Fuzzy Optim Decis Making (2015) 14:265–287
DOI 10.1007/s10700-014-9200-6

Multistage production distribution under uncertain demands with integrated discrete particle swarm optimization and extended priority-based hybrid genetic algorithm

Thitipong Jamrus · Chen-Fu Chien · Mitsuo Gen · Kanchana Sethanan

Received: 31 July 2013 / Accepted: 29 December 2014 / Published online: 8 January 2015

Abstract Production distribution systems are increasingly crucial because of shortened product life cycles, increasing competition, and uncertainty introduced by globalization. Production distribution involves a multistage supply chain network that consists of factories, distribution centers, retailers, and various customers. Customer demands fluctuate and are unpredictable, thereby causing an imprecise customer quantity demand in each period in the production distribution model, and increasing inventory and related costs. Most studies have addressed the production distribution problem with certain demands or a single period. To fill the gap, this study aims to integrate the extended priority-based discrete particle swarm optimization and novel extended priority-based hybrid genetic algorithm for solving flexible multistage production distribution under uncertain demands in multiple periods. In particular, triangular fuzzy demands are considered for minimizing the total cost, including transportation costs, inventory costs, shortage costs, and ordering costs, in the multistage and multi-time-period supply chain. For validation, we designed numerical experiments to compare the proposed approaches with LINGO computational software (for small problems)

T. Jamrus · C.-F. Chien (✉)
Department of Industrial Engineering and Engineering Management,
National Tsing Hua University, Hsinchu 30013, Taiwan
e-mail: cfchien@mx.nthu.edu.tw

T. Jamrus
e-mail: thitipong1987@gmail.com

M. Gen
Fuzzy Logic System Institute, Fukuoka 820-0067, Japan
e-mail: mitsuogen@gmail.com

K. Sethanan
Department of Industrial Engineering, Khon Kaen University, Khon Kaen 40002, Thailand
e-mail: skanch@kku.ac.th

and conventional genetic algorithms (for normal problems) in real settings. The experimental results demonstrated practical viability of the proposed approaches.

Keywords Triangular fuzzy demand · Multistage and multi-time-period (MSMT) supply chain · Extended priority-based hybrid genetic algorithm (EP-HGA) · Extended priority-based discrete particle swarm optimization (EP-DPSO)

1 Introduction

Production distribution and demand fulfillment is increasingly challenging because of shortening product life cycles, rapid technological change, increasing competition, and uncertainty introduced by globalization (Chien et al. 2013). The manufacturing paradigm is shifting, in which the conventional vertically integrated supply chain has been altered by collaborations among many fragmented yet complementary and specialized companies in production distribution for value constellations (Chien et al. 2010). The design of a long-term efficient production distribution network is a crucial strategic decision that affects supply chain performance and the manufacturing network (Xu et al. 2008).

The present production distribution problem with the multistage and multi-time-period (MSMT) supply chain consists of factories, distribution centers, retailers, and various customers. The present problem is crucial in supply chain management, in which an optimal platform is required to solve it systemically for efficient and effective supply chain management (Lin et al. 2009). In a flexible multistage logistics network, products can be delivered to customers from plants or distribution centers instead of retailers, thereby decreasing delivery time and increasing customer demand satisfaction. Customer demands in practice are typically imprecise in each period of multi-time-period production distribution models, thus causing supplier production distribution complexity in meeting fluctuating demands in a multi-time period. The supply chain must be prepared for demand unpredictability, disruption, or customer delivery delay, in which the multi-time period or infinite-horizon production distribution problem is unobtainable over the planning horizon until supply is insufficient to meet demand. Hence, decision makers attempting to manage this problem should consider uncertain situations such as the amount of inventory and production for meeting fluctuating demands in each period and providing flexibility. However, such consideration can be costly and risky (Chien et al. 2012). Indeed, the major goal is to determine the processes to minimize maximum potential regret and risk including the total cost (Chien and Zheng 2012).

In practice, the data in production distribution systems and supply chains are characteristically fuzzy, whereas customer demand is typically uncertainly quantified in supply chain procedures. Thus, conventional models with constant demand may not be appropriate for managing the fuzziness of production distribution systems that are necessary for effectively determining solutions. Thus, we addressed the MSMT problem with uncertain demands.

This study aims to develop a flexible MSMT production distribution under uncertain demands. We propose extended priority-based discrete particle swarm optimization

(EP-DPSO) and new extended priority-based hybrid genetic algorithm approach (nEP-HGA) approach for solving uncertain demands in the flexible MSMT supply chain model. For validation, we compared the proposed approach with both the traditional GA approach and LINGO computational software. The approaches were solved under triangular fuzzy demands to satisfy demands with the minimal total cost by considering the transportation costs, inventory costs, shortage costs, and ordering costs of the supply chain network.

The remainder of this paper is organized as follows: Sect. 2 reviews the literature on the production distribution problem. Section 3 introduces the uncertain function of the triangular fuzzy number. Section 4 describes the formulation of the mathematical model for MSMT production distribution. Section 5 describes the proposed solution methodology of the DPSO approach combined with EP-based encoding/decoding and the nEP-HGA. Section 6 describes the proposed adaptive auto tuning that entails using the FLC for enhancing the evolutionary process. Section 7 presents examples of numerical experiments of the model for comparing EP-DPSDO and the nEP-HGA with LINGO and the GA. Section 8 concludes the paper with a discussion on contributions and future research directions.

2 Literature review

Various studies have considered different time-period classifications for each complexity of the problem that can be divided into two parts: a single period and a multi-time period. Several researchers have solved single-time-period production distribution problems. For example, Chan et al. (2005) proposed an HGA in multi-factory supply chain models by using the analytic hierarchy process (AHP) for assigning weightings among costs, day, and use, and combined the GA for allocating jobs into suitable production plants. Xu et al. (2008) discussed the supply-chain network design problem for shipping costs and customer demands by using random fuzzy variables. Using spanning tree-based genetic algorithms (st-GAs) and applying the Prüfer number to solve this problem, they determined multi-objectives for minimizing the total cost and maximizing customer services. Costa et al. (2010) proposed an innovative encoding-decoding procedure with the GA for determining the location and opening of facilities in a single-product three-stage supply chain network. They proposed a new, efficient chromosome representation procedure based on a parsimonious permutation decoding of the network string representation for reducing infeasible transportation trees.

However, actual production distribution problems involve the multi-time-period that demands period fluctuations and customer quantity demands (Chamnanlor et al. 2014). Wang and Hsu (2012) applied the possibilistic mean and mean square imprecision index of the shortage and surplus for uncertain factors in the closed-loop logistics model. Zeballos et al. (2014) proposed mixed-integer linear programming for 10 layers (five stages) of closed-loop supply chains with uncertain supply and demand. Their objective function minimizes the expected costs of facilities, purchasing, storage, transportation, and emissions, minus the expected revenue of returned products. Sethanan et al. (2013) considered production planning, ordering, and inventory in analyzing costs for solving the supply chain problem.

Table 1 Related studies

Study	No. of states	Objective	Demands classification	Time classification	Method
Chan et al. (2005)	2	Min. total cost	Certainty	Single period	AHP and GA
Varthanan et al. (2013)	2	Min. total cost	Uncertainty	Multi period	DPSO
Costa et al. (2010)	4	Min. total cost	Certainty	Single period	New encoding/ decoding and GA
Xu et al. (2008)	4	Min. total cost/ Max. service level	Uncertainty	Single period	Spanning tree and Prüfer number
Zeballos et al. (2014)	5	Min. expected cost	Uncertainty	Multi period	mixed-integer linear programming

In this study, we propose an approach integrating discrete particle swarm optimization, the extended priority based-encoding/decoding (nEP-HGA), and a fuzzy logic controller (FLC) for solving complex factor-production distribution problems such as uncertain demands in the MSMT supply chain model. For example, Duenas and Petrovic (2008) proposed a multi-objective genetic algorithm for single machine scheduling under fuzziness. We formulated this problem with triangular fuzzy customer demand to determine which customers required delivery in each period, and selected the distribution center or retailers for each customer in the production distribution network for satisfying customer demands, and for comparing the proposed approach with the traditional genetic algorithm (GA) approach. The objective function in this problem is the minimal total cost under triangular fuzzy customer demands that involve considering transportation costs, inventory costs, and ordering costs.

Most existing studies focused on single-time-period production distribution problems that have both unpredictable and certain factors. However, the complexity of production distribution problems includes imprecise periods and uncertain customer demands. Several studies have proposed various approaches for solving this problem. Table 1 lists review of related studies and their classifications. The proposed approach aims to minimize total costs of the multistage and multi-time-period in production distribution problems.

