Train Timetable Problem on a Single-Line Railway with Fuzzy Passenger Demand

Lixing Yang, Keping Li, and Ziyu Gao

Abstract—The aim of train timetable problem is to determine arrival and departure times at each station so that no collisions will happen between different trains and the resources can be utilized effectively. Due to uncertainty of real systems, train timetables have to be made under the uncertain environment in most circumstances. This paper mainly investigates a passenger train timetable problem with fuzzy passenger demand on a single-line railway, in which two objectives, i.e., fuzzy total passengers’ time and total delay time, are considered. As a result, an expected value goal programming model is constructed for the problem. A branch and bound algorithm based on the fuzzy simulation is designed in order to obtain an optimal solution. At last, some numerical experiments are given to show applications of the model and the algorithm.

Index Terms—Train Timetable Problem; Fuzzy Variable; Simulation; Goal Programming; Branch and Bound Algorithm

I. INTRODUCTION

Train timetable problem involves how to determine the plan of train arrival and departure times at the important points, such as crossings, stations and yards, such that conflicts between different trains will not occur and the transportation efficiency is optimized. Because the physical railway network is shared by a large number of trains, train timetable problem is one of the most challenging and difficult problems in railway planning, which has also been drawing the attention of researchers for decades. Generally, there are two methods for making a timetable, i.e., manual way and computer-based way. Manual method has been used more than 150 years on almost all railways. And it is still being used on some railways nowadays. Due to complexity of this work, by manual way, operators can only obtain a feasible train timetable without optimization of resources. With the development of computer technique in 1990s, it is possible for operators to make an optimal train timetable with the help of computers. Up to now, computer-based way has been playing an important role in making optimal train timetables.

In literature, many researchers have investigated this problem and developed a lot of scheduling techniques. Roughly speaking, the available scheduling techniques can be divided into three classes, i.e., simulation, expert systems and mathematical programming. For the development of simulation and expert systems, we may refer to Frank[7], Rudd & Storey[29], Petersen & Taylor[28], Iida[14], Komaya & Fukuda[16], Komaya[17].

The most popular technique to make train schedules is to use mathematical programming, although it may take probably long time to get an approximately optimal schedule. Mathematical programming method was initialized by Amit & Goldfarb[2] in 1971. Up to now, a good many models and algorithms have been presented by researchers, for instance, an SCAN system by Jovanovic & Harker[15], heuristic algorithms by Kraay, Harker & Chen[18], Carey & Lockwood[4], Kraay & Harker[19], Higgins, Kozen & Ferrera[12], [13], Sahin[30], branch and bound algorithms by Higgins, Kozen & Ferrera[11], Chierici, Cordone & Maja[6], Zhou & Zhong[33], a lagrangian relaxation approach by Brannlund et.al[3], a lagrangian based heuristic procedure by Castelli, Pesenti & Ukovich[5], a backtracking algorithm by Adenso-Diaz et.al[1], a multi-objective programming by Ghoseiri et.al[9].

In literature, most of works are mainly done for three kinds of railways, i.e., single-line railway, multi-line railway and railway network. In China, there exist a lot of single-line tracks due to the limitation of physical conditions, for instance, the Qinghai-Tibet railway. Over the single-line railways, it is stipulated that trains overtake and cross each other only at the stations. When conflicts occur, train dispatchers will resolve them on a time-distance graph using experience and knowledge of prevailing conditions. Thus, how to make train timetables scientifically is still an important issue today.

In this paper, we shall investigate passenger train timetable problem over a single-line railroad. As we know, due to uncertainty of the real systems, we always meet uncertain factors in the process of decision-making. As for the passenger train timetable problem, the number of passengers getting on/off the train at each station is actually uncertain in different periods. So it is more suitable to treat it as an uncertain variable than a fixed quantity. If available sample data are enough, it is natural...
to treat the number of passengers as a random variable by statistical methods. However, if the data are scare, a common way is to treat it as a fuzzy variable according to the experience or expert’s evaluation. In this paper, the number of passengers getting on/off the train at each station is assumed to be a fuzzy variable. This is a new idea that has not been seen in literature so far.

In this problem, we shall minimize two objectives, that is, the total passengers’ time and the total delay time. Thus the problem discussed in this paper can be formulated as a multi-objective programming problem. In order to obtain a Pareto solution of this problem, we shall treat the mathematical model as a goal programming model, in which the expected value of total passengers’ time with penalty function and the total delay time should not exceed some predetermined targets. This is also a new idea for making an optimal train timetable.

The rest of this paper is organized as follows. Section II introduces some definitions and theorems in credibility theory. Then in Section III, an expected value goal programming model is constructed for the train timetable problem. In order to solve the model, a branch and bound algorithm based on the fuzzy simulation is designed to seek the approximate optimal solution in Section IV. At last, some numerical experiments are performed to show applications of the model and the algorithm.

II. Preliminaries

This paper investigates the train timetable problem with fuzzy parameters. In the following, we shall briefly introduce some basic knowledge of credibility theory, which is a branch of mathematics studying the behavior of fuzzy phenomena.

Credibility measure was presented by Liu and Liu [22] in 2002, which is defined as the average of possibility measure and necessity measure. Credibility theory was founded by Liu [24] in 2004, and it was developed on the basis of credibility measure and axiomatic foundations. In 2006, Li and Liu [21] obtained a sufficient and necessary condition for credibility measure. That is, a set function Cr on the power set Θ of a nonempty set Θ is a credibility measure if and only if it satisfies the following four conditions:

(i) cr(∅) = 1;
(ii) cr(A) ≤ cr(B) whenever A ⊆ B;
(iii) cr(A) + cr(Aᶜ) = 1 for any A ∈ Θ;
(iv) cr(∪ₐ Aₐ) = supₐ cr(Aₐ) for any events {Aₐ} with supₐ cr(Aₐ) < 0.5.

Definition 2.1: [24] Let Θ be a non-empty set, Θ the power set of Θ and cr a credibility measure. Then the triplet (Θ, Θ, cr) is called a credibility space.

