Fourth party logistics routing problem model with fuzzy duration time and cost discount

Yan Cui, Min Huang, Shengxiang Yang, Loo Hay Lee, Xingwei Wang

Abstract

In this paper, from the viewpoint of a fourth party logistics (4PL) provider, a multi-source single-destination 4PL routing problem with fuzzy duration time and cost discount (M-S 4PLRFPFC) is described considering the comprehensive ability of 3PL suppliers and nodes. A chance-constrained programming model is established for the M-S 4PLRFPFC. Next, a memetic algorithm (MA) with a fuzzy simulation method is designed to solve the problem. Based on a set of problem instances as the test bed, experiments are performed to compare the performance of the proposed MA with those of the enumeration method and a standard genetic algorithm (SGA). The experimental results show that the proposed MA obtains the same results as the enumeration method and that it is than the SGA.

Keywords:
Fourth party logistics, Routing problem, Chance constrained programming, Memetic algorithm, Local search

1. Introduction

With the development of science and technology, many companies have outsourced their logistics to special logistics providers known as third party logistics (3PL). Trebilcock [23] showed that nearly 75% of the Fortune 500 companies have relied on 3PL services since 1996. Additionally, 83% of the top 500 manufacturers in the United States had already adopted 3PL by 2003 [16].

3PL has been widely accepted by many companies. However, most 3PL providers only provide transportation and warehousing services. The integration, operation and monitoring of the complex resources of the supply chain and the maximization of the current and long-term benefits of the supply chain members remains an open question. The initial concept of fourth party logistics (4PL) was introduced by Accenture, which is an integrator that assembles the resources, capability, and technology of its own organization and other organizations to design, build, and run comprehensive supply chain solutions [2]. Because the essence and core superiority of 4PL lies in its ability to integrate the supply chain resources, it encourages strategic alliances among 3PLs and manages the logistic process within the entire supply chain. Many articles, such as [1,3,7,19,20], have discussed this concept, revealing that 4PL will likely surpass 3PL and become the main logistics service mode in the following years.

As 4PL plays an important role in modern logistics, many studies have been conducted on 4PL ever since its introduction. These studies can be classified into two classes, one regarding the perspective of 4PL, which focuses on its macro-framework, such as its historical inevitability, advantages, and development prerequisites [3,11,19,22]: and one regarding the optimization of operational problems under the guidance of 4PL, such as the 4PL routing problem (4PLRP) [4,12,13], the selection of 3PL suppliers [5,26], and the operational problems in the 4PL supply chain [14,15,24,25].

The routing problem is one of the most important issues in logistics and has received extensive attention from researchers over the past several decades. When studying the routing problem in 4PL, researchers found that it is more difficult than that in 3PL because more issues, such as the selection of 3PL providers, the cost and time factors, and the capacity and reputation of 3PL providers, should be considered. In recent years, the 4PLRP has attracted increasing attention from researchers. There have been studies in the literature that apply different intelligent methods to solve the 4PLRP. Chen et al. [4] proposed a genetic algorithm (GA) for solving the 4PLRP with ten nodes. Li et al. [15] introduced some main optimization processes in the logistics operation, including routing optimization, job decomposition, and job assignment, and depicted the relationships between these processes.
Huang et al. [12] considered the node-edge property with three different scales and developed a mathematical model for the 4PLRP. They designed an immune algorithm embedded with the Dijkstra algorithm to solve the 4PLRP model developed. To enhance the capability of seeking optimum solutions, Huang et al. [13] also proposed a hybrid immune algorithm by combining the memory base with the immune operator, which was shown to be better than the original immune algorithm.

Many studies have been devoted to the 4PLRP, but some problems remain. In the previous studies, the parameters are fixed and deterministic, which is not realistic. In a real-world situation, the driving conditions could be affected by many factors, such as weather and human factors. In the traditional way, researchers tend to use the stochastic optimization methods. However, under some conditions, it is difficult to describe the parameters of the problem as random variables because of insufficient data [17]. For example, the duration time from one point to another is often not precise enough, e.g., “between 30 and 40 min”, or “approximately 1 h”. Generally, we can use fuzzy variables to address these uncertainty parameters. Additionally, all of the previous studies focus on the single-source single-destination problem, while in real-world situations, such simple models are rarely or never representative. Many large manufacturing enterprises need to assemble products with many parts, which may be made by their subsidiaries from different areas. Therefore, such enterprises will be more inclined to resort to powerful logistics, such as 4PL, to serve their customers on the single-source single-destination problem, while in real-world situations, such simple models are rarely or never representative.

The rest of this paper is organized as follows. The next section introduces the credibility theory and describes the M-S 4PLRPFC and the process of the fuzzy simulation. Section 3 describes the proposed MA for solving the M-S 4PLRPFC in detail. Section 4 presents an experimental study investigating the performance of the proposed algorithm on some test M-S 4PLRPFC instances. Finally, Section 5 concludes this paper with some discussions on relevant future work.

2. Description of the M-S 4PLRPFC

In this section, the information of credibility space is first introduced. Then, the M-S 4PLRPFC is described. And last, fuzzy simulation method is developed according to the uncertainty theory.

2.1. Credibility space

Credibility theory was founded by Liu [17] as a branch of mathematics for studying the behavior of fuzzy phenomena. The emphasis is that credibility theory is based on following credibility measure. Let $\Theta$ be a nonempty set, and $P$ the power set of $\Theta$. Each element in $P$ is called an event. In order to present an axiomatic definition of credibility, it is necessary to assign to each event $A$ a number $Cr(A)$ which indicates the credibility that $A$ will occur.

Definition 1. The set function $Cr$ is called a credibility measure if it satisfies the following axioms:

- Axiom 1. (Normality) $Cr(\emptyset) = 1$.
- Axiom 2. (Monotonicity) $Cr(A) \leq Cr(B)$ whenever $A \subseteq B$.
- Axiom 3. (Self-Duality) $Cr(A) + Cr(A^c) = 1$ for any event $A$.
- Axiom 4. (Maximality) $Cr(A \cup B) = \sup_n Cr(A_n)$ for any event $A_n$ with $\sup_n Cr(A_n) < 0.5$.

Definition 2. Let $\Theta$ be a nonempty set, $P$ the power set of $\Theta$, and $Cr$ is a credibility measure. Then the triplet $(\Theta, P, Cr)$ is called a credibility space.

Definition 3. A fuzzy variable is defined as a function from a credibility space $(\Theta, P, Cr)$ to the set of real numbers.

Definition 4. Let $\xi$ be a fuzzy variable defined on the credibility space $(\Theta, P, Cr)$. Then its membership function is derived from the credibility measure by

$$\mu(x) = \{2Cr(\xi = x)\} \wedge 1, \quad x \in \mathbb{R}$$

Membership function represents the degree of possibility that the fuzzy variable $\xi$ takes some prescribed value. From credibility inversion theorem which was proved in [6], it has following formulation:

$$Cr(\xi \in B) = \frac{1}{2} \left( \sup_{x \in B} \mu(x) + 1 - \sup_{x \in B^c} \mu(x) \right)$$

for any set of real numbers.

Definition 5. (Critical value): Let $\xi$ be a fuzzy variable, and $x \in (0, 1]$. Then
\( \xi_{\text{sup}}(x) = \sup \{ r | \text{Cr}(x) \geq r \} \geq \alpha \)

is called the \( \alpha - \) optimistic value to \( \xi \); and

\( \xi_{\text{inf}}(x) = \inf \{ r | \text{Cr}(x) \leq r \} \geq \alpha \)

is called the \( \alpha - \) pessimistic value to \( \xi \).

### 2.2. Problem description

As shown in Fig. 1, the M-S 4PLRPFc can be represented by a multi-graph \( G(V,E) \), where \( V(|V|=n) \) is the set of nodes, which represent cities or ports, and \( E \) is the set of edges, which connect pairs of nodes and represent different 3PL suppliers that are available to provide the transportation service between connected nodes. The variables used in the representation of the M-S 4PLRPFc are defined as follows.

**Decision variables**

\( x_{ijk} \)

1 if the \( k \)th edge between node \( i \) and node \( j \) for transportation if it undertakes \( Q \) tasks for the 4PL

\( y_{ij} \)

1 if node \( i \) undertakes the corresponding task; 0 otherwise

**Parameters**

\( C_{ijk}(Q) \)

The unit cost quoted by the \( k \)th 3PL supplier between node \( i \) and node \( j \) for transportation if it undertakes \( Q \) tasks for the 4PL

\( C_{ij}(Q') \)

The unit cost quoted by node \( i \) if it undertakes \( Q' \) tasks for the 4PL

\( B^s \)

The total number of goods transported from the \( s \)th source node

\( r_{ij} \)

The number of 3PL suppliers (edges) between node \( i \) and node \( j \)

\( \zeta_{ijk} \)

The fuzzy duration time of the \( k \)th 3PL supplier between node \( i \) and node \( j \)

\( \eta_i \)

The fuzzy duration time of node \( i \), such as processing, loading and unloading, changing vehicles, and storage

\( T^s \)

The due date of the task transported from the \( s \)th source required by the customer

\( b_i^s \)

The number of goods received by node \( i \) when transported from the \( s \)th node, which is defined as follows:

\[
\begin{cases}
  -B^s, & \text{if } i \text{ is a source node} \\
  B^s, & \text{if } i \text{ is the destination node} \\
  0, & \text{otherwise}
\end{cases}
\]

\( P_{ijk} \)

The transportation capacity of the \( k \)th 3PL supplier between node \( i \) and node \( j \)

\( P_i^s \)

The processing capacity of node \( i \)

\( AA^s \)

The reputation that the selected 3PL suppliers are required by the customer when goods are transported from the \( s \)th source node

\( A_{ijk} \)

The reputation of the \( k \)th 3PL supplier between node \( i \) and node \( j \)

\( A_i^s \)

The reputation of node \( i \)

Using the definition of variables above, the mathematical model for the M-S 4PLRPFc can be constructed. The objective of this problem is to find routes from sources to the destination with the minimum cost and is subject to constraints on the due date, 3PL supplier’s reputation, and load requirement given by the customer. The mathematical model can be presented as follows:

\[
Z = \min \left\{ \sum_{i=1}^{S} \sum_{j=1}^{n} \sum_{k=1}^{r} C_{ijk}(Q)B^s x_{ijk} + \sum_{i=1}^{n} C_{ij}(Q')B^s y_i \right\}
\]  \( (1) \)

s.t.

\[
\text{Cr} \left( \sum_{i=1}^{S} \sum_{j=1}^{n} \sum_{k=1}^{r} \zeta_{ijk} x_{ijk} + \sum_{i=1}^{n} \eta_i y_i \right) \geq \alpha \quad s \in \{1,2, \ldots, S\}
\]  \( (2) \)

\[
\sum_{i=1}^{S} \sum_{j=1}^{n} B^s x_{ijk} - \sum_{i=1}^{n} B^s y_i = b_i^s, \quad j \in \{1,2, \ldots, n\}, \quad s \in \{1,2, \ldots, S\}
\]  \( (3) \)

\[
\sum_{i=1}^{n} x_{ij} = y_j^s, \quad j \in \{2,3, \ldots, n\}, \quad s \in \{1,2, \ldots, S\}
\]  \( (4) \)

\[
\sum_{i=1}^{n} x_{ij} = y_j^s, \quad i \in \{1,2, \ldots, n-1\}, \quad s \in \{1,2, \ldots, S\}
\]  \( (5) \)

\[
\sum_{s=1}^{S} B^s y_i \leq P_i^s, \quad i \in \{1,2, \ldots, n\}
\]  \( (6) \)

\[
AA^s x_{ijk} \leq A_{ijk}, \quad i \in \{1,2, \ldots, n\}, \quad j \in \{1,2, \ldots, r\}, \quad s \in \{1,2, \ldots, S\}
\]  \( (7) \)

\[
AA^s y_i \leq A_i^s, \quad i \in \{1,2, \ldots, s\}, \quad s \in \{1,2, \ldots, S\}
\]  \( (8) \)

\[
x_{ijk} = 0 \text{ or } 1, \quad i \in \{1,2, \ldots, n\}, \quad k \in \{1,2, \ldots, r\}, \quad s \in \{1,2, \ldots, S\}
\]  \( (9) \)

\[
y_j^s = 0 \text{ or } 1, \quad i \in \{1,2, \ldots, n\}, \quad s \in \{1,2, \ldots, S\}
\]  \( (10) \)

In the formulation, Eq. (1) is the objective function, i.e., minimizing the sum of the cost over the entire transportation job, where

\[
Q = \sum_{i=1}^{S} x_{ijk}, \quad k \in \{1,2, \ldots, r\}, \quad s \in \{1,2, \ldots, S\}
\]  \( Q = \sum_{i=1}^{S} y_i \) \( (i \in \{1,2, \ldots, n\}) \). Eq. (2) ensures that the credibility that the time needed on the route is not more than the due date \( T^s \) required by the customer is not less than \( \alpha \), where \( \alpha \) is a defined confidence level (0 < \( \alpha \) < 1). Eq. (3) maintains a balance of the network flow. Eqs. (4) and (5) ensure that the selected nodes and edges are made up of routes from source nodes. Eq. (6) ensures that the capacity constraint of the selected 3PL supplier is less than the transportation capacity \( B^s \) required by the customer. Eq. (7) ensures that the capacity of any node on the route is not less than \( B^s \), which is required by the customer. Eq. (9) ensures that the reputation of the selected 3PL supplier is not less than \( AA^s \), which is required by the customer. Eq. (10) ensures that the reputation of any node is not less than \( AA^s \), which is required by the customer. Finally, Eqs. (11) and (12) define \( x_{ijk} \) and \( y_i^s \) respectively, as 0–1 decision variables.