3 Mathematical model for multistage and multi-time-period supply chain

The indices, parameters, and decision variables are listed as follows:
Indices

i index of plants ($i = 1, 2, \dots, I$)
j index of distribution centers ($j = 1, 2, \dots, J$)
k index of retailers ($k = 1, 2, \dots, K$)
l index of customers ($l = 1, 2, \dots, L$)
t index of time period ($t = 1, 2, \dots, T$)

Parameters

I number of plants
J number of distribution centers

K	number of retailers
L	number of customers
T	total time period
P_i	plant i
DC_j	distribution center j
R_k	retailer k
$\tilde{d}_l(t)$	customer uncertain demand l at time t
c_{ij}^1	unit shipping cost of product from plant i to distribution center j
c_{ik}^2	unit shipping cost of product from plant i to retailer k
c_{jk}^3	unit shipping cost of product from distribution center j to retailer k
c_{jl}^4	unit shipping cost of product from distribution center j to customer l
c_{kl}^5	unit shipping cost of product from retailer k to customer l
f_j^1	fix charge of products transportation in distribution center j
f_k^2	fix charge of products transportation to retailer k
h_j^1	unit holding cost of inventory per period at distribution center j
h_k^2	unit holding cost of inventory per period at retailer k
$pr_i(t)$	amount of products from plant i at time t
t_{ij}^{L1}	lead time for transporting products from plant i to distribution center j
t_{ik}^{L2}	lead time for transporting products from plant i to retailer k
t_{jk}^{L3}	lead time for transporting products from distribution center j to retailer k
t_{jl}^{L4}	lead time for transporting products from distribution center j to customer l
t_{kl}^{L5}	lead time for transporting products from retailer k to customer l
u_j^1	upper bound of capacity of distribution center j
u_k^2	upper bound of capacity of retailer k
α_x	$= \begin{cases} 1, & x \neq 0 \\ 0, & \text{otherwise} \end{cases}$ α_x is the step function

Decision variables

$x_{ij}^1(t)$	amount of transporting product from plant i to distribution center j at time t
$x_{ik}^2(t)$	amount of transporting product from plant i to retailer k at time t
$x_{jk}^3(t)$	amount of transporting product from distribution center j to retailer k at time t
$x_{jl}^4(t)$	amount of transporting product from distribution center j to customer l at time t
$x_{kl}^5(t)$	amount of transporting product from retailer k to customer l at time t
$y_j^1(t)$	amount of inventory in distribution center j at time t
$y_k^2(t)$	amount of inventory in retailer k at time t

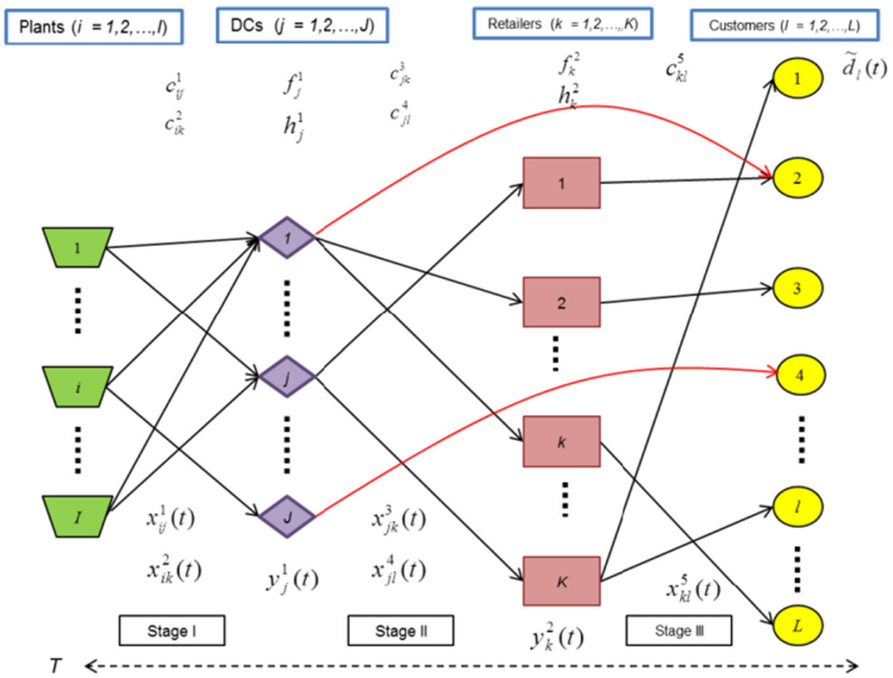


Fig. 1 Flexible multistage logistics network model

The proposed model was formulated based on a mixed integer programming model for a multistage supply chain network. This problem involves two deliveries—normal delivery from the first stage to the next closing stage and direct delivery—for transporting the product from distribution centers to customers instead of from retailers (Fig. 1).

The total cost function of any of the following cost items is calculated as the sum of an item's costs over the planning horizon T. The objective function is to minimize costs that consist of ordering costs, holding costs, and transportation costs. The detailed cost items and constraints are shown as follows:

$$\begin{aligned}\min z = \sum_{t=0}^{T}\Bigg[&\sum_{i=1}^{I}\sum_{j=1}^{J} c^1_{ij}x^1_{ij}(t) + \sum_{i=1}^{I}\sum_{k=1}^{K} c^2_{ik}x^2_{ik}(t) + \sum_{j=1}^{J}\sum_{k=1}^{K} c^3_{jk}x^3_{jk}(t)\\ &+ \sum_{j=1}^{J}\sum_{l=1}^{L} c^4_{jl}x^4_{jl}(t) + \sum_{k=1}^{k}\sum_{l=1}^{L} c^5_{kl}x^5_{kl}(t) + \sum_{i=1}^{I}\sum_{j=1}^{J} f^1_j\alpha(x^1_{ij}(t))\\ &+ \sum_{i=1}^{I}\sum_{k=1}^{K} f^2_k\alpha(x^2_{ik}(t)) + \sum_{j=1}^{J}\sum_{k=1}^{K} f^2_k\alpha(x^3_{jk}(t))\\ &+ \sum_{j=1}^{J} h^1_j y^1_j(t) + \sum_{k=1}^{K} h^2_k y^2_k(t)\Bigg] \qquad (1)\end{aligned}$$

s. t.

$$y_j^1(t-1)+\sum_{i=1}^{I} x_{ij}^1(t-t_{ij}^{L3})-\sum_{k=1}^{K} x_{jk}^3(t-t_{jk}^{L3})=y_j^1(t),\quad \forall j,t \tag{2}$$

$$y_k^2(t-1)+\sum_{i=1}^{I} x_{ik}^2(t-t_{ik}^{L2})+\sum_{j=1}^{J} x_{jk}^3(t-t_{jk}^{L3})$$

$$-\sum_{l=1}^{L} x_{kl}^5(t-t_{kl}^{L5})=y_k^2(t),\quad \forall k,t \tag{3}$$

$$\sum_{t=1}^{T}\left(\sum_{j=1}^{J} x_{ij}^1(t)+\sum_{k=1}^{K} x_{ik}^2(t)\right)\leq \sum_{t=1}^{T} pr_i(t),\quad \forall i,t \tag{4}$$

$$\sum_{j=1}^{J} x_{jl}^4(t-t_{jl}^{L4})+\sum_{k=1}^{K} x_{kl}^5(t-t_{kl}^{L5})\geq \tilde{d}_l(t),\quad \forall l,t \tag{5}$$

$$y_j^1(t)\leq u_j^1\quad \forall j,t \tag{6}$$

$$y_k^2(t)\leq u_k^2\quad \forall k,t \tag{7}$$

$$x_{ij}^1(t),x_{ik}^2(t),x_{jk}^3(t),x_{jl}^4(t),x_{kl}^5(t),y_j^1(t),y_k^2(t)\geq 0,\quad \forall i,j,k,l,t \tag{8}$$

The objective function (1) is to minimize total costs, which are shipping transportation costs (1st through 5th terms), fixed charge of product transportation costs (6th through 8th terms), and holding costs (9th through 10th terms). Constraint (2) specifies the balanced inventory of distribution centers, and Constraint (3) specifies the balanced inventory of retailers. Constraint (4) ensures that the total quantities of transporting products do not exceed the production limit of the plant. Constraint (5) is the demand satisfaction constraint. Constraints (6) and (7) ensure that the distribution centers and retailer do not exceed their capacities. Finally, Constraint (8) is the variable domain constraint.

4 Fuzzy numbers

Fuzzy theory (Zadeh, 1965) proposes mathematical techniques for deriving particular solutions. Chien et al. (2011) developed a comprehensive modular framework to derive various configurations of fuzzy numbers for fuzzy ranking. Kumar (2012) summarized the definitions of the arithmetic operations of fuzzy numbers as follows:

If X is a collection of objects, then the fuzzy subset $\tilde{a}$ of X is defined as a set of ordered pairs:

$$\tilde{a}=\{(x),\mu_{\tilde{a}}(x)|x\in X\}$$

where $\mu_{\tilde{a}}(x)$ is the membership function for the fuzzy set $\tilde{a}$. The membership function maps each element of X to a membership grade or membership value between 0 and 1.

A real fuzzy number $\tilde{a} = [a_L\, a_M\, a_U]$, when defined as a triangular fuzzy number when $a_L \leq a_M \leq a_U$, is a fuzzy subset of the set of real numbers R, with membership function $\mu_{\tilde{a}}$ satisfying the following conditions:

$\mu_{\tilde{a}}$ is continuous from R to the closed interval [0,1];
$\mu_{\tilde{a}}$ is strictly increasing and continuous $[a_L\, a_M]$.

The membership function $\mu_{\tilde{a}}$ is strictly decreasing and continuous $[a_M\, a_U]$, where a_M and a_U are real numbers.