A fuzzy variable is defined as a function ξ from a credibility space (Θ, Θ, cr) to the set of real numbers to describe fuzzy phenomena. For any fuzzy variable ξ, we can deduce its membership function µξ(x) by credibility measure via the following way:

\[ µ_ξ(x) = (2cr(ξ = x)) ∧ 1. \]

For any set B of real numbers, the credibility measure of fuzzy event {ξ ∈ B} can be computed by the following formula:

\[ cr(ξ ∈ B) = \frac{1}{2} \left( \sup_{x ∈ B} µ_ξ(x) + 1 - \sup_{x ∈ B} µ_ξ(x) \right). \]

Liu & Liu [22] presented a definition of expected value operator on the basis of credibility measure to measure the mean value of a fuzzy variable.

Definition 2.2: [22] Let ξ be a fuzzy variable. Then the expected value of ξ is defined as

\[ E[ξ] = \int_0^{+∞} cr[ξ ≥ r]dr - \int_{-∞}^0 cr[ξ ≤ r]dr \]

provided at least one of the two integrals is finite.

Example 2.1: Let ξ be a fuzzy variable with the following membership function

\[ µ_ξ(x) = \begin{cases} 1, & \text{if } x ∈ [a, b] \\ 0, & \text{otherwise}. \end{cases} \]

Then the expected value of ξ is

\[ E[ξ] = \frac{a + b}{2}. \]

Example 2.2: Let ξ be a trapezoidal fuzzy variable with the following membership function

\[ µ_ξ(x) = \begin{cases} \frac{x - a}{b - a}, & \text{if } a ≤ x ≤ b \\ 1, & \text{if } b ≤ x ≤ c \\ \frac{x - d}{c - d}, & \text{if } c ≤ x ≤ d \\ 0, & \text{otherwise}. \end{cases} \]

Then the expected value of ξ is

\[ E[ξ] = \frac{a + b + c + d}{4}. \]

To rank fuzzy variables, expected values can be used as the representation values of different fuzzy variables. In this ranking criterion, the smaller the expected value is, the smaller the corresponding fuzzy variable is. Additionally, expected value operator is also linear. If a and b are real numbers, and ξ and η are independent fuzzy variables, then we have \( E[aξ + bη] = aE[ξ] + bE[η] \). For more details of expected value operator, we may refer to Liu[23], [24], [25].

Recently, many researchers further developed credibility theory and its applications. For instance, Li and Liu [20] presented a new definition of entropy of fuzzy variable on the basis of credibility measure; Liu [20] gave a survey of entropy of fuzzy variables; Gao [8] applied credibility theory to game theory; Liu & Zhu [27] explored some inequalities between moments of credibility distributions;
Wang, Tang & Zhao[31] investigated continuity and convexity of the expected value function of a fuzzy mapping; Yang & Liu[32] investigated a solid transportation problem under the fuzzy environment by using credibility measure.

Based on the credibility theory and uncertain programming, we shall investigate the train timetable problem on a single-line railway in the following sections.

III. Mathematical Model

A single-line railroad is a directed way on which the movement of trains is restricted in one-dimension. As shown in Figure 1, there are many stations located along the railroad, which are numbered consecutively with the index values 1, 2, ···, n along the outbound direction. The links between the different stations are also numbered consecutively with the index numbers 1, 2, ···, n − 1. In the railway system, the main role of each link is for trains’ traversing, and the other operations of trains, such as meeting and passing, meeting and overtaking, loading/unloading passengers, are carried out at the stations. We suppose that each station has two or more tracks (main track + side tracks) for the convenience of trains’ operations.

![Figure 1. Single-Line Railway](image)

In the following, we construct a mathematical model for train timetable problem on the single-line railway.

A. Parameters and Decision Variables

1. Parameters

- $E$: set of outbound trains;
- $W$: set of inbound trains;
- $N_k$: the number of sub-journeys of train $k$ ($k \in E \cup W$);
- $k_n$: the $n$th sub-journey of train $k$ ($k \in E \cup W, n = 1, 2, ···, N_k$);
- $S$: set of all stations along the railway;
- $S_k$: set of all stations on the path of train $k$;
- $S_k^i$: journey origin station of train $k$ ($k \in E \cup W$);
- $S_k^f$: journey destination station of train $k$ ($k \in E \cup W$);
- $L$: set of all links on the railway;
- $L_k$: set of all links on the path of train $k$ ($k \in E \cup W, L_k \subset L$);
- $P_k$: number of passengers in train $k$ when arriving at the initial station of sub-journey $k_n$, which is treated as a fuzzy variable;
- $a_k$: number of passengers getting off train $k$ at the initial station of sub-journey $k_n$, which is treated as a fuzzy variable;
- $b_k$: number of passengers getting on train $k$ at the initial station of sub-journey $k_n$, which is treated as a fuzzy variable;
- $t_{kn}$: time at which sub-journey $k_n$ starts;
- $t_{ak}$: required stopping time for allowing passengers to get off the train at the initial station of sub-journey $k_n$;
- $t_{bk}$: required stopping time for allowing passengers to get on the train at the initial station of sub-journey $k_n$;
- $t_{kl}$: minimum trip time on link $l$ for train $k$ ($k \in E \cup W, l \in L_k$);
- $s_{kl}$: minimum dwell time at station $t$ for train $k$;
- $d_k^*$: predetermined preferred departure time of train $k$;
- $h_l$: minimum headway between two trains on link $l \in L$;
- $M$: large arbitrary constant;
- $H$: time horizon (for instance, 24 hours).

2. Decision Variables

- $A_{kl} = \begin{cases} 1, & \text{if train } i \text{ traverses through link } l \text{ after train } k \ (i \in E, k \in W, l \in L_i \cap L_k) \\ 0, & \text{otherwise} \end{cases}$
- $B_{kl} = \begin{cases} 1, & \text{if train } i \text{ traverses through link } l \text{ after train } k \ (i, k \in E, l \in L_i \cap L_k) \\ 0, & \text{otherwise} \end{cases}$
- $C_{kl} = \begin{cases} 1, & \text{if train } i \text{ traverses through link } l \text{ after train } k \ (i, k \in E, l \in L_i \cap L_k) \\ 0, & \text{otherwise} \end{cases}$
- $a_{k}^t$: time at which train $k$ enters link $l$ ($k \in E \cup W, l \in L_k$);
- $d_{k}^t$: time at which train $k$ departs link $l$ ($k \in E \cup W, l \in L_k$).