### 2.3. Fuzzy simulation

In our problem, the constraint in Eq. (2) contains fuzzy parameters. In this subsection, we use fuzzy simulation [17] to estimate it, which is given as follows:

Suppose \( x' \) and \( y' \) are two decision vectors, which are composed of \( x_{ijk} \) and \( y_i^s \) \( (i,j \in \{1,2, \ldots, n\}, k \in \{1,2, \ldots, r\}, s = 1,2, \ldots, S) \) respectively.

Let

\[
f(x',y',\xi,\eta) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{r} \zeta_{ijk} x_{ijk} + \sum_{i=1}^{n} \eta_i y_i^s
\]  \( (12) \)

where \( \xi \) and \( \eta \) are fuzzy vectors composed of \( \zeta_{ijk} \) and \( \eta_i \) \( (i,j \in \{1,2, \ldots, n\}, k \in \{1,2, \ldots, r\}, s = 1,2, \ldots, S) \) respectively. And we
denote that $\mu_{\xi_{ijk}}$ is the membership function of $\xi_{ijk}$ and $\mu_{\eta_{ij}}$ is the membership function of $\eta_{ij}$, respectively.

In the following we will show how to simulate the following fuzzy function:

$$U: (\mathbf{x}', \mathbf{y}') \rightarrow \min \{f^s \mid \text{Cr}(f^s(\mathbf{x}', \mathbf{y}', \xi, \eta) \leq f^s) \geq x\}$$

where $0 < x \leq 1$, $s = 1, 2, \ldots, S$.

Firstly, generate $\theta_{ijk}^q$ and $\sigma_i^q$ from the $\varepsilon$-level sets of fuzzy variables $\xi_{ijk}$ and $\eta_{ij}$ ($i,j \in \{1,2,\ldots,n\}, k \in \{1,2,\ldots,r_{ij}\}$) respectively, where $\varepsilon$ is a sufficiently small positive number, $q = 1,2,\ldots,N$, and $N$ is a sufficiently large number. Set $v_q = \min_{i,k} \{\mu_{\xi_{ijk}}(\theta_{ijk}^q)\} \wedge \min_{i} \{\mu_{\eta_{ij}}(\sigma_i^q)\}$ ($q = 1,2,\ldots,N$). According to the concept of credibility measure, for any number $r$, we set [17]

$$L^s(r) = \frac{1}{2} \left( \max_{i,q,n} \{v_q f_0^s(\mathbf{x}', \mathbf{y}') \leq r \} + \min_{i,q,n} \{1 - v_q f_0^s(\mathbf{x}', \mathbf{y}') > r \} \right)$$

It follows from monotonicity that we may employ bisection search to find the maximal value $r$ such that $L^s(r) \geq x$. This value is an estimation of $f^s$. The process can be run as follows:

**Step 1:** Set $s := 1$.

**Step 2:** Generate $\theta_{ijk}^q$ and $\sigma_i^q$ from the $\varepsilon$-level sets of fuzzy variables $\xi_{ijk}$ and $\eta_{ij}$ ($i,j \in \{1,2,\ldots,n\}, k \in \{1,2,\ldots,r_{ij}\}$) respectively, where $\varepsilon$ is a sufficiently small positive number, and $q = 1,2,\ldots,N$.

**Step 3:** Set $v_q = \min_{i,k} \{\mu_{\xi_{ijk}}(\theta_{ijk}^q)\} \wedge \min_{i} \{\mu_{\eta_{ij}}(\sigma_i^q)\}$ ($q = 1,2,\ldots,N$) for $q = 1,2,\ldots,N$.

**Step 4:** Find the minimal value $r$ such that $L^s(r) \geq x$.

**Step 5:** Set $f^s := r$.

**Step 6:** If $s \geq S$, output $f^s$ ($s = 1,2,\ldots,S$), otherwise $s := s + 1$ and return to Step 2.

Then Eq. (2) could be rewritten as follows:

$$\tilde{f}^s \leq T^s$$

where $s = 1,2,\ldots,S$.

### 3. Proposed memetic algorithm for the M-S 4PLRPFC

In this section, the memetic algorithm based on the fuzzy simulation is designed to solve the M-S 4PLRPFC. In the following, we first describe the MA in detail and then introduce an enumeration algorithm, which is used to check the quality of the solutions obtained by the MA.

#### 3.1. The memetic algorithm

MAs are improved GAs in which a local search is used to intensify the search [21]. In this paper, an MA with a double arrays encoding method is proposed. In the main loop of the algorithm, the selection, crossover, mutation, and local search operators are independently applied to each individual in the current population. The major components of the proposed MA will be described in detail in the corresponding sub-sections.
### 3.1.1. Adaptive double arrays encoding

According to the characteristics of the M-S 4PLRPFC, an encoding based on two ordered arrays is adopted to represent a chromosome in the MA (see Fig. 2), where the first array consists of cities (nodes) encoded by natural numbers and the second array consists of 3PL suppliers (edges) encoded by integers. The chromosome length is variant. For the multi-source problem, there will be multiple routes representing the whole job. The path _s_ ( _s_ = 1, 2, ..., _S_) array is a permutation of the cities on the route starting from the _s_ th source node, and _v_s_ and _v_m_ are the _s_ th source node and the destination, respectively. _v_2, ..., _v_(m−1) are the intermediate cities between _v_s_ and _v_m_. This array means that transporting goods from _v_s_ to _v_m_ requires _v_2, ..., _v_(m−1) as ports. The _PL_ array is a list of 3PL providers for the above transportation. For example, if _e_v_s_v_k_ = 2, then if we start from the _s_ th source node _v_s_ and transport goods from city _v_s_ to city _v_k_, the 3PL supplier with index 2 between the two cities is selected. Additionally, _k_i_ ( _i_ = 1, 2, ..., _m_−1) means that the _i_ th 3PL supplier on the route selected for the transportation from node _v_s_ to _v_(i+1). The two arrays together form an individual on which genetic operators will be applied. Each individual corresponds to _S_ chromosomes in the MA (see Fig. 2), where the first array consists of cities (nodes) encoded by natural numbers and the second array consists of 3PL suppliers (edges) encoded by integers. The chromosome length is variant. For the multi-source problem, there will be multiple routes representing the whole job. The path _s_ ( _s_ = 1, 2, ..., _S_) array is a permutation of the cities on the route starting from the _s_ th source node, and _v_s_ and _v_m_ are the _s_ th source node and the destination, respectively. _v_2, ..., _v_(m−1) are the intermediate cities between _v_s_ and _v_m_. This array means that transporting goods from _v_s_ to _v_m_ requires _v_2, ..., _v_(m−1) as ports. The _PL_ array is a list of 3PL providers for the above transportation. For example, if _e_v_s_v_k_ = 2, then if we start from the _s_ th source node _v_s_ and transport goods from city _v_s_ to city _v_k_, the 3PL supplier with index 2 between the two cities is selected. Additionally, _k_i_ ( _i_ = 1, 2, ..., _m_−1) means that the _i_ th 3PL supplier on the route selected for the transportation from node _v_s_ to _v_(i+1). The two arrays together form an individual on which genetic operators will be applied. Each individual corresponds to _S_ chromosomes in the MA (see Fig. 2), where the first array consists of cities (nodes) encoded by natural numbers and the second array consists of 3PL suppliers (edges) encoded by integers. The chromosome length is variant. For the multi-source problem, there will be multiple routes representing the whole job. The path _s_ ( _s_ = 1, 2, ..., _S_) array is a permutation of the cities on the route starting from the _s_ th source node, and _v_s_ and _v_m_ are the _s_ th source node and the destination, respectively. _v_2, ..., _v_(m−1) are the intermediate cities between _v_s_ and _v_m_. This array means that transporting goods from _v_s_ to _v_m_ requires _v_2, ..., _v_(m−1) as ports. The _PL_ array is a list of 3PL providers for the above transportation. For example, if _e_v_s_v_k_ = 2, then if we start from the _s_ th source node _v_s_ and transport goods from city _v_s_ to city _v_k_, the 3PL supplier with index 2 between the two cities is selected. Additionally, _k_i_ ( _i_ = 1, 2, ..., _m_−1) means that the _i_ th 3PL supplier on the route selected for the transportation from node _v_s_ to _v_(i+1). The two arrays together form an individual on which genetic operators will be applied. Each individual corresponds to _S_ chromosomes in the MA (see Fig. 2), where the first array consists of cities (nodes) encoded by natural numbers and the second array consists of 3PL suppliers (edges) encoded by integers. The chromosome length is variant. For the multi-source problem, there will be multiple routes representing the whole job. The path _s_ ( _s_ = 1, 2, ..., _S_) array is a permutation of the cities on the route starting from the _s_ th source node, and _v_s_ and _v_m_ are the _s_ th source node and the destination, respectively. _v_2, ..., _v_(m−1) are the intermediate cities between _v_s_ and _v_m_. This array means that transporting goods from _v_s_ to _v_m_ requires _v_2, ..., _v_(m−1) as ports. The _PL_ array is a list of 3PL providers for the above transportation. For example, if _e_v_s_v_k_ = 2, then if we start from the _s_ th source node _v_s_ and transport goods from city _v_s_ to city _v_k_, the 3PL supplier with index 2 between the two cities is selected. Additionally, _k_i_ ( _i_ = 1, 2, ..., _m_−1) means that the _i_ th 3PL supplier on the route selected for the transportation from node _v_s_ to _v_(i+1).