Let $[\tilde{a} = a_L\, a_M\, a_U]$ and $\tilde{b} = [b_L\, b_M\, b_U]$ be two triangular fuzzy numbers; then, define the arithmetic operations of $\tilde{a}$ and $\tilde{b}$ as

$$\begin{aligned} \text{Addition: } \tilde{a} + \tilde{b} &= \{[a_L + b_L][a_M + b_M][a_U + b_U]\} \\ \text{Subtraction: } \tilde{a} - \tilde{b} &= \{[a_L - b_L][a_M - b_M][a_U - b_U]\} \end{aligned}$$

The triangular fuzzy number is frequently used for practical purposes (Li et al. 1996, 2014). Chien et al. (2011) developed a comprehensive modular fuzzy ranking framework that can derive various configurations of fuzzy numbers for fuzzy ranking. In particular, the membership function μã(x) of the triangular fuzzy number ã is expressed as follows:

$$\mu_{\tilde{a}}(x) = \begin{cases} 0 & \text{for } x \leq a_L \\ (x - a_L)/(a_M - a_L) & \text{for } a_L \leq x \leq a_M \\ 1 & \text{for } x = a_M \\ (a_U - x)/(a_U - a_M) & \text{for } a_M \leq x \leq a_U \\ 0 & \text{for } x \geq a_U \end{cases}$$

According to t definition of a triangular fuzzy number as show in Fig. 2, let ã = (aL(r), aU(r)), $(0 \leq r \leq 1)$ be a fuzzy number. Then, calculate the average value of the triangular fuzzy number $\bar{a}_0$ as follows:

$$\bar{a}_0 = \frac{1}{2}\int_0^1 \{a_L(r) + a_U(r)\}\mathrm{d}r = \frac{1}{4}[2a_M + a_L + a_U]$$

If ā ω = (aL(r), aU(r)) = {aL + r(aM − aL)/ ω, aU + r(aM − aU)/ ω} is an arbitrary triangular fuzzy number at a decision level higher than "α" for α, ω ∈ [0, 1], the value ā ω can be calculated as follows:

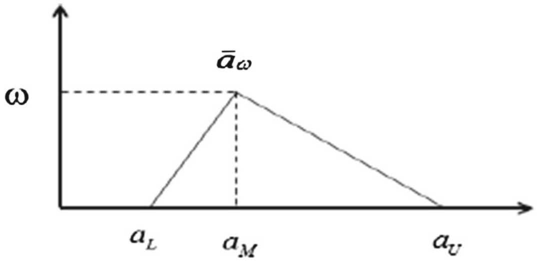
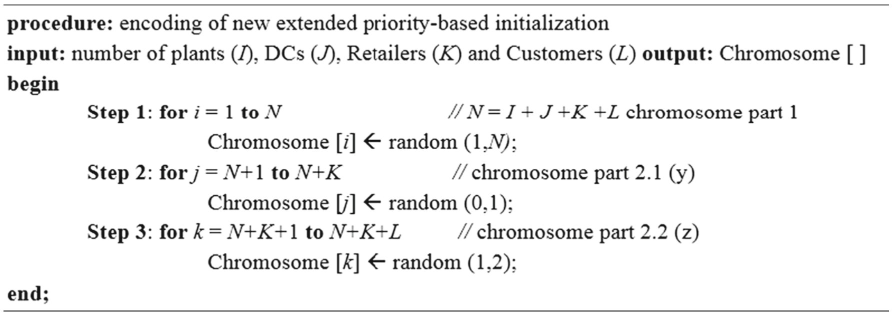


Fig. 2 A triangular fuzzy number



```
procedure: encoding of new extended priority-based initialization
input: number of plants ($I$), DCs ($J$), Retailers ($K$) and Customers ($L$) output: Chromosome [ ]
begin
        Step 1: for $i = 1$ to $N$                    // $N = I + J + K + L$ chromosome part 1
                    Chromosome [$i$] ← random (1,$N$);
        Step 2: for $j = N+1$ to $N+K$                // chromosome part 2.1 (y)
                    Chromosome [$j$] ← random (0,1);
        Step 3: for $k = N+K+1$ to $N+K+L$            // chromosome part 2.2 (z)
                    Chromosome [$k$] ← random (1,2);
end;
```

Fig. 3 Procedure for initialization by using new extended priority-based encoding

If ω > α, then

$$\bar{a}_\omega = \frac{1}{2}\int_{\alpha}^{\omega} \{a_L(r) + a_U(r)\}\,\mathrm{d}r = \frac{1}{4\omega}\left[2a_M(\omega^2 - \alpha^2) + (a_L + a_U)(\omega - \alpha)^2\right]$$

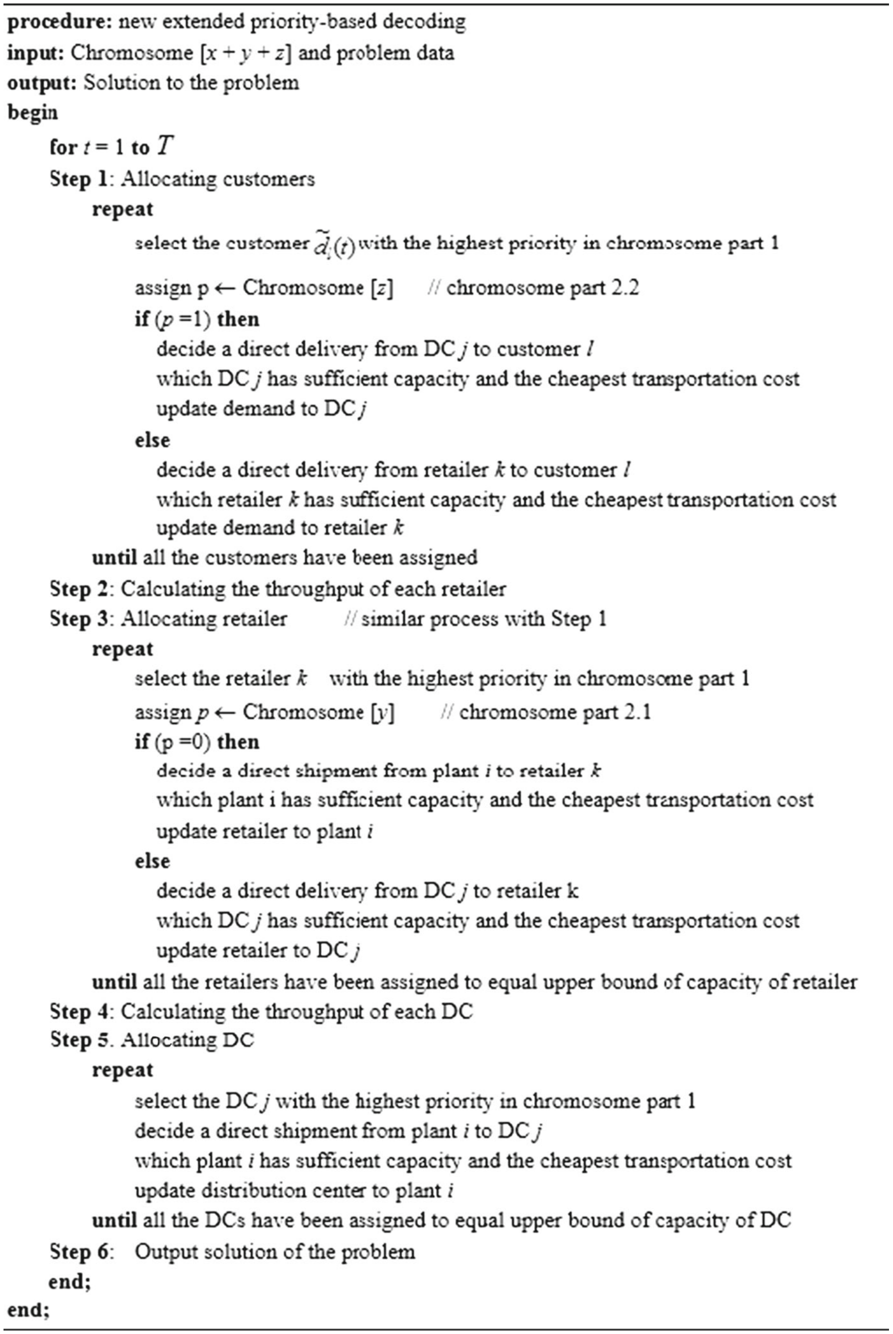
5 Discrete particle swarm optimization and extended priority-based hybrid genetic algorithm

5.1 Extended priority-based hybrid genetic algorithm

The priority-based GA is an indirect approach for solving some network problems (Gen and Cheng 2000; Gen et al. 2008; Chou et al. 2014). This study proposed a chromosome consisting of two types of information: the locus, the position of the gene within the chromosome structure, and the allele, the value the gene takes (Lin et al. 2009). The position of a gene is used to represent the value that is used to represent the priority of the source depot for constructing candidates. The second part of the chromosome consists of two parts: the first part assigns retailers to a plant or distribution center, in which each locus is denoted as an integer ranging from 0 to 1. The second part involves a customer assigning a distribution center or retailer, for which each locus is denoted as an integer ranging from 1 to 2. The encoding and decoding procedures of the chromosome are shown in Figs. 3 and 4, respectively.

