniform ~ [10,30] to Uniform ~ [25,40]</td>
<td>Second</td>
<td>Processing time</td>
</tr>
<tr>
<td>$tt^l_{mt}$</td>
<td>Uniform ~ [150,500] to Uniform ~ [3500,4000]</td>
<td>Second</td>
<td>Processing time</td>
</tr>
<tr>
<td>$s^l_{it}$</td>
<td>Uniform ~ [150,300] to Uniform ~ [400,1000]</td>
<td>Ton</td>
<td>Capacity</td>
</tr>
<tr>
<td>$A^k_{mt}$</td>
<td>Uniform ~ [100,590] to Uniform ~ [5000,10000]</td>
<td>Ton</td>
<td>Capacity</td>
</tr>
<tr>
<td>$WP^k_{mt}$</td>
<td>Uniform ~ [80,350] to Uniform ~ [1500,2500]</td>
<td>Ton</td>
<td>Capacity</td>
</tr>
<tr>
<td>$WR^k_{mt}$</td>
<td>Uniform ~ [50,200] to Uniform ~ [100,400]</td>
<td>Ton</td>
<td>Capacity</td>
</tr>
<tr>
<td>$W^l_{it}$</td>
<td>Uniform ~ [400,600] to Uniform ~ [2500,3500]</td>
<td>Ton</td>
<td>Capacity</td>
</tr>
<tr>
<td>$R^k_{mt}$</td>
<td>Uniform ~ [150,250] to Uniform ~ [10,20]</td>
<td>Ton</td>
<td>Capacity</td>
</tr>
<tr>
<td>$p^l_{it}$</td>
<td>Uniform ~ [2,4] to Uniform ~ [3,7]</td>
<td>-</td>
<td>Bill of material</td>
</tr>
</tbody>
</table>
\[ L_d \]

<table>
<thead>
<tr>
<th>( L_d )</th>
<th>Uniform (-1,8)</th>
<th>Uniform (-1,15)</th>
<th>Day</th>
<th>Delay time</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>Large number</td>
<td>Large number</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Description of large test problems

| Problem | Indices \( d \) \| m \| j \| i \| k \| l \| t | Number of variables | Number of constraints |
|---------|----------------|----------------|------|--------------|
|         | Total          | Binary         |      |              |
| L1      | 3              | 4              | 5    | 3            | 3              | 2              | 3              | 1651          | 35            | 947        |
| L2      | 4              | 5              | 4    | 5            | 3              | 2              | 4              | 2689          | 56            | 1465       |
| L3      | 4              | 5              | 6    | 8            | 4              | 3              | 5              | 6586          | 110           | 3623       |
| L4      | 5              | 6              | 6    | 9            | 5              | 4              | 6              | 12086         | 191           | 6350       |
| L5      | 5              | 6              | 7    | 10           | 5              | 5              | 7              | 18301         | 193           | 9654       |
| L6      | 5              | 8              | 8    | 12           | 4              | 5              | 7              | 19491         | 209           | 9662       |
| L7      | 6              | 7              | 8    | 15           | 5              | 4              | 8              | 26319         | 259           | 13791      |
| L8      | 7              | 8              | 8    | 12           | 6              | 6              | 8              | 24323         | 399           | 16775      |
| L9      | 6              | 9              | 9    | 15           | 5              | 5              | 9              | 36317         | 420           | 19526      |
| L10     | 8              | 6              | 15   | 20           | 8              | 5              | 10             | 90331         | 311           | 46683      |

Table 5. Obtained results by LINGO, GPSO and simulated-based HIPSO on small test problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Fitness value (mean)</th>
<th>RDP%</th>
<th>Number of iterations to converge</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GPSO</td>
<td>Sim-HIPSO</td>
<td>Max</td>
<td>Mean</td>
</tr>
<tr>
<td>S1</td>
<td>21287</td>
<td>29276</td>
<td>22499</td>
<td>5.7</td>
</tr>
<tr>
<td>S2</td>
<td>20142</td>
<td>23821</td>
<td>21374</td>
<td>6.1</td>
</tr>
<tr>
<td>S3</td>
<td>49020</td>
<td>56679</td>
<td>51643</td>
<td>5.4</td>
</tr>
<tr>
<td>S4</td>
<td>18094</td>
<td>28876</td>
<td>19288</td>
<td>6.6</td>
</tr>
<tr>
<td>S5</td>
<td>17376</td>
<td>28839</td>
<td>19384</td>
<td>11.6</td>
</tr>
<tr>
<td>S6</td>
<td>15994</td>
<td>21265</td>
<td>16703</td>
<td>4.4</td>
</tr>
<tr>
<td>S7</td>
<td>22777</td>
<td>26004</td>
<td>23990</td>
<td>5.3</td>
</tr>
<tr>
<td>S8</td>
<td>18009</td>
<td>22694</td>
<td>20231</td>
<td>12.3</td>
</tr>
<tr>
<td>S9</td>
<td>23697</td>
<td>31153</td>
<td>24872</td>
<td>5.0</td>
</tr>
<tr>
<td>S10</td>
<td>34623</td>
<td>47003</td>
<td>37486</td>
<td>8.3</td>
</tr>
</tbody>
</table>