The initialization process involves the selection nodes and edges on the route. Taking Fig. 1 as an example, the corresponding adjacent matrix _D_ with node _v_s_ = 1 as the _s_ th source node and node _v_m_ = 8 as the destination is given as follows:

\[
D = (d_{ij})_{8 	imes 8} = \begin{bmatrix}
0 & 3 & 4 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 3 & 0 & 2 & 2 & 0 & 0 \\
0 & 2 & 0 & 3 & 2 & 2 & 0 & 0 \\
0 & 3 & 3 & 0 & 2 & 3 & 2 & 0 \\
0 & 0 & 3 & 2 & 0 & 2 & 3 & 0 \\
0 & 2 & 2 & 3 & 2 & 0 & 2 & 3 \\
0 & 2 & 2 & 2 & 3 & 2 & 0 & 3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

where _d_{ij} ∈ D_ represents whether node _i_ is linked to node _j_ directly in the multi-graph _G(V,E)_. If there is no edge between node _i_ and node _j_, _d_{ij} is set to zero; otherwise, _d_{ij} is set to the number of edges between node _i_ and node _j_. Furthermore, during the transportation process, goods are not allowed to return to the source city after leaving and do not continue to travel after arriving at the destination.

The multi-graph _G(V,E) can be converted into a simple graph as follows. An edge is randomly selected from available edges between any adjacent nodes in the multi-graph so that a sub-graph _G(V,E)_, where _V = V_ and _E_ is the set of edges in the sub-graph, is obtained, which can be described as a matrix _F_ ( _F_ = _f_{ij} ∈ E_). Fig. 3a is a simple graph generated as above. For example, the element _f_{12} = 2_ indicates that in the simple graph _G_, when goods are transported between city 1 and city 2, the second 3PL supplier is chosen to complete the task. A feasible solution is generated based on the simple graph. We start from the source node _v_s_ and choose an adjacent node randomly as the second node _v_2_ (using Fig. 3a as an example, _v_2_ = 2). If _v_s_ and the 3PL supplier represented by _f_{v_s_v_2_} satisfy the capacity and reputation required by the customer, we use _f_{v_s_v_2_} to perform the transportation between node _v_s_ and _v_2_, and set _P_i := P_i – B_i, P_{s} := P_{s} – B_i, F(∗, v_2) := 0_ (where _F(∗, v_2) denotes all elements in the _v_2_ th column of _F_); otherwise, we set _P_i := P_i – B_i, P_{s} := P_{s} – B_i, F(∗, v_2) := 0_ (where _F(∗, v_2) denotes all elements in the _v_2_ th column of _F_); otherwise, we set

\[
\begin{align*}
F := & \begin{bmatrix}
0 & 2 & 0 & 1 & 1 & 2 & 0 \\
0 & 2 & 0 & 1 & 3 & 2 & 0 \\
0 & 2 & 0 & 1 & 2 & 2 & 0 \\
0 & 2 & 2 & 1 & 0 & 1 & 2 \\
0 & 2 & 1 & 3 & 1 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \\
\text{Fig. 3. Route generation.}
\end{align*}
\]
\[ f_{v_s} := 0 \] and choose another adjacent node to examine as above. During this process, if \( f(v_s) = 0 \), which means that there is no node adjacent to the current node \( v_s \), we generate a new simple graph and repeat the above process until we reach the destination, which means a route is generated. This process is repeated for \( S \) times starting from each of the \( S \) source nodes. Then, an initial solution is generated. We record the used time of the nodes and edges and calculate the cost of the \( S \) routes. From Fig. 3, we can see that \( v_s = 7 \) (Fig. 3b), \( v_s = 4 \) (Fig. 3c), \( v_s = 5 \) (Fig. 3d), \( v_s = 6 \) (Fig. 3e) and \( v_s = 8 \) (Fig. 3f).

3.1.3. Fitness function

A fitness function is usually designed according to the objective function and constraints of the problem. In this paper, the time constraint is used as a penalty added to the objective function. The capacity and reputation constraints are adjusted after the crossover operation in Section 3.1.6. Thus, the objective function of a chromosome is represented as follows:

\[
 f(R) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{l} C_{ijk}E_{ijk} + \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij}y_j + \beta \cdot \max(0, T - T^*)
\]

where \( \beta (\beta \geq 1) \) is a penalty factor, which can be set according to a particular problem. Usually, \( \beta \) is set to be a given value and it is defined in Section 4.2.