5.2 Proposed chromosome encoding procedure

The encoding representation designed in this paper consists of the number of plants, distribution centers, retailers, and customers. Figure 5 shows an example of encoding new extended priority-based initialization with two plants, three distribution centers, three retailers, and five customers, based on the random permutation, in which the priority of each gene is to allocate each stage so that the total number of genes is 13 in chromosome part 1. Chromosome part 2 is composed of chromosome part 2.1, which assigns retailers to a plant or distribution center that has three genes. The value is 0 in



```
procedure: new extended priority-based decoding
input: Chromosome [$x + y + z$] and problem data
output: Solution to the problem
begin
    for $t$ = 1 to $T$
    Step 1: Allocating customers
        repeat
            select the customer $\widetilde{d}_l(t)$ with the highest priority in chromosome part 1
            assign p ← Chromosome [$z$]     // chromosome part 2.2
            if ($p$ =1) then
              decide a direct delivery from DC $j$ to customer $l$
              which DC $j$ has sufficient capacity and the cheapest transportation cost
              update demand to DC $j$
            else
              decide a direct delivery from retailer $k$ to customer $l$
              which retailer $k$ has sufficient capacity and the cheapest transportation cost
              update demand to retailer $k$
        until all the customers have been assigned
    Step 2: Calculating the throughput of each retailer
    Step 3: Allocating retailer          // similar process with Step 1
        repeat
            select the retailer $k$   with the highest priority in chromosome part 1
            assign $p$ ← Chromosome [$y$]       // chromosome part 2.1
            if (p =0) then
              decide a direct shipment from plant $i$ to retailer $k$
              which plant i has sufficient capacity and the cheapest transportation cost
              update retailer to plant $i$
            else
              decide a direct delivery from DC $j$ to retailer k
              which DC $j$ has sufficient capacity and the cheapest transportation cost
              update retailer to DC $j$
        until all the retailers have been assigned to equal upper bound of capacity of retailer
    Step 4: Calculating the throughput of each DC
    Step 5. Allocating DC
        repeat
            select the DC $j$ with the highest priority in chromosome part 1
            decide a direct shipment from plant $i$ to DC $j$
            which plant $i$ has sufficient capacity and the cheapest transportation cost
            update distribution center to plant $i$
        until all the DCs have been assigned to equal upper bound of capacity of DC
    Step 6:   Output solution of the problem
    end;
end;
```

Fig. 4 Procedure for new extended priority-based decoding

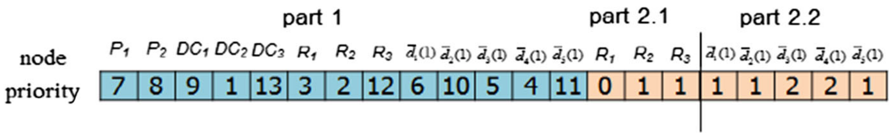

part 1
part 2.1
part 2.2
node
priority
P1 P2 DC1 DC2 DC3 R1 R2 R3 d̄1(1) d̄2(1) d̄3(1) d̄4(1) d̄5(1) R1 R2 R3 d̄1(1) d̄2(1) d̄3(1) d̄4(1) d̄5(1)
7 8 9 1 13 3 2 12 6 10 5 4 11 0 1 1 1 1 2 2 1

Fig. 5 Example of the new extended priority-based encoding

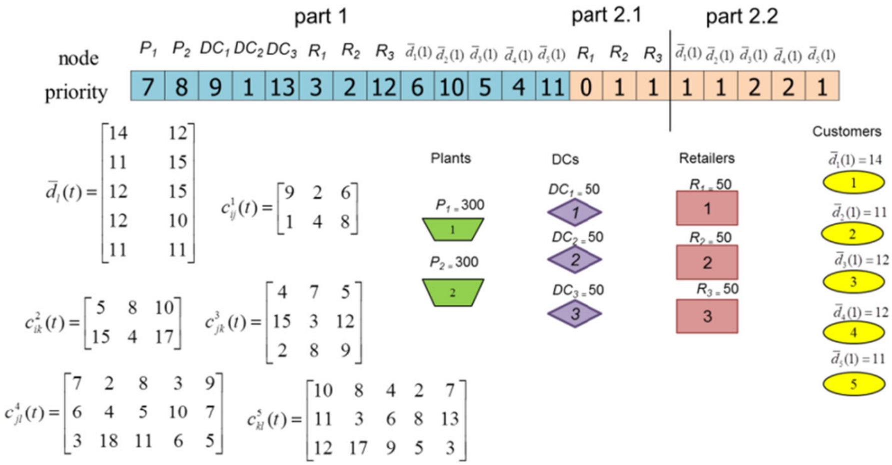

part 1
part 2.1
part 2.2
node
priority
P1 P2 DC1 DC2 DC3 R1 R2 R3 d̄1(1) d̄2(1) d̄3(1) d̄4(1) d̄5(1) R1 R2 R3 d̄1(1) d̄2(1) d̄3(1) d̄4(1) d̄5(1)
7 8 9 1 13 3 2 12 6 10 5 4 11 0 1 1 1 1 2 2 1
$\bar{d}_l(t)=\begin{bmatrix}14&12\\11&15\\12&15\\12&10\\11&11\end{bmatrix}$
$c^1_{ij}(t)=\begin{bmatrix}9&2&6\\1&4&8\end{bmatrix}$
$c^2_{ik}(t)=\begin{bmatrix}5&8&10\\15&4&17\end{bmatrix}$
$c^3_{jk}(t)=\begin{bmatrix}4&7&5\\15&3&12\\2&8&9\end{bmatrix}$
$c^4_{jl}(t)=\begin{bmatrix}7&2&8&3&9\\6&4&5&10&7\\3&18&11&6&5\end{bmatrix}$
$c^5_{kl}(t)=\begin{bmatrix}10&8&4&2&7\\11&3&6&8&13\\12&17&9&5&3\end{bmatrix}$
Plants
P1 = 300
1
P2 = 300
2
DCs
DC1 = 50
1
DC2 = 50
2
DC3 = 50
3
Retailers
R1 = 50
1
R2 = 50
2
R3 = 50
3
Customers
d̄1(1) = 14
1
d̄2(1) = 11
2
d̄3(1) = 12
3
d̄4(1) = 12
4
d̄5(1) = 11
5

Fig. 6 Example of the flexible multistage logistics network chromosome encoding/decoding ($t = 1$)

the first gene (k1) in chromosome part 2.1, which means a direct shipment from plant i to the first retailer, and the value of 1 in the second and third gene means a direct delivery from distribution center j to k2 and k3. The final part of this chromosome is part 2.2, which allocates customers from the distribution center or retailer, so that two integer numbers range from 1 to 2.

In this example, the gene value of the first, second, and fifth customers are 1, which means a direct delivery from distribution center j to the first, second, and fifth customers. The third and fourth customers are 2, which means allocating a direct delivery from retailer k to them.

5.3 Proposed chromosome decoding procedure

The proposed three-stage decoding procedure (Fig. 6) is described as follows: The first stage (Step 1–2 in Fig. 4) is the retailer or distribution center that delivers products for customer demand allocation in the period of triangular fuzzy customer demands; the values are obtained according to the proposed Sect. 4 ($\omega = 1$ and $\alpha = 0$). Table 2 presents examples of fuzzy demands. The allocation among retailers, distribution centers, and customers is obtained by decoding chromosome part 1 and part 2.2, which follows a priority-based procedure: Select the customer with the highest priority in chromosome part 1 and determine delivery from retailers or distribution centers according to chromosome part 2.2. A gene number of 1 means a direct delivery from

Table 2 Example triangular fuzzy demands for the test problem

L	$t = 1$		$t = 2$	
	$\tilde{d}_l(1)$	$\bar{d}_l(1)$	$\tilde{d}_l(2)$	$\bar{d}_l(2)$
1	(10,15,16)	14	(10,11,16)	12
2	(9,10,15)	11	(11,16,17)	15
3	(9,12,15)	12	(11,16,17)	15
4	(10,11,16)	12	(8,9,14)	10
5	(8,10,16)	11	(9,10,15)	11

distribution center j to customer l, whereas a gene number of 2 means a direct delivery from retailer k to customer l; the highest priority of the customer corresponds to the minimum transportation cost; the customer allocation t continues until the maximum capacity has been reached and all customers are allocated; then, the throughput of each retailer is calculated.

The retailer allocation from the distribution center or plant is presented in Steps 3–4, a stage at which each retailer lost stock from allocating demand; however, this stage is similar to the first stage, except that different retailers are positioned according to the chromosome and gene number. The allocation among plants, distribution centers, and retailers is obtained by decoding chromosome parts 1 and 2.1, in which chromosome part 1 follows a priority-based procedure: Select the retailer with the highest priority in chromosome part 1 and determine delivery from plants or distribution centers according to chromosome part 2.1, in which a gene number equal to 0 means a direct delivery from plant i to retailer k; then, select a suitable transportation cost to minimize costs; allocation continues until the maximum capacity has been reached and all retailers are allocated; then, calculate the throughput of each distribution center.