Table 6. Obtained results by GPSO and simulated-based HIPSO on large test problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Fitness value (mean)</th>
<th>RDP%</th>
<th>Number of iterations to converge</th>
<th>CPU time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GPSO</td>
<td>Sim-HIPSO</td>
<td>Max</td>
<td>Mean</td>
</tr>
<tr>
<td>L1</td>
<td>65054</td>
<td>61299</td>
<td>5.77</td>
<td>19</td>
</tr>
<tr>
<td>L2</td>
<td>79429</td>
<td>72918</td>
<td>8.20</td>
<td>31</td>
</tr>
<tr>
<td>L3</td>
<td>85819</td>
<td>80611</td>
<td>6.07</td>
<td>48</td>
</tr>
<tr>
<td>L4</td>
<td>99633</td>
<td>95466</td>
<td>4.18</td>
<td>61</td>
</tr>
<tr>
<td>L5</td>
<td>1157173</td>
<td>1056480</td>
<td>8.70</td>
<td>83</td>
</tr>
<tr>
<td>L6</td>
<td>1581107</td>
<td>1475062</td>
<td>6.71</td>
<td>143</td>
</tr>
<tr>
<td>L7</td>
<td>1768096</td>
<td>1552178</td>
<td>12.21</td>
<td>177</td>
</tr>
<tr>
<td>L8</td>
<td>5617377</td>
<td>5016614</td>
<td>10.69</td>
<td>256</td>
</tr>
<tr>
<td>L9</td>
<td>6774706</td>
<td>6127052</td>
<td>9.56</td>
<td>415</td>
</tr>
<tr>
<td>L10</td>
<td>8758024</td>
<td>8197365</td>
<td>6.40</td>
<td>438</td>
</tr>
</tbody>
</table>
Figure 1. An example of an $\alpha$-cut.

Figure 2. An architecture of a four-echelon supply chain network.
Figure 3. Decentralized PPDP model based on a tri-level structure.
Figure 4. A flow chart for the proposed simulated-based HIPSO algorithm

Figure 5. Comparative convergence results between the simulated HIPSO and GPSO algorithms.
Figure 6. Comparative ODP% and RDP% results for different problems.
Figure 7. Results for the small-scale case study considered in this example.
Biographies:

Raheleh Nourifar is a PhD candidate of Industrial Engineering at Babol Mazandaran University of Science and Technology and Faculty member of university college of Sarouye. He has graduated in BSc. and MSc. of industrial engineering from Mazandaran University of Science and Technology, Babol, Iran, in 2006 and 2008, respectively. His research interests are supply chain management, multi-criteria decision making techniques and network programming.

Iraj Mahdavi is the full Professor of Industrial Engineering at Mazandaran University of Science and Technology, Babol, Iran. He received his Ph.D. from India in Production Engineering. He is also in the editorial board of five journals and scientific committee member of international conferences. He was awarded as the best researcher of engineering area in Iran and among the best professors of Iran. He has published over 350 research papers. His research interests include production planning, supply chain management, fuzzy networks, digital management and intelligent operations management. He also edited and authored 2 books in electronic supply network design and multi-objective optimization, respectively.

Nezam Mahdavi-Amiri is a full professor of Mathematical Sciences at Sharif University of Technology. He received his Ph.D. degree from Johns Hopkins University in Mathematical Sciences in 1981. He has been on the editorial board of several mathematical and computational journals in Iran including Bulletin of the Iranian Mathematical Society (Style- Language Editor), the Iranian Journal of Operations Research (Editor-in-Chief) and the CSI Journal of Computer Science and Engineering. He was also the Editor-in-Chief of Mathematical Thought and Culture (in Persian), a journal of Mathematical Society of Iran, and on the Executive Council of the Iranian Mathematical Society as well as the Computer Society of Iran. He is currently Vice President of the Operations Research Society of Iran and its representative to IFORS. His research interests include Optimization, Numerical Analysis and Scientific Computing, Matrix Computations, Fuzzy Modeling and Computing.

Mohammad Mahdi Paydar is the Assistant Professor of Industrial Engineering at Babol Noshirvani University of Technology. He received his PhD in Industrial Engineering from Iran University of Science and Technology. His research interests are cellular manufacturing systems, supply chain design and modelling of manufacturing applications. He has published articles in journals such as Computers and Industrial Engineering, Computers and Operations Research, Expert Systems with Applications, Computers and Chemical Engineering, International Journal of Advanced Manufacturing Technology, Journal of Manufacturing Systems, International Journal of Production Research and International Journal of Operational Research and 25 papers in international conferences.