



Model and algorithm for 4PLRP with uncertain delivery time



Min Huang^a, Liang Ren^{a,*}, Loo Hay Lee^b, Xingwei Wang^a, Hanbin Kuang^a,
Haibo Shi^c

^a College of Information Science and Engineering, State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, Liaoning 110819, China

^b Department of Industrial and Systems Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore

^c Shenyang Institute of Automation (SIA), Chinese Academy of Sciences, Shenyang, Liaoning 110819, China

ARTICLE INFO

Article history:

Received 24 January 2015

Revised 13 October 2015

Accepted 17 October 2015

Available online 23 October 2015

Keywords:

Fourth party logistics routing problem

Uncertainty theory

Multi-graph

Intelligent algorithm

Shortest path problem

ABSTRACT

To address the challenge of logistics routing decision under uncertain environment, this paper studies a fourth party logistics routing problem (4PLRP) with uncertain delivery time (4PLRPU). A novel 4PLRPU model based on uncertainty theory is proposed by describing the delivery time of a third party logistics (3PL) provider as an uncertain variable. After that, the model is transformed into an equivalent deterministic model, and several improved genetic algorithms are designed to get solutions. To handle the problem of infeasible solutions in the proposed 4PLRPU, an improved node-based genetic algorithm (INGA) and an improved distance-based genetic algorithm (IDGA) are developed to reduce the computing time required to repair infeasible solutions, and an improved genetic algorithm based on the simple graph and Dijkstra algorithm (SDGA) is proposed to avoid the generation of infeasible solutions. Numerical experiments are conducted to investigate the performance of the proposed algorithms and verify the effectiveness of the proposed 4PLRPU model. The results show that INGA and SDGA are more effective than the standard genetic algorithm and IDGA at solving large-scale problems. Additionally, compared with the expected value model, the 4PLRPU model is more robust.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

In supply chain environment, it is accepted that customers' expectations of their suppliers will increase continually [3]. To meet customers' needs, companies turn to third party logistics (3PL) providers. Currently, 3PL has been widely accepted. However, in addition to the professional strength and skills in traditional logistics fields, such as transportation, warehousing, customs clearance and distribution, 3PL providers realize that they must strengthen information technology, knowledge management and strategic alliances to satisfy the requirements of the modern logistics.

There is an increasing attention on the development of the fourth party logistics (4PL), which strives to the integration and management of the logistics supply chain. The concept of 4PL was put forward by Andersen Consulting [17]: "It 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." Compared with 3PL, the inevitabilities and advantages of 4PL are gradually and widely becoming recognized [45]. 3PL providers are turning to 4PL to build closer relationships amongst the participants along the

* Corresponding author. Tel.: +86 02483691272.

E-mail address: neurenliang@163.com (L. Ren).

supply chain, to support cost cutting initiatives, to develop the flexibility to deal with supply and demand uncertainties and, ultimately, to have a positive impact on the bottom-line [44].

Since 4PL was introduced, many scholars have devoted research time and effort. First, researchers discussed the concepts of 4PL and revealed that 4PL is emerging as a breakthrough solution to modern supply chain challenges [3,13]. Second, some literature has illustrated the relationship between 3PL and 4PL. Some have studied performance evaluation problems between 4PL and 3PL providers [6,8,30], and others have studied the contract design to benefit them both [26,31]. Third, researchers have focused on the 4PL routing problem (4PLRP). Huang et al. [23] studied the basic 4PLRP. Zhang et al. [49] researched the 3PL selection problem in the 4PLRP. Huang et al. [24] discussed optimization of the 4PLRP with soft time windows. Cui et al. [10] and Huang et al. [25] studied the 4PLRP with fuzzy duration time. Li and Huang [32] worked on the multiple transportation modes selection problem in the 4PLRP.

The 4PLRP is a fundamental and unavoidable problem in 4PL's actual operation. Its core content is to manage and optimize the distribution process from the source to the destination; this directly relates to the operational efficiency [27]. The 4PLRP includes two main issues: 3PL providers' selection and path selection. Some scholars [25] deal with these two issues integrally, and use the graph theory to simplify the 4PLRP and clarify the problem's description. Then, the complexity of a graph structure directly relates to the difficulty of the problem. A 4PLRP case can be described as a complex multi-graph network including many alternative 3PL providers, routes and transit cities. Note that in this multi-graph network, more than one edge may exist between two nodes, which dramatically increases the complexity of the problem.

Most of the previous literature on 4PL is worked under deterministic environments [23], but decisions in reality are usually made in the state of indeterminacy [36]. For modeling indeterminacy, there are two types of situations. One is to estimate the probability distribution with enough samples. In this situation, the indeterminacy can be described as randomness based on probability theory [29]. The other is that no samples or not enough samples can be collected to estimate a probability distribution. In this situation, people have to invite some domain experts to evaluate the belief degree that events will occur. Perhaps some people think that the belief degree should be modeled by subjective probability or fuzzy set theory. Tversky and Kahneman [43] claimed that, for both positive and negative prospects, people overweight low probabilities and underweight moderate and high probabilities. Liu [36] claimed that human beings usually overweight unlikely events. Then, the belief degree function will have a much larger variance than the long-run cumulative frequency. Hence, the belief degree cannot be treated as a probability distribution. At the same time, fuzzy set theory is not self-consistent in mathematics and may lead to wrong results in practice [36]. Therefore, the belief degree can neither be described as a stochastic number nor a fuzzy number.

Thus, a new methodology based on belief degree, called uncertainty theory [36], is incorporated into our model to describe the delivery time of 3PL providers. Uncertainty theory, proposed in 2007 and refined in 2010 by Liu [33,35], is another efficient tool for handling nondeterministic information with expert data [16]. Currently, uncertainty theory has become a branch of axiomatic mathematics for modeling human uncertainty and has been applied to many areas, including economics [14,48], finance [9,19,28] and management [16,46,47]. For example, Mou et al. [40] researched on an emergency scheduling problem in a transportation network in which the truck times and transportation costs were assumed as uncertain variables. Han et al. [20] studied the maximum flow problem in an uncertain network in which the edge capacities were assumed as uncertain variables. Gao [15] studied the α -shortest path problem and the most shortest path problem in a simple graph based on uncertainty theory.

In this paper, the uncertain programming model of 4PLRP with uncertain delivery time (4PLRPU) is established, where the delivery time is described as an uncertain variable. Because the uncertain measure is included, the 4PLRPU model is difficult to solve. Therefore, the 4PLRPU model is equivalently transformed into a deterministic model, which is regarded as a constrained shortest path problem (CSPP) in a multi-graph. To solve the CSPP, the basic and popular early approaches are the k-shortest path algorithm [12] and dynamic programming [4,21]. However, these approaches are mainly designed for simple graphs. Huang et al. [25] showed that these approaches are not effective enough to solve CSPP in a multi-graph. Another approach widely used in recent years is the evolutionary algorithm (EA) [5]. There are different types of EAs [2,18,41], in which the genetic algorithm [22] is one of the most classic EAs with good performance in solving CSPP. Therefore, to solve the proposed model after equivalent transformation, an improved enumeration algorithm (ENUM), genetic algorithm (GA) and three improved GAs are designed from different perspectives. The infeasible solution that is not a connected route from the source to the destination is one of the key factors impacting the search efficiency for the proposed problem. Therefore, an improved node-based genetic algorithm (INGA) and an improved distance-based genetic algorithm (IDGA) are proposed to reduce the running time and computing cost of repairing this type of infeasible solutions, and an improved genetic algorithm based on the simple graph and Dijkstra algorithm (SDGA) is proposed to avoid this type of infeasible solution. Numerical experiments are conducted to investigate the performance of the proposed algorithms. Comparison and analysis show the effectiveness of the proposed model and algorithms.

The remainder of this paper is organized as follows: Section 2 introduces basic concepts and properties of uncertainty theory used throughout this paper. Section 3 presents the problem description and formulation of the proposed 4PLRPU. Section 4 introduces the proposed algorithms in detail. In Section 5, numerical examples are used to investigate the performance of the proposed algorithms and the robustness of the 4PLRPU model. Finally, Section 6 provides a brief summary of this paper.

2. Preliminary

In this section, basic concepts of the uncertainty theory are presented briefly.

Let Γ be a nonempty set, and \mathcal{L} a σ -algebra over Γ . Each element $\Lambda \in \mathcal{L}$ is assigned a number $\mathcal{M}\{\Lambda\}$. In order to ensure that the number $\mathcal{M}\{\Lambda\}$ has certain mathematical properties, Liu [33] presented the following three axioms:

Axiom 1. (Normality Axiom) $\mathcal{M}\{\Gamma\} = 1$ for the universal set Γ .

Axiom 2. (Duality Axiom) $\mathcal{M}\{\Lambda\} + \mathcal{M}\{\Lambda^c\} = 1$ for any event Λ .

Axiom 3. (Subadditivity Axiom) For every countable sequence of events $\Lambda_1, \Lambda_2, \dots$, we have

$$\mathcal{M}\left\{\bigcup_{i=1}^{\infty} \Lambda_i\right\} \leq \sum_{i=1}^{\infty} \mathcal{M}\{\Lambda_i\}.$$

Definition 1 (Liu [33]). The set function \mathcal{M} is called an uncertain measure if it satisfies the normality, self-duality, and countable subadditivity axioms.

Definition 2 (Liu [33]). An uncertain variable is a measurable function ξ from an uncertain space $(\Gamma, \mathcal{L}, \mathcal{M})$ to the set of real number, i.e., $\{\xi \in \mathcal{B}\}$ is an event for any Borel set \mathcal{B} .

Definition 3 (Liu [33]). Let $\xi_1, \xi_2, \dots, \xi_n$ be uncertain variables, and f a real-valued measurable function. Then $\xi = f(\xi_1, \xi_2, \dots, \xi_n)$ is an uncertain variable defined by

$$\xi(\gamma) = f(\xi_1(\gamma), \xi_2(\gamma), \dots, \xi_n(\gamma)), \forall \gamma \in \Gamma.$$

Definition 4 (Liu [33]). The uncertainty distribution Φ of an uncertain variable ξ is defined by

$$\Phi(x) = \mathcal{M}\{\xi \leq x\}$$

for any real number x .

Definition 5 (Liu [33]). An uncertain variable ξ is called normal if it has a normal uncertainty distribution

$$\Phi(x) = \left(1 + \exp\left(\frac{\pi(e-x)}{\sqrt{3}\sigma}\right)\right)^{-1}, x \in R$$

denoted by $\mathcal{N}(e, \sigma)$ where e and σ are real numbers with $\sigma > 0$.

Definition 6 (Liu [33]). Let ξ be an uncertain variable. Then the expected value of ξ is defined by

$$E[\xi] = \int_0^{+\infty} \mathcal{M}\{\xi \geq r\}dr - \int_{-\infty}^0 \mathcal{M}\{\xi \leq r\}dr$$

provided that at least one of the two integrals is finite.

Definition 7 (Liu [33]). Let ξ be an uncertain variable with finite expected value e . Then the variance of ξ is

$$V[\xi] = E[(\xi - e)^2].$$

Theorem 1 (Liu [34]). Assume that the constraint function $g(x, \xi_1, \xi_2, \dots, \xi_n)$ is strictly increasing with respect to $\xi_1, \xi_2, \dots, \xi_k$ and strictly decreasing with respect to $\xi_{k+1}, \xi_{k+2}, \dots, \xi_n$. If $g(x, \xi_1, \xi_2, \dots, \xi_n)$ are independent uncertain variables with uncertainty distribution $\Phi_1, \Phi_2, \dots, \Phi_n$, respectively, then the chance constraint

$$\mathcal{M}\{g(x, \xi_1, \xi_2, \dots, \xi_n) \leq 0\} \geq \alpha$$

holds if and only if

$$g(x, \Phi_1^{-1}(\alpha), \dots, \Phi_k^{-1}(\alpha), \Phi_{k+1}^{-1}(1-\alpha), \dots, \Phi_n^{-1}(1-\alpha)) \leq 0,$$

where $\Phi_k^{-1}(\alpha)$ is the inverse function of Φ_k under confidence level α , and called inverse uncertainty distribution of ξ_k .

Theorem 2 (Liu [35]). Let ξ_1 and ξ_2 be independent normal uncertain variables $\mathcal{N}(e_1, \sigma_1)$ and $\mathcal{N}(e_2, \sigma_2)$, respectively. Then the sum $\xi_1 + \xi_2$ is also a normal uncertain variable $\mathcal{N}(e_1 + e_2, \sigma_1 + \sigma_2)$, i.e.,

$$\mathcal{N}(e_1, \sigma_1) + \mathcal{N}(e_2, \sigma_2) = \mathcal{N}(e_1 + e_2, \sigma_1 + \sigma_2).$$

3. Problem description and formulation

The 4PLRP can be defined as selecting logistics companies for a 4PL to optimize the supply chain [25]. One key problem of 4PLRPU in this paper is choosing 3PL providers to finish the transportation task from the source to the destination to ensure cost minimization and the satisfaction of customer's time constraint.

Assume that a 4PL enterprise undertakes a task from the source to the destination. The transportation network and candidate 3PL providers are determined. To more concisely and clearly describe the problem, an undirected multi-graph is used to describe the transportation network as is shown in Fig. 1. V is the set of nodes, which represent depots, material warehouse, sale location, etc. $|V| = n$ is the number of nodes. Node v_s represents the source and node v_t represents the destination. Other nodes are transit points. E is the set of edges, which represent different 3PL providers that are available to provide the transportation service

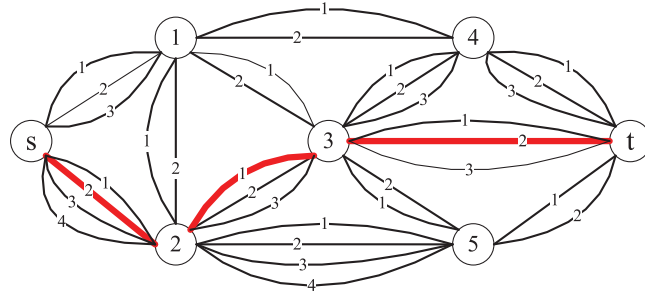


Fig. 1. Multi-graph of the 4PLRP. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

between connected nodes. In such a multi-graph, there is more than one edge (3PL provider) between each pair of nodes. Nodes and edges have properties of cost and time. Under uncertain environment, the 3PL provider's delivery time is obtained from some domain experts; therefore, an uncertain variable is used to address it. The 4PL needs to select a feasible route which includes interconnected nodes and edges from the source to the destination, as red lines show in Fig. 1.

To formulate the problem, parameters are defined as follows.

Parameters	
r_{ij}	The number of edges (3PL providers) between node i and node j .
e_{ijk}	The k^{th} edge between node i and node j , where $i, j \in (1, 2, \dots, n)$, and k is the index number of edges. For example, e_{122} is the second edge between node v_1 and node v_2 .
C_{ijk}	The cost of the 3PL provider represented by e_{ijk} .
$T_{ijk} + t_{ijk}$	The delivery time of the 3PL provider represented by e_{ijk} . T_{ijk} is the basic time, t_{ijk} is an uncertain variable, which represents time perturbation.
C_j	The cost of node j , including the cost of processing, inventory, loading and unloading.
T_j	The time of node j .
T_0	The customer's required due date.
α	The confidence level.

To generalize more, we assume that t_{ijk} are independent uncertain variables with uncertainty distribution of $\Phi_{ijk}(t)$ and subject to normal uncertainty distribution of $\mathcal{N}(\mu_{ijk}, \sigma_{ijk}^2)$, where μ_{ijk} and σ_{ijk} are the expected value and the standard variance, and σ_{ijk}^2 is the variance. The confidence level $\alpha \in (0.5, 1)$.

With the definition of parameters, the mathematical model for the proposed 4PLRPU can be built. Note that 3PL provider's delivery time for each edge consists of two parts T_{ijk} and t_{ijk} , where T_{ijk} is a constant, and t_{ijk} is an uncertain variable. The goal of this problem is to find a route from the source to the destination at minimum cost, and is also subject to time constraint, which requests the belief degree of completing the task before the due date be larger than the required confidence level. Then, the uncertain programming model of 4PLRPU is presented as follows:

$$\min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{r_{ij}} C_{ijk} x_{ijk}(R) + \sum_{j=1}^n C'_j y_j(R) \tag{1}$$

s.t.

$$\mathcal{M} \left\{ \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{r_{ij}} (T_{ijk} + t_{ijk}) x_{ijk}(R) + \sum_{j=1}^n T'_j y_j(R) \leq T_0 \right\} \geq \alpha \tag{2}$$

$$x_{ijk}(R) = \begin{cases} 1, & \text{if } e_{ijk} \in R \\ 0, & \text{else} \end{cases} \tag{3}$$

$$y_j(R) = \begin{cases} 1, & \text{if } v_j \in R \\ 0, & \text{else} \end{cases} \tag{4}$$

$$R = (v_s, \dots, v_i, k, v_j, \dots, v_t) \in G \tag{5}$$

In the formulation, Eq. (1) is the objective function, i.e. minimizes the total cost, including node costs and edge costs. Eq. (2) means that the total time needed on the route should not be more than the due date T_0 with at least the given confidence level α . $x_{ijk}(R)$ and $y_j(R)$ in Eqs. (3) and (4) are 0–1 variables. Eq. (3) represents whether edge e_{ijk} is included in route R , and Eq. (4) represents whether node v_j is included in route R . Eq. (5) is used to ensure the selected route is legal, which means that nodes and edges of the route R must be interconnected one by one, and must start from the source and end at the destination. Furthermore, there is no loop in the route R .

The 4PLRPU model with uncertain measure \mathcal{M} is difficult to solve, therefore, based on uncertainty theory (Theorem 1), Eq. (2) can be equivalently written as Eq. (6), where the chance constraint is transformed into a deterministic one. $\Phi_{ijk}^{-1}(\alpha)$ is the inverse

uncertainty distribution of $\Phi_{ijk}(t)$.

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{r_{ij}} \Phi_{ijk}^{-1}(\alpha) x_{ijk}(R) + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{r_{ij}} T_{ijk} x_{ijk}(R) + \sum_{j=1}^n T'_j y_j(R) \leq T_0 \tag{6}$$

Since $t_{ijk} \sim \mathcal{N}(\mu_{ijk}, \sigma_{ijk})$, we have

$$\Phi_{ijk}^{-1}(\alpha) = \mu_{ijk} + \frac{\sqrt{3}\sigma_{ijk}}{\pi} \ln \frac{\alpha}{1-\alpha}.$$

4. Algorithm design

Because the 4PLRPU model with uncertain measure and uncertain variables is transformed into a deterministic one, the model can be considered as a CSPP in a multi-graph. Liu et al. [37] stated that the CSPP is an NP-hard problem. Furthermore, for the deterministic equivalent problem of the 4PLRPU, there is more than one 3PL provider between two nodes; therefore, the number of alternative solutions grows exponentially, which increases the difficulty in solving the problem. Thus, EA is introduced to solve this deterministic equivalent problem. GA [22] is one of the most widely used EAs in recent decades to solve constrained optimization problems [42]. Because of the strong search capability, controllable search process and easy improvability, many scholars [37,39] use GA to solve the CSPP. In this research, several improved GAs are designed to solve our problem.

When solving the 4PLRPU model using GA, the infeasible solution that is not a route from the source to the destination is one of the key factors impacting the search efficiency. Repairing these infeasible solutions is complicated and time-consuming. Therefore, IDGA and INGA are proposed to reduce the running time and computing cost of repairing this type of infeasible solution and SDGA is proposed to avoid this type of infeasible solution. Furthermore, we try to transform the complex problem into multiple simple problems and solve these sub-problems using classic methods. INGA is designed from the perspective of the k-th shortest path, and SDGA is designed from the perspective of the constrained Dijkstra algorithm [11].

To evaluate the efficiency of the proposed algorithms, ENUM is designed to search for the optimal solutions of small-scale problems, and is treated as a benchmark for the other algorithms.

4.1. ENUM

Because the multi-graph has a huge solution space, the general enumerative algorithm may not be effective. To obtain the optimal solution of the proposed 4PLRPU, ENUM is designed in two steps, which are briefly described as follows:

- (1) Simple graphs are enumerated based on the multi-graph instead of enumerating routes directly. The simple graph should contain all of the nodes on the multi-graph. For each pair of nodes, if there are edges between them in the multi-graph, only one should be chosen into the simple graph.
- (2) The constrained shortest path problem (CSPP) on every simple graph is solved with the constrained Dijkstra algorithm [11].

4.2. GA

With the advantages of strong search capability, controllable search process, easy provision of near-optimal solutions, easy improvability, etc., GA is considered as one of the most popular EAs.

First, an upper triangular matrix ($F = (f_{ij})_{n \times n}$) of dimension $n \times n$ is used to represent the multi-graph. Each element $f_{ij} \in F$ indicates whether node i is linked to node j directly in the multi-graph. If there is no edge between node i and node j , f_{ij} is set to zero; otherwise, f_{ij} is set to the number of edges between node i and node j . For example, the corresponding matrix of the multi-graph in Fig. 1 is shown in Eq. (7). In matrix F_1 ($F_1 = (f_{ij})_{7 \times 7}$), the element value indicates the number of edges between two nodes. Therefore, the multi-graph can be simply represented by a nonzero digital string (342223432332), which is constructed according to the nonzero elements in the matrix from left to right and top to bottom.

$$F_1 \begin{matrix} v_s & 1 & 2 & 3 & 4 & 5 & v_t \\ v_s & \begin{bmatrix} 0 & 3 & 4 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 2 & 2 & 0 & 0 \\ 2 & 0 & 0 & 0 & 3 & 0 & 4 & 0 \\ 3 & 0 & 0 & 0 & 0 & 3 & 2 & 3 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ v_t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \tag{7}$$

Then, when a route is selected, it can also be represented by an upper triangular matrix. Each element f_{ij} represents the edge between node i and node j that is selected by the task or which edge exists on the route. If there is no edge selected between node i and node j , f_{ij} is set to zero; otherwise, f_{ij} is set to the marked number of the selected edge. Only one edge can be selected between two adjacent nodes. Therefore, the route can also be simply represented by a digital string. This string is the code of

a solution (Chromosome). For example, the red line route in Fig. 1 can be represented as (020001000200); the corresponding matrix F_2 is shown in Eq. (8). Each number of the code (Gene) denotes the selected edge between two nodes at the corresponding position. $f_{13} = 2$ means the second edge between v_s and v_2 is selected.

$$F_2 \begin{matrix} & v_s & 1 & 2 & 3 & 4 & 5 & v_t \\ v_s & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ v_t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \quad (8)$$

Because it is a basic GA, the detailed process will not be described here.

4.3. IDGA

There are two crucial issues affecting the search efficiency of GA to solve the proposed 4PLRPU. One is how to address infeasible solutions (illegal routes). There are two common ways: discard these infeasible solutions directly and reselect randomly among the breakpoints. However, neither approach makes good use of the original individuals. The proposed IDGA is designed to reduce the running time and computing cost of repairing this type of infeasible solution; The other is how to reduce the time spent on backtracking in the complex multi-graph network. Researchers are always exploring how to search faster and broader. Therefore, IDGA is designed to improve the standard GA from the following aspects:

(1) Repair strategy.

- (a) To repair non-connected nodes in an illegal route, the first node v_i whose following node is not connected and the last node v_j whose prior node is not connected are determined. In this process, repeated nodes are avoided.
- (b) The distance matrix ($F = (f_{ij})_{n \times n}$) of the multi-graph is determined. Each element f_{ij} represents the relative distance between node i and node j . If node i and node j are connected directly, f_{ij} is set to 1; otherwise, f_{ij} is set to the minimum number of edges needed to reach node j from node i . For example, the distance matrix F_3 is shown in Eq. (9) for the multi-graph of Fig. 1. $f_{12} = 1$ means that v_s and v_1 are adjacent nodes. $f_{14} = 2$ means that the relative distance of v_s and v_3 is 2, and at least two edges are needed to reach v_3 from v_s .

$$F_3 \begin{matrix} & v_s & 1 & 2 & 3 & 4 & 5 & v_t \\ v_s & 0 & 1 & 1 & 2 & 2 & 2 & 3 \\ 1 & 1 & 0 & 1 & 1 & 1 & 2 & 2 \\ 2 & 1 & 1 & 0 & 1 & 2 & 1 & 2 \\ 3 & 2 & 1 & 1 & 0 & 1 & 1 & 1 \\ 4 & 2 & 1 & 2 & 1 & 0 & 2 & 1 \\ 5 & 2 & 2 & 1 & 1 & 2 & 0 & 1 \\ v_t & 3 & 2 & 2 & 1 & 1 & 1 & 0 \end{matrix} \quad (9)$$

- (c) Beginning from v_i , instead of choosing the following edge (multiple edges for every two nodes), we choose the following node v_k first based on the distance matrix (as Eq. (9)) shows) of the multi-graph. The distance matrix is used to calculate the probability of each possible node according to the reciprocal and the roulette wheel selection scheme. For example, in the multi-graph of Fig. 1, suppose that $v_i = v_s$ and $v_j = v_4$; then, the following node v_k may be chosen between v_1 and v_2 . From Eq. (9), we know that the distance from v_1 to v_4 (v_j) is 1 and the distance from v_2 to v_4 is 2. Therefore, the probabilities of selection for v_1 and v_2 are $2/3$ and $1/3$, respectively. Then, v_k can be decided. The edge between v_i and v_k is selected randomly.

- (2) Immigration strategy. We calculate the standard deviation (msd) of the current population. If $msd < c$ (c is a constant), we generate N new individuals based on the distance matrix ($v_j = v_t$) and then insert them into the current population according to the reciprocal and the roulette wheel selection scheme.
- (3) Backtracking. Backtracking greatly delays the running time of the algorithm for large-scale problems, so the initialization or repair of an individual will start again if the number of times of backtracking $bt > b$ (b is a constant).

4.4. INGA

The multi-graph increases the difficulty in searching for feasible routes and solving the model. Therefore, INGA is designed from the perspective of the simple route and the k-th shortest path algorithm [12]. INGA is divided into two stages. The first stage chooses nodes, and the second stage adds edges between them.

4.4.1. The first stage

First, the multi-graph is simplified into a simple graph. The simple graph has all nodes of the multi-graph. If two nodes are connected directly in the multi-graph, one edge without weight will be added between the two nodes in the simple graph.

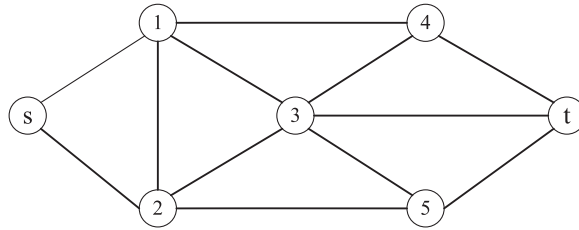


Fig. 2. Simple graph of the first stage for INGA.

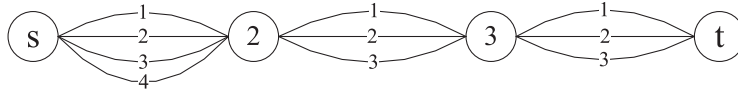


Fig. 3. Graph of the second stage for INGA.

Then, we choose routes in the simple graph based on GA without considering the constraints of the proposed model. Because weight is not considered in this stage, the distance matrix (introduced in IDGA) is used to generate the initial population.

For example, Fig. 2 is the corresponding simple graph of the multi-graph in Fig. 1, and $\{v_s, v_2, v_3, v_t\}$ is a route of the first stage. In fact, what we choose in the first stage is a set of nodes that can be connected into a route from the source to the destination because the edges on the route of this stage are virtual and have no weight. The edges between those nodes will be chosen in the second stage.

GA is used to search for individuals in the simple graph of the first stage in which each individual represents a set of nodes. Because the constraints are not considered, this process is simpler than the GA introduced in Section 4.2. Here, we briefly introduce the initialization and repair process of the first stage.

Initialization is done to find routes from the source to the destination in the simple graph. We begin from node v_s , and then, the following node will be selected based on the distance between each possible node and the destination v_t . Probabilities are calculated with the reciprocal of their distance. For example, starting from node v_s , v_1 and v_2 are possible nodes, and both of their distances to the destination are 2. Their probabilities can be calculated, and they are 0.5 and 0.5, respectively. Then, the roulette wheel selection scheme is used.

After the crossover operation and mutation operation in the first stage, we need to check the feasibility of each individual. If the individual is infeasible (disconnected), we try to repair it according to the following process. If there are repeated nodes on the individual, we keep the first one and delete others. We then check from the first node v_s until arriving at the breakpoint v_i . We examine whether v_i can be connected directly with other nodes behind v_i in the individual. If that happens, we connect them and repeat this process. Otherwise, we check from the last node v_t until arriving at the breakpoint v_j . We then generate a feasible route to connect v_i and v_j for the individual according to the distance matrix.

4.4.2. The second stage

Each individual from the first stage is a set of nodes that can be connected into a route from the source to the destination. At the beginning, the corresponding edges on the multi-graph between the nodes of each individual are found. The edges and nodes of each individual can form a simplified graph. Then, the constrained shortest path problem will be solved on the graph. For example, Fig. 3 is the graph of the individual $\{v_s, v_2, v_3, v_t\}$. The k -th shortest path algorithm is implemented according to the idea of DA (deletion algorithm) [1]. Traditional methods of the k -th shortest path problem can only be applied in a simple graph; Azevedo et al. [1] extended it to the multi-graphs. Detailed steps of the second stage are described as follows.

- Step 1. Compute the current shortest path p_k ($k \in \{1, 2, \dots, K\}$). Let $p_k = p_1$, that is, $k = 1$.
- Step 2. If $k = K$ or no candidate path is left, the second stage terminates. Otherwise, advance to Step 3.
- Step 3. Determine the first node v_h^m on p_k such that the in-degree is larger than 1. If v_h^m does not have an extended node, advance to Step 4. Otherwise, determine the first node v_i^l on p_k without extended nodes and advance to Step 5. v_h^m represents the m -th extension of v_h .
- Step 4. Add an extended node v_h^{m+1} for v_h^m in graph \bar{G} of the second stage. If v_h^m is not the last node of p_k , mark the next node of v_h^m on p_k as v_i^l . (v_{h-1}^l, v_h^m, s_j) represents an edge that begins at node v_{h-1}^l and ends at node v_h^m , and its number is s_j . (v_{h-1}^l, v_h^m, s_j) and $(v_{h-1}^l, v_h^{m+1}, s_j')$ have equal weights. The set of the incoming edges of v_h^{m+1} are:

$$I(v_h^{m+1}) \leftarrow \{(v_{h-1}^l, v_h^{m+1}, s_j') | (v_{h-1}^l, v_h^m, s_j) \in I(v_h^m), (v_{h-1}^l, v_h^m, s_j) \notin p_k\}.$$

- Step 5. Add an extended node for each node on p_k from v_i^l to the end $(v_j^m, j \in \{1, 2, \dots, n\})$. If v_{j-1}^l (the last node before v_j^m on p_k) has no extended nodes, the sets of the incoming edges are:

$$I(v_j^m) \leftarrow \{(v_{j-1}^l, v_j^m, s_q') | (v_{j-1}^l, v_j^{m-1}, s_q) \in I(v_j^{m-1}), (v_{j-1}^l, v_j^{m-1}, s_q) \notin p_k\}.$$

Table 1
Detailed information of nodes and edges.

Type	Number	Weight	Time	Type	Number	Weight	Time
Node	v_s	10	6	Edge	$v_2 \xrightarrow{2} v_3$	12	12
Edge	$v_s \xrightarrow{1} v_2$	18	16	Edge	$v_2 \xrightarrow{3} v_3$	11	13
Edge	$v_s \xrightarrow{2} v_2$	17	17	Node	v_3	9	7
Edge	$v_s \xrightarrow{3} v_2$	19	14	Edge	$v_3 \xrightarrow{1} v_t$	12	9
Edge	$v_s \xrightarrow{4} v_2$	15	20	Edge	$v_3 \xrightarrow{2} v_t$	13	8
Node	v_2	6	5	Edge	$v_3 \xrightarrow{3} v_t$	11	10
Edge	$v_2 \xrightarrow{1} v_3$	10	14	Node	v_t	7	5

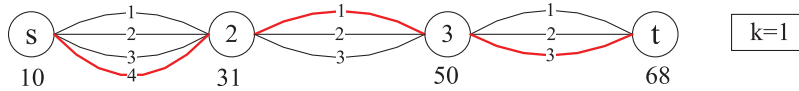


Fig. 4. The graph when $k = 1$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

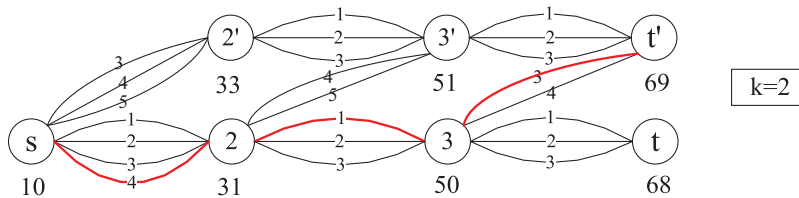


Fig. 5. The graph when $k = 2$. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

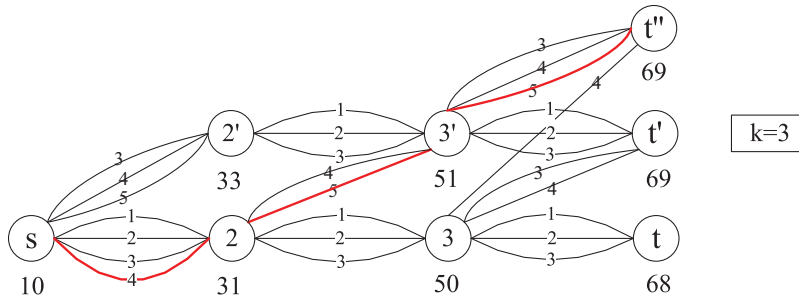


Fig. 6. The graph when $k = 3$.

If v_{j-1}^t has extended node v_{j-1}^{t+1} , the sets of the incoming edges are:

$$I(v_j^m) \leftarrow \{(v_{j-1}^l, v_j^m, s_q) | (v_{j-1}^l, v_j^{m-1}, s_q) \in I(v_j^{m-1}), (v_{j-1}^l, v_h^{m-1}, s_q) \notin p_k\} \cup \{(v_{j-1}^{t+1}, v_j^m, s_q) | (v_{j-1}, v_j, s_q) \in \bar{G}\}.$$

Step 6. Update the current shortest path p_k . $k = k + 1$. Return to Step 2.

Taking Fig. 3 as an example, assume that $K = 3$, $T_0 = 65$; detailed weight information (cost in the proposed model) of the nodes and edges is shown in Table 1 (randomly generated).

In Table 1, rows of “Node” represent the information of nodes. For example, the weight of v_s is 10. Rows of “Edge” represent the information of corresponding edges. For example, $v_s \xrightarrow{1} v_2$ represents the first edge from v_s to v_2 , and its weight is 18. Detailed steps of the second stage are as follows.

- (1) First, it is easy to find the current shortest path p_k , as red lines show in Fig. 4 when $k = 1$. The total weight of each node from v_s is also obtained. Now, $W_1 = 68$ (weight of the 1st shortest route), $T_1 = 67$, and $T_1 > T_0$.
- (2) According to p_k , the extended nodes and their incoming edges are added based on Fig. 4, which is shown in Fig. 5. The weights of these extended nodes are computed and the current shortest path p_k is found, as the red lines in Fig. 5 show, $k = 2$. Now, $W_2 = 69$, $T_2 = 66$, and $T_2 > T_0$.
- (3) Similar steps are performed to continue. $k = 3$ and $k = K$. $W_3 = 69$, $T_3 = 66$, and $T_3 > T_0$. The current shortest path p_3 is shown in Fig. 6. Therefore, when $K = 3$, the individual $\{v_s, v_2, v_3, v_t\}$ of the first stage is interpreted as an infeasible route. The second stage of this individual is finished.

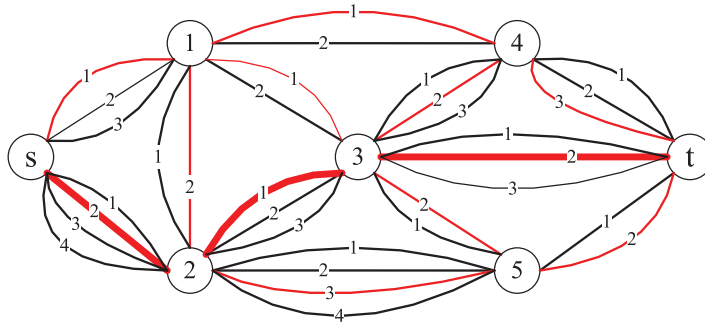


Fig. 7. Multi-graph of 4PLRP for SDGA. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article).

4.5. SDGA

According to the three previously proposed algorithms, feasible routes are selected directly or indirectly from the multi-graph. The three algorithms inevitably encounter the problem of multiple edges between two nodes. In this situation, one idea is to select a simple graph, and to solve the problem in the simple graph with classic methods. According to this idea, SDGA, which uses GA to select simple graphs and uses the constrained Dijkstra algorithm [11] to solve the proposed model in the simple graphs, is designed. Both the problems of infeasible routes and multiple edges in GA are avoided in SDGA.

In SDGA, every individual is a simple graph and the initial population is generated randomly. A legal individual is a simple graph that contains all nodes of the multi-graph and one edge between every two adjacent nodes. First, a linkage matrix (F_1 in Eq. (7)) of dimension $n \times n$ is also used to represent the multi-graph, as shown in Fig. 7, whose corresponding code is (342223432332). In Fig. 7, F_4 (Eq. (10)) represents an individual (a simple graph) that is constructed by the red lines. The number “1” in row one and column two indicates that the first edge between v_s and v_1 is chosen to construct the simple graph and that the upper bound of the number is 3 (only 3 edges in the corresponding position of F_1). The corresponding code of F_4 is (122111322232).

The advantage of this coding method is that it is easy to avoid illegal codes both in the crossover operation and mutation operation in GA. In the crossover operation, any two elements that may exchange position begin in the same node and end at the same node also. Therefore, new codes are also legal codes (i.e., they can form a simple graph). For example, consider parent codes 122|111322|232 and 242|121321|231 and assume that the crossover points are between 3 and 9. Then, the new codes are 122|121321|232 and 242|111322|231. In the mutation operation, the upper bound of each code is treated as the module; thus, a legal code will be ensured.

$$F_4 \begin{matrix} v_s & 1 & 2 & 3 & 4 & 5 & v_t \\ v_s & \begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 & 3 \\ 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \end{matrix} \tag{10}$$

Then, the problem of each individual can be considered as a constrained shortest path problem, and the constrained Dijkstra algorithm can be used to solve this problem. Due to limited space, details of the process will not be introduced here.

5. Numerical experiments

To investigate the performance of the proposed algorithms and verify the effectiveness of the proposed 4PLRPU model, three scales of examples are given in this section. Then, the performances of the proposed algorithms and the effect of some key parameters on the performance of the algorithms are analyzed. Furthermore, the robustness of the 4PLRPU model is illustrated by comparing with the expected value model of 4PLRP.

5.1. The 4PLRPU examples

As is introduced previously, the 4PLRPU can be represented by an undirected multi-graph. The multi-graphs are generated randomly in a rectangle to represent examples [7]. Detailed steps are shown as follows:

Step 1. Generate $n - 2$ nodes (a total of n nodes on the multi-graph including the source and the destination) randomly in a $d \times d$ square area. $(0, 0)$ and (d, d) represent the source and destination, respectively.

Table 2
Experimental results of the algorithms for the 7 nodes example.

Algorithm	Node	LCode	NP	NG	Pc	Pm	Best	Bad	Avg	Msd	RTime (s)
ENUM	7	12	–	–	–	–	68	–	–	–	24.78
GA	7	12	50	12	0.90	0.05	68	68	68	0.00	0.98
IDGA	7	12	50	11	0.90	0.05	68	68	68	0.00	0.27
INGA	7	12	50	6	0.90	0.05	68	68	68	0.00	0.24
SDGA	7	12	50	5	0.90	0.05	68	68	68	0.00	0.05

Table 3
Experimental results of the algorithms for the 15 nodes example.

Algorithm	Node	LCode	NP	NG	Pc	Pm	Best	Bad	Avg	Msd	RTime
ENUM	15	36	–	–	–	–	–	–	–	–	–
GA	15	36	200	30	0.90	0.05	107	107	107	0.00	5.62 s
IDGA	15	36	50	15	0.90	0.05	107	107	107	0.00	0.99 s
INGA	15	36	50	14	0.90	0.05	107	107	107	0.00	1.37 s
SDGA	15	36	50	20	0.90	0.05	107	107	107	0.00	2.41 s

Step 2. If the Euclidean distance between two nodes is less than or equal to D_s , edges exist between them.

Step 3. Generate a random number $r_{ij} \in [a_1, b_1]$ which represents the number of edges between the two nodes.

Step 4. Randomly generate cost C_{ijk} and time T_{ijk} for each edge, where $C_{ijk} \in [a_2, b_2]$ and $T_{ijk} \in [a_3, b_3]$.

Step 5. Randomly generate cost C'_j and time T'_j for each node, where $C'_j \in [a_4, b_4]$ and $T'_j \in [a_5, b_5]$.

To illustrate the ideas discussed above and compare more clearly, three examples with 7 nodes, 15 nodes and 30 nodes ($n = 7, n = 15, n = 30$) are generated. $d = 1$, and $D_s \in [0.5, 0.75]$. Cui et al. [25] consider two test examples in which one is a real logistics network and the other is generated randomly. Based on these, other parameters are set as follows:

$$a_1 = 2, \quad b_1 = 4, \quad a_2 = 6, \quad b_2 = 30, \quad a_3 = 2, \quad b_3 = 22, \quad a_4 = 5, \quad b_4 = 15, \quad a_5 = 4, \quad b_5 = 9.$$

The investigated algorithms are coded with Microsoft Visual Studio 2008 and run on an Intel Core-2 Duo 3.0 GHz PC.

5.2. Performance analysis of algorithms

ENUM serves as a basic standard to evaluate the other algorithms' performances. After setting an appropriate μ, σ, T_0 and α , three examples with different scales (7 nodes, 15 nodes and 30 nodes) are tested.

5.2.1. Numerical results of algorithms

Without loss of generality, supposing that $\mu = 0, \sigma = 1.20, \alpha = 0.95$, the experimental results of the 7 nodes example are shown in Table 2. "Algorithm" represents the proposed five algorithms ENUM, GA, IDGA, INGA and SDGA. "Node" represents the number of nodes and "LCode" represents the length of code. These two parameters reflect the scale of the problem. The algorithms are run 100 times, and the parameters considered in the algorithms are population size "NP", the number of generation "NG", crossover rate "Pc", mutation rate "Pm", the best solution "Best", the worst solution "Bad", the average value "Avg", standard deviation "Msd" in Eq. (11) and algorithm running time "RTime"(once). "-" indicates that the algorithm does not have this parameter.

$$\text{Msd}(S) = \sqrt{\frac{\sum_{i=1}^n (S_i - \bar{S})^2}{n-1}} \quad (11)$$

In Eq. (11), \bar{S} is the mean, S_i is the best solution in the i -th running, and $n = 100$.

Here, the due date T_0 is set to 73, which can be changed based on the customer's requirement. In INGA, the parameter k (i.e., the second stage, k -th shortest path algorithm) is set to 8. It can be observed from Table 2 that both GA and the three improved GAs can find the optimal solution quickly for small-scale problems (they obtain the same solution as the enumeration algorithm). For small-scale problems, the three improved GAs need less running time than GA.

Expanding the scale of the problem, a 15 nodes example is tested, and the experimental results are presented in Table 3. In spite of using the improved strategy, ENUM cannot determine the optimal solution within an acceptable time (more than 12 h). In this case, four GAs are compared with each other.

We can see from Table 3 that all of the proposed intelligent algorithms obtain solutions with the same value (they may choose different routes, but the cost is the same) within an acceptable running time. For medium-scale problems, the average running time of IDGA is the least under the premise of guaranteeing solution quality and the standard deviation, which illustrates the effectiveness of strategies in IDGA. Compared with GA, the three improved algorithms converge to the solution faster and need smaller population size and generation.

Table 4
Experimental results of the algorithms for the 30 nodes example.

Algorithm	Node	LCode	NP	NG	Pc	Pm	Best	Bad	Avg	Msd	RTime
ENUM	30	71	–	–	–	–	–	–	–	–	–
GA	30	71	2000	35	0.90	0.05	182	183	182.02	0.14	35 min 22 s
IDGA	30	71	300	40	0.90	0.05	182	182	182	0.00	182.70 s
INGA	30	71	200	20	0.95	0.05	182	182	182	0.00	15.31 s
SDGA	30	71	100	25	0.90	0.05	182	182	182	0.00	3.97 s

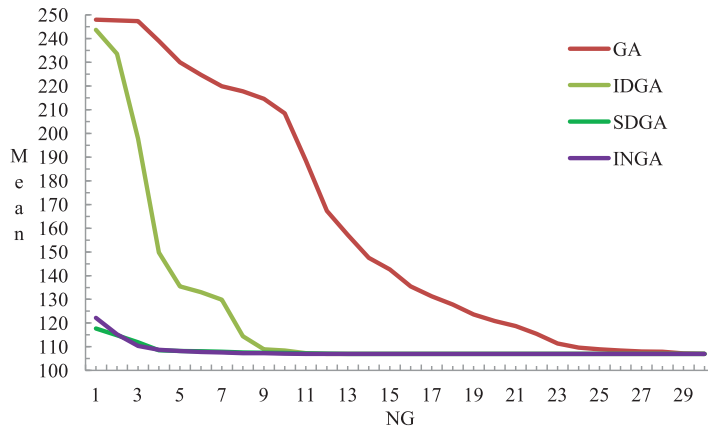


Fig. 8. Variation diagram of average cost for the 15 nodes example.

Then, the large-scale problem with 30 nodes and 198 edges is tested to investigate the performances of the proposed algorithms. In this case, each code contains 71 genes, which poses a challenge for general algorithms. The experimental results are shown in Table 4.

GA can get the same best solution as the other improved algorithms but needs more running time. The proposed IDGA greatly reduces the running time, and its standard deviation is smaller than the GA's. For large-scale problems, the improvement is obvious. However, the running time of IDGA is also more than 3 min, which is not efficient enough. From Table 4, we can see that SDGA and INGA search faster than GA and IDGA and that SDGA's running time is smaller than INGA's. In addition to reducing the running time of repairing infeasible solutions, INGA and SDGA also simplify the complexity of the multi-graph, which improves the efficiency of the algorithms.

5.2.2. Analysis of algorithm convergence

To see the algorithms' search processes more clearly, the mean value and good rate of algorithms are tested along with the parameter of population generation (NG). Then, how the algorithms converge to the best solution can be shown.

$$\text{Mean}(c) = \frac{1}{n} \sum_{i=1}^n c_i \tag{12}$$

$$\text{Rate}(c) = \frac{m}{n} \tag{13}$$

c_i is the cost of the i -th individual of the current generation, n is the population size, and m is the number of individuals whose cost is equal to the best solution.

Fig. 8 shows the variation diagram of average cost changes with NG for the 15 nodes example. To make a clearer comparison, the population size (NP) is set to 200 for all algorithms. It can be seen that the convergence processes of INGA and SDGA are quite similar and that they converge faster than IDGA and GA. Because the Dijkstra algorithm and k -th shortest path algorithm are used, respectively, SDGA and INGA have obvious advantages in the initial solutions. It is obvious that SDGA and INGA converge faster than IDGA, and they all converge faster than GA.

Fig. 9 shows the variation diagram of the 30 nodes example. "1" is the detailed figure of part "2". Because of difficulties in solving large-scale problems, especially for GA and IDGA, the population size (NP) is no longer equal. The NP of GA, IDGA, INGA and SDGA are set to 2000, 300, 200 and 100, respectively. Generally, the three improvement strategies of IDGA, INGA and SDGA are efficient, and INGA and SDGA are more obvious. INGA reduces the complexity of multiple edges between neighboring nodes, and SDGA solves the problem of illegal coding that often appears in GA and its improved algorithms. It can be observed in Fig. 9 that the improved algorithms significantly accelerate the search process. The three improved GAs' NP are much smaller than that of GA; however, they can also converge to the best solution with less generations.

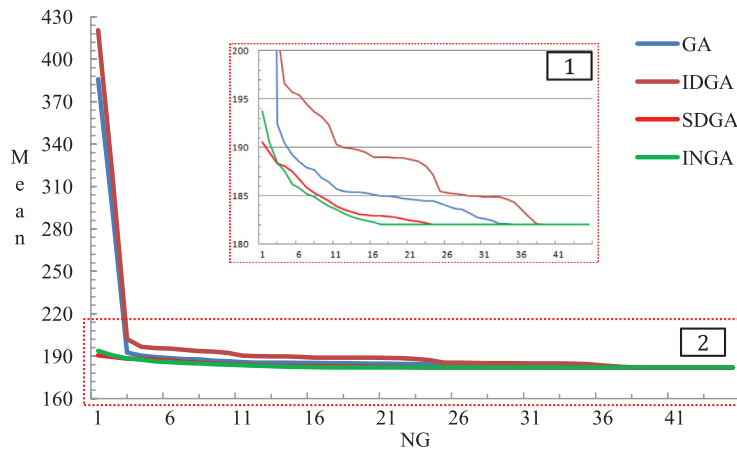


Fig. 9. Variation diagram of average cost for the 30 nodes example.

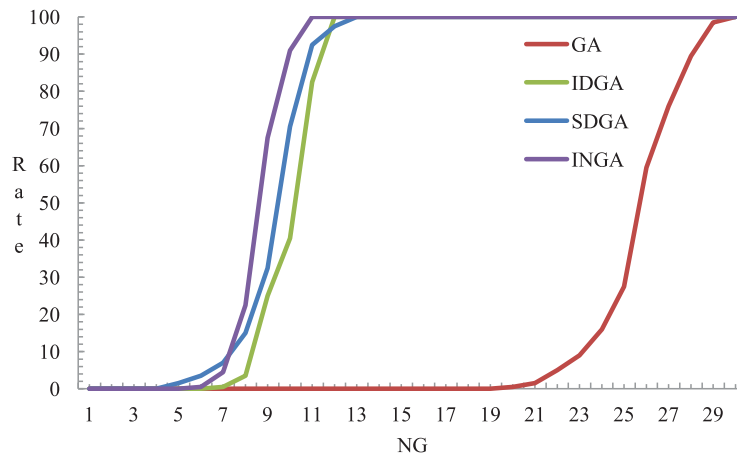


Fig. 10. Variation diagram of good rates for the 15 nodes example.

Figs. 10 and 11 are good rate variation diagrams, along with NG, for the 15 nodes example and the 30 nodes example. As introduced above, NP is set to 200 in Fig. 10, and the NP of GA, IDGA, INGA and SDGA are set to 2000, 300, 200 and 100, respectively, in Fig. 11. We can also see that the three improved algorithms show higher efficiency than GA. With the increase in the problem scale from 7 nodes to 30 nodes, INGA and SDGA show advantages gradually. Generally, SDGA always needs a smaller NP, and its initial solutions are better than those of INGA. However, INGA needs fewer generations to converge to the best solution. SDGA and INGA perform better than the other proposed algorithms when solving large-scale problems for the proposed 4PLRPU.

5.2.3. The effects of some key problem parameters on the algorithms

Some key parameters also impact the algorithms' efficiency, such as the due date T_0 from the customer, the problem scale represented by the number of nodes in the proposed model, the transportation environment represented by μ and σ , and the confidence level α asked by the customer. In this section, the effect of T_0 is tested for the 15 nodes example first, and then, the four GAs are compared to solve a larger scale problem with 100 nodes. The experimental results are shown in Fig. 12 and Table 5.

It can be observed from Fig. 12 that T_0 obviously impacts SDGA. As T_0 increases, the running time of SDGA decreases quickly. The constrained Dijkstra algorithm is easily affected by the tightness of the constraints. However, once T_0 increases, the running time of INGA and IDGA will also decrease, but not very clearly. In particular, the influence on INGA is even more obvious because the idea of the k-th shortest path is used. It is similar to SDGA to some extent, and a more strict constraint often leads to a larger K.

Table 5 shows the experimental results of the 100 nodes example, where $T_0 = 1000$ and $dev = 1$. It can be seen that the length of code "LCode" for the four algorithms is a very large number and that it is very hard to search for the optimal solution. Here, the average value of the objective function is used to show the performances of the algorithms. In this condition, GA and IDGA are not very effective. INGA can help us solve large-scale problems but is less effective than SDGA. SDGA's solution has the smallest "Avg" and uses the least running time.

Overall, the proposed GA and the three improved GAs can find optimal solutions accurately and quickly and are all effective for solving the small-scale problems of 4PLRPU. When the problem scale increases, IDGA's improvement strategies play a role and IDGA has a better performance than GA. INGA and SDGA have obvious advantages in the initial population, especially for

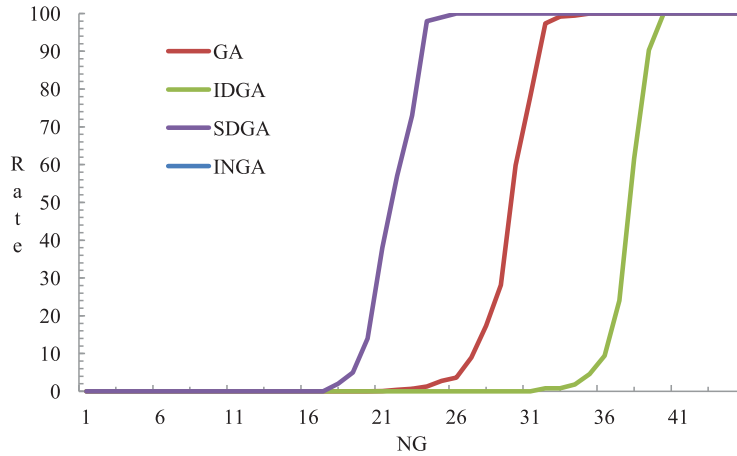


Fig. 11. Variation diagram of good rates for the 30 nodes example.

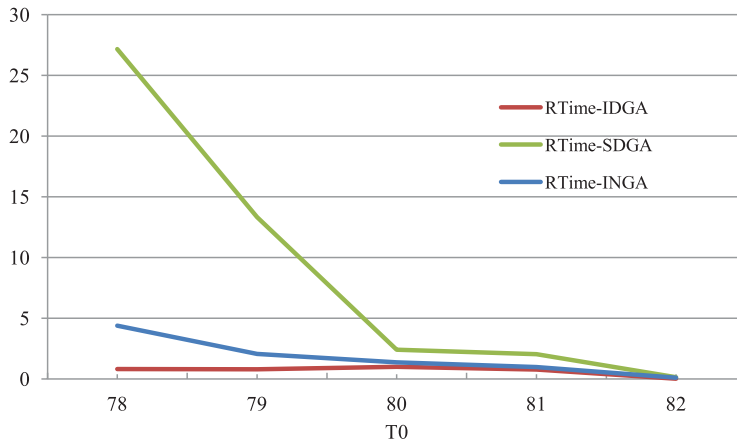


Fig. 12. The effect of parameter T_0 .

Table 5
Experimental results of the algorithms for the 100 nodes example.

Algorithm	Node	LCode	NP	NG	Pc	Pm	Best	Avg	RTime
GA	100	11020	5000	500	0.95	0.15	609	612.4532	75 min 36 s
IDGA	100	11020	3000	500	0.95	0.15	584	584.7463	43 min 52 s
INGA	100	11020	3000	250	0.95	0.10	578	578.2156	136.37 s
SDGA	100	11020	3000	100	0.95	0.15	575	575.4770	31.43 s

large-scale problems. The running times of INGA and SDGA are always smaller than those of the other algorithms, and at the same time, they ensure the solution quality. INGA simplifies the complexity of multiple edges in multi-graph and limits the problem of illegal routes, and SDGA avoids the problem of illegal routes. The improvements are significant.

5.3. Robustness analysis of the 4PLRPU model

Many existing studies on 4PL established deterministic models considering the delivery time as a deterministic value [23,24,38]. These models can be regarded as expected value models under an uncertain environment. To illustrate the effectiveness of the proposed 4PLRPU model under an uncertain environment, the corresponding expected value model of 4PLRP, called 4PLRPE, is developed by replacing Eq. (2) with Eq. (14). Furthermore, the three examples with different scales are tested, and the results of both models are compared in this section.

$$E \left[\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{r_{ij}} (T_{ijk} + t_{ijk})x_{ijk}(R) + \sum_{j=1}^n T'_j y_j(R) \right] \leq T_0 \tag{14}$$

Table 6
Comparison of the 4PLRPE and 4PLRPU model

Model	Algorithm	Node	T_0	α	σ	Best	Robustness
4PLRPE	SDGA	7	66	–	1.2	69	0.5007
4PLRPU	SDGA	7	66	0.95	1.2	73	0.9538
4PLRPE	SDGA	7	66	–	2.0	69	0.4955
4PLRPU	SDGA	7	66	0.95	2.0	80	0.9514
4PLRPE	SDGA	7	70	–	1.2	68	0.8199
4PLRPU	SDGA	7	70	0.95	1.2	70	0.9546
4PLRPE	SDGA	7	70	–	2.0	68	0.7160
4PLRPU	SDGA	7	70	0.95	2.0	73	0.9519
4PLRPE	SDGA	15	75	–	1.2	103	0.6013
4PLRPU	SDGA	15	75	0.95	1.2	110	0.9550
4PLRPE	SDGA	15	75	–	2.0	103	0.5586
4PLRPU	SDGA	15	75	0.95	2.0	114	0.9520
4PLRPE	SDGA	15	80	–	1.2	100	0.6784
4PLRPU	SDGA	15	80	0.95	1.2	107	0.9787
4PLRPE	SDGA	15	80	–	2.0	100	0.6112
4PLRPU	SDGA	15	80	0.95	2.0	110	0.9510
4PLRPE	SDGA	30	140	–	1.2	177	0.6895
4PLRPU	SDGA	30	140	0.95	1.2	186	0.9548
4PLRPE	SDGA	30	140	–	2.0	177	0.6176
4PLRPU	SDGA	30	140	0.95	2.0	196	0.9537
4PLRPE	SDGA	30	143	–	1.2	174	0.6404
4PLRPU	SDGA	30	143	0.95	1.2	183	0.9553
4PLRPE	SDGA	30	143	–	2.0	174	0.5886
4PLRPU	SDGA	30	143	0.95	2.0	191	0.9492

To more clearly analyze the effectiveness of the 4PLRPU model under an uncertain environment, the robustness of solution “Robustness” is defined as shown in Eq. (15).

$$Robustness = \frac{N_f}{N_s} \tag{15}$$

Given a solution (a route R) of the problem, its distribution function of the total delivery time can be easily obtained, and the robustness is calculated via Monte Carlo Simulation according to the distribution function. In Eq. (15), N_s is the number of samples generated according to the distribution function of route R . N_f is the number of feasible samples that satisfy the due date T_0 required by the customer.

The experimental results are presented in Table 6. It can be seen that the 4PLRPU model considers the tardiness risk under an uncertain environment and improves the robustness of solutions. With the same T_0 , the 4PLRPU model chooses routes with higher robustness and higher cost to guarantee the predetermined confidence level α . However, although the 4PLRPE chooses routes with lower costs, its robustness is far lower than that of the 4PLRPU model. The 4PLRPE only considers expectations and has a higher tardiness risk. From Table 6, we can also see that, with the increase in σ , the robustness is reduced and the 4PLRPE is influenced obviously by σ , whereas the 4PLRPU model is not. Taking 7 nodes as an example, when T_0 is set to 66 and $\sigma = 1.2$, the 4PLRPE chooses the route $R_1 = (v_s \xrightarrow{4} v_2 \xrightarrow{3} v_3 \xrightarrow{3} v_t)$. R_1 has a lower total cost; however, the robustness is only 0.5007. If the confidence level $\alpha = 0.95$ is considered, R_1 cannot meet the customer’s demand. The route that the 4PLRPU model selects is $R_2 = (v_s \xrightarrow{3} v_2 \xrightarrow{3} v_3 \xrightarrow{3} v_t)$. Although the cost of R_2 is slightly higher than that of R_1 , the robustness of R_2 is 0.9538. Overall, compared with the expected value model, the 4PLRPU model can obtain higher robustness with little cost increase.

6. Conclusion

Because of the lack of historical data regarding delivery time in uncertain 4PL environments, routing decisions are usually made in the state of indeterminacy. To address this type of uncertainty, the delivery time is described as an uncertain variable, and a novel 4PLRPU uncertain programming model is developed based on uncertainty theory.

To solve the proposed 4PLRPU model, it is first transformed into an equivalent deterministic model, and then, IDGA, INGA and SDGA are designed according to the characteristics of the problem. Numerical analysis shows the effectiveness of the algorithms and the model. For small-scale problems, GA, IDGA, INGA and SDGA obtain the optimal solutions quickly, which are calculated by the ENUM. For medium-scale problems, IDGA, INGA and SDGA show the advantages in searching speed compared with GA, and all of them can obtain the same best solutions. Further expanding the scale of the problem, compared with GA and IDGA, INGA gradually shows its obvious advantages in convergence and efficiency, because INGA’s computing time required to repair infeasible solutions is greatly reduced by decreasing the complexity of multiple edges. Additionally, compared with GA and IDGA, SDGA shows obvious advantages in effectiveness and efficiency due to avoiding infeasible solutions. Finally, compared with the

expected value model of 4PLRP, the results indicate that the 4PLRPU model is more robust. In general, the proposed 4PLRPU model, as well as INGA and SDGA, provides a useful routing decision methodology for 4PL under uncertain environments.

In the future, the work in this paper can be extended by considering more uncertain factors or more tasks.

Acknowledgment

This work is supported by the National Science Foundation for Distinguished Young Scholars of China under Grant nos. 71325002 and 61225012; the National Natural Science Foundation of China under Grant no. 71071028; the Fundamental Research Funds for State Key Laboratory of Synthetical Automation for Process Industries under Grant no. 2013ZCX11.

References

- [1] J.A. Azevedo, M.E.O.S. Costa, J.J.E.R. Madeira, E.Q.V.M. Martins, An algorithm for the ranking of shortest paths, *Eur. J. Oper. Res.* 69 (1) (1993) 97–106.
- [2] T. Bäck, P. Schwefel, An overview of evolutionary algorithms for parameter optimization, *Evol. Comput.* 1 (1) (1993) 1–23.
- [3] D.J. Bade, J.K. Mueller, New for the millennium 4PL, *Transp. Distrib.* 40 (2) (1999) 78–80.
- [4] N. Bolanda, J. Dethridge, I. Dumitrescu, Accelerated label setting algorithms for the elementary resource constrained shortest path problem, *Oper. Res. Lett.* 34 (1) (2006) 58–68.
- [5] J. Branke, Evolutionary algorithms for dynamic optimization problems: a survey. Technical report 387, Institute AIFB, University of Karlsruhe, 1999.
- [6] G. Büyükköçkan, O.F. glu, M.S. Ersoy, Evaluation of 4PL operating models: a decision making approach based on 2-additive choquet integral, *Int J. Prod. Econ.* 121 (1) (2009) 112–120.
- [7] X.J. Chen, J. Hu, X.D. Hu, A new model for path planning with interval data, *Comput. Oper. Res.* 36 (6) (2009) 1893–1899.
- [8] K.H. Chen, C.T. Su, Activity assigning of fourth party logistics by particle swarm optimization-based preemptive fuzzy integer goal programming, *Expert Syst. Appl.* 37 (5) (2010) 3630–3637.
- [9] X.W. Chen, J.W. Gao, Uncertain term structure model of interest rate, *Soft Comput.* 17 (4) (2013) 597–604.
- [10] Y. Cui, M. Huang, S.X. Yang, L.H. Lee, X.W. Wang, Fourth party logistics routing problem model with fuzzy duration time and cost discount, *Knowl.-Based Syst.* 50 (2013) 14–24.
- [11] E.W. Dijkstra, A note on two problems in connexion with graphs, *Numer. Math.* 1 (1) (1959) 269–271.
- [12] D. Eppstein, Finding the k shortest paths, *SIAM J. Comput.* 28 (2) (1998) 652–673.
- [13] T. Foster, 4PLs: the next generation for supply chain outsourcing, *Logist. Manag. Distrib. Rep.* 38 (4) (1999) 35.
- [14] J.W. Gao, Uncertain bimatrix game with applications, *Fuzzy Optim. Decision Mak.* 12 (1) (2013) 65–78.
- [15] Y. Gao, Shortest path problem with uncertain arc lengths, *Comput. Math. Appl.* 62 (6) (2011) 2591–2600.
- [16] Y. Gao, L.X. Yang, S.K. Li, S. Kar, On distribution function of the diameter in uncertain graph, *Inf. Sci.* 296 (2015) 61–74.
- [17] J. Gattorna, Strategic Supply Chain Alignment: Best Practice in Supply Chain Management, Gower Publishing Company, 1998.
- [18] F. Glover, E. Taillard, D.D. Werra, A user's guide to tabu search, *Ann. Oper. Res.* 41 (1–4) (1993) 3–28.
- [19] C. Guo, J.W. Gao, Optimal dealer pricing under transaction uncertainty, *J. Intell. Manuf.* (2015), doi:10.1007/s10845-014-1002-8.
- [20] S.W. Han, Z.X. Peng, S.Q. Wang, The maximum flow problem of uncertain network, *Inf. Sci.* 265 (5) (2014) 167–175.
- [21] G.Y. Handler, I. Zang, A dual algorithm for the constrained shortest path problem, *Networks* 10 (4) (1980) 293–309.
- [22] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan press, Ann Arbor, 1975.
- [23] M. Huang, W. Tong, Q. Wang, X. Xu, X.W. Wang, Immune algorithm based routing optimization in fourth-party logistics, in: *Proceedings of IEEE Congress on Evolutionary Computation*, Vancouver, Canada, 2006, pp. 3029–3034.
- [24] M. Huang, G.H. Bo, X.W. Wang, W.H. Ip, The optimization of routing in fourth-party logistics with soft time windows using harmony search, in: *Sixth International Conference on Natural Computation*, Yantai, China, 2010, pp. 4344–4348.
- [25] M. Huang, Y. Cui, S.X. Yang, X.W. Wang, Fourth party logistics routing problem with fuzzy duration time, *Int. J. Prod. Econ.* 145 (1) (2013a) 107–116.
- [26] M. Huang, J. Tu, C.H. Xu, Delivery time contract design in outsourcing logistics: a fourth party logistics perspective, *Inf. - Int. Interdiscip. J.* 16 (7) (2013b) 4577–4594.
- [27] M. Huang, L. Ren, L.H. Lee, X.W. Wang, 4PL routing optimization under emergency conditions, *Knowl.-Based Syst.* (2015), doi:10.1016/j.knosys.2015.06.023.
- [28] D.Y. Jiao, K. Yao, An interest rate model in uncertain environment, *Soft Comput.* 19 (3) (2015) 775–780.
- [29] A.N. Kolmogorov, *Grundbegriffe der Wahrscheinlichkeitsrechnung*, Julius Springer, Berlin, 1933.
- [30] F. Krakovics, J.L. Eugenio, P. Mendes Jr, R.L. Santos, Defining and calibrating performance indicators of a 4PL in the chemical industry in Brazil, *Int. J. Prod. Econ.* 115 (2) (2008) 502–514.
- [31] F.C. Li, Y. Wang, P. Hang, Analysis of outsourcing pricing for fourth party logistics enterprise based on bilateral bargaining game with asymmetric information, in: *Proceedings of IEEE International Conference on Service Systems and Service Management*, Tokyo, Japan, 2010, pp. 1–6.
- [32] G.H. Li, M. Huang, The optimization model and its algorithms for multiple transportation modes selection in the fourth party logistics, *Adv. Mater. Res.* 383 (2011) 2249–2255.
- [33] B.D. Liu, *Uncertainty theory*, second ed., Springer-Verlag, Berlin, 2007.
- [34] B.D. Liu, *Theory and practice of uncertain programming*, second ed., Springer-Verlag, Berlin, 2009.
- [35] B.D. Liu, *Uncertainty theory: a branch of mathematics for modeling human uncertainty*, Springer-Verlag, Berlin, 2010.
- [36] B.D. Liu, *Uncertainty theory*, fourth ed., Springer-Verlag, Berlin, 2013.
- [37] L.Z. Liu, H.B. Mu, X.F. Yang, R.C. He, Y.Z. Li, An oriented spanning tree based genetic algorithm for multi-criteria shortest path problems, *Appl. Soft Comput.* 12 (1) (2012) 506–515.
- [38] Q. Liu, C.Y. Zhang, K.R. Zhu, Y.Q. Rao, Novel multi-objective resource allocation and activity scheduling for fourth party logistics, *Comput. Oper. Res.* 44 (2014) 42–51.
- [39] C. Mohamed, J. Bassem, L. Taicir, A genetic algorithms to solve the bicriteria shortest path problem, *Electron. Notes Discret. Math.* 36 (1) (2010) 851–858.
- [40] D.Y. Mou, W.L. Zhao, X.D. Chang, A transportation problem with uncertain truck times and unit costs, *Ind. Eng. Manag. Syst.* 12 (1) (2013) 30–35.
- [41] S. Oreski, G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment, *Expert Syst. Appl.* 41 (4) (2014) 2052–2064.
- [42] H.C. Tsai, Integrating the artificial bee colony and bees algorithm to face constrained optimization problems, *Inf. Sci.* 258 (3) (2014) 80–93.
- [43] A. Tversky, D. Kahneman, Advances in prospect theory: cumulative representation of uncertainty, *J. Risk Uncertain.* 5 (4) (1992) 297–323.
- [44] A. Win, The value a 4PL provider can contribute to an organisation, *Int. J. Phys. Distrib. Logist. Manag.* 38 (9) (2008) 674–684.
- [45] J.M. Yao, Decision optimization analysis on supply chain resource integration in fourth party logistics, *J. Manuf. Syst.* 29 (4) (2010) 121–129.
- [46] K. Yao, J.W. Gao, Uncertain random alternating renewal process with application to interval availability, *IEEE Trans. Fuzzy Syst.* (2015), doi:10.1109/TFUZZ.2014.2360551.
- [47] K. Yao, X. Li, Uncertain alternating renewal process and its application, *IEEE Trans. Fuzzy Syst.* 20 (6) (2012) 1154–1160.
- [48] X.F. Yang, J.W. Gao, Uncertain differential game with application to capitalism, *J. Uncertain. Anal. Appl.* 1 (17) (2013) 1–11.
- [49] H. Zhang, X. Li, W.H. Liu, B. Li, An application of the AHP in 3PL vendor selection of a 4PL system, in: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Hague, Netherlands, 2004, pp. 1255–1260.