At the final stage (Steps 5–6), the plants delivers direct shipment products to each distribution center that causes each distribution center to lose stock from allocating demands and retailers. This step entails selecting the distribution j with the highest priority in chromosome part 1 and involves determining a direct shipment from plant i to DC j, in which plant i has sufficient capacity and the lowest transportation cost until all distribution centers have been assigned to the equal upper bound of distribution center capacity; then, the total cost from the encoding procedure is calculated. Finally, in the first period of this flexible multistage logistics network problem, the number of products of each stage is presented, as shown in Table 3, and the allocation, transportation cost, inventory cost, and total cost of each stage (Table 4) are traced as an example of Periods 1 and 2. Customers in the next period allocate the total number products of each stage from the first period in a continuous decoding procedure.

5.4 Particle swarm optimization

PSO is based on the simulation of social iterations proposed by Kennedy and Eberhart (1995), and is initialized with a population of random candidate solutions as particles. The position of each kth particle $x_k(t)$ is a potential result (potential solution) of the problem under study. Each particle remembers the best position that it has found

Table 3 Total number of products at each stage in Steps 1–6 ($t = 1$)

Step	Total number			
	Demand ($l = 1, 2, 3, \ldots, 5$)	Retailer ($k = 1, 2, 3$)	Distribution center ($j = 1, 2, 3$)	Plant ($i = 1, 2$)
Initialization	(14,11,12,12,11)	(50,50,50)	(50,50,50)	(300,300)
1–2	(0,0,0,0,0)	(26,50,50)	(39,50,25)	(300,300)
3–4	(0,0,0,0,0)	(50,50,50)	(39,50,25)	(276,300)
5–6	(0,0,0,0,0)	(50,50,50)	(50,50,50)	(251,289)

thus far during the search process $h_{\text{best}}(t)$ and knows the global best position of swarm $g_{\text{best}}(t)$. We can calculate the particle's fitness by inputting its position into a designated objective function that is expressed according to the following equations:

$$v_k(t+1) = v_k(t) + b_1 r_1 (h_{\text{best}}(t) - x_k(t)) + b_2 r_2 (g_{\text{best}}(t) - x_k(t)) \tag{9}$$

$$x_k(t+1) = x_k(t) + v_k(t+1) \tag{10}$$

where $v_k(t)$ is the kth particle's flying velocity at the tth iteration; b_1 and b_2 are positive constants, called the acceleration constants; and $r_1, r_2 \in [0, 1]$ are uniform random numbers. The PSO procedure is summarized in Fig. 7:

5.5 Genetic algorithm operators

5.5.1 Crossover

The crossover operator is exchanged between two chromosomes from one generation to the next generation. In this research, we used one cut-point crossover that randomly selects a locus and exchanges the subsequences before and after the locus between two parent chromosomes to two new offspring chromosomes that cut one point of each chromosome part, as shown in Fig. 8.

After the crossover operation, improving chromosomes/offspring is necessary by checking and repairing the number of sequence constraints if the genes in chromosome part 1 are the same number. Subsequently, the first number must be selected and the lacking number in chromosome part 1 must be replaced.

5.5.2 Mutation

The mutation operator is applied to each child solution resulting from the crossover operator. In this study, we used two mutation operators, which are swapping and inverse mutations, as shown in Fig. 9. The swapping mutation operates in chromosome part 1 that selects a substring and swap within this chromosome to decrease the checking and repairing process of the offspring chromosome and to modify the gene. We also

Table 4 Tracing allocation and total cost at each stage in Steps 1–6 for Periods 1 and 2

Stage	Demand (Steps 1–2)		Retailer (Steps 3–4)		Distribution center (Steps 5–6)	
	$t = 1$	$t = 2$	$t = 1$	$t = 2$	$t = 1$	$t = 2$
Allocating	$DC_3 \rightarrow \bar{d}_5(1)$, $DC_1 \rightarrow \bar{d}_2(1)$, $DC_3 \rightarrow \bar{d}_1(1)$, $R_1 \rightarrow \bar{d}_3(1)$, $R_1 \rightarrow \bar{d}_4(1)$	$DC_3 \rightarrow \bar{d}_5(2)$, $DC_1 \rightarrow \bar{d}_2(2)$, $DC_3 \rightarrow \bar{d}_1(2)$, $R_1 \rightarrow \bar{d}_3(1)$, $R_1 \rightarrow \bar{d}_4(1)$	$P_1 \rightarrow R_1$	$P_1 \rightarrow R_1$	$P1 \rightarrow DC_3$	$P_1 \rightarrow DC_3$ $P_2 \rightarrow DC_1$
Transportation cost ($)	$x_{jl}^4(1) + x_{kl}^5(1) = 191$	$x_{jl}^4(2) + x_{kl}^5(2) = 201$	$x_{ik}^2(1) = 120$	$x_{ik}^2(2) = 125$	$x_{ij}^1(1) = 161$	$x_{ij}^1(2) = 195$
Inventory cost ($)	–	–	$y_k^2(1) = 0$	$y_k^2(2) = 0$	$y_j^1(1) = 0$	$y_j^1(2) = 0$
Total cost ($)	$\tilde{d}_l(1) : z = 191 + 120 + 161 + 0 + 0 = \$472$			$\tilde{d}_l(2) : z = 201 + 125 + 195 + 0 + 0 = \$521$		

```
procedure: particle swarm optimization
input: $f(x)$, $v_k(0)$, $h_{best}(0)$, $g_{best}(0)$, $b_1$, $b_2$, $maxIter$
output: the best solution
begin
    $t \leftarrow 0$;
    initialize ($v_k$, $x_k$) for each particle $k$;
    evaluate $x_k(t)$ by decoding routine;
    while $t \le maxIter$ do
        for each particle $x_k$ in swarm do
            update velocity $v_k$ using (13);
            update position $x_k$ using (14);
            find the local best ever position of the particle $h_{best}$;
            evaluate $x_k(t)$ by decoding routine;
        find the global best position of the particle $g_{best}$;
        $t = t+1$;
    return $g_{best}$
end;
```

Fig. 7 Procedure for particle swarm optimization

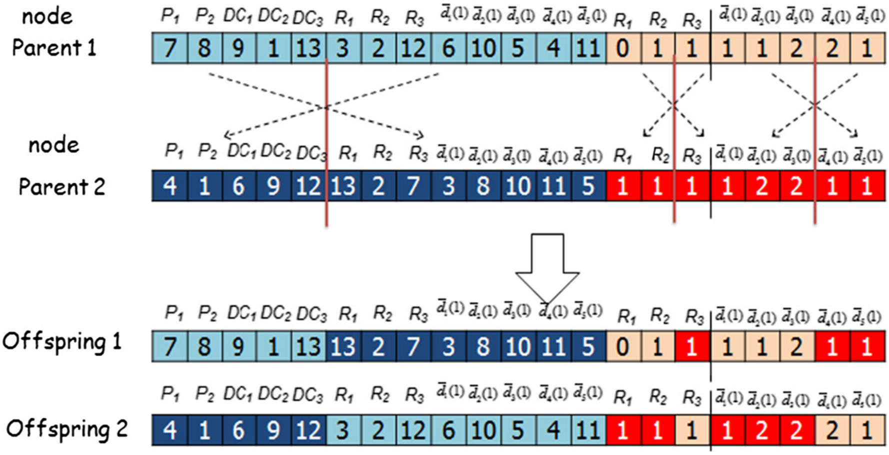

node
Parent 1
P_1 P_2 DC_1 DC_2 DC_3 R_1 R_2 R_3 $\bar{d}_1(1)$ $\bar{d}_2(1)$ $\bar{d}_3(1)$ $\bar{d}_4(1)$ $\bar{d}_5(1)$ R_1 R_2 R_3 $\bar{d}_1(1)$ $\bar{d}_2(1)$ $\bar{d}_3(1)$ $\bar{d}_4(1)$ $\bar{d}_5(1)$
7 8 9 1 13 3 2 12 6 10 5 4 11 0 1 1 1 1 2 2 1
node
Parent 2
P_1 P_2 DC_1 DC_2 DC_3 R_1 R_2 R_3 $\bar{d}_1(1)$ $\bar{d}_2(1)$ $\bar{d}_3(1)$ $\bar{d}_4(1)$ $\bar{d}_5(1)$ R_1 R_2 R_3 $\bar{d}_1(1)$ $\bar{d}_2(1)$ $\bar{d}_3(1)$ $\bar{d}_4(1)$ $\bar{d}_5(1)$
4 1 6 9 12 13 2 7 3 8 10 11 5 1 1 1 1 2 2 1 1
Offspring 1
P_1 P_2 DC_1 DC_2 DC_3 R_1 R_2 R_3 $\bar{d}_1(1)$ $\bar{d}_2(1)$ $\bar{d}_3(1)$ $\bar{d}_4(1)$ $\bar{d}_5(1)$ R_1 R_2 R_3 $\bar{d}_1(1)$ $\bar{d}_2(1)$ $\bar{d}_3(1)$ $\bar{d}_4(1)$ $\bar{d}_5(1)$
7 8 9 1 13 13 2 7 3 8 10 11 5 0 1 1 1 1 2 1 1
Offspring 2
P_1 P_2 DC_1 DC_2 DC_3 R_1 R_2 R_3 $\bar{d}_1(1)$ $\bar{d}_2(1)$ $\bar{d}_3(1)$ $\bar{d}_4(1)$ $\bar{d}_5(1)$ R_1 R_2 R_3 $\bar{d}_1(1)$ $\bar{d}_2(1)$ $\bar{d}_3(1)$ $\bar{d}_4(1)$ $\bar{d}_5(1)$
4 1 6 9 12 3 2 12 6 10 5 4 11 1 1 1 1 2 2 2 1

Fig. 8 Crossover operation

selected a string in chromosome Parts 2.1 and 2.2 at random for inverse mutation, and inversed them into another number with a randomly selected locus.