B. Objective Functions

From a viewpoint of market, the level of service of a train timetable is a crucial factor that affects traveler’s decision in choosing desirable transportation modes. For the passenger train timetable problem, it is a good idea to minimize the total passengers’ time in all trains, which can be regarded as a scale of service for railway companies. This idea is also adopted in [9]. In this paper, we shall use this idea as an evaluation function of train timetable while the number of passengers in the train is represented by fuzzy variable other than crisp quantity according to the real conditions.

1. Total Passengers’ Time

To compute the total passengers’ time, we need to know the number of passengers in each train at every time. In the previous works, the number of passengers getting off/on the train at each station is always treated as a fixed quantity in mathematical model. In fact, it should not be fixed in nature due to uncertainty of the real system. In this sense, we shall treat the number of passengers getting off/on at each station as a fuzzy variable. Moreover, we assume that the membership function of the fuzzy variable is obtained in advance. Thus, the number of passengers in each train will be represented by a fuzzy variable at every time. For convenience, we use a fuzzy-variable-valued function $f_k(t)$ to denote the relationship between the operation time and
the number of passengers in train $k$, where fuzzy-variable-valued function refers to the function whose value assumes fuzzy variable. We assume that the total passengers’ time is the sum of each passenger time in all trains. Since the number of passengers getting off/on train at each station is assumed to be a fuzzy variable, the total passengers’ time should also be a fuzzy variable.

On the strategy planning level, the decision-maker should give some predetermined stations for each train, at which the train is scheduled to stop for allowing passengers to get on/off. When the train arrives at a predetermined station, a new sub-journey will start. Equivalently, starting a new sub-journey is the same as terminating the mined station, a new sub-journey will start. Equivalently, should also be a fuzzy variable.

The total passengers’ time in train $k$ at time $t_{k}$, denoted by $u_{k}$, is

\[
u_{k} = \frac{(f_{k}(t_{k}) + f_{k}(t_{k} + t_{a_{k}})) \times t_{a_{k}}}{2}.
\]

We can see that it is also a fuzzy variable. (Note that the operation of unloading passengers will not be involved in the first sub-journey of each train.)

- After the operation of unloading passengers, some passengers on the platform will board the train in time window $[t_{k} + t_{a_{k}}, t_{k} + t_{a_{k}} + t_{b_{k}}]$. The total passengers’ time in this interval is also the area below the fuzzy-variable-valued curve $f_{k}(t)$. We adopt the same treatment mentioned above. That is, this area is treated as the trapezoid with two parallel sides whose lengths are represented as fuzzy variables (see Figure 2). Thus the following formula is employed to compute the total passengers’ time in this time window:

\[
\frac{(f_{k}(t_{k}) + f_{k}(t_{k} + t_{a_{k}} + t_{b_{k}})) \times t_{b_{k}}}{2},
\]

which is also a fuzzy variable.

- After the above operations, the number of passengers in train $k$ will not vary until the train arrives at the initial station of the next sub-journey, i.e., at the time $t_{k_{o+1}}$. It is easy to see that the total passengers’ time in this interval is the length of this interval multiplied by the number of passengers. Actually, it is also the area below the curve $f_{k}(t)$, i.e., the rectangle with two parallel sides whose lengths are fuzzy variables. The total passengers’ time in this interval can be formulated as follows:

\[
f_{k}(t_{k} + t_{a_{k}} + t_{b_{k}}) \times (t_{k_{o+1}} - t_{k} - t_{a_{k}} - t_{b_{k}}),
\]

which is also a fuzzy variable.

![Figure 2. Variation of Passengers in Sub-Journey $k_n$](image)

Then the total passengers’ time in interval $[t_{k}, t_{k_{o+1}}]$, denoted by $u_{k}$, is

\[
u_{k} = \frac{(f_{k}(t_{k}) + f_{k}(t_{k} + t_{a_{k}})) \times t_{a_{k}}}{2} + \frac{(f_{k}(t_{k} + t_{a_{k}}) + f_{k}(t_{k} + t_{a_{k}} + t_{b_{k}})) \times t_{b_{k}}}{2} + f_{k}(t_{k} + t_{a_{k}} + t_{b_{k}}) \times (t_{k_{o+1}} - t_{k} - t_{a_{k}} - t_{b_{k}}).
\]

The total passengers’ time in train $k$, denoted by $U(T_k)$, is

\[
U(T_k) = \sum_{n=1}^{N_k} u_{k_n},
\]

where $T_k$ is the timetable of train $k$. The total passengers’ time on the railway is computed as follows:

\[
U(T) = \sum_{k \in EUW} U(T_k),
\]
where $T$ denotes the timetable of all trains.

In addition, there is a predetermined preferred departure time for each train. In the optimal train timetable, it is required that the actual departure time of each train should be close to its preferred departure time. For this purpose, we add a penalty function in the total passengers’ time. That is, we have

$$ F(T) = \sum_{k \in E \cup W} U(T_k) + N \cdot P(T). \tag{1} $$

In equation (1), notation “$N$” is a large positive number and function $P(T)$ is defined by

$$ P(T) = \sum_{k \in E} (d_{kS}^o - d_k^*) + \sum_{k \in W} (d_{kS}^o - d_k^*), $$

where $d_k^*$ denotes the planned departure time of train $k$. For a timetable, if all trains depart from their origin stations at the preferred times, then the penalty value in equation (1) will disappear. Otherwise, penalty value will be added to the total passengers’ time. By minimizing this objective, we can ensure that the actual departure time of each train in optimal timetable is as close to its preferred departure time as possible.

2. Total Delay time

Another objective in this problem is to minimize the total delay time. In fact, if train $k$ traverses on the railway without obstacles, then we can obtain the minimal end time of its journey, denoted by $F_k$. Thus the delay time of train $k$ on the railway is

$$ D_k = \begin{cases} d_{kS}^o - F_k, & k \in E, \\ d_{kS}^o - F_k, & k \in W, \end{cases} $$

where $d_{kS}^o$ denotes the actual arrival time of train $k$ at its destination station in the timetable. Then the total delay time of trains is formulated as

$$ G(T) = \sum_{k \in E} (d_{kS}^o - F_k) + \sum_{k \in W} (d_{kS}^o - F_k). $$

3. Goal Programming Model

Note that this problem is a multi-objective programming problem. We shall construct a goal programming model in the following to deal with it.

As mentioned above, we know that the total passengers’ time is a fuzzy variable for any train timetable $T$. In the process of seeking the optimal train timetable, it is meaningless to minimize the total passengers’ time since we cannot rank fuzzy variables directly. To rank fuzzy variables in a reasonable way, we shall use expected value criterion in the mathematical model.

For the problem discussed in this paper, we shall treat $E[F(T)]$ as an objective function. Thus, the first objective is to minimize $E[F(T)]$ other than $F(T)$ directly in the model. Generally, for these two objectives, the optimal values cannot be obtained at the same solution since one objective does not dominate the other one.

In the real conditions, due to the competition between different transportation corporations, operators should first consider the level of service in the process of making train timetables. Thus at the first priority level, we shall minimize the expected value of the total passengers’ time. In the following, we give the priority structure of objective functions in goal programming model.

At the first priority level of goal programming, the expected total passengers’ time with penalty value should not exceed a given target $\bar{T}$, that is,

$$ E[F(T)] + d_1^* - d_1^* = \bar{T}, $$

in which $d_1^*$ will be minimized.

At the second priority level, the total delay time should not exceed a given target $\bar{D}$. Thus, we have

$$ G(T) + d_2^* - d_2^* = \bar{D}, $$

in which $d_2^*$ will be minimized.

We construct a goal programming model as follows:

```
\text{lexmin} \quad V = \{d_1^*, \, d_2^*\} \\
\text{s.t.} \quad E[F(T)] + d_1^* - d_1^* = \bar{T} \\
\quad G(T) + d_2^* - d_2^* = \bar{D} \quad (2) \\
\quad T \text{ is feasible} \\
\quad d_1^*, d_2^*, d_3^*, d_4^* \geq 0.
```

In this model, the notation “lexmin” represents lexicographically minimizing the objective vector. To guarantee the feasibility of train timetable $T$, we introduce system constraints in the next subsection.

C. System Constraints

The discussed problem is to design a timetable for trains such that no conflicts will happen when trains traverse on the railway. Thus, it is necessary for us to check the feasibility of each timetable. In this subsection, we shall introduce system constraints of train timetable problem.

(1) Trip time on links constraints

In the real railway system, the speed of each train is always confined to a given range due to the different physical conditions of links. When a train travels with the maximal allowable speed, we can compute the minimal trip time on each link. Therefore, in system constraints, it is required that the time traveling on the link should not be less than the minimal trip time. Suppose that the minimal trip time of train $k$ on link $l$ is denoted by $t_{kl}$. Then we have the following constraints:

$$ d_{kl} - a_{kl} \geq t_{kl}, \forall k \in E \cup W, l \in L_k. $$

(2) Leaving and entering links constraints

In order to guarantee the safety of trains, the headway between adjacent trains in the same direction needs to be considered. For simplicity, we suppose that the velocity of
train $k$ on link $l$ is a fixed quantity. Thus in mathematical model, it is sufficient to consider the headway between the adjacent trains when they enter and leave the links.

For outbound trains $i$ and $k$, we have the following constraints when they leave link $l$:
\[d_{il} + h_i \leq d_{il} + M(1 - B_{il}), \forall i, k \in E, l \in L_k \cap L_i,\]
\[d_{il} + h_i \leq d_{il} + M(1 - B_{il}), \forall i, k \in E, l \in L_k \cap L_i.\]
For inbound trains $i$ and $k$, we have the following constraints when they leave link $l$:
\[d_{il} + h_i \leq d_{il} + M(1 - C_{il}), \forall i, k \in W, l \in L_k \cap L_i,\]
\[d_{il} + h_i \leq d_{il} + M(1 - C_{il}), \forall i, k \in W, l \in L_k \cap L_i.\]
For outbound trains $i$ and $k$, we have the following constraints when they enter link $l$:
\[a_{il} + h_i \leq a_{il} + M(1 - B_{il}), \forall i, k \in E, l \in L_k \cap L_i,\]
\[a_{il} + h_i \leq a_{il} + M(1 - B_{il}), \forall i, k \in E, l \in L_k \cap L_i.\]
For inbound trains $i$ and $k$, we have the following constraints when they enter link $l$:
\[a_{il} + h_i \leq a_{il} + M(1 - C_{il}), \forall i, k \in W, l \in L_k \cap L_i,\]
\[a_{il} + h_i \leq a_{il} + M(1 - C_{il}), \forall i, k \in W, l \in L_k \cap L_i.\]
(3) Meeting and passing constraints
To ensure that no collision occurs between two adjacent trains of opposite directions, we stipulate that meeting and passing operation is carried out at the station. Then it should satisfy
\[d_{il} \leq a_{il} + M(1 - A_{il}), i \in E, k \in W, l \in L_k \cap L_i,\]
\[d_{il} \leq a_{il} + M(1 - A_{il}), i \in E, k \in W, l \in L_k \cap L_i.\]
(4) Dwell time at the station constraints
For each passenger train, it is usually required some time for passengers leaving/boarding the train or for some operations being carried out at some stations. Assume that the minimal dwell time at station $i$ is $s_{ki}$ for train $k$. Then we have the following constraints:
\[a_{kl} - d_{k(i-1)} \geq s_{ki}, i \in S_k / \{S_k^D, S_k^P\}, k \in E,\]
\[a_{k(i-1)} - d_{ki} \geq s_{ki}, i \in S_k / \{S_k^D, S_k^P\}, k \in W.\]
(5) Time window constraints
Usually, it is stipulated that trains should not depart from their origin stations earlier than the preferred departure times and should not arrive later at their destinations than the given time horizon. Then we have the following constraints:
\[a_{kS_k^P} \geq d^*_k, k \in E,\]
\[a_{k(S_k^P)} \geq d^*_k, k \in W,\]
\[d_{kS_k^P} \leq H, k \in E,\]
\[d_{kS_k^P} \leq H, k \in W,\]
where $d^*_k$ denotes the preferred departure time of train $k$.

IV. Algorithm

As shown in the model, we need to calculate the expected value of fuzzy variable when seeking the optimal solution. Generally, there are two ways to do this work, i.e., analytic method and simulation. If the involved fuzzy datum is a special fuzzy variable, such as interval fuzzy variable, triangular fuzzy variable or trapezoidal fuzzy number, then we can compute expected value by analytic method. Otherwise, if analytic methods are invalid, simulation technique may be the most suitable way to calculate expected value although it is time-consuming.

We can see from Section I that different algorithms have been proposed for train timetable problem in literature. In this paper, we formulate this problem as a fuzzy mixed integer goal programming model. Up to now, no algorithm has been devoted to solving this kind of models.

In this section, we shall design a fuzzy simulation based branch and bound algorithm to seek the approximate optimal solution of the problem, in which branch and bound algorithm is employed to seek the feasible train timetable and simulation algorithm is used to compute objective value if analytic methods are invalid. For the fuzzy simulation algorithm, we can refer to Liu[23],[24]. Although the main process of branch and bound algorithm is similar to that of paper [11], the detailed techniques are different in nature, such as the techniques of making bound, computing objective value and resolving conflicts. The following are some details of the designed algorithm.

A. Representation of Solution

Assume that there are $K$ trains to be arranged to travel over the railroad, including $I$ outbound trains and $K - I$ inbound trains. We use the numbers $1, 2, \ldots, I$ and $I + 1, I + 2, \ldots, K$ to denote the indexes of outbound trains and inbound trains, respectively.

In computer, we use a matrix with the following form to represent a train timetable:
\[
\begin{bmatrix}
  a_{11} & d_{11} & \cdots & a_{1(n-1)} & d_{1(n-1)} \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  a_{I1} & d_{I1} & \cdots & a_{I(n-1)} & d_{I(n-1)} \\
  d_{(I+1)1} & a_{(I+1)1} & \cdots & d_{(I+1)(n-1)} & a_{(I+1)(n-1)} \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  d_{K1} & a_{K1} & \cdots & d_{K(n-1)} & a_{K(n-1)}
\end{bmatrix}
\]

In this matrix, the vector in the $k$th row denotes a schedule of train $k$. Generally, not all the trains need to be scheduled from station 1 to station $n$. If train $k$ does not pass line $l$ in its journey, then we have $a_{kl} = d_{kl} = 0$ in the above matrix.

B. Generate Initial Solution

This algorithm starts with an initial solution. In order to find the optimal solution within the minimum time, we
stipulate that the initial solution is the train timetable that all trains traverse on the railroad without obstacles. That is, when all trains traverse on the railroad with the maximized allowable speeds, the corresponding timetable is treated as the initial solution. If this timetable is feasible, it is the optimal solution obviously. However, the initial solution is not a feasible solution in most circumstances since some conflicts will occur between different trains.

C. Find the First Conflict

To obtain a feasible timetable, all the conflicts should be resolved if the initial timetable is infeasible. In the algorithm, the conflicts will be resolved one by one and this process always begins with the first conflict in order to save the computational time. By the following procedure, we can obtain a feasible train timetable with the minimum computational time.

step:1. Let T be a timetable;
step:2. If there is no conflict in T, stop (T is feasible); otherwise, find the first conflict in T;
step:3. Update T by resolving the first conflict. go to step 2.

In the algorithm, we use the following way to compute the conflict time of two trains. Assume that trains i and j in different directions will cause a conflict on link l. Figure 3 shows possible conflict that may appear for trains i and j, where notations a, b, c and d denote the times that the trains enter or depart link l.

For this case, we can compute the conflict time e by the following formula:

\[ e = \frac{a(d - c) - c(b - a)}{d - c + a - b}. \]

It is easy to see that the value of e is completely dependent on the parameters a, b, c and d.

In Figure 4, we shall show conflicts that may occur between trains in the same directions.

D. Solve Conflict

For simplicity, we only introduce how to resolve the conflict for two outbound trains, and the other cases can be resolved similarly. Suppose that outbound trains i and j will be scheduled from station 1 to station n. Their timetables are listed as follows:

\[ (a_1, d_1, a_2, d_2, \ldots, a_n, d_n, a_{n-1}, d_{n-1}), \]
\[ (a_j, d_j, a_j, d_j, \ldots, a_{j-1}, d_{j-1}, a_{j-1}, d_{j-1}). \]

Assume that a conflict occurs on link l such that \( a_i \leq a_j \leq d_j \leq d_i \). There are two alternatives to resolve this conflict, that is, train i is delayed first or train j is delayed first. When train j is delayed, we set \( t = h_i + d_i - d_j \). Thus the resolved timetables for trains i and j, respectively, are:

\[ (a_1, d_1, a_2, d_2, \ldots, a_i, d_i, \ldots, a_{n-1}, d_{n-1}), \]
\[ (a_j, d_j, a_j, d_j, \ldots, a_j + t, d_j + t, \ldots, a_{j-1} + t, d_{j-1} + t). \]

The process of resolving this case on link l is shown in Figure 5.
If objective vector \( V^{(1)} \) is better than \( V^{(2)} \), then we use notation \( V^{(1)} \succeq V^{(2)} \) or \( V^{(2)} \preceq V^{(1)} \) to denote this relationship. Also, the train timetable corresponding to \( V^{(1)} \) is better than that corresponding to \( V^{(2)} \).

F. Branch and Bound Algorithm

In computer, we use a binary tree to record the searching process when seeking the optimal timetable, in which each train timetable is represented by a node. For an infeasible node, if we resolve its first conflict, two new timetables will be obtained, which are denoted by two son nodes in the binary tree. Thus, each node is obtained by delaying one of the trains involved in the first conflict of its father node. Then, the objective value of each node is not less than that of its father node if parameter \( N \) in the penalty function is large enough. Based on this property, we shall completely use the objective value to make bound other than calculate the lower bound estimate adopted in [11], which can simplify the computation to some degree. In the searching process, we always use the best objective value so far encountered as the upper bound of the objective.

The following notations are employed in the algorithm:
- \( T \): train timetable which is not necessarily feasible;
- \( T.\) : the left branch of node \( T \);
- \( T.\) : the right branch of node \( T \);
- \( T.\) : the father node of \( T \);
- \( U \) : the upper bound in the algorithm;
- \( V(T) \) : the objective value of train timetable \( T \), which can be calculated by analytic method or fuzzy simulation algorithm if analytic methods are invalid;
- Bestsol: the best train timetable encountered so far.

In the following, we list the procedure of the branch and bound algorithm.

**step 1.** Generate the initial train timetable \( T \) when all trains travel on the links with the maximal allowable speeds without obstacles, let \( T.\) = 1, \( T.\) = NULL, set the upper bound \( U \) of objective vector.

**step 2.** If \( T.\) \leq 1, go to Step 3; otherwise, go to Step 7;

**step 3.** If \( T \) is feasible and \( V(T) \leq U \), let \( \text{Bestsol}=T \), \( U=V(T) \), go to Step 7. If \( V(T) \leq U \) and \( T \) is infeasible, then go to Step 4. For other cases, go to Step 7;

**step 4.** Calculate conflict time for each conflict in \( T \), then select the first conflict. Pick out the trains \( i, j \) involved and the link \( l \) where this conflict occurs;

**step 5.** There are two alternatives to resolve the first conflict, that is, either train \( i \)
or train \( j \) is delayed. For each alternative, update the timetable by using the method presented in Subsection D. Then we can get two branches of \( T \), denoted by \( T \mathrm{ left} \) and \( T \mathrm{ right} \); let \( (T \mathrm{ left}) \mathrm{ counter} =1 \), \( (T \mathrm{ right}) \mathrm{ counter} =1 \).

**Step 6.** Let \( T \leftarrow T \mathrm{ left} \), go to Step 3; step 7. If \( T \mathrm{ previous} = \mathrm{ NULL} \), go to Step 9; otherwise, let \( T' \leftarrow T \mathrm{ previous} \), go to Step 8; step 8. If \( T \) is the left branch of \( T' \), then \( T \leftarrow T' \), \( \mathrm{ right} \); otherwise, if \( T \) is the right branch of \( T' \), let \( T \leftarrow T' \), \( T \mathrm{ counter} ++ \).

Go to Step 2;

**Step 9.** Output \( \text{Bestsol} \), stop.

The searching process of this algorithm is shown in Figure 7. The algorithm begins with the root node (initial solution). Each node represents a train schedule, which is obtained by resolving the earliest conflict of its father node. After resolving the conflict, each node will produce two son nodes, that is, left branch and right branch. This process goes on according to the arrow direction. When the root node is reached two times, the searching process will be finished. The corresponding best feasible train schedule will be treated as an approximate optimal solution.

![Figure 7. The Searching Process of the Algorithm](image)

### V. Numerical Example

In this section, we shall present a numerical example to show applications of the model and the algorithm. Consider a railroad with 6 stations and 5 links (see Figure 8).

![Figure 8. Railroad in the Example](image)

The stations and links are numbered consecutively in the outbound direction, that is, with the notations 1, 2, 3, 4, 5, and 1, 2, 3, 4, 5, respectively. The distances of different links are \( l_1 = 130 \text{km} \), \( l_2 = 80 \text{km} \), \( l_3 = 200 \text{km} \), \( l_4 = 150 \text{km} \), \( l_5 = 120 \text{km} \). The maximum allowable speeds on links are \( V_1 = 100 \text{km/h} \), \( V_2 = 80 \text{km/h} \), \( V_3 = 130 \text{km/h} \), \( V_4 = 120 \text{km/h} \), \( V_5 = 100 \text{km/h} \). There are 10 trains to be scheduled on the railroad. Trains 1, 2, 3, 4, 5 are outbound from station 1 to station 6 and trains 6, 7, 8, 9, 10 are inbound from station 6 to station 1. Thus, the minimum trip times (unit: minute) on each link are \( t_{k1} = 78 \text{min} \), \( t_{k2} = 60 \text{min} \), \( t_{k3} = 80 \text{min} \), \( t_{k4} = 75 \text{min} \), \( t_{k5} = 72 \text{min} \), \( k = 1, 2, \ldots, 10 \).

The number of passengers getting on/off each train at each station and the operation time for getting on/off the train are listed in Tables I-II, where the notation with form \( (a, b, c) \) denotes a triangular fuzzy variable. Suppose that every train has enough capacity to accommodate its passengers. In addition, we set the parameter \( N \) in the first goal constraint as 1600.

On the strategy planning level, the decision-maker determines the preferred departure time for each train. For this example, the preferred times that passengers are allowed to get on each train are listed as follows: train 1: 0:00, train 2: 1:00, train 3: 2:00, train 4: 3:00, train 5: 4:00, train 6: 0:00, train 7: 1:00, train 8: 2:00, train 9: 3:00, train 10: 4:00. We suppose that the headway between adjacent trains is 10 minutes on each link.

For the convenience of expression, we use “minute” as a basic unit to schedule the trains in the algorithm. That is, the integer 0 denotes the time 0:00, 60 denotes the time 1:00, and the representations of other times can be deduced by analogy. For simplicity, we omit the passengers’ time of getting off each train at each station. The initial solution is listed as follows after considering the operation time of boarding the train at the origin station:

\[
\begin{bmatrix}
20 & 98 & 98 & 158 & 178 & 258 & 258 & 333 & 353 & 425 \\
80 & 158 & 166 & 226 & 226 & 306 & 321 & 396 & 396 & 468 \\
140 & 218 & 218 & 278 & 298 & 378 & 378 & 453 & 453 & 525 \\
190 & 268 & 268 & 328 & 338 & 418 & 418 & 493 & 501 & 573 \\
260 & 338 & 338 & 398 & 398 & 478 & 498 & 573 & 596 & 668 \\
420 & 342 & 327 & 267 & 267 & 187 & 167 & 92 & 92 & 20 \\
482 & 404 & 404 & 344 & 319 & 239 & 239 & 164 & 152 & 80 \\
523 & 445 & 445 & 385 & 365 & 287 & 287 & 212 & 212 & 140 \\
578 & 500 & 487 & 427 & 427 & 347 & 337 & 262 & 262 & 190 \\
\end{bmatrix}
\]

We give the corresponding train timetable graph in Figure 9:

![Figure 9. The Initial Train Timetable Graph](image)

We can see from Figure 9 that the initial solution is not feasible since there exist 23 conflicts in the timetable.
For the convenience of understanding, we present an optimal train timetable graph in Figure 10, for which the capacity of each station is 6, the first target is 61700 hrs and the second target is 12.3 hrs. We can see from Figure 10 that the number of trains stopping at each station at any time is not larger than 6, which shows that capacity constraint is not violated for any station. In this optimal timetable, all the trains leave their origin stations at the preferred departure times. Trains 1 and 2 reach the destination station almost at the same time, and the same occurs for trains 3, 4 and 5. We can see from the graph that trains 3 and 4 are delayed at station 3 for more than one hour to avoid the conflicts with trains 6, 7, 8 and 9. In fact, to decrease the waiting time at the third station, trains 3 and 4 can start from the origin station later than the preferred departure times, if the regulations will not cause new conflicts. Equivalently, decreasing the waiting time at the third station will increase the delay time at the origin station. If the decision-maker do not care about the deviations from the preferred departure times, the actual departure times of train 3 and 4 can be regulated. For the inbound trains, trains 6, 7, 8 and 9 reach the destination station almost at the same time. At station 4, train 6 is delayed more than two hours to avoid the conflicts with trains 1 and 2. We can regulate the departure time of train

For the convenience of understanding, we present an optimal train timetable graph in Figure 10, for which the capacity of each station is 6, the first target is 61700 hrs and the second target is 12.3 hrs. We can see from Figure 10 that the number of trains stopping at each station at any time is not larger than 6, which shows that capacity constraint is not violated for any station. In this optimal timetable, all the trains leave their origin stations at the preferred departure times. Trains 1 and 2 reach the destination station almost at the same time, and the same occurs for trains 3, 4 and 5. We can see from the graph that trains 3 and 4 are delayed at station 3 for more than one hour to avoid the conflicts with trains 6, 7, 8 and 9. In fact, to decrease the waiting time at the third station, trains 3 and 4 can start from the origin station later than the preferred departure times, if the regulations will not cause new conflicts. Equivalently, decreasing the waiting time at the third station will increase the delay time at the origin station. If the decision-maker do not care about the deviations from the preferred departure times, the actual departure times of train 3 and 4 can be regulated. For the inbound trains, trains 6, 7, 8 and 9 reach the destination station almost at the same time. At station 4, train 6 is delayed more than two hours to avoid the conflicts with trains 1 and 2. We can regulate the departure time of train
### Table II

**The Number of Passengers Getting on/off at Each Station for Inbound Trains**

<table>
<thead>
<tr>
<th>Train</th>
<th>Station</th>
<th>Get Off Time (unit: min)</th>
<th>Get On Time (unit: min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>(800, 830, 850)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>(200, 215, 230)</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(300, 330, 350)</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>0</td>
<td>(800, 850, 870)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>(200, 230, 250)</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(400, 425, 450)</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>0</td>
<td>(900, 950, 970)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>(300, 325, 350)</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>0</td>
<td>(200, 225, 250)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>(130, 140, 150)</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(100, 125, 150)</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>0</td>
<td>(700, 725, 750)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>(330, 340, 350)</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(200, 225, 250)</td>
<td>8</td>
</tr>
</tbody>
</table>

6 if needed. The similar regulation can also be considered for train 10 at station 5 if no new conflicts appear.

Figure 10. The Optimal Train Timetable Graph

In addition, we can investigate the sensitivity of two objective functions with respect to different station capacities. If only one objective is considered, we can obtain two different mathematical models. One model is with the total passengers’ time and the other is with the total delay time. We can solve the models by using branch and bound algorithm. The optimal objective values and the other corresponding objectives are listed in Table IV and Table V. We can see that when we use different station capacities, the optimal train timetables of two models are different, which implies that the first objective does not necessarily dominate the second objective.

Figures 11 an 12 show the variations of two optimal objective values. As for the first objective, with the increase of station capacity, the expected value of the total passengers’ time will decrease. It is easy to see that the optimal EVTPT is more sensitive for station capacities from 2 to 5 than the other cases. When station capacity is larger than 5, the optimal expected value of the total passengers’ time will not vary. Additionally, the optimal total delay time will decrease with the increase of station capacity. In this process, the optimal total delay time turns insensitive more and more. When station capacity is larger than 5, the optimal total delay time does not vary any more. In fact, for other numerical examples, we can also analyze the sensitivity of objectives by this way.

Figure 11. Sensitivity of Optimal EVTPT
For the expected value of the total passengers’ time, we introduce the following formula to compute the satisfaction degree:

$$\mu_1 = 1 - \frac{\text{EVTPT}_{op} - \text{EVTPT}_{in}}{\text{EVTPT}_{op}}$$

where the notation “EVTPT\text{\text{in}}” represents the expected value of the total passengers’ time in the initial solution, and “EVTPT\text{\text{op}}" represents the expected value of the total passengers’ time in the approximate optimal solution.

We can also see that the above formula is the ratio of the expected total passengers’ time of the optimal solution to that of the initial solution. It is obvious that the closer \(\text{EVTPT}_{op}\) is to \(\text{EVTPT}_{in}\), the larger the satisfaction degree is, which coincides with the common sense. Moreover, if \(\text{EVTPT}_{op} = \text{EVTPT}_{in}\), then the satisfaction degree \(\mu_1 = 1\).

For the second objective, we shall use two criteria to evaluate the performance of the optimal solution.

**Criterion 1:** The total traveling time of all trains. We first introduce the following notations:

- \(\text{TIT}_k\): the traveling time of train \(k\) in the initial solution;
- \(\text{TIT}'_k\): the actual traveling time of train \(k\) in the optimal solution.

We use the following formula to calculate the satisfaction degree of the solution:

$$\mu_{21} = 1 - \frac{\sum_{k=1}^{K} \text{TIT}'_k - \sum_{k=1}^{K} \text{TIT}_k}{\sum_{k=1}^{K} \text{TIT}_k}.$$
It is easy to see that if the actual total traversing time is close to the initial total traversing time, then the satisfaction degree is close to 1.

**Criterion 2:** The time to clear the line. We introduce some notations in the following:

- $J_a$: the departure time of the earliest train in the initial solution;
- $J_a'$: the arrival time of the last train in the initial solution;
- $J_d$: the arrival time of the last train in the optimal solution;
- $J_d'$: the arrival time of the last train in the optimal solution.

The time to clear the line in the initial solution and the optimal solution, respectively, can be computed by:

$$J_a - J_d, J_a' - J_d'.$$

We introduce the following formula to show the satisfaction degree of the train timetable:

$$\mu_{22} = \frac{J_a - J_d}{J_a' - J_d}.$$  \(\text{Figure 13. Variation of Satisfaction Degree}\)

This formula also complies with the common sense, that is, the greater $\mu_{22}$ is, the more optimal the corresponding solution is.

For the results in Tables IV and V, we obtain the satisfaction degree of each optimal solution in Table VI. Figure 13 show the variations of satisfaction degrees with respect to different station capacities.

It is easy to see from Table VI that the satisfaction degrees are not less than 0.8121 for the first objective when station capacity is not less than 3. This is a good result for the train timetable problem on a single-line railway. For the second objective, the satisfaction degrees of two criteria are not less than 0.7833 when station capacity is not less than 3. This is a good result for decision-makers. But when each station's capacity is 2, the values of $\mu_1$ and $\mu_{21}$ are below 0.7700 and the value of $\mu_{22}$ is 0.6708. The reason for this fact is that the decrease of station capacity causes some trains to be delayed more time at some stations. Thus the total passengers’ time, the total traversing time and the time to clear the line increase accordingly.

**VI. Conclusion**

This paper mainly investigated a passenger train timetable problem under the fuzzy environment, in which the number of passengers getting on/off the train at each station was treated as a fuzzy variable other than a crisp quantity. A expected value mixed integer goal programming model was constructed for the problem. In order to obtain the optimal solution, a simulation based branch and bound algorithm was also designed to solve the model, in which branch and bound algorithm is used to seek feasible train timetables and simulation algorithm is used to compute objective if analytic methods are invalid. At last, some numerical experiments were performed to show applications of the model and the algorithm, in which the fuzzy data are supposed to be triangular fuzzy variables and the objective function is computed by analytic method. The computational results imply that the designed branch and bound algorithm is effective in solving the problem.

**Acknowledgement**

The authors would like to thank the editor and the anonymous referees for their valuable comments and suggestions.

**References**


TABLE VI
THE SATISFACTION DEGREE OF EACH OPTIMAL SOLUTION

<table>
<thead>
<tr>
<th>Station</th>
<th>Capacity</th>
<th>μ₁</th>
<th>μ₂₁</th>
<th>μ₂₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.7610</td>
<td>0.7691</td>
<td>0.6708</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.8121</td>
<td>0.7833</td>
<td>0.7931</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.8380</td>
<td>0.8294</td>
<td>0.8594</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.8444</td>
<td>0.8428</td>
<td>0.8471</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.8444</td>
<td>0.8428</td>
<td>0.8471</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.8444</td>
<td>0.8428</td>
<td>0.8471</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.8444</td>
<td>0.8428</td>
<td>0.8471</td>
<td></td>
</tr>
</tbody>
</table>

Keping Li received the M.S. degree in theoretical physics in 1997 from the Normal University of GuangXi, and received the Ph.D. degree in theoretical physics in 2002 from the NanKai University, Tianjin, China. He is an associate professor in Beijing Jiaotong University’s State Key Laboratory of Rail Traffic Control and Safety. His current research interests include stochastic programming, fuzzy programming, intelligent systems, and applications in transportation problem and rail traffic control systems. He has published more than 20 papers in national conferences, international conferences and premier journals.

Ziyou Gao was born in 1963. In January 1994 he got the Ph.D. degree at Institute of Applied Mathematics of Chinese Academy of Sciences, majoring in Operations Research and Control Theory. From October 1993 to September 1995, as a post doctor studied in Transportation Simulating Center in Northern Jiaotong University, he majored in Transportation Management. As a professor in 1996 and as a doctor advisor in 1997 he has been working in the Northern Jiaotong University (Beijing Jiaotong University). Now he assumes Director of State Key Laboratory of Rail Traffic Control and Safety. In his part-time, he assumes the following offices: Vice-Chairman of China Transportation Systems Engineering Society, advanced member of China Railway Academy, Co-editor of Traffic and Transportation Systems Engineer and Information(a journal), an editorial board of Transportmetrica (an international journal). In 2003, he was elected as a foreign member of the academy, Russian Academy of Natural Sciences.


Jiaotong University (Beijing Jiaotong University), Beijing, China. He is an associate professor in Beijing Jiaotong University’s State Key Laboratory of Rail Traffic Control and Safety. His current research interests include stochastic programming, fuzzy programming, intelligent systems, and applications in transportation problem and rail traffic control systems. He has published more than 20 papers in national conferences, international conferences and premier journals.

Lixing Yang received the B.S. and M.S. degrees from the Department of Mathematics, Hebei University, Baoding, China, in 1999 and 2002, respectively. In 2005, he received Ph.D. degree from the Department of Mathematical Sciences, Tsinghua University, Beijing, China. Since 2005, he has been with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China, where he is a Lecturer. His current research interests include stochastic programming, fuzzy programming, intelligent systems, and applications in transportation problem and rail traffic control systems. He has published more than 20 papers in national conferences, international conferences and premier journals.