To calculate the fitness of individuals, we first sort the population size (PS) individuals in the population in the order of decreasing objective value (i.e., the better the chromosome is, the smaller the ordinal number it has). Supposing that the parameter \( \gamma \in (0,1) \) is given, the rank-based evaluation function for calculating the fitness of each chromosome \( i \) is given as follows:

\[
 \text{eval}(l_i) = \gamma(1 - \gamma)^{|i| - 1}, \quad i = 1, \ldots, \text{PS}
\]

3.1.4. Selection mechanism

The selection process is based on the roulette wheel selection scheme, which is a fitness-proportional selection. We spin the roulette wheel PS times. Each time we select a single individual into the mating pool to undergo crossover and mutation. The selection process is as follows:

1. **Step 1**: Calculate the cumulative probability \( q_i \) for each individual \( l_i \) (\( i = 1, \ldots, \text{PS} \)):

\[
 q_0 = 0, \quad q_i = \sum_{j=1}^{i} \text{eval}(l_j), \quad i = 1, \ldots, \text{PS}
\]

2. **Step 2**: Generate a random number \( r \in [0, q_{\text{PS}}] \).
3. **Step 3**: Select the individual \( l_i \) such that \( q_{i-1} < r \leq q_i \) (\( i = 1, \ldots, \text{PS} \)).
4. **Step 4**: Repeat Step 2 and Step 3 PS times to select PS individuals.

3.1.5. Crossover operation

In every generation, crossover is applied to parents with a crossover probability \( P_c \). Due to the characteristics of the encoding method, an individual represents multiple routes. The crossover operation occurs separately on every route. Suppose that \( A \) and \( B \) are two individuals, as shown in Fig. 4, to be exchanged as follows:

Let us take \( A_1 \) (\( A_1 = \text{path}_1 \)) and \( B_2 \) (\( B_2 = \text{path}_2 \)) as an example to show how crossover is operated. Firstly, we determine the number of nodes on \( \text{path}_1 \) and \( \text{path}_2 \), and record the smaller value as \( L \). Secondly, we generate a random crossover point \( h \) between 2 and \( L - 2 \). Thirdly, two children are generated by exchanging the genetic materials of \( \text{path}_1^{h} \) and \( \text{path}_2^{h} \), after the crossover point \( h \). The crossover process for the node array and the 3PL suppliers array are as shown in Figs. 5 and 6, respectively. Finally, we must examine the legality of the children using the following rule. If there are two nodes that are the same on the route, we delete the part between them, keep one of the two nodes, and calculate the time and cost for the new route; otherwise, if the new child is still a route from the source to the destination, quit with success. If not, we examine whether the node before the crossover point is connected with the one behind it. If so, we delete the part between them and randomly add a corresponding 3PL supplier; otherwise, quit with failure. If the child is infeasible, we use the corresponding parent to replace it.

3.1.6. Adjustment to infeasible 3PL providers

Due to the multiple sources in the problem, after the crossover operation, some genes of the individual (some nodes (intermediate cites) or 3PL providers on the route) may not have enough capacity, may have undertaken more tasks, or may have a reputation that cannot satisfy the customer’s requirement. To address the problem that some genes cannot meet the requirements of capacity or reputation, which is caused by the crossover operation, we redistribute the capacity and examine the reputation of all the genes as follows.

Firstly, we redistribute the capacity used by nodes and edges and set \( P_{ijk} \) to their initial values. For each route, e.g., the \( s \)th (\( s \in [1,2,\ldots,S] \)) route, we examine whether the capacity or reputation remaining on both the nodes and edges can satisfy the customer’s requirement. For nodes, if both the capacity and reputation meet the customer’s requirement, we set \( P_{ijk} := P_{ijk} - B^* \); otherwise, we generate a corresponding route again using the method mentioned in Section 3.1.2. For each edge, if both its capacity and reputation meet the customer’s requirement, we set \( P_{ijk} := P_{ijk} - B^* \); otherwise, we examine whether there are any other 3PL providers that can satisfy the requirement. Such 3PL providers exist, we randomly select one and set \( P_{ijk} := P_{ijk} - B^* \); otherwise, we generate a corresponding route again using the method mentioned in Section 3.1.2.

When a new feasible solution is obtained, we calculate the time and cost for nodes and edges for this solution.

3.1.7. Mutation operation

Mutation is carried out on the PL array of individuals according to a probability \( P_m \) as follows:

1. **Step 1**: Calculate the total number of nodes in the path array of an individual, denoted as \( P_l \).
2. **Step 2**: Generate an integer \( p \) randomly between 1 and \( P_l - 1 \).
3. **Step 3**: Choose another 3PL supplier \( k \) randomly from those that have sufficient capacity and reputation to replace the one \( (k) \) between node \( p \) and node \( p + 1 \) of the individual. Set \( P_{p+p+1,k} := P_{p+p+1,k} + B^* \) and \( P_{p+1,k} := P_{p+1,k} - B^* \).

3.1.8. Local search

Considering the cost discount for multi-task, local search (LS) is employed to adjust routes by re-organizing some nodes or 3PL suppliers. Here, two LS operators are designed: one edge-based LS operator and one node-based LS operator. For the edge-based LS operator, the usage of each edge on an individual is checked. If the edge is only used once, we check whether other routes from different sources have used some edges with the same end nodes as those of this edge. If so, we calculate the number of such edges and check whether any of them satisfy the capacity and reputation requirement of the customer. If there are such edges, we choose one of them randomly to replace the original edge and change the capacity and usage of relevant two 3PL providers; otherwise, we make no change. We check all the edges of an individual that are only used once and perform the edge exchanging operation as above to obtain a better multi-task discount and hence an improved solution.
For the node-based LS scheme, we take a chromosome $I$ as an example. We choose two routes randomly from $I$, denoted as $I_1$ and $I_2$. We take the first route as a template and find the common nodes ($v_{v_1}, \ldots, v_{v_o}$, $o$ is the number of common nodes) between $I_1$ and $I_2$. If $o \geq 2$, we try to change the route of $I_2$ between $v_{v_i}$ and $v_{v_i+1}$ ($i = 1, \ldots, o - 1$) as follows:

Step 1: Set $i := 0$.
Step 2: $i := i + 1$. If $i < o$, go to Step 3; otherwise, go to Step 5.
Step 3: If there is any node between $v_{v_i}$ and $v_{v_i+1}$, go to Step 4; otherwise, check whether the edge between $v_{v_i}$ and $v_{v_i+1}$ used by $I_1$ still has the capacity and reputation that can satisfy the requirement given by the customer for $I_2$. If not, go to Step 2; otherwise, we use that edge to replace the edge between $v_{v_i}$ and $v_{v_i+1}$ in $I_2$, update the capacity and reputation of the two involved edges in $I_1$ and $I_2$, and then go to Step 2.
Step 4: Check whether each node between $v_{v_i}$ and $v_{v_i+1}$ in $I_1$ has enough remaining capacity and reputation to satisfy the requirement given by the customer for $I_2$. If there is any node that cannot satisfy the requirement, go to Step 2; otherwise, check whether each edge between $v_{v_i}$ and $v_{v_i+1}$ used by $I_1$ can satisfy the requirement of $I_2$.
Step 4.1: If so, we replace the nodes and edges between $v_{v_i}$ and $v_{v_i+1}$ on $I_2$ by the corresponding nodes and edges between $v_{v_i}$ and $v_{v_i+1}$ on $I_1$, renew the corresponding capacity and reputation of involved nodes and edges, and then go to Step 2.
Step 4.2: Otherwise, for each edge between $v_{v_i}$ and $v_{v_i+1}$ on $I_1$ that cannot satisfy the requirement of $I_2$, check whether there is any other 3PL provider with the same end nodes as those of this edge that can satisfy the requirement of $I_2$. If there is no such 3PL provider, go to Step 2; otherwise, we use such a 3PL provider instead for that edge.

Eventually, we form a new sub-route between $v_{v_i}$ and $v_{v_i+1}$ in $I_1$ that can satisfy the requirement of $I_2$. Use this sub-route to replace the nodes and edges between $v_{v_i}$ and $v_{v_i+1}$ on $I_2$, renew the capacity and reputation of involved nodes and edges, and then go to Step 2.

Step 5: Calculate the fitness of the newly created solution, denoted $I_0$. If $I_0$ is better than the original solution $I$, we replace $I$ with $I_0$.
and update the individual if a better solution arises according to the fitness value.

3.1.9. Framework of the proposed MA
The framework of the proposed MA can be summarized as follows:

Step 1: Initialize the adjacent matrix according to the model graph.
Step 2: Generate the initial population of PS individuals, where PS denotes the population size.
Step 3: Select individuals by the roulette wheel selection scheme.
Step 4: Perform crossover operations on pairs of selected individuals with a crossover probability Pc.
Step 5: Update the capacity infeasible 3PL providers with feasible ones.
Step 6: Perform mutation operations to individuals with a mutation probability Pm.
Step 7: Perform local search operations to each individual and update happens when a better solution arises according to the fitness value.
Step 8: Calculate the fitness value of each solution and update the elitism if a better solution arises.
Step 9: Check whether the maximum allowable number of generations, denoted NG, is reached. If not, go to Step 3; otherwise, go to Step 10.
Step 10: Stop the process and output the best solution.

3.2. The enumeration method
To check the quality of the solutions obtained by the MA, the enumeration method (EM) is used. Because the solution space is very large, the enumeration method is described as follows:

Firstly, produce all the routes from the first source to the destination on a simple graph generated in Section 3.1.2 that satisfy the requirements of capacity and reputation. Find all the routes from the first source on the original multi-graph by exchanging the 3PL supplier with all the other legal ones and record the usage of the nodes and edges. Secondly, for the route generated from the first step, we remove the 3PL suppliers and nodes’ capacity information on G and produce all the routes from the second source node to the destination. This process continues until all the routes from the S source nodes are generated. Finally, all the combinations of S routes are checked and ordered according to the fitness function. The combination with the lowest fitness function value is the best solution.

4. Numerical experiments
In this section, we present experiments used to investigate the performance of the proposed MA for the M-S 4PLRPFC. The experiments are based on two kinds of 4PLRPFC instances as the test bed to compare the solution quality of the proposed MA against the EM and a standard GA (SGA). All the investigated algorithms were encoded in Matlab 7.0 and run on a Core 2 2.83 GHz PC.

4.1. The M-S 4PLRPFC examples
We use two kinds of M-S 4PLRPFCs as the test examples in this study. For the first 8-node M-S 4PLRPFC, we assume that a 4PL company undertakes transnational transportation from Beijing, Tianjin and Shanghai (the source cities) to Canberra (the destination city) and the intermediate cities between them are Hong Kong, Darwin, Brisbane, and Sydney. If there is any business between two cities, an edge is added (see Fig. 7), where each city has the properties of cost, time, and capacity. Because several 3PL supplier companies may exist for transportation between any two cities, there

<table>
<thead>
<tr>
<th>Node</th>
<th>Cost</th>
<th>Time</th>
<th>Capacity</th>
<th>Reputation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>8</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>9</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 1 The data of nodes for the 8-node problem.

<table>
<thead>
<tr>
<th>S</th>
<th>E</th>
<th>E'</th>
<th>C</th>
<th>P</th>
<th>A</th>
<th>Time</th>
<th>S</th>
<th>E</th>
<th>E'</th>
<th>C</th>
<th>P</th>
<th>A</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>8</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>8</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>5</td>
<td>10</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>17</td>
</tr>
</tbody>
</table>

Table 2 The data of edges for the 8-node problem, where “S” and “E” mean the start and end nodes, respectively, and “E’” means the edge between the start and end nodes, and “C”, “P”, and “A” are respectively the cost, transportation capacity, and reputation of the corresponding 3PL supplier.
may be multiple edges between them (one edge stands for a 3PL supplier). This results in a multi-graph, where each edge and node has properties of cost, time, capacity, and reputation. Here, the multi-graph for the M-S 4PLRPFC shown in Fig. 7 is shown in Fig. 1, where each city is assigned an index. Suppose the fuzzy time is represented by triangular fuzzy numbers and assume the following definition:

\[ C_{ijk} \]: the unit cost quoted by the kth 3PL supplier between node i and node j for transportation, if it only undertakes one task for the 4PL.

\[ C_{i} \]: the unit cost quoted by node i for processing, inventory, loading, unloading, etc., if it only undertakes one task for the 4PL.

Additionally,

\[
C_{ijk}(Q) = \begin{cases} 
C_{ijk}, & Q = 0 \text{or} 1 \\
0.9 \cdot C_{ijk}, & Q = 2 \\
0.8 \cdot C_{ijk}, & Q \geq 3
\end{cases}
\]

The node and edge data are given in Table 1 and Table 2, respectively.

### Table 3
Comparison of MA with different \( s \), where 
\( (s = 1, 2, 3) \), “n” means the number of nodes, and “C” means the cost of the best solution, whose Path and 3PL arrays are as shown.

<table>
<thead>
<tr>
<th>( s )</th>
<th>( n )</th>
<th>( B^* )</th>
<th>( T^* )</th>
<th>C</th>
<th>Path</th>
<th>3PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>8</td>
<td>2,3,4</td>
<td>26,25,24</td>
<td>86</td>
<td>[1,3,6,8,2,6,8,3,6,8]</td>
<td>[1,1,2,1,1,1]</td>
</tr>
<tr>
<td>0.7</td>
<td>8</td>
<td>2,3,4</td>
<td>26,25,24</td>
<td>84</td>
<td>[1,2,6,8,2,6,8,3,6,8]</td>
<td>[1,1,2,1,2,1]</td>
</tr>
<tr>
<td>0.6</td>
<td>8</td>
<td>2,3,4</td>
<td>26,25,24</td>
<td>75</td>
<td>[1,2,6,8,2,6,8,2,6,8]</td>
<td>[1,1,2,1,1,1]</td>
</tr>
<tr>
<td>0.8</td>
<td>30</td>
<td>3,3,4</td>
<td>136,133,125</td>
<td>507</td>
<td>2,5,8,11,15,19,22,26,30</td>
<td>[1,1,2,1,2,3,1]</td>
</tr>
<tr>
<td>0.7</td>
<td>30</td>
<td>3,3,4</td>
<td>136,133,125</td>
<td>481</td>
<td>2,5,9,14,18,22,26,30</td>
<td>[1,1,2,1,2,3,1]</td>
</tr>
<tr>
<td>0.6</td>
<td>30</td>
<td>3,3,4</td>
<td>136,133,125</td>
<td>469</td>
<td>2,5,9,14,18,22,26,30</td>
<td>[1,1,3,3,3,3,1]</td>
</tr>
<tr>
<td>0.8</td>
<td>50</td>
<td>3,5,3</td>
<td>252,250,245</td>
<td>653</td>
<td>2,5,9,13,16,19,23,26,30</td>
<td>[1,1,2,2,2,1,1,2,1]</td>
</tr>
<tr>
<td>0.7</td>
<td>50</td>
<td>3,5,3</td>
<td>252,250,245</td>
<td>644</td>
<td>2,5,9,13,16,19,23,26,30</td>
<td>[1,1,2,2,2,1,1,2,1]</td>
</tr>
<tr>
<td>0.6</td>
<td>50</td>
<td>3,5,3</td>
<td>252,250,245</td>
<td>637</td>
<td>2,4,9,13,18,22,26,34,41,45,50</td>
<td>[2,1,2,1,1,1,2,1]</td>
</tr>
</tbody>
</table>

### Table 4
Comparison of cost in non-discount and discount, where “\( \triangle \)” means cost1 − cost2.

<table>
<thead>
<tr>
<th>Non-discount Nodes</th>
<th>Path</th>
<th>PL</th>
<th>Cost1</th>
<th>Discount Path</th>
<th>PL</th>
<th>Cost2</th>
<th>( \triangle )</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>[1,3,6,8,2,6,8,3,6,8]</td>
<td>[1,2,2,3,1,1,1]</td>
<td>101</td>
<td>[1,3,6,8,2,6,8,3,6,8]</td>
<td>[1,2,1,1,1,1]</td>
<td>86</td>
<td>15</td>
</tr>
<tr>
<td>30</td>
<td>[1,5,9,13,16,19,23,26,30]</td>
<td>[1,3,1,1,1,1,1,1]</td>
<td>564</td>
<td>[1,5,9,13,16,19,23,26,30]</td>
<td>[1,3,2,3,2,1,1]</td>
<td>507</td>
<td>57</td>
</tr>
<tr>
<td>50</td>
<td>[2,4,8,13,18,22,24,30,33,38,42,47,50]</td>
<td>[3,2,1,1,4,3,1,1,1,1]</td>
<td>743</td>
<td>[2,4,9,13,18,22,24,30,33,38,42,47,50]</td>
<td>[4,1,3,2,2,1,2,1,2,1,2,1,2,1]</td>
<td>653</td>
<td>90</td>
</tr>
</tbody>
</table>

The other M-S 4PLRPFs are small scale M-S 4PLRPF with 30 and 50 nodes and large scale M-S 4PLRPs with 100, 500, 1000, 1500 and 2000 nodes. All of them are generated randomly. We use \( H := [a, b] \) to mean that \( H \) is an integer number randomly generated between \( a \) and \( b \). Supposing the fuzzy times used in these examples are also triangular fuzzy numbers, the parameters are set as follows:

\[ r_j := [2, 5], \text{if } |i - j| 
\leq 5 \text{ and } i \neq n \text{, and } (i \neq j) \text{ otherwise, } r_j := 0. \]

\[ C_{ijk} := \begin{cases} 
C_{ij}, \text{ if } j = k \\
C_{ij} + C_{jk}, \text{ otherwise }
\end{cases} \]

\[ C_{ij} := \begin{cases} 
C_{ij}, \text{ if } j = k \\
C_{ij} + C_{jk}, \text{ otherwise }
\end{cases} \]

The data are generated using the above method with \( n = 30, 50, 100, 500, 1000, 1500 \) and \( 2000 \). Set \( S = 3, i,j = 1,2, \ldots, n, k = 1,2, \ldots, r_{ij} \) and
Comparison of Two Algorithms with Different Sizes.

Table 5
Comparison of Two Algorithms with Different Sizes.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Nodes</th>
<th>Edges</th>
<th>$T^*$ ($s = 1, 2, 3$)</th>
<th>PS</th>
<th>NG</th>
<th>Best</th>
<th>Bad</th>
<th>Avg</th>
<th>msd</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGA</td>
<td>8</td>
<td>46</td>
<td>26,25,24</td>
<td>30</td>
<td>30</td>
<td>86</td>
<td>90</td>
<td>87</td>
<td>1.1</td>
<td>&lt;1</td>
</tr>
<tr>
<td>MA</td>
<td>8</td>
<td>46</td>
<td>26,25,24</td>
<td>30</td>
<td>30</td>
<td>86</td>
<td>88</td>
<td>86</td>
<td>0.5</td>
<td>&lt;1</td>
</tr>
<tr>
<td>EM</td>
<td>8</td>
<td>46</td>
<td>26,25,24</td>
<td>–</td>
<td>–</td>
<td>86</td>
<td>86</td>
<td>86</td>
<td>–</td>
<td>522</td>
</tr>
<tr>
<td>SGA</td>
<td>30</td>
<td>766</td>
<td>136,133,125</td>
<td>80</td>
<td>40</td>
<td>507</td>
<td>523</td>
<td>515</td>
<td>7.2</td>
<td>2</td>
</tr>
<tr>
<td>MA</td>
<td>30</td>
<td>766</td>
<td>136,133,125</td>
<td>50</td>
<td>40</td>
<td>507</td>
<td>514</td>
<td>510</td>
<td>1.8</td>
<td>4</td>
</tr>
<tr>
<td>EM</td>
<td>30</td>
<td>766</td>
<td>136,133,125</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>SGA</td>
<td>50</td>
<td>1414</td>
<td>252,250,245</td>
<td>100</td>
<td>60</td>
<td>653</td>
<td>678</td>
<td>662</td>
<td>10.6</td>
<td>10</td>
</tr>
<tr>
<td>MA</td>
<td>50</td>
<td>1414</td>
<td>252,250,245</td>
<td>80</td>
<td>60</td>
<td>653</td>
<td>663</td>
<td>657</td>
<td>3.4</td>
<td>19</td>
</tr>
<tr>
<td>EM</td>
<td>50</td>
<td>1414</td>
<td>252,250,245</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>SGA</td>
<td>100</td>
<td>3104</td>
<td>600,610,620</td>
<td>200</td>
<td>100</td>
<td>2049</td>
<td>2105</td>
<td>2188</td>
<td>52.6</td>
<td>62</td>
</tr>
<tr>
<td>MA</td>
<td>100</td>
<td>3104</td>
<td>1800,1900,2000</td>
<td>100</td>
<td>100</td>
<td>2049</td>
<td>2131</td>
<td>2085</td>
<td>14.2</td>
<td>130</td>
</tr>
<tr>
<td>SGA</td>
<td>100</td>
<td>51700</td>
<td>1800,1900,2000</td>
<td>400</td>
<td>200</td>
<td>3862</td>
<td>4077</td>
<td>3963</td>
<td>23.6</td>
<td>246</td>
</tr>
<tr>
<td>MA</td>
<td>100</td>
<td>51700</td>
<td>1800,1900,2000</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>EM</td>
<td>100</td>
<td>51700</td>
<td>1800,1900,2000</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

$s = 1, 2, \ldots, S$. The nodes indexed 1, 2 and 3 are the sources and the nodes indexed 30, 50, 100, 500, 1000, 1500, and 2000 are respectively as the destination node.

4.2. Performance analysis

4.2.1. Effect of $\alpha$ on the decision

In this section, the effect of $\alpha$ on the decision is tested with the small scale examples, which have 8, 30 and 50 nodes, respectively. Record the average time taken by the algorithm for one run. The parameters considered are the number of generations (NG), PS, K, $P_c$, and $P_m$. The algorithms were run 100 times and the best solution (Best), the worst solution (Bad), the average value (Avg), and the mean square deviation of $T(msd)$ were recorded. Avg and msd were computed by

$$\text{Avg}(T) = \frac{1}{n} \sum_{i=1}^{n} T_i$$

and

$$\text{msd}(T) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (T_i - \text{Avg}(T))^2}$$

where $n$ is the number of runs and $T_i$ is the best solution in the $i$th run.

The parameters are set as follows:

$\beta = 20$, $\gamma = 0.05$, $PS = 20$, $NG = 30$, $P_c = 0.5$, $P_m = 0.2$

The experimental results are shown in Table 3. Using 8-node M-S 4PLRPFC as an example, if the due date is set to 26, 25, or 24 days, the capacity is set to be 2, 3, and 4 and the reputation is 2, 3, and 3 for each source node. When the credibility parameter $\alpha$ is set to 0.8, the total cost is 85.8 ($\approx 88$) and $C_r(T \in T^*) = 0.8$. In the 4PL operation, this means that when goods are to be transported from Beijing to Canberra within 26 days with a credibility 0.8, the best choice is to use Shanghai and Brisbane as interim ports, and the 3PL suppliers with indexes 1, 1, and 2 are chosen for the three links from Beijing to Shanghai, Shanghai to Brisbane, and Brisbane to Canberra, respectively. In the same way, when goods are to be transported from Tianjin and Shanghai to Canberra within 25 and 24 days, respectively, Brisbane is used as the interim port, and the 3PL suppliers with the same index 1, 1 and 1 are chosen for the corresponding links from Tianjin to Brisbane, Brisbane to Canberra, Shanghai to Brisbane, and Brisbane to Canberra, respectively. It costs the 4PL company 86 to finish the entire job.

From Table 3, it can also be seen that given a due date ($T_s$) for each route, when $\alpha$ is changed, the most suitable routes are different. Given the same due date, the cost needed on the routes will be less with the requirement of a lower credibility level. Therefore, the customers could select which $\alpha$ is the best considering their time requirements.

4.2.2. Effect of the cost discount on the decision

The results of cost in non-discount and discount with $n = 8, 30$ and 50 are tested. From Table 4 we can see that, as the 3PL providers and transit nodes contribute a cost discount when they undertake multiple tasks, 4PL providers would prefer to select a 3PL provider or node to perform multiple jobs to reduce the total cost and accelerate the transfer links, allowing them to gain more profit.

4.2.3. Effect of the problem scale on the algorithm

The purpose of the experiments in this section is to evaluate the efficiency of the designed MA for solving M-S 4PLRPFC with different scale examples. The experiments include the above three small scale problems and five randomly generated large scale problems with the number of (nodes, edges) equal to (100, 3104), (500, 15,700), (1000, 31,345), (1500, 47,119), and (2000, 63,250), requiring 1, 2, 12, 38 and 89s, respectively, to generate the large networks. The parameters of the test algorithms were set based on the balance between the operator Avg and msd.

The first four examples are executed for 50 runs, and the other examples are executed for 20 runs. To compare the running time of each algorithm, the time values computed in the following do not include the time required to generate the network.

In Table 5, ‘−’ means that a solution could not be obtained within an acceptable time. According to the EM designed in Section 3.2, for the 8-node problem, there are approximately $189800$ combinations in one simple graph and over $6.0 \times 10^{10}$ combinations in the multi-graph, and its optimal result is the same as that obtained by MA and SGA. When EM is used in problems with more
nodes, i.e., the 30- and 50-node problems, it cannot produce an optimal result within an acceptable time. With a credibility level 0.8, from Table 5, it can be seen that the solution obtained by MA and SGA is optimal and that MA and SGA use much less time than EM.

For the relatively large scale problems, from Table 5, it can be seen that SGA can obtain the same best solution as the MA for the problems with 100, 500, and 1000 nodes. However, the average quality of the obtained solutions and the deviations, even the best solution obtained by the designed MA are better than those obtained by SGA. When the number of nodes increases to 1500, SGA cannot find the same best solution as MA. Therefore, the proposed MA can achieve an optimal solution with a higher probability. In summary, the EM can guarantee the optimal solution when the problem size is small, but it cannot obtain the solution within an acceptable time when the problem size is larger. Although SGA can solve the larger problem quickly, it can only produce a poor-quality optimal solution. When we add a local search operator in the proposed algorithm, the MA is able to efficiently improve the performance of complex combinatorial optimization problems.

5. Conclusion

With the increasing intensity of market competition and rapid development of science and technology, an increasing number of enterprises have begun to realize the importance of the fourth party logistics (4PL) and have dedicated increased attention to the 4PL routing problem (4PLRP). In this paper, the multi-source single destination 4PLRP with fuzzy duration time and cost discount (M-S 4PLRPFC) was discussed, which can be described as a selection of the shortest path problem with constraints of fuzzy variants in a multi-graph. A fuzzy programming model was built for this problem based on the fuzzy theory, and a fuzzy simulation method is provided to solve it.

Based on the fuzzy simulation method, an memetic algorithm (MA) is proposed. A double arrays encoding method is used for the representation of the solutions. A standard GA is then employed to search for satisfactory solutions, and a local search algorithm is employed by integrating 3PL suppliers and nodes to obtain better solutions.

Numerical experiments were carried out to investigate the performance of the proposed MA. The numerical experiments show that the proposed MA yields the same results as the enumeration method and that the MA is a better algorithm for solving the M-S 4PLRPFC than the standard GA. The experimental results indicate that the proposed MA is an efficient method for solving the M-S 4PLRPFC.

For the future work, other fuzzy factors, such as fuzzy cost may be handled. Moreover, more realistic problem such as multi-destination problem may be taken into consideration.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant Nos. 71071028, 71021061, 70931001 and 61070162, the National Science Foundation for Distinguished Young Scholars of China under Grant No. 61225012; the Specialized Research Fund of the Doctoral Program of Higher Education for the Priority Development Areas under Grant No.20120042130003; Specialized Research Fund for the Doctoral Program of Higher Education under Grant Nos. 20110042110024 and 20100042110025; the Specialized Development Fund for the Internet of Things from the ministry of industry and information technology of the P.R. China; the Fundamental Research Funds for the Central Universities under Grant Nos. N110204003 and N120104001; the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

References