5.6 Discrete particle swarm operator

Most PSO applications involve using continuous function value optimization, but few studies have considered using DPSO. The displacement of each particle depends on

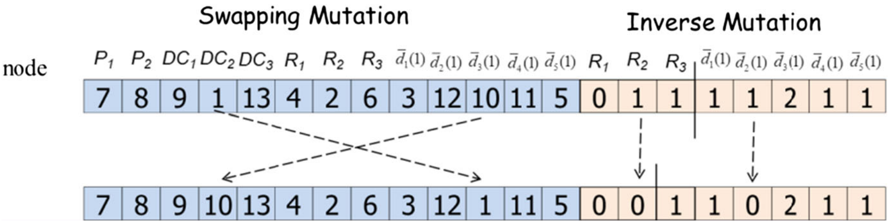


Fig. 9 Mutation operation

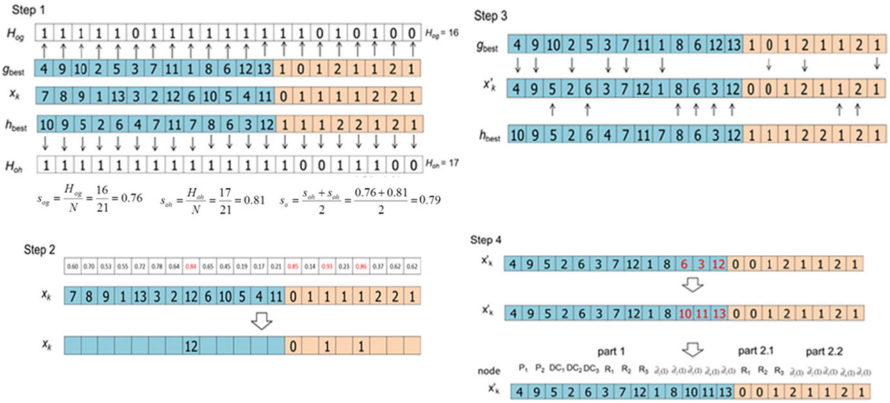


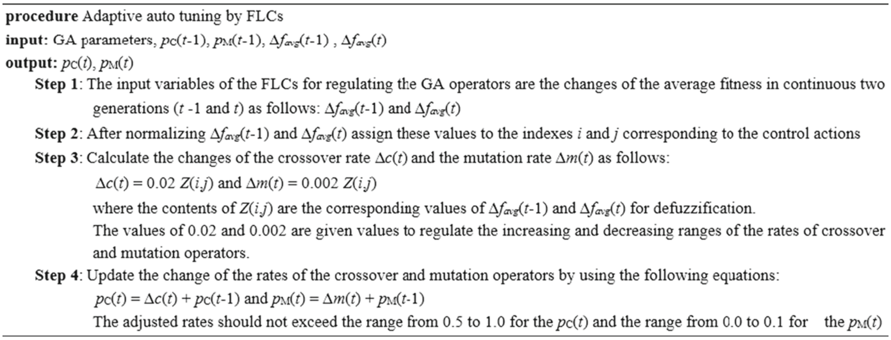
Fig. 10 Example of discrete particle swarm operator

its h_{best} and g_{best}, in which the particle velocity represents the probability of a particle to change toward h_{best} and g_{best}. The particle velocity is similarly dependent. In the proposed algorithm, the similarity between two positions is based on the Hamming distance proposed by Shao et al. (2013). The Hamming distance between two particle positions is the number of corresponding components that are different. The displacement operator is summarized as follows (Fig. 10).

6 Adaptive auto tuning by using the fuzzy logic controller

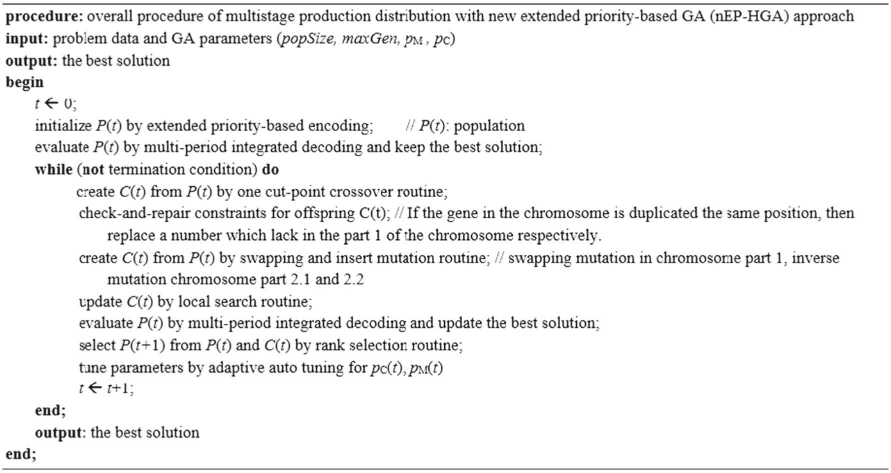
The conventional GA approach cannot guarantee an optimal platform in all cases because the GA uses unknown genetic parameters such as the crossover rate and mutation rate. Adaptive autotuning can adaptively regulate genetic parameters during GA search processes. Yun and Gen (2003) proposed a heuristic updating strategy for crossover and mutation rates to consider changes of average fitness in the GA population of two continuous generations. For example, in the minimization problem, we can set the change of the average fitness at generation t, $\Delta f_{avg}(t)$ as follows:

$$\Delta f_{avg}(t) = \overline{f_{popSize}}(t) - \overline{f_{offSize}}(t) = \frac{1}{popSize}\sum_{k=1}^{popSize} f_k(t) - \frac{1}{offSize}\sum_{k=1}^{offSize} f_k(t) \tag{11}$$



```
procedure Adaptive auto tuning by FLCs
input: GA parameters, $p_C(t-1)$, $p_M(t-1)$, $\Delta f_{avg}(t-1)$ , $\Delta f_{avg}(t)$
output: $p_C(t)$, $p_M(t)$
    Step 1: The input variables of the FLCs for regulating the GA operators are the changes of the average fitness in continuous two
            generations ($t$ -1 and $t$) as follows: $\Delta f_{avg}(t-1)$ and $\Delta f_{avg}(t)$
    Step 2: After normalizing $\Delta f_{avg}(t-1)$ and $\Delta f_{avg}(t)$ assign these values to the indexes $i$ and $j$ corresponding to the control actions
    Step 3: Calculate the changes of the crossover rate $\Delta c(t)$ and the mutation rate $\Delta m(t)$ as follows:
            $\Delta c(t) = 0.02\ Z(i,j)$ and $\Delta m(t) = 0.002\ Z(i,j)$
            where the contents of $Z(i,j)$ are the corresponding values of $\Delta f_{avg}(t-1)$ and $\Delta f_{avg}(t)$ for defuzzification.
            The values of 0.02 and 0.002 are given values to regulate the increasing and decreasing ranges of the rates of crossover
            and mutation operators.
    Step 4: Update the change of the rates of the crossover and mutation operators by using the following equations:
            $p_C(t) = \Delta c(t) + p_C(t-1)$ and $p_M(t) = \Delta m(t) + p_M(t-1)$
            The adjusted rates should not exceed the range from 0.5 to 1.0 for the $p_C(t)$ and the range from 0.0 to 0.1 for  the $p_M(t)$
```

Fig. 11 Procedure for adaptive auto tuning by using FLCs



```
procedure: overall procedure of multistage production distribution with new extended priority-based GA (nEP-HGA) approach
input: problem data and GA parameters (popSize, maxGen, $p_M$ , $p_C$)
output: the best solution
begin
    $t \leftarrow 0$;
    initialize $P(t)$ by extended priority-based encoding;      // $P(t)$: population
    evaluate $P(t)$ by multi-period integrated decoding and keep the best solution;
    while (not termination condition) do
        create $C(t)$ from $P(t)$ by one cut-point crossover routine;
        check-and-repair constraints for offspring C(t); // If the gene in the chromosome is duplicated the same position, then
            replace a number which lack in the part 1 of the chromosome respectively.
        create $C(t)$ from $P(t)$ by swapping and insert mutation routine; // swapping mutation in chromosome part 1, inverse
            mutation chromosome part 2.1 and 2.2
        update $C(t)$ by local search routine;
        evaluate $P(t)$ by multi-period integrated decoding and update the best solution;
        select $P(t+1)$ from $P(t)$ and $C(t)$ by rank selection routine;
        tune parameters by adaptive auto tuning for $p_C(t)$, $p_M(t)$
        $t \leftarrow t+1$;
    end;
    output: the best solution
end;
```

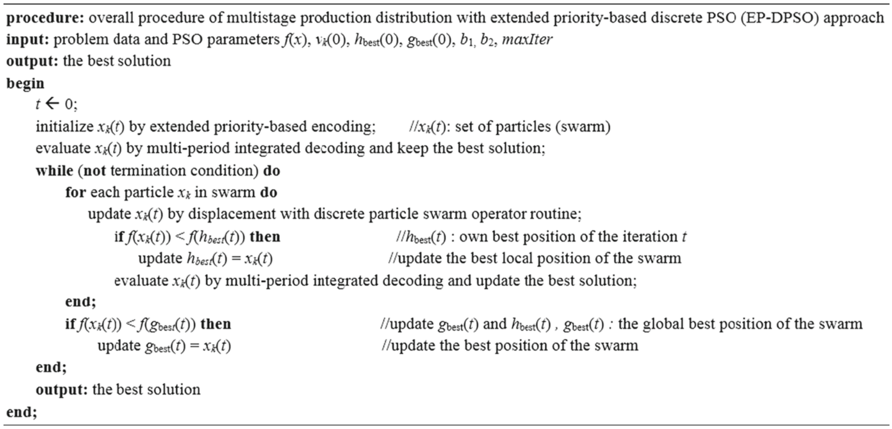
Fig. 12 Overall nEP-HGA procedure for the MSMT problem

where *popSize* is the population size that satisfies constraints, and *offSize* is the offspring size that satisfies constraints. The adaptive auto-tuning process using FLCs is shown in Fig. 11.

Thus, we present the nEP-HGA that combines adaptive autotuning with FLCs. We also propose the EP-DPSO for priority-based encoding and decoding with DPSO that involves using the Hamming distance for solving the flexible MSMT supply chain. The overall procedure for the proposed nEP-HGA and EP-DPSO is outlined in Figs. 12 and 13, respectively.

7 Numerical experiment and experimental design

To demonstrate the efficiency and effectiveness of the nEP-HGA in the fuzzy demand environment, the GA parameters are designed tuning parameters according to the adaptive autotuning routine for $p_C(t)$, $p_M(t)$. The numerical experiments are 10 times the $popSize = 50$ and $maxGen = 500$. The DSPO parameters for numerical experi-



```
procedure: overall procedure of multistage production distribution with extended priority-based discrete PSO (EP-DPSO) approach
input: problem data and PSO parameters $f(x)$, $v_k(0)$, $h_{best}(0)$, $g_{best}(0)$, $b_1$, $b_2$, $maxIter$
output: the best solution
begin
    $t \leftarrow 0$;
    initialize $x_k(t)$ by extended priority-based encoding;    //$x_k(t)$: set of particles (swarm)
    evaluate $x_k(t)$ by multi-period integrated decoding and keep the best solution;
    while (not termination condition) do
        for each particle $x_k$ in swarm do
            update $x_k(t)$ by displacement with discrete particle swarm operator routine;
                if $f(x_k(t)) < f(h_{best}(t))$ then    //$h_{best}(t)$ : own best position of the iteration $t$
                    update $h_{best}(t) = x_k(t)$    //update the best local position of the swarm
                evaluate $x_k(t)$ by multi-period integrated decoding and update the best solution;
        end;
        if $f(x_k(t)) < f(g_{best}(t))$ then    //update $g_{best}(t)$ and $h_{best}(t)$ , $g_{best}(t)$ : the global best position of the swarm
            update $g_{best}(t) = x_k(t)$    //update the best position of the swarm
    end;
    output: the best solution
end;
```

Fig. 13 Overall EP-DPSO procedure for the MSMT problem

Table 5 Size of tested problems

Problem No.	No. of plants I	No. of DCs J	No. of retailers K	No. of customers L	Total planning time T
1	2	3	3	5	8
2	3	4	5	18	12
3	3	5	8	30	15

ments are the number of particles = 50 and $maxIter = 500$. The proposed algorithms were run using MatLab on a 2.10 GHz PC, with 8 G-Byte of RAM, for testing and evaluation. For illustration, we generated three problems (Table 5) involving the various numbers of plants, distribution centers, retailers, customers, and planning times. Uncertain demands generated in the a_L and a_U fuzzy values interval are [10, 20] for each problem. We assumed the fixed charge of product transportation and the unit holding cost of inventory per period at each node to be equal to 1. Table 6 provides information on the unit shipping cost for each node of a small-scaled MSMT problem (Problem 1).

We formulated the problems as a mixed-integer programming model for a multistage supply chain network, as shown in Table 7. The results of tested problems for deriving an optimal solution were followed, generating three problems presented as the number of constraints, decision variables, and the objective function results of each problem. Problem 3 yielded a high number of variables, in which the location-allocation problem was NP-hard and highly complex. Therefore, we propose that the EP-DPSO and nEP-HGA can determine delivery to customers in each period and select the distribution center or retailers for each customer in the production-distribution network for satisfying customer demands and solving both small and large problems.

We tested the EP-DPSO and nEP-HGA performance using three sizes of test problems. The computational test compared the best and average total cost of each solution for solving each problem, and the results of the optimal solution, traditional GA, the nEP-HGA, and EP-DPSO for flexible multiple stages and multi-time periods are shown

Table 6 Unit shipping cost of each node ($/unit)

C^1_{ij}	1	2	3		
1	9	2	6		
2	1	4	8		
C^2_{ik}	1	2	3		
1	5	8	10		
2	15	4	17		
C^3_{jk}	1	2	3		
1	4	7	5		
2	15	3	12		
3	2	8	9		
C^4_{jl}	1	2	3	4	5
1	7	2	8	3	9
2	6	4	5	10	7
3	3	18	11	6	5
C^5_{kl}	1	2	3	4	5
1	10	8	4	2	7
2	11	3	6	8	13
3	12	17	9	5	3

Table 7 Results of tested problems by applying optimal solution (LINGO)

Problem no.	No. of constraints	No. of decision variables	Objective function (K$)
1	153	456	1,094
2	469	2,616	4,428
3	886	7,230	Exceed capacity

Table 8 Comparison of the best and average total cost of each solution from solving each problem [K·$]

Solutions problem no.	Optimal solution By LINGO	Traditional MSMT by GA		Flexible MSMT by nEP-HGA		Flexible MSMT by EP-DPSO	
	Best solution	Best solution	AV. solution	Best solution	AV. solution	Best solution	AV. solution
1	094	094	1,094.0	094	094.0	094	094.0
2	4,428	4,887	4,987.9	4,428	4,473.9	4,428	4,443.0
3	Exceed capacity	9,323	9,460.6	8,416	8,601.9	8,416	8,506.7

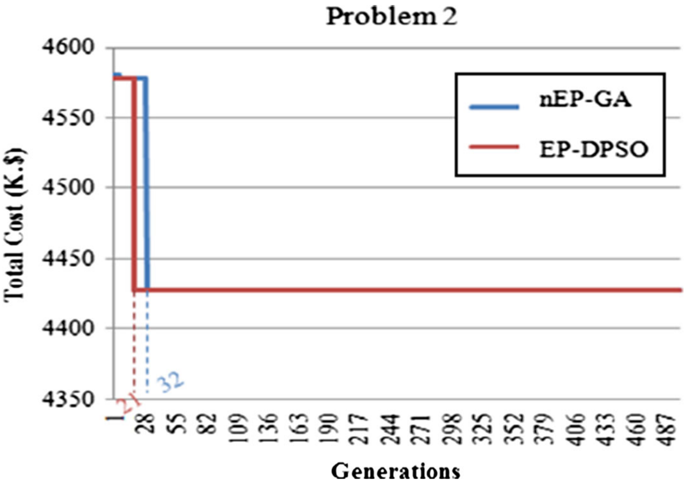


Fig. 14 Comparison of the results of using the nEP-HGA and EP-DPSO for Problem 2

in Table 8. Thus, the nEP-HGA and EP-DPSO determined that total costs equal the optimal solution and have enhanced results compared with the traditional GA, whereas in the comparison results between the nEP-HGA and EP-DPSO for each problem, the EP-DPSO spends fewer number of iterations than the number of generations in the nEP-HGA for finding the best solution for large problems (Problems 2 and 3; Fig. 14). The proposed EP-DPSO outperforms the traditional GA and nEP-HGA, particularly for large problems. Time is the CPU time in seconds for every run, and the example of the EP-DPSO experiment facilitated determining the best solution at 35.9 s at the 21st iteration in Problem 2. However, the CPU time of EP-DPSO found a solution that is less than the nEP-HGA, in which the best solution is at 54.5 s at the 32nd iteration. The result of EP-DPSO outperformed the nEP-HGA in Problem 3. As shown in Fig. 15, the CPU time and times for EP-DPSO and nEP-HGA runs are 272.4 s (80 iteration) and 619.9 s (179th generation), respectively. However, in small problems such as Problem 1, the results of EP-DPSO and nEP-HGA are similar. However, they can determine which real world demands must be optimized for long-term efficient operation of the entire supply chain.

