

# An Automated Test Data Generation Method for On-board Subsystem

Ning Xu, Xiao-Yu Zhao\*, Yi-Nan Li, Yu Zhang, Ya-Qing Liu, Fei Wang

**Abstract**—On-board subsystem is a key of guaranteeing traffic security and enhancing operational efficiency for high-speed railway (HSR). Thus, the test of the subsystem is the main means of ensuring both the functionality and reliability of the train control system. To deal with the low efficiency of existing manual mode, an automated test data generation method combining back propagation neural network (BPNN) and multi-population genetic algorithm (MPGA) is presented. First, colored petri net (CPN) models are designed according to the relevant requirements specifications of the on-board subsystem. The feedback information of each model is taken as the fitness function. Second, the BPNN method is adopted to simulate the solving process of individual fitness value, which is to reduce the running cost of the designed model. Third, to overcome the shortcomings of genetic algorithm (GA), the MPGA is used to search for test data in the input space. Finally, we take the startup procedure of the on-board subsystem as a case study. The results reveal that the novel method has the advantages of short running time, higher efficiency, and better stability compared with the traditional GA.

**Index Terms**—Train control system, on-board subsystem, test data generation, BPNN, MPGA.

## I. INTRODUCTION

THE train control system, which ensures operation security and efficiency of trains in high-speed railway (HSR) lines, is a classic safety-critical system (SCS). In order to guarantee the safety, reliability and stability of the SCS, it must be thoroughly tested to avoid immeasurable catastrophic consequences [1], [2]. In this paper, the Chinese Train Control System Level 3 (CTCS-3) is the research object. The CTCS-3 system realizes the bi-directional track-train communication by using the Global System for Mobile Communications-Railway (GSM-R). The CTCS-3 system

Manuscript received September 21, 2022; revised May 16, 2023. This work was supported by the Foundation of China Academy of Railway Sciences Co., Ltd. (Grant No. 2021YJ085), the Foundation of China State Railway Grope Co., Ltd. (Grant No. J2022G005) and the Foundation of Signal & Communication Research Institute (Grant No. 2022HT04).

Ning Xu is a professor of Signal & Communication Research Institute, China Academy of Railway Sciences Co., Ltd., Beijing 100081, China (e-mail: jsxuning@aliyun.com).

Xiao-Yu Zhao is an associate professor of Signal & Communication Research Institute and also of Simulation Testing Laboratory for Train Control System Engineering of High-speed Railway, China Academy of Railway Sciences Co., Ltd., Beijing 100081, China (\*corresponding author, phone: +8615011267969; e-mail: zhaoxiaoyu059@126.com).

Yi-Nan Li is a professor of Signal & Communication Research Institute, China Academy of Railway Sciences Co., Ltd., Beijing 100081, China (e-mail: liyinan\_cars@sina.com).

Yu Zhang is an engineer of Signal & Communication Research Institute and also of Simulation Testing Laboratory for Train Control System Engineering of High-speed Railway, China Academy of Railway Sciences Co., Ltd., Beijing 100081, China (e-mail: bjjuzhang@126.com).

Ya-Qing Liu is a research associate of Signal & Communication Research Institute, China Academy of Railway Sciences Co., Ltd., Beijing 100081, China (e-mail: alidaidaia@163.com).

Fei Wang is a research associate of Signal & Communication Research Institute, China Academy of Railway Sciences Co., Ltd., Beijing 100081, China (e-mail: wangfeisunlight@126.com).

must satisfy the relevant requirements specifications [3], [4], and the test cases [5] are suitable for functional test and interoperability test of the CTCS-3 system. There are two subsystems in the CTCS-3 system, i.e., the on-board subsystem and the track-side subsystem. The on-board subsystem refers specifically to the Automatic Train Protection (ATP). The ATP is a computer-based system that exchanges information with the track-side subsystem to supervise the movement of its train. As the system structure and functional logic of ATP become more complex, the limitations of existing test methods are more obvious. The existing test methods require manual preparation of test sequences, preparation of test data, and verification of test results. The methods also require testers to have high professional knowledge and practical experience. It takes too much time and high cost to design the corresponding test data manually. Therefore, the test data generation for ATP is studied in detail in the paper to improve test efficiency.

Model-based testing (MBT) [6], [7] is a hot method in the field of automated testing and can solve the above problems well. The MBT-based test data generation problem is described below. Given the test target path of a model, when the test data searched in the input space are taken as the input data, the path traversed happens to be the test target path. By designing a suitable fitness function, we can replace the above-mentioned problem with a mathematical function optimization problem. The meta-heuristic search algorithm is employed to settle the above optimization problem and generate test data that meets the test coverage. There are many used meta-heuristic search algorithms, such as genetic algorithm (GA) [8], particle swarm optimization algorithm (PSO) [9], and ant colony optimization algorithm (ACO) [10]. Among the above meta-heuristic search algorithms, the GA-based method with test case generation has received extensive attention from scholars. However, considering the problem of MBT-based test data generation for the on-board subsystem, the GA has the following problems.

- The traditional GA is prone to precocious state, and there is blindness and randomness in the search process.
- Considering each set of test data, we need to spend a lot of time to run the model to get individual fitness value, reducing the generation efficiency.

To solve the problems above, an automated test data generation method based on back propagation neural network (BPNN) and multi-population genetic algorithm (MPGA) for the on-board subsystem is proposed. The main contributions of our work are described as below.

- In order to make up for the deficiencies of the traditional GA, the MPGA is applied to realize the multi-population parallel search, that is, the test data is searched in the input space of colored petri net (CPN)

model.

- The feedback information of the CPN model is taken as the fitness function. The BPNN method is adopted to simulate the solving process of individual fitness value, which shortens the running time of the CPN model. The above methods will greatly reduce the preparation time of test data.

The rest of this paper is arranged as below. Section II reviews the literatures about test methods. Section III describes the proposed novel method in detail, including mathematical model of test data generation problem, CPN executable model design, BPNN design, and MPGA design. Section IV presents experiments to analyze the performance of the proposed novel method. The paper is concluded and further research directions are given in Section V.

## II. RELATED WORK

It is necessary to conduct laboratory simulation tests, field tests, inter-operability tests, and other safety tests for the on-board subsystem before formal operation. As for test methodologies, most scholars focus on test case generation, test sequence optimization, fault test, etc. In terms of the test case generation, we generally get test cases by functional features [11] or MBT-based methods [12], [13]. Rao [14] proposed a combinatorial test suite generation approach based on the input parameter features of the on-board subsystem. For Target Speed Monitor (TSM) profile of the on-board subsystem, Zheng [15] presented a complete test case generation approach by using equivalence partitioning. Dou [16] used the scene method to design test cases in view of the level transition function of the on-board subsystem. Gu [17] designed test cases based on the obtained operation profile for the on-board subsystem. On the test case optimization side, Dou [18] put forward a test sequence optimization method based on simulated annealing (SA) optimization improved circle (IC) algorithm. Considering the functional requirements specification, Zheng [19] developed an automated test sequence optimization technology for the train control system. Zhao [20] adopted the Chinese postman algorithm to optimize test sequences and raised a multi-objective test sequence set generation strategy [21] with improved max-min ant colony algorithm (IMMAC). In the field of fault test, Hao [22] designed a fault injection test method for the actual equipment of the train control system. Combining time automata with input and output (TAIO) and mutation analysis method, Wei [23] put forward an automated test suite generation method for the CTCS-3 system. However, none of the above studies involve the test data generation.

The MBT realizes system testing by designing models, designing test cases, and executing the test cases. Different from other modeling methods, such as finite state machine (FSM) [24] and Unified Modeling Language (UML) [25], CPN [26], [27] has intuitive graphical description and rigorous mathematical foundation. Base on hierarchical modeling and data processing functions, CPN can accurately describe the synchronous, parallel, and asynchronous behavior of any complex system. Zhou [28] adopted UML activity diagram and hierarchical timed CPN (HTCPN) to model the level transition process of the CTCS-3 system. Wang [29] established a CPN-based level transition model, and studied

the real-time performance of mode transition under different block mode-communication (BM) design operation intervals. Here, we use CPN to design models of the on-board subsystem according to the relevant requirements specifications [3], [4].

The GA is an effective method to automatically generate test data. Shi [30] established an automated test data generation tool with dynamic variable parameters by the theory and characteristics of GA. Pan [31] introduced a GA-based method that can automatically identify irrelevant input variables from the test path. Aiming at improving the efficiency of test data generation, Fan [32] introduced the concept of program balance based on GA to optimize the evolutionary process of individuals. Xia [33] designed a test data generation method based on the negative selection GA. Considering maximum difference of population paths, Avdeenok [34] designed a novel fitness function based on GA with an additional component, as so to search for the optimal test data set for multi-path coverage. Esnaashari [35] took reinforce learning as a memetic algorithm (MA) and extended the algorithm to GA, which was to enhance generation efficiency. However, the test data generation based on GA takes too much time due to the repeated running of the test model. Therefore, the BPNN method is adopted here to simulate the solving process of individual fitness value.

## III. AUTOMATED TEST DATA GENERATION METHOD

The proposed method consists of three parts, including the CPN modeling part, the BPNN part, and the MPGA part, as shown in Fig. 1.

The detailed processes are presented as below.

- The CPN models of the on-board subsystem are designed according to the relevant requirements specifications [3], [4] and the CPN modeling rules [21], [27]. A certain number of input data is randomly generated, and the corresponding fitness values are obtained by running the CPN models.
- The BPNN method is introduced to simulate the solving process of individual fitness value, and the training of BPNN is carried out through the input data and corresponding fitness values.
- The MPGA is used to search for test data in the input space. When the conditions are met, each subpopulation shares information through migration strategy. Otherwise, each subpopulation performs genetic operations to form a new subpopulation, and the MPGA loops through the parallel search until the input data is searched or the cutoff condition is met.

### A. Mathematical Model of Test Data Generation Problem

The generation problems with test data are usually described as function optimization problems, where common fitness functions [36] are layer proximity degree and branch distance. In this paper, the dynamic execution of the model is realized by a parser. The feedback information is obtained dynamically to calculate the fitness value and guide the automated test data generation.

The fitness function proposed in [37] is taken as the fitness function in the paper. Hypothetically, the test target path is  $P$ , the path length of  $P$  is  $L_P$ , the test data is

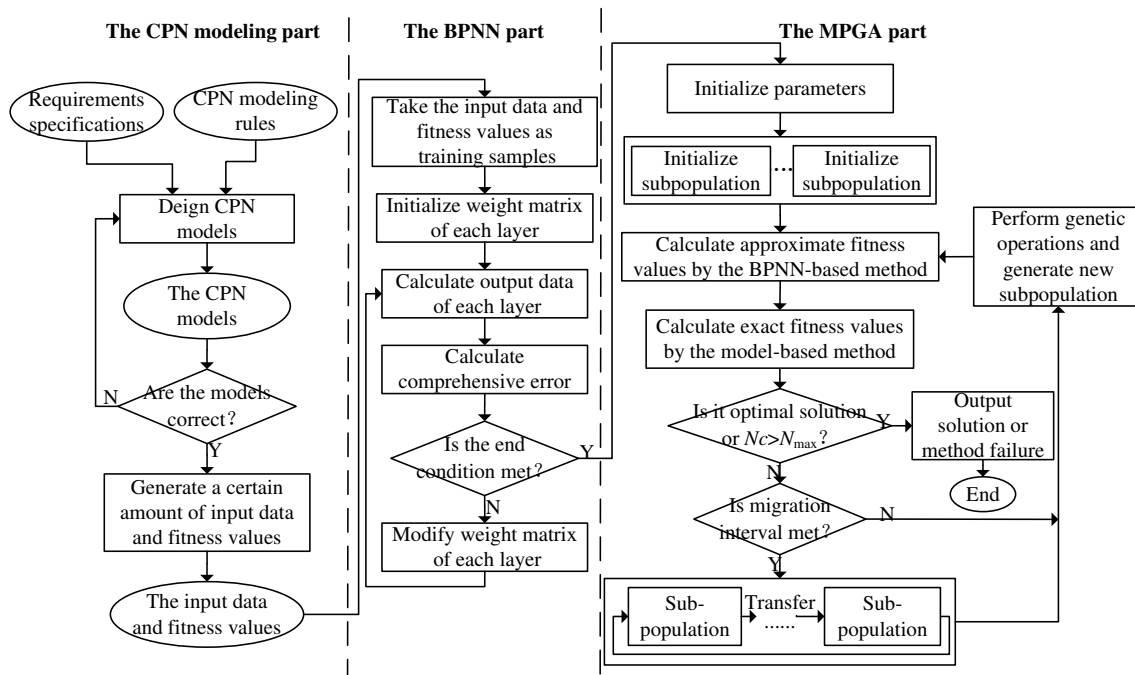


Fig. 1. Flow chart of the automated test data generation.

$\mathbf{X} = (x_1, x_2, \dots, x_n)^T$ , and the path length of the path triggered sequentially from the initial transition by  $\mathbf{X}$  is  $L_{SP}$ . The fitness function is

$$F(\mathbf{X}) = \frac{L_{SP}}{L_P} \times 100 \quad (1)$$

This fitness function does not involve the calculation error caused by the data type and the calculation cost is relatively small. The larger the fitness value, the more transitions are covered in the test target path. When the fitness value is 100, it means that the test data can trigger all transitions of the test target path sequentially, i.e., the test data can completely cover the test target path.

### B. CPN Executable Model Design

In this paper, a parser is developed to analyze the guard function, executive function, and arc expression of CPN model. For different test data, the parser supports automated execution of the model without manually modifying the token value of the place. It enables the model to read the input data automatically, run the model automatically, and calculate (1) based on the feedback information. The guard function, executive function and arc expression are all string formats with on program semantic information. Therefore, the key to model design is to analyze and execute the model's expressions. If the guard function is parsed and verified correctly, the transition is performed automatically. If the transition is triggered, the executive function is parsed and run automatically, and the tokens enter subsequent places.

### C. BPNN Design

In the paper, we adopt the BPNN method to simulate the solving process of the fitness function as (1). Firstly,

we randomly select a certain amount of input data, and run the CPN model to obtain the corresponding fitness values. Secondly, we use the input data and fitness values as samples to train the BPNN, thereby reducing the running time of the CPN model.

#### 1) Construction of BPNN:

According to (1), we design a BPNN with  $n$  input variables, one output variable, and a hidden layer with  $l$  neuron. The input vector of the input layer (IL)  $\mathbf{X} = (x_1, x_2, \dots, x_n)^T$  corresponds to the input data, the output vector of the output layer (OL)  $y$  corresponds to the fitness value, the output vector of the hidden layer (HL) is  $\mathbf{O} = (o_1, o_2, \dots, o_l)^T$ , and the weight matrixes from the IL to the HL and from HL to the OL are  $\mathbf{V} = [v_{ij}]$  and  $\mathbf{W} = (w_1, w_2, \dots, w_n)^T$ , respectively.

For the BPNN designed in the paper, the following principles are specified.

- The number of nodes of the HL affects the learning efficiency and generalization ability. The number of nodes here is set as  $l = \sqrt{\mu + \zeta} + r$ , where  $\mu$  is the number of nodes of the IL,  $\zeta$  is the number of nodes of the OL, and  $r$  is a random integer in the range [1-10].
- The number of training samples is about 10 times that of the network connection weight.
- In order to make the training samples more representative, the selected test input data should be evenly distributed in the input field.
- The input data needs to be normalized to eliminate the impact of different input data on BPNN.

The normalized function is

$$x_n = \frac{x_n - x_{\min}}{x_{\max} - x_{\min}} \quad (2)$$

where  $x_{\min} = \min(x_n)$ ,  $x_{\max} = \max(x_n)$ .

### 2) Output of Layer Neuron:

The input and output of the  $j$ -th neuron of the HL are described as below

$$net_j = \sum_{i=1}^n v_{ij}x_i \quad (3)$$

$$o_j = f(net_j) = f\left(\sum_{i=1}^n v_{ij}x_i\right) \quad (4)$$

where  $x_i$  is the input of the  $i$ -th neuron of the IL, and  $v_{ij}$  is the weight from the  $i$ -th neuron of the IL to the  $j$ -th neuron of the HL.

The input and output of the OL are described as below

$$r = \sum_{j=1}^l w_j o_j \quad (5)$$

$$y = f(r) = f\left(\sum_{j=1}^l w_j o_j\right) \quad (6)$$

where  $w_j$  is the weight from the  $j$ -th neuron of the HL to the OL.

The activation function used here is

$$f(u) = \frac{1}{1 + e^{-u}}. \quad (7)$$

### 3) Modification of Weights of Each Layer:

For the training sample  $(\mathbf{X}, d)$ ,  $\mathbf{X}$  notes the input data and  $d$  notes the fitness value expected here. Suppose that  $y$  notes the actual output, the output error is described as below

$$E = \frac{1}{2}(y - d)^2 \quad (8)$$

Simplifying (8) with (4) and (6), we can get

$$\begin{aligned} E &= \frac{1}{2} \left[ f\left(\sum_{j=1}^l w_j o_j\right) - d \right]^2 \\ &= \frac{1}{2} \left\{ f\left[\sum_{j=1}^l w_j f\left(\sum_{i=1}^n v_{ij}x_i\right)\right] - d \right\}^2 \end{aligned} \quad (9)$$

For the training samples  $\{(\mathbf{X}^1, d^1), \dots, (\mathbf{X}^z, d^z)\}$ , the comprehensive error is

$$E_z = \sum_{k=1}^z E^k = \frac{1}{2} \sum_{k=1}^z (y^k - d^k)^2 \quad (10)$$

Simplifying (10) with (4) and (6), we can get

$$\begin{aligned} E_z &= \frac{1}{2} \sum_{k=1}^z \left[ f\left(\sum_{j=1}^l w_j o_j^k\right) - d^k \right]^2 \\ &= \frac{1}{2} \sum_{k=1}^z \left\{ f\left[\sum_{j=1}^l w_j f\left(\sum_{i=1}^n v_{ij}x_i^k\right)\right] - d^k \right\}^2 \end{aligned} \quad (11)$$

The weight correction equations [38] from the IL to the HL and from the HL to the OL are described as below, respectively,

$$\begin{aligned} \Delta v_{ij}(t+1) &= -\eta_1 \frac{\partial E_T(t)}{\partial v_{ij}(t)}, \\ \Delta w_j(t+1) &= -\eta_2 \frac{\partial E_T(t)}{\partial w_j(t)} \end{aligned} \quad (12)$$

### D. MPGA Design

In the paper, we use the MPGA to implement the multi-population parallel search. When the preset conditions are met, each subpopulation shares information according to the migration strategy. Otherwise, each subpopulation performs genetic operations to form a new subpopulation, and the MPGA loops through the parallel search until the test input data is obtained in the input space or the cutoff condition is satisfied.

#### 1) Population Initialization:

For the randomly generated initial population  $P(t) = \{P_1(t), P_2(t), \dots, P_h(t)\}$ ,  $h$  represents the number of subpopulations,  $h_{cp}$  represents the number of individuals in each subpopulation, and  $h * h_{cp}$  represents the total number of individuals in the population.

#### 2) Migration Strategy:

Migration strategy refers to the migration process of individuals among sub-populations to achieve the multi-population co-evolution. According to [39], we set a fixed migration interval  $T_{\text{interval}}$  and migration rate *Rate*, randomly select some individuals during the migration process, and eliminate poor individuals in adjacent subpopulation.

#### 3) Genetic Manipulation:

Each subpopulation performs genetic operations independently, including roulette wheel selection, single point crossover, and signal point mutation, as shown in Fig. 2.

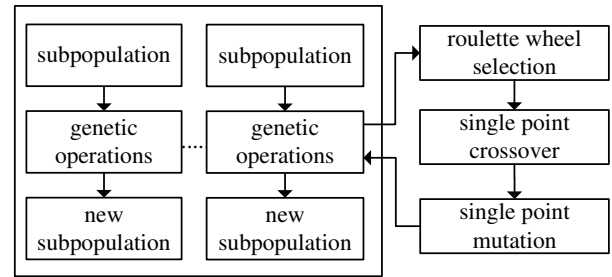


Fig. 2. Flow chart of multi-population genetic operation.

#### a) Selection Operation

The roulette wheel selection is used. The probability  $P_{ii}$  is

$$p_{ii} = \frac{f(ii)}{\sum_{jj=1}^{h_{cp}} f(jj)} \quad (13)$$

where  $f(ii)$  is the fitness value of the  $ii$ -th individual.

#### b) Crossover Operation

The single point crossover is adopted. The crossover probability  $P_c(Nc)$  is

$$P_c(Nc) = P_c(1) + c * f_{\text{rand}}(h_{cp}, 1) \quad (14)$$

where  $Nc$  is the number of iterations (NOI),  $P_c(1)$  is the initial crossover probability,  $c$  is the interval length of crossover operation,  $f_{\text{rand}}(h_{cp}, 1)$  is the generation function of random number, and the empirical interval [40] is [0.7, 0.9]. The above equation is updated as

$$P_c(Nc) = 0.7 + (0.9 - 0.7) * f_{\text{rand}}(h_{cp}, 1) \quad (15)$$

#### c) Mutation Operation

The signal point mutation is accepted. The mutation probability  $P_m(Nc)$  is

$$P_m(Nc) = P_m(1) + m * f_{\text{rand}}(h_{cp}, 1) \quad (16)$$

where  $P_m(1)$  is the initial mutation probability,  $m$  is the interval length of mutation operation, and the empirical interval [40] is [0.05, 0.001]. The above equation is updated as

$$P_m(Nc) = 0.001 + (0.05 - 0.001) * f_{\text{rand}}(h_{cp}, 1) \quad (17)$$

## IV. EXPERIMENT

### A. Experimental Design

The ATP consists of an on-board host and several on-board peripherals. The composition of the ATP is presented in this section, as shown in Fig. 3. The on-board host includes vital computer, speed measurement unit, Balise Transmission Module (BTM) host, wireless transmission unit, Track Circuit Receiver (TCR), Juridical Recorder Unit (JRU), Train Interface Unit (TIU), isolation switch, and redundancy switch. The on-board peripherals include Driver-Machine Interface (DMI), BTM antenna, GSM-R antenna, TCR antenna, and speed sensors. The operation scenarios of the CTCS-3 system include registration and startup, cancelation, movement authority, temporary speed restriction, auto-passing neutral section, Radio Block Center (RBC) handover, level transition, degraded operation, disaster protection, manual route release, shunting operation, special route, reconnection and disconnection, and entering and driving Electrical Multiple Units (EMU) depot. The case study of the startup procedure of the ATP is employed in the paper.

Fig. 4 shows the startup procedure model of the ATP, which is established by the relevant requirements specifications and the CPN modeling rules. To make the designed CPN model more concise, the driver selects the CTCS-3 level by default. After the driver inputs the identity and the telephone number of the RBC, ATP will request a secure connection with the RBC. If the correct system version is not received by the ATP, i.e. the variable  $bOverTime$  is true, and it will repeat the above operation. When a defined number exceeds 3, the driver shall be informed that no connection was established. At this point, the driver is allowed to select On Sight (OS) mode or re-enter the level, the identity, and the telephone number of the RBC.

Given the tokens of the "StartState" place and the "Comm-Total" place, we obtain the state space of the startup procedure model, as given in Fig. 5. In this case study, we try to select the paths that are difficult to cover as the selected target paths and ensure that these paths are feasible, with the aim of evaluating the feasibility of the novel method proposed. We randomly select four target paths from the designed CPN

model and calculate the corresponding target path length, as shown in TABLE I. The selected target paths are sorted according to path complexity, namely, the larger the number of the selected target path, the more complex the selected target path.

### B. Experimental Results

#### 1) Comparison of Fitness Value Solving Methods:

Because of the simplicity of the designed CPN model based on the startup procedure of the ATP, the required data is also simplified. In the paper, according to the relevant principles involved, the number of nodes of the HL, the NOI, the learning rate and the target value of the BPNN are set to 9, 1000, 0.01, and  $1e-7$ , respectively.

In order to compare and analyze the fitness value solving methods, 100 sets of input data are randomly generated for each selected target path in TABLE I, and the fitness values are calculated by the model-based method and the BPNN-based method. The comparison of the fitness value solving methods under the selected target paths are shown as Fig. 6.

It can be seen that the same conclusion can be drawn in Fig. 6 (a)-(d). Although the fitness values obtained by the above two methods are not equal, the trend is consistent. This indicates that the BPNN-based method can replace the model-based method to solve individual fitness value. If the model-based method is adopted to recalculate the accurate fitness value for individuals with good fitness, the error of the fitness value will not be introduced. Thus, it is possible and effective to calculate the fitness value by the BPNN-based method.

#### 2) Comparison of Parameters of MPGA:

The performance and efficiency of the MPGA largely depend on the value of the parameters. A reasonable value can accelerate the global convergence speed and shorten the running time. In the paper, the simple variable method [39] is used to control the relevant parameters. Three groups of comparison experiