

Research Article

Networked Timetable Stability Improvement Based on a Bilevel Optimization Programming Model

Xuelei Meng,¹ Bingmou Cui,¹ Limin Jia,² Yong Qin,² and Jie Xu²

¹ School of Traffic and Transportation, Lanzhou Jiaotong University, P.O. Box 405, Anning West Road, Anning District, Lanzhou, Gansu 730070, China

² State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, No. 3 Shangyuancun, Haidian District, Beijing 100044, China

Correspondence should be addressed to Xuelei Meng; mengxuelei@gmail.com

Received 28 November 2013; Revised 13 January 2014; Accepted 23 January 2014; Published 4 March 2014

Academic Editor: Wuhong Wang

Copyright © 2014 Xuelei Meng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Train timetable stability is the possibility to recover the status of the trains to serve as arranged according to the original timetable when the trains are disturbed. To improve the train timetable stability from the network perspective, the bilevel programming model is constructed, in which the upper level programming is to optimize the timetable stability on the network level and the lower is to improve the timetable stability on the dispatching railway segments. Timetable stability on the network level is defined with the variances of the utilization coefficients of the section capacity and station capacity. Weights of stations and sections are decided by the capacity index number and the degrees. The lower level programming focuses on the buffer time distribution plan of the trains operating on the sections and stations, taking the operating rules of the trains as constraints. A novel particle swarm algorithm is proposed and designed for the bilevel programming model. The computing case proves the feasibility of the model and the efficiency of the algorithm. The method outlined in this paper can be embedded in the networked train operation dispatching system.

1. Introduction

Train timetable is the fundamental file for organizing the railway traffic, which determines the inbound and outbound time of trains. Railways are typically operated according to a planned (predetermined) timetable, and the quality of the timetable determines the quality of the railway service. So it is most important to map a high quality timetable for all kinds of trains.

But there is a dilemma that we place as much as possible trains on the timetable chart, and simultaneously we should enhance the possibility to adjust the timetable when disruptions occur. The randomly occurring disturbances may cause train delays and even disrupt the entire train operation plan. In the railway network, every station and section are planned to serve the trains according to the schedule, often compactly. So a slightly delayed train may cause a domino effect of secondary delays over the thorough network. Although the buffer times added to the minimum

running time in the sections and minimum dwell at stations in scheduled timetables may absorb some train delays and assure some degree of timetable stability, the large buffer time will reduce the capacity of the railway.

Therefore, to ensure both the capacity and the order of the train operation, a reliable, stable, robust timetable, and the feasible efficient rescheduling of the planned timetable must be worked out. A superior quality timetable cannot only decide the inbound and outbound time at stations, and the more important, can offer the possibility to recover the operation according to the planned timetable when the trains are disturbed by accidents randomly. Timetable stability is the index to measure the *possibility*. Timetable stability is related to the train number assigned to the railway sections and the buffer time distributed to each station and section, the probability that the train is disrupted at the stations and in the sections. In this paper, the buffer time refers to the time added to the minimum running time in a section. It equals the planned period of running minus the minimum running

time in the section. And it also refers to the time added to the dwelling time at a station, which equals the planned dwelling time at the station minus the minimum dwelling time.

So when assigning the trains paths and mapping out the train schedule, the train number assigned to the sections and buffer time distribution should be designed carefully, not only considering the section capacity and station capacity, but also the minimal running time at each section and the minimal dwell on the station.

We define the networked timetable stability quantitatively, considering that the railway network with the goal is to optimize the timetable stability to offer more possibilities to reschedule the trains on the railway network to deal with disturbs in the train operation process.

The outline of the paper is as follows. First, Section 2 proposes the literature review on timetable stability improvement. Section 3 builds the bilevel optimization model for the networked timetable stability. Then Section 4 introduces the hybrid fuzzy particle swarm algorithm, improving the velocity equation. Section 5 applies hybrid fuzzy particle swarm optimization algorithm in solving the bilevel model for improving the networked timetable stability. The computing case is presented in Section 6. Finally, Section 7 gives some conclusions.

2. Literature Review

It is a hot topic now to assure the reliability, safety, and stability of the traffic control system as discussed in [1–3]. The timetable stability optimizing is a relatively new issue in the field of railway operation research. The research work was published in the 1990s.

The research experienced two developing periods. In the earliest period, the focus was the timetable on the dispatching section, according to the operation mode of the railway and the basis to study the train timetable stability is formed. A discrete dynamic system model was built to describe the timetable with the max-plus algebra based on the discussion of the timetable periodicity and analyzed the timetable stability, as proposed by Goverde [4]. Carey and Carville developed a simulation model to test the schedule performance and reliability for train stations in [5]. Hansen pointed out that the effect of the stochastic disturbance on trains relied on the adjustment of the running time and buffer time in the timetable and assessed the advantages and the disadvantages of the capacity and stability of evaluating model [6]. These researchers promoted the train timetable stability theory from the perspective of the running time in railway sections, the dwelling time on railway stations, and the buffer time for running and dwelling. De Kort et al. proposed a method to evaluate the capacity determined by the timetable and took the timetable stability as a part of the capacity; see [7]. The goal was tantamount to place train running lines as much as possible, while taking the timetable into consideration at the same time. Goverde presented a method based on max-plus algebra to analyze the timetable stability. He proved the feasibility of the method with data of the Netherlands national railway timetable; see [8]. We

defined and qualified timetable stability and took it as a goal when rescheduling trains on the dispatching sections in [9]. So it is easy to understand that the time is the key factor when studying the timetable of a dispatching railway section. Focusing on the delay time, the behind schedule ratio, the buffer time, and time deviation, researchers studied the timetable adjustability, equilibrium, stability, using statistics theories, max-plus algebra, and so forth.

Research on timetable stability progressively expanded to the railway network, for the study focusing on the timetable stability of dispatching cannot suit the networked timetable design and optimization. Engelhardt-Funke and Kolonko considered a network of periodically running railway lines. They built a model to analyze stability and investments in railway networks and designed an innovative evolutionary algorithm to solve the problem in [10]. Goverde analyzed the dependence of the timetable on the busy degree of the railway network. He again hired the max-plus algebra, to analyze the timetable stability of the railway network. On this basis, he proposed a novel method to generate the paths for the trains on a large-scale railway network; see [11]. Vromans built a complex linear programming model to optimize the timetable on the railway network level, taking the total delayed time as the optimizing goal. And they designed the stochastic optimization algorithm for the model; see [12]. Delorme et al. presented a station capacity evaluating model and evaluated the stability on the key parts of the railway network: stations; see [13]. We analyzed the complex characteristic analysis of passenger train flow network in the former study work [14] and have done some research work to support the networked train timetable stability optimization, from transportation capacity calculation [15], paths generating [16], and line planning [17], which can be seen as the constraint of timetable stability optimizing.

And we can see that the networked timetable stability is related to not only time but also the utilization coefficient capacity of the railway network, as discussed in [11–13]. That is to say, the networked timetable stability study requires the combination of the railway network capacity utilization and the buffer time distribution of the buffer time in the sections and at the stations. However, most of the publications are about the stability of the timetable for a definite dispatching railway section. And there are limited publications about the networked timetable stability. Furthermore, the research on the timetable is in the stage of evaluating, mostly qualitative, not the quantization of the timetable stability.

3. The Bilevel Optimization Programming Model for Networked Timetable Stability Improvement

Networked timetable stability must be studied from two levels. The upper level is to study the relation between the trains flow and the capacity of the sections and stations and the ability to recover the timetable when an emergency occurs determined by the relation. The lower level is to study the distribution plan of the buffer time for each train in

the sections running process and the stations dwelling, to eliminate the negative effects of the disturbs.

The goal of the upper programming is to decide the number of trains assigned on each railway section and at the stations. The fundamental restriction is that the number of trains assigned to the sections and stations must not exceed the capacity of the sections and the stations. And the number of trains received by the stations must be equal to the total number of the trains running through the sections which are connected to the relative station.

The lower programming is to determine the buffer time distribution plan. The running time through a whole section planned in the timetable is more than that it requires if it runs at its highest speed. So there is a period of time called buffer time that can be distributed for the sections running and stations dwelling to absorb the delay caused by the random disturbances. The restriction is that buffer time allocated to each station and section must be longer than or equal to zero.

3.1. The Timetable Stability Improvement Programming on the Network Level. To define the timetable stability on the network level, the load on the sections and stations is the key factor. So the load index numbers must be defined first.

Definition 1. The load index number of a station on the railway network is

$$Z_{ST} = \text{Var} (e^{-\rho_i} w_{ST,i}^{\rho_i - \rho}), \quad (1)$$

where Var is the function to calculate the variance of a vector, ρ_i is the load of the i th station; the bigger ρ_i is, the smaller the stability value is. $\rho_i = F_i/B_i$, B_i is the receiving and sending capacity of the station. F_i is the number of the receiving and sending trains by the i th station according to the trains distribution plan. ρ is a threshold value of a station load. $w_{ST,i}$ is the weight of the i th station, and K is the number of the stations on the railway network.

The index number of the capacity of a station is

$$IX_{ST,i} = \frac{B_i D_i}{\sum_{i=1}^K B_i D_i}, \quad (2)$$

where D_i is the degree of the i th station.

Then the station weight is

$$w_{ST,i} = \frac{IX_{ST,i}}{\sum_{i=1}^K IX_{ST,i}}. \quad (3)$$

Definition 2. The load index number of a section on the railway network is

$$Z_{SE} = \text{Var} (e^{-\lambda_i} w_{SE,i}^{\lambda_i - \lambda}), \quad (4)$$

where Var is the function to calculate the variance of a vector, λ_i is the load of the i th section; the bigger λ_i is, the smaller the stability value is. $\lambda_i = G_i/C_i$, C_i is the capacity of the section. G_i is the number of the trains running through i th section according to the trains distribution plan. λ is a threshold value

of a section load. w_i is the weight of the i th section, and L is the number of the sections on the railway network.

The index number of the capacity of a section is

$$IX_{SE,i} = \frac{C_i}{\sum_{i=1}^L C_i}. \quad (5)$$

Then the weight of the section is

$$w_{SE,i} = \frac{IX_{SE,i}}{\sum_{i=1}^L IX_{SE,i}}. \quad (6)$$

Then with the load index numbers of the stations and sections, the timetable stability on the network level is defined as

$$S_{NET} = e^{-Z_{ST}} \times e^{-Z_{SE}} = e^{-(Z_{ST} + Z_{SE})}. \quad (7)$$

The goal of the upper programming is to optimize the timetable stability on the network level, so S_{NET} is taken as the optimization goal. That is to say, the goal is to maximize the timetable on the network level: S_{NET} .

Restrictions require that the number of the trains running through a section cannot be greater than the number of the trains that the section capacity allows. Likewise, the total trains number going through a station cannot exceed the station capacity of receiving and sending off trains.

And the total numbers of the trains distributed on the sections connected to the station must be equal to the number of arriving trains at the station:

$$\begin{aligned} F_i &\leq B_i, \\ G_i &\leq C_i, \\ F_i &= \sum_{l=1}^U G_{i,l}, \end{aligned} \quad (8)$$

where U is the number of sections connected to station i .

3.2. The Timetable Stability Improvement Programming on the Dispatching Section Level. Take it for granted that there are M trains going through section p , which is the result of the upper programming. The running times of all the M trains form a vector $T_R^p = \{t_{R,i}^p\}_M$. The minimum running time of all the M trains forms a vector $T_R^{p,\min} = \{t_{R,i}^{p,\min}\}_M$. Then the margin vector of the M trains is $\Delta T_R^p = \{\Delta t_{R,i}^p\}_M = \{t_{R,i}^p - t_{R,i}^{p,\min}\}_M$. Set the $A_R^p = \{\Delta t_{R,i}^p / t_{R,i}^p\}_M$ to be the running adjustability vector. To evaluate the equilibrium of the distribution of the buffer time in the sections, the running adjustability dispersion is defined as

$$\text{Var} (A_R^p) = \text{Var} \left(\frac{\Delta t_{R,i}^p}{t_{R,i}^p} \right). \quad (9)$$

The smaller the value of the $\text{Var}(A_R^p)$ is, the more balanced the buffer time distribution plan is, and the timetable is more stable.

Likewise, take it for granted that there are N trains going through station q , with stop or without stop. The planned

dwelling time according to the timetable of the N trains forms a vector $T_D^q = \{t_{D,i}^q\}_N$. The minimal dwelling time of all the N trains at q also forms a vector $T_D^{p,\min} = \{t_{D,i}^{p,\min}\}_N$. Then the margin vector of the N trains is $\Delta T_D^q = \{\Delta t_{D,i}^q\}_N = \{t_{D,i}^q - t_{D,i}^{q,\min}\}_N$. Set the $A_D^q = \{\Delta t_{D,i}^q / t_{D,i}^q\}_N$ to be the dwelling adjustability vector. To evaluate the equilibrium of the distribution of the buffer time at stations, the adjustability dispersion is defined as

$$\text{Var}(A_D^q) = \text{Var}\left(\frac{\Delta t_{D,i}^q}{t_{D,i}^q}\right). \quad (10)$$

The smaller the value of the $\text{Var}(A_D^q)$ is, the more balanced the buffer time distribution plan is, and the timetable is more stable.

On the basis of considering of the running adjustability dispersion and the dwelling adjustability dispersion, the timetable stability on the dispatching section level is defined as

$$S_{\text{DIS}} = e^{-\text{Var}(A_R^p) \times \text{Var}(A_D^q)} = e^{-\text{Var}(\Delta t_{R,i}^p / t_{R,i}^p) \times \text{Var}(\Delta t_{D,i}^q / t_{D,i}^q)}. \quad (11)$$

Then we take the timetable stability on the network level as the optimizing goal of the upper programming:

$$\max S_{\text{DIS}}. \quad (12)$$

When rescheduling the trains on the sections, the minimum running time and the minimum dwelling time must be considered. The rescheduled running time and dwelling time must be longer than the minimum time, which is described in (13) and (14). And the margins between inbound times of different trains at the same stations must be bigger than the minimum interval time to ensure the safety of train operation. This constraint is defined in (15). Likewise, the margins between outbound times of the trains at a station have the same characteristic, as shown in (16). And the number of the trains dwelling at a same station cannot be bigger than the number of the tracks in a station, described in (17):

$$t_{R,i}^p \geq t_{R,i}^{p,\min}, \quad (13)$$

$$t_{R,i}^q \geq t_{R,i}^{q,\min}, \quad (14)$$

$$\left| a_{i,j}^p - a_{l,j}^p \right| > I_{a-a}, \quad l \neq i, \quad (15)$$

$$\left| d_{i,j}^p - d_{l,j}^p \right| > I_{d-d}, \quad l \neq i, \quad (16)$$

$$LN_j - \sum_{p \in P} \sum_{k=1}^{\text{TN}} n_{p,j}^k \geq 0. \quad (17)$$

3.3. The Networked Timetable Stability Definition. Depending on the analysis in Sections 3.1 and 3.2, networked timetable stability is defined with the following:

$$S = S_{\text{NET}} \times S_{\text{DIS}}. \quad (18)$$

We can see that the networked timetable stability is directly related to timetable stability on the network level and the dispatching section level. The programming in Sections 3.1 and 3.2 can optimize the networked timetable stability, through optimizing the stability on the two levels.

4. The Hybrid Fuzzy Particle Swarm Optimization Algorithm

4.1. Fuzzy Particles Swarm Optimization Algorithm. Considerable attention has been paid to fuzzy particle swarm optimization (FPSO) recently. Abdelbar et al. proposed the FPSO [18]. Abdelbar and Abdelshahid brought forward the instinct-based particle swarm optimization with local search applied to satisfiability in [19]. Abdelshahid analyzed variations of particle swarm optimization and gave evaluation on maximum satisfiability; see [20]. Mendes et al. proposed a fully informed particle swarm in [21]. Bajpai and Singh studied the problem of fuzzy adaptive particle swarm optimization (FAPSO) for bidding strategy in uniform price spot market in [22]. Saber et al. attempted to solve the problem of unit commitment computation by FAPSO in [23]. Esmin studied the problem of generating fuzzy rules and fitting fuzzy membership functions using hybrid particle swarm optimization (HPSO); see [24, 25].

Computation in the PSO paradigm is based on a collection (called a swarm) of fairly primitive processing elements (called particles). The neighborhood of each particle is the set of particles with which it is adjacent. The two most common neighborhood structures are p_g , in which the entire swarm is considered a single neighborhood, and p_i , in which the particles are arranged in a ring, and each particle's neighborhood consists of itself, its immediate ring-neighbor to the right, and its immediate ring-neighbor to the left.

PSO can be used to solve a discrete combinatorial optimization problem whose candidate solutions can be represented as vectors of bits; i is supposed to be a given instance of such a problem. Let N denote the number of elements in the solution vector for i . Each particle i would contain two N -dimensional vectors: a Boolean vector x_i , which represents a candidate solution to i and is called particle i 's state, and a real vector v_i , called the velocity of the particle. In the biological insect-swarm analogy, the velocity vector represents how fast, and in which direction, the particle is flying for each dimension of the problem being solved.

Let $K(i)$ denote the neighbors of particle i , and let p_i denote the best solution ever found by particle i . In each time iteration, each particle i adjusts its velocity based on

$$v_{i+1} = \omega v_i + c_1 r_1 (p_i - x_i) + c_2 r_2 (p_g - x_i), \quad (19)$$

$$x_{i+1} = x_i + v_{i+1},$$

where ω , called inertia, is a parameter within the range $[0, 1]$ and is often decreased over time as discussed in [26]; c_1 and c_2 are two constants, often chosen so that $c_1 + c_2 = 4$, which control the degree to which the particle follows the herd thus stressing exploitation (higher values of c_2), or goes its own way thus stressing exploration (higher values of c_1); r_1 and r_2 are uniformly random number generator function that returns values within the interval $(0, 1)$; and p_g is the particle in i 's neighborhood with the current neighborhood-best candidate solution.

Fuzzy PSO differs from standard PSO in only one respect: in each neighborhood, instead of only the best particle in

the neighborhood being allowed to influence its neighbors, several particles in each neighborhood can be allowed to influence others to a degree that depends on their degree of charisma, where charisma is a fuzzy variable. Before building a model, there are two essential questions that should be answered. The first question is how many particles in each neighborhood have nonzero charisma. The second is what membership function (MF) will be utilized to determine level of charisma for each of the k selected particles.

The answer to the first question is that the k best particles in each neighborhood are selected to be charismatic, where k is a user-set parameter, k can be adjusted according to the required precision of the solution.

The answer to the other question is that there are numerous possible functions for charisma MF. Popular MF choices include triangle, trapezoidal, Gaussian, Bell, and Sigmoid MFs; see [27]. The Bell, Gaussian, sigmoid, Trapezoidal, and Triangular MFs

$$\text{bell}(x; c, \sigma) = \frac{1}{1 + ((x - c) / \sigma)^2},$$

$$\text{gaussian}(x; c, \sigma) = \exp\left(-\frac{1}{2}\left(\frac{x - c}{\sigma}\right)^2\right),$$

$$\text{sig}(x; c, \sigma) = \frac{1}{1 + \exp(-\sigma(x - c))},$$

$$\text{triangle}(x; a, b, c) = \max\left(\min\left(\frac{x - a}{b - a}, \frac{c - x}{c - b}\right), 0\right),$$

$$a < b < c,$$

$$\text{trapezoid}(x; a, b, c, d) = \max\left(\min\left(\frac{x - a}{b - a}, 1, \frac{d - x}{d - c}\right), 0\right),$$

$$a < b \leq c < d.$$

(20)

Let h be one of the k -best particles in a given neighborhood, and let $f(p_g)$ refer to the fitness of the very-best particle for the neighborhood under consideration. The charisma $\varphi(h)$ is defined as

$$\varphi(h) = \frac{1}{1 + ((f(p_h) - f(p_g)) / \beta)^2} \quad (21)$$

if the MF is based on Bell function.

The charisma $\varphi(h)$ is defined as

$$\varphi(h) = \exp\left(-\frac{1}{2}\left(\frac{f(p_h) - f(p_g)}{\beta}\right)^2\right) \quad (22)$$

if the MF is based on Gaussian function.

The charisma $\varphi(h)$ is defined as

$$\varphi(h) = \frac{1}{1 + \exp(-\beta(f(p_h) - f(p_g)))} \quad (23)$$

if the MF is based on Sigmoid function.

The charisma $\varphi(h)$ is defined as

$$\varphi(h) = \max\left(\min\left(\frac{f(p_h) - f(p_g^0)}{f(p_g) - f(p_g^0)}, \frac{f(p_g^1) - f(p_h)}{f(p_g^1) - f(p_g)}\right), 0\right) \quad (24)$$

if the MF is based on Triangle function.

The charisma $\varphi(h)$ is defined as

$$\varphi(h) = \max\left(\min\left(\frac{f(p_h) - f(p_g^0)}{f(p_g) - f(p_g^0)}, 1, \frac{f(p_g^1) - f(p_h)}{f(p_g^1) - f(p_g)}\right), 0\right) \quad (25)$$

if the MF is based on Trapezoid function.

Because $(p_h) \leq f(p_g)$, $\varphi(h)$ is a decreasing function that is 1 when $f(p_h) = f(p_g)$, and asymptotically approaches zero as $f(p_h)$ moves away from $f(p_g)$. To avoid dependence on the scale of the fitness function, β is defined as $\beta = f(p_g) / \ell$ where ℓ is a user-specified parameter. For a fixed $f(p_h)$, the larger the value of ℓ , the smaller the charisma $\varphi(h)$ will be. $f(p_g^0)$ and $f(p_g^1)$ are two functional values that decide the verge of the triangle and trapezoid function.

In Fuzzy PSO, velocity equation is

$$v_{i+1} = \omega v_i + c_1 r_1 (p_i - x_i) + \sum_{h \in B(i,k)} \varphi(h) c_2 r_2 (p_h - x_i), \quad (26)$$

where $B(i, k)$ denotes the k -best particles in the neighborhood of particle i . Each particle i is influenced by its own best solution p_i and the best solutions obtained by the k charismatic particles in its neighborhood, with the effect of each weighted by its charisma $\varphi(h)$. It can be seen that if k is 1, this model reduces to the standard PSO model.

4.2. Hybrid Fuzzy PSO. Hybrid rule requires selecting two particles from the alternative particles at a certain rate. Then the intersecting operation work needs to be done to generate the descendant particles. The positions and velocities of the descendant particles are as follows: according to the intersecting rule, inheriting from the FPSO, see [28]

$$v_{i+1}^1 = \omega v_i^1 + c_1 r_1 (p_i - x_i^1) + \sum_{h \in B(i,k)} \varphi(h) c_2 r_2 (p_h^1 - x_i^1),$$

$$v_{i+1}^2 = \omega v_i^2 + c_1 r_1 (p_i - x_i^2) + \sum_{h \in B(i,k)} \varphi(h) c_2 r_2 (p_h^2 - x_i^2),$$

$$v_{i+1}^1 = \frac{v_{i+1}^1 + v_{i+1}^2}{|v_{i+1}^1 + v_{i+1}^2|} |v_{i+1}^1|,$$

$$v_{i+1}^2 = \frac{v_{i+1}^1 + v_{i+1}^2}{|v_{i+1}^1 + v_{i+1}^2|} |v_{i+1}^2|,$$

$$x_{i+1}^1 = x_i^1 + v_{i+1}^1,$$

$$x_{i+1}^2 = x_i^2 + v_{i+1}^2,$$

$$x_{i+1}^1 = p \times x_{i+1}^1 + (1 - p) \times x_{i+1}^2,$$

$$x_{i+1}^2 = p \times x_{i+1}^2 + (1 - p) \times x_{i+1}^1. \quad (27)$$

Thus, the HFPSO is built. In the model, x is a position vector with D dimensions. x_i^j stands for the particle j of the i th generation particles. p is a random variable vector with D dimensions which obeys the equal distribution. Each dimension of p is in $[0, 1]$.

4.3. Adaptability and Applicability HFPSO. It can be seen that the network timetable stability optimizing model is a nonlinear one and it is an NP-hard problem. Generally, it is very difficult to solve the problem with mathematical approaches. Evolutionary algorithms are often hired to solve the problem for their characteristics. First, its rule of the algorithm is easy to apply. Second, the particles have the memory ability which results in convergent speed and there are various methods to avoid the local optimum. Thirdly, the parameters which need to select are fewer, and there is considerable research work on the parameters selecting.

In addition, the HFPSO hires the fuzzy theory and the hybrid handling method when designing the algorithm. Thus it has the ability to improve the computing precision when solving the optimization problem. And it utilizes intersecting tactics to generate the new generation of particles to avoid the precipitate of the solution. It is adaptive to solve the timetable stability optimization. And its easy computing rule determines the applicability in the solving of timetable stability optimization problem.

5. Hybrid Fuzzy Particle Swarm Algorithm for the Model

5.1. The Particle Swarm Design for the Upper Level Programming. The size of the particle swarm is set to be 30 to give consideration to both the calculating degree of accuracy and computational efficiency. For F_i and G_i are the decision variables, L is the number of the sections on the railway network. K is the number of the stations on the railway network. So a particle can be designed as

$$pa_{UP} = \{F_1, F_2, \dots, F_K, G_1, G_2, \dots, G_L\}. \quad (28)$$

5.2. The Particle Swarm Design for the Lower Level Programming. The size of the particle swarm is also set to be 30. $\Delta t_{R,i}^p$ and $\Delta t_{D,i}^q$ are the decision variables. So a particle can be designed as

$$pa_{LO} = \left\{ \begin{aligned} &\Delta t_{R,1}^1, \Delta t_{R,2}^1, \dots, \Delta t_{R,M_1}^1, \\ &\Delta t_{R,1}^2, \Delta t_{R,2}^2, \dots, \Delta t_{R,M_2}^2, \dots \\ &\Delta t_{R,1}^i, \Delta t_{R,2}^i, \dots, \Delta t_{R,M_i}^i, \dots \\ &\Delta t_{R,1}^L, \Delta t_{R,2}^L, \dots, \Delta t_{R,M_L}^L, \\ &\Delta t_{D,1}^1, \Delta t_{D,2}^1, \dots, \Delta t_{D,M_1}^1, \\ &\Delta t_{D,1}^2, \Delta t_{D,2}^2, \dots, \Delta t_{D,M_2}^2, \dots \\ &\Delta t_{D,1}^j, \Delta t_{D,2}^j, \dots, \Delta t_{D,M_j}^j, \dots \\ &\Delta t_{D,1}^K, \Delta t_{D,2}^K, \dots, \Delta t_{D,M_K}^K \end{aligned} \right\}, \quad (29)$$

where M_i is the number of trains going through section i and M_j is the number of the trains going through station j .

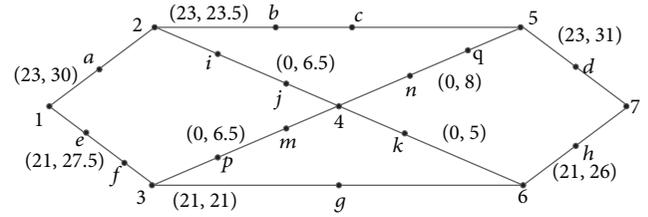


FIGURE 1: The railway network in the computing case and the original distribution plan of the trains.

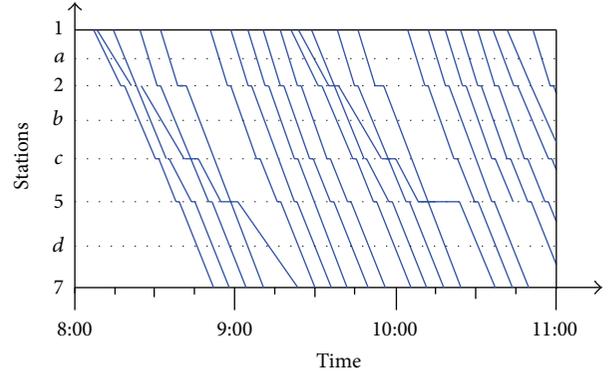


FIGURE 2: Planned operation diagram on path 1-2-5-7.

6. Experimental Results and Discussion

6.1. Computing Case Assumptions. There are 22 stations and 10 sections in the network, as indicated in Figure 1. We assume that the 44 trains operate on the network from station 1 to station 7. The numbers in parentheses beside the edges are the numbers of the trains distributed on the sections according to the planned timetable and the section capacity.

And the planned operation diagram on path 1-2-5-7 is shown in Figure 2, with 23 trains. The planned operation diagram on path 1-3-6-7 is illustrated in Figure 3.

According to the networked timetable stability definition in Section 3, the timetable stability value of the planned timetable can be calculated out. Detailed computing data are presented in Table 1.

According to Table 1(a), the Z_{ST} can be calculated with $Z_{ST} = \text{Var}(e^{-\rho_i} w_{ST,i}^{\rho_i - \rho}) = 1.047978$, while Z_{SC} can be calculated out with $Z_{SE} = \text{Var}(e^{-\lambda_i} w_i^{\lambda_i - \lambda}) = 7.1940742$. Then the goal of the upper programming $S_{NET} = e^{-Z_{ST}} \times e^{-Z_{SE}} = e^{-(Z_{ST} + Z_{SE})} = 0.000263$.

6.2. The Upper Level Computing Results and Discussion. We reallocate all the 44 trains on the modest railway network, as shown in Figure 4, according to the computing results of the upper level programming. The numbers in parentheses beside the edges are the numbers of the trains allocated on the sections according to the rescheduled timetable and the section capacity. Based on the capacity of every section and station, the computing result of the upper level programming is presented in Table 2.

TABLE 1: Computing results of the planned timetable stability.

(a) Related computing results of Z_{ST}

Key nodes	Trains number through station	Nodes capacity	Load	Capacity index	Nodes degree	Degree index	Stations weight
1	44	66.0	0.6667	0.3099	2.0000	0.1111	0.2245
2	23	34.5	0.6667	0.1620	3.0000	0.1667	0.1760
3	21	31.5	0.6667	0.1479	3.0000	0.1667	0.1607
4	0	15.0	0.0000	0.0704	4.0000	0.2222	0.1020
5	23	36.0	0.6389	0.1690	3.0000	0.1667	0.1837
6	21	30.0	0.7000	0.1408	3.0000	0.1667	0.1531

(b) Related computing results of Z_{SC}

Section	Trains number through section	Sections capacity	Load	Capacity index	Sections weight
1-2	23	30.0	0.7667	0.1622	0.1622
1-3	21	27.5	0.7636	0.1486	0.1486
2-4	0	6.5	0.0000	0.0351	0.0351
2-5	23	23.5	0.9787	0.1270	0.1270
3-4	0	6.5	0.0000	0.0351	0.0351
3-6	21	21.0	1.0000	0.1135	0.1135
4-5	0	8.0	0.0000	0.0432	0.0432
4-6	0	5.0	0.0000	0.0270	0.0270
5-7	23	31.0	0.7419	0.1676	0.1676
6-7	21	26.0	0.8077	0.1405	0.1405

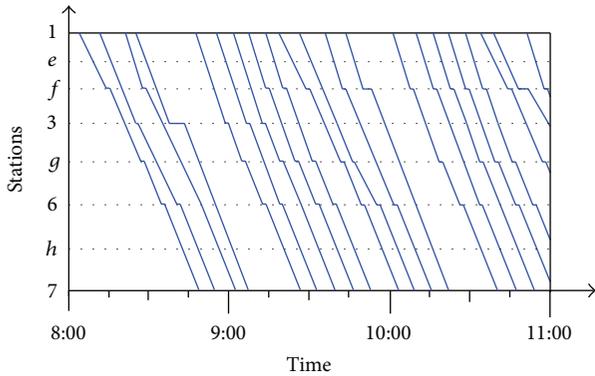


FIGURE 3: Planned operation diagram on path 1-3-6-7.

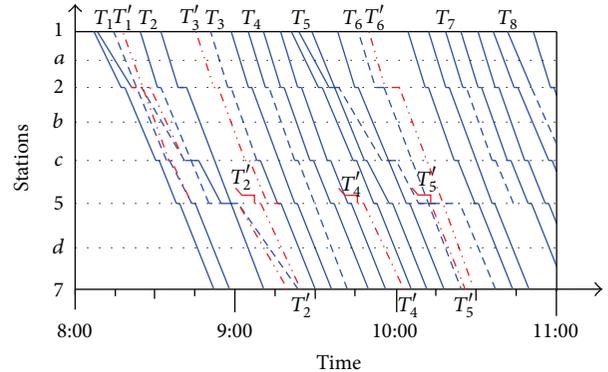


FIGURE 5: Rescheduled timetable of path 1-2-5-7 according to the computing results of the lower level programming.

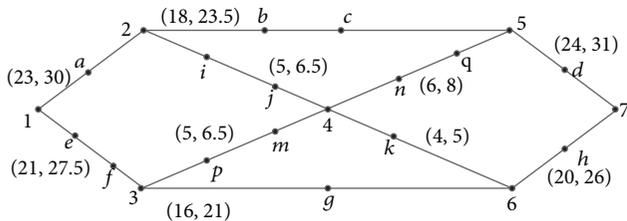


FIGURE 4: Distribution plan of the trains according to the computing results.

According to Table 2(a), the Z_{ST} can be calculated with $Z_{ST} = \text{Var}(e^{-\rho_i} w_{ST,i}^{\rho_i - \rho}) = 0.000223814$, while Z_{SC} can be calculated out with $Z_{SE} = \text{Var}(e^{-\lambda_i} w_i^{\lambda_i - \lambda}) = 0.002432614$. Then the goal of the upper programming $S_{NET} = e^{-Z_{ST}} \times e^{-Z_{SE}} = e^{-(Z_{ST} + Z_{SE})} = 0.997$.

6.3. *The Lower Level Computing Results and Discussion.* According to the computing results of the lower level programming, we adjust the timetable, moving some of the running lines of the trains. The two-dot chain line is the newly planned running trajectory and the dotted line is the previously planned trajectory. The timetable on path 1-2-5-7 is shown in Figure 5. Figure 6 is the timetable of path 2-4-5. Figures 7 and 8 are the timetables on paths 1-3-6-7 and 3-4-6, respectively.

From Figures 5 and 6 we can see that eight trains are rescheduled on path 1-2-5-7, which are numbered $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8$. T_1, T_3 , and T_6 run on the original path as planned, but the inbound and outbound times at the stations are changed. T_2, T_4, T_5, T_7 , and T_8 modify the path when they arrive at station 2. They run through path 2-4-5, with arriving time 8:32, 9:12, 9:41, 10:25, and 10:48, respectively at station 2.

TABLE 2: Computing results of rescheduled timetable stability.

(a) Related computing results of Z_{ST}

Key nodes	Trains number through station	Nodes capacity	Load	Capacity index	Nodes degree	Degree index	Stations weight
1	44	66.0	0.6667	0.3099	2	0.1111	0.2245
2	23	34.5	0.6667	0.1620	3	0.1667	0.1760
3	21	31.5	0.6667	0.1479	3	0.1667	0.1607
4	10	15.0	0.6667	0.0704	4	0.2222	0.1020
5	24	36.0	0.6667	0.1690	3	0.1667	0.1837
6	20	30.0	0.6667	0.1408	3	0.1667	0.1531

(b) Related computing results of Z_{SC}

Section	Trains number through section	Sections capacity	Load	Capacity index	Sections weight
1-2	23	30.0	0.7667	0.1622	0.1622
1-3	21	27.5	0.7636	0.1486	0.1486
2-4	5	6.5	0.7692	0.0351	0.0351
2-5	18	23.5	0.7660	0.1270	0.1270
3-4	5	6.5	0.7692	0.0351	0.0351
3-6	16	21.0	0.7619	0.1135	0.1135
4-5	6	8.0	0.7500	0.0432	0.0432
4-6	4	5.0	0.8000	0.0270	0.0270
5-7	24	31.0	0.7742	0.1676	0.1676
6-7	20	26.0	0.7692	0.1405	0.1405

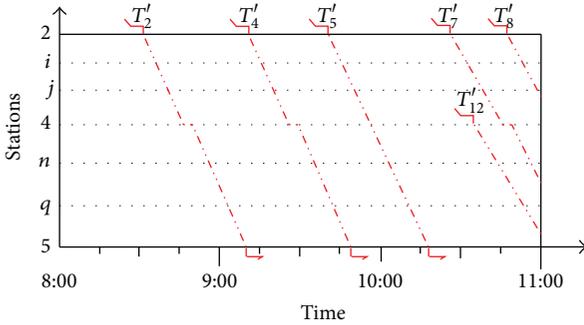


FIGURE 6: Rescheduled timetable of path 2-4-5 according to the computing results of the lower level programming.

T_2 , T_4 , and T_5 leave arrive station 5 at 9:10, 9:48, and 10:18, respectively, then finish the rest travel along Sections 5-7.

From Figures 7 and 8 we can see that eight trains are rescheduled on path 1-3-6-7, which are numbered T_9 , T_{10} , T_{11} , T_{12} , T_{13} , T_{14} , T_{15} , T_{16} . T_{14} , T_{15} , and T_{16} run on the original path as planned, but the inbound and outbound time at the stations are changed. T_9 , T_{10} , T_{11} , T_{12} , and T_{13} change the path when they arrive at station 3. T_9 , T_{10} , T_{11} , and T_{13} run along path 3-4-6 with the arriving time 8:22, 9:16, 9:44, and 10:43 at station 3, respectively. T_9 , T_{10} , and T_{11} arrive at station 6 at 8:53, 9:47, and 10:15, respectively. T_{12} runs along the path 3-4-5. It arrives at station 3 at 10:30 and at station 4 at 10:42. From Figure 4 we can see that it runs along path 4-5 to finish the rest travel.

The timetable stability on the dispatching section level is $S_{DIS} = e^{-\text{Var}(A_R^p) \times \text{Var}(A_D^q)} = e^{-\text{Var}(\Delta t_{R,i}^p / t_{R,i}^p) \times \text{Var}(\Delta t_{D,i}^q / t_{D,i}^q)} = 0.879$.

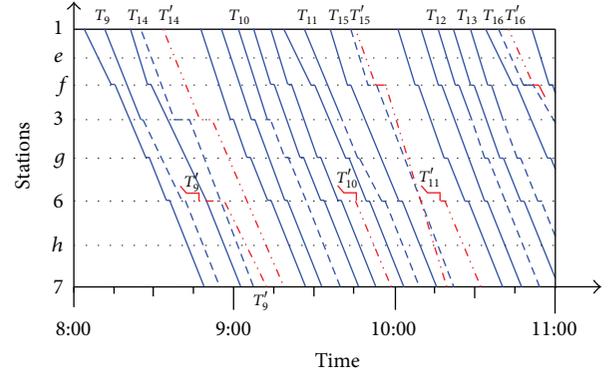


FIGURE 7: Rescheduled timetable of path 1-3-6-7 according to the computing results of the lower level programming.

Then, the networked timetable is $S = S_{NET} \times S_{DIS} = 0.997 \times 0.879 = 0.876$.

7. Conclusion

The bilevel programming model is appropriate for the networked timetable stability optimizing. It comprises the timetable stability of the network level and the dispatching section level. Better solution can be attained via hybrid fuzzy particle swarm algorithm in networked timetable optimizing. The timetable is more stable, which means that it is more feasible for rescheduling in the case of disruption, when it is optimized by hybrid fuzzy particle swarm algorithm. The timetable rearranged based on the timetable stability with bilevel networked programming model can make the

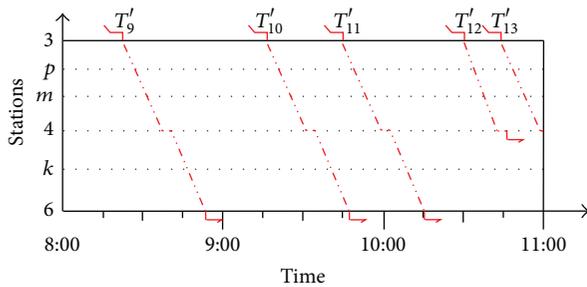


FIGURE 8: Rescheduled timetable of path 3-4-6 according to the computing results of the lower level programming.

real train movements very close to, if not the same with, the planned schedule, which is very practical in the daily dispatching work.

The results also show that hybrid fuzzy particle algorithm has significant global searching ability and high speed and it is very effective to solve the problems of timetable stability optimizing. The novel method described in this paper can be embedded in the decision support tool for timetable designers and train dispatchers.

We can do some microcosmic research work on the timetable optimizing based on the railway network in the future based on the method set out in the present paper, enlarging the research field, adding the inbound time, and outbound time of the trains at stations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is financially supported by National Natural Science Foundation of China (Grant 61263027), Fundamental Research Funds of Gansu Province (Grant 620030), and New Teacher Project of Research Fund for the Doctoral Program of Higher Education of China (20126204120002). The authors wish to thank anonymous referees and the editor for their comments and suggestions.

References

- [1] H. Guo, W. Wang, W. Guo, X. Jiang, and H. Bubb, "Reliability analysis of pedestrian safety crossing in urban traffic environment," *Safety Science*, vol. 50, no. 4, pp. 968–973, 2012.
- [2] W. Wang, W. Zhang, H. Guo, H. Bubb, and K. Ikeuchi, "A safety-based approaching behavioural model with various driving characteristics," *Transportation Research Part C*, vol. 19, no. 6, pp. 1202–1214, 2011.
- [3] W. Wang, Y. Mao, J. Jin et al., "Driver's various information process and multi-ruled decision-making mechanism: a fundamental of intelligent driving shaping model," *International Journal of Computational Intelligence Systems*, vol. 4, no. 3, pp. 297–305, 2011.
- [4] R. M. P. Goverde, "Max-plus algebra approach to railway timetable design," in *Proceedings of the 6th International Conference on Computer Aided Design, Manufacture and Operation in the Railway and Other Advanced Mass Transit Systems*, pp. 339–350, September 1998.
- [5] M. Carey and S. Carville, "Testing schedule performance and reliability for train stations," *Journal of the Operational Research Society*, vol. 51, no. 6, pp. 666–682, 2000.
- [6] I. A. Hansen, "Station capacity and stability of train operations," in *Proceeding of the 7th International Conference on Computers in Railways. Computers in Railways VII*, pp. 809–816, Bologna, Italy, September 2000.
- [7] A. F. De Kort, B. Heidergott, and H. Ayhan, "A probabilistic (max, +) approach for determining railway infrastructure capacity," *European Journal of Operational Research*, vol. 148, no. 3, pp. 644–661, 2003.
- [8] R. M. P. Goverde, "Railway timetable stability analysis using max-plus system theory," *Transportation Research Part B*, vol. 41, no. 2, pp. 179–201, 2007.
- [9] X. Meng, L. Jia, and Y. Qin, "Train timetable optimizing and rescheduling based on improved particle swarm algorithm," *Transportation Research Record*, no. 2197, pp. 71–79, 2010.
- [10] O. Engelhardt-Funke and M. Kolonko, "Analysing stability and investments in railway networks using advanced evolutionary algorithm," *International Transactions in Operational Research*, vol. 11, no. 4, pp. 381–394, 2004.
- [11] R. M. P. Goverde, *Punctuality of railway operations and timetable stability analysis [Ph.D. thesis]*, TRAIL Research School, Delft University of Technology, Delft, The Netherlands, 2005.
- [12] M. J. Vromans, *Reliability of railway systems [Ph.D. thesis]*, TRAIL Research School, Erasmus University Rotterdam, Rotterdam, The Netherlands, 2005.
- [13] X. Delorme, X. Gandibleux, and J. Rodriguez, "Stability evaluation of a railway timetable at station level," *European Journal of Operational Research*, vol. 195, no. 3, pp. 780–790, 2009.
- [14] X. Meng, L. Jia, J. Xie, Y. Qin, and J. Xu, "Complex characteristic analysis of passenger train flow network," in *Proceedings of the Chinese Control and Decision Conference (CCDC '10)*, pp. 2533–2536, Xuzhou, China, May 2010.
- [15] X.-L. Meng, L.-M. Jia, Y. Qin, J. Xu, and L. Wang, "Calculation of railway transport capacity in an emergency based on Markov process," *Journal of Beijing Institute of Technology*, vol. 21, no. 1, pp. 77–80, 2012.
- [16] X.-L. Meng, L.-M. Jia, C.-X. Chen, J. Xu, L. Wang, and J.-X. Xie, "Paths generating in emergency on China new railway network," *Journal of Beijing Institute of Technology*, vol. 19, no. 2, pp. 84–88, 2010.
- [17] L. Wang, L.-M. Jia, Y. Qin, J. Xu, and W.-T. Mo, "A two-layer optimization model for high-speed railway line planning," *Journal of Zhejiang University: Science A*, vol. 12, no. 12, pp. 902–912, 2011.
- [18] A. M. Abdelbar, S. Abdelshahid, and D. C. Wunsch II, "Fuzzy PSO: a generalization of particle swarm optimization," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '05)*, pp. 1086–1091, Montreal, Canada, August 2005.
- [19] A. M. Abdelbar and S. Abdelshahid, "Instinct-based PSO with local search applied to satisfiability," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 2291–2295, July 2004.
- [20] S. Abdelshahid, *Variations of particle swarm optimization and their experimental evaluation on maximum satisfiability [M.S. thesis]*, Department of Computer Science, American University in Cairo, May 2004.

- [21] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
- [22] P. Bajpai and S. N. Singh, "Fuzzy adaptive particle swarm optimization for bidding strategy in uniform price spot market," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2152–2160, 2007.
- [23] A. Y. Saber, T. Senjyu, A. Yona, and T. Funabashi, "Unit commitment computation by fuzzy adaptive particle swarm optimisation," *IET Generation, Transmission and Distribution*, vol. 1, no. 3, pp. 456–465, 2007.
- [24] A. A. A. Esmin, "Generating Fuzzy rules from examples using the particle swarm optimization algorithm," in *Proceedings of the 7th International Conference on Hybrid Intelligent Systems (HIS '07)*, pp. 340–343, September 2007.
- [25] A. A. A. Esmin and G. Lambert-Torres, "Fitting fuzzy membership functions using hybrid particle swarm optimization," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 2112–2119, Vancouver, Canada, July 2006.
- [26] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, May 1998.
- [27] J. S. R. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, 1996.
- [28] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 84–89, May 1998.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