The developed heuristics were compared to the solutions from the traditional procedures, which were based on factorial design according to fuzzy customer demands. The combination experiments were conducted in quintuplicate. The parameters included the number of problems (Table 5) and the fluctuation of fuzzy a_L and a_U customer demands (20, 40, and 60 % of customer demands, which were an average of $\bar{a}_{\omega}$). The samples totaled 45 and the results are shown in the percentage of improvement presented in Eq. (12), where R_GA is the result of the traditional MSMT that entails using the GA, and R_DPSO is the result of the flexible MSMT that involves using EP-DPSO.

$$\%\text{improvement} = \frac{(R\_DPSO - R\_GA)}{R\_GA} \times 100 \tag{12}$$

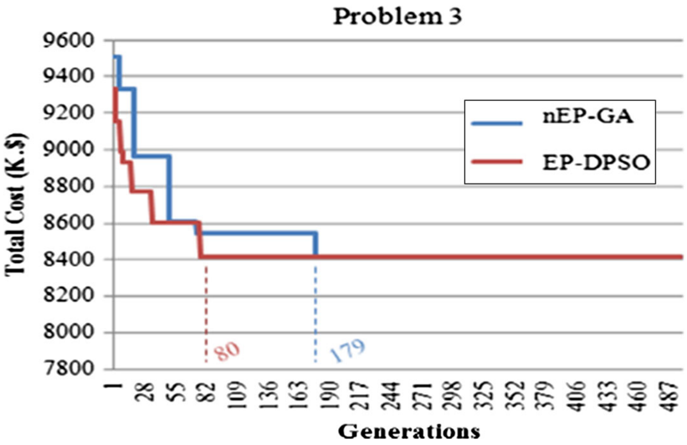


Fig. 15 Comparison of the results of using the nEP-HGA and EP-DPSO for Problem 3

Table 9 Experimental design and results

No.	Factor		Average total cost (K $)		Percentage of improvement
	Problem	% uncertain demands	Traditional MSMT by GA	Flexible MSMT by EP-DPSO	
1–5	1	20	1,095.7	1,092.4	0.3
6–10	1	40	1,412.5	1,410.5	0.1
11–15	1	60	1,798.1	1,708.6	4.8
16–20	2	20	4,995.9	4,362.2	12.7
21–25	2	40	5,698.2	5,125.6	10.0
26–30	2	60	7,014.2	6,280.5	10.5
31–35	3	20	9,546.0	8,444.1	11.5
36–40	3	40	11,498.7	10,082.9	12.3
41–45	3	60	13,633.1	12,108.5	11.2
Average			6,299.2	5,623.9	8.2

Table 9 shows an 8.2 % improvement from the flexible multistage network model when the data were analyzed using ANOVA. The percentage improvement differed significantly from actual situation practices at the 95 % reliability level and $p < 0.05$. The parameter explaining this difference was the fluctuation of demand at 60 % of an average of $\bar{a}_{\omega}$, which yielded the highest average improvement.

8 Conclusion

Because globalization has introduced increased competition and unpredictable factors, customer demands are typically imprecise in each period and customer quantity demands in production-distribution models affect inventory costs and related costs, causing unsatisfactory planning. In this research, we formulated a MSMT supply chain network problem under uncertain demands as a mixed-integer programming model. However, the location-allocation sub-problem in the formulated model is NP-hard and highly complex. The DPSO approach combined with the EP-HGA using a fuzzy number is frequently used for practical purposes and is relatively easy to employ with triangular fuzzy numbers for solving large-scale problems with reasonable computational time, compared with the LINGO computational software.

Using uncertain customer demands in a flexible MSMT supply chain model, we combined the DPSO with the encoding/decoding nEP-HGA to determine delivery to customers in each period by using a triangular fuzzy number of customer demands. We selected the distribution center and retailers for each customer in the production-distribution network for satisfying uncertain customer demands. The experimental results prove that the EP-DPSO and nEP-HGA can be applied for solving all problem types with the best solution under uncertain customer demands. The proposed EP-DPSO outperforms the traditional GA and nEP-HGA in large problems and improves the traditional priority-based MSMT. This research can extend to various supply chain environment problems by providing an enhanced method with large echelons.

Acknowledgments This research is supported by the Ministry of Science and Technology, Taiwan (NSC100-2628-E-007-017-MY3; NSC101-2811-E-007-004; NSC102-2622-E-007-013; NSC102-2811-E-007-005), the Advanced Manufacturing and Service Management Research Center of National Tsing Hua University (102N2075E1), Fuzzy Logic Systems Institute, Japan (JSPS-the Grant-in-Aid for Scientific Research C No. 24510219) and the Faculty of Engineering, Khon Kaen University, Thailand.

References

Chamnanlor, C., Sethanan, K., Chien, C.-F., & Gen, M. (2014). Re-entrant flow shop scheduling problem with time windows using hybrid genetic algorithm based on auto-tuning strategy. *International Journal of Production Research, 52*(9), 2612–2629.

Chan, F. T. S., Chung, S. H., & Wadhwa, S. (2005). A hybrid genetic algorithm for production and distribution. *Omega*, *55*, 345–355.

Chien, C.-F., Chen, Y.-J., & Peng, J.-T. (2010). Manufacturing intelligence for semiconductor demand forecast based on technology diffusion and product life cycle. *International Journal of Production Economics*, *128*(2), 496–509.

Chien, C.-F., Chen, J., & Wei, C. (2011). Constructing a comprehensive modular fuzzy ranking framework and illustrations. *Journal of Quality*, *18*(4), 333–350.

Chien, C.-F., Wu, C., & Chiang, Y. (2012). Coordinated capacity migration and expansion planning for semiconductor manufacturing under demand uncertainties. *International Journal of Production Economics*, *135*(2), 860–869.

Chien, C.-F., Wu, J.-Z., & Wu, C.-C. (2013). A two-stage stochastic programming approach for new tape-out allocation decisions for demand fulfillment planning in semiconductor manufacturing. *Flexible Services and Manufacturing Journal*, *25*(3), 286–309.

Chien, C.-F., & Zheng, J.-N. (2012). Mini–max regret strategy for robust capacity expansion decisions in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, *23*(6), 2151–2159.

Chou, C.-W., Chien, C.-F., & Gen, M. (2014). A multiobjective hybrid genetic algorithm for TFT-LCD module assembly scheduling. *IEEE Transactions on Automation Science and Engineering, 11*(3), 692–705.

Costa, A., Celano, G., Fichera, S., & Trovato, E. (2010). A new efficient encoding/decoding procedure for the design of a supply chain network with genetic algorithms. *Computers & Industrial Engineering*, *59*, 986–999.

Duenas, A., & Petrovic, D. (2008). Multi-objective genetic algorithm for single machine scheduling problem under fuzziness. *Fuzzy Optimization and Decision Making*, *7*(1), 87–104.

Gen, M., & Cheng, R. W. (2000). *Genetic algorithm and engineering optimization*. New York: Wiley.

Gen, M., Cheng, R., & Lin, L. (2008). *Network models and optimization: Multiobjective genetic algorithm approach*. London: Springer.

Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. In *Proceeding of the 1998 IEEE international conference on neural network* (pp. 1942–1948).

Kumar, B. R. (2012). On fuzzy transportation problem using triangular fuzzy numbers with modified revised simplex method. *International Journal of Engineering Science and Technology*, *4*, 285–294.

Li, X., Chien, C.-F., Yang, L., & Gao, Z. (2014). The train fueling cost minimization problem with fuzzy fuel prices. *Flexible Service and Manufacturing Journal*, *26*, 249–267.

Li, Y., Gen, M. & Ida, K. (1996). Evolutionary computation for multicriteria solid transportation problem with fuzzy numbers. In *Proceeding of IEEE international conference on evolutionary computation* (pp. 596–601).

Lin, L., Gen, M., & Wang, X. (2009). Integrated multistage logistics network design by using hybrid evolutionary algorithm. *Computers & Industrial Engineering*, *56*, 854–873.

Sethanan, K., Jamrus, T., & Boonthavee, P. (2013). Solving production scheduling of egg production in hen farms with capacity of slaughterhouse by heuristic algorithms. *IACSIT International Journal of Engineering and Technology*, *5*, 50–53.

Shao, X., Liu, W., Liu, Q. & Zhang, C. (2013). Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem. *The International Journal of Advanced Manufacturing Technology, 67*, 2885–2901.

Varthanan, P. A., Murugan, N., & Kumar, G. M. (2013). An AHP based heuristic DPSO algorithm for generating multi criteria production-distribution plan. *Journal of Manufacturing Systems*, *32*, 632–647.

Wang, H. F., & Hsu, H. W. (2012). A possibilistic approach to the modeling and resolution of uncertain closed-loop logistics. *Fuzzy Optimization and Decision Making*, *11*, 177–208.

Xu, J., Liu, Q., & Wang, R. (2008). A class of multi-objective supply chain networks optimal model under random fuzzy environment and its application to the industry of Chinese liquor. *Information Sciences*, *178*, 2022–2043.

Yun, Y., & Gen, M. (2003). Performance analysis of adaptive genetic algorithms with fuzzy logic and heuristics. *Fuzzy Optimization and Decision Making*, *2*, 161–175.

Zeballos, L. J., Mendez, C. A., Barbosa-Povoa, A. P., & Novais, A. Q. (2014). Multi-period design and planning of closed-loop supply chains with uncertain supply and demand. *Computers & Chemical Engineering*, *66*, 151–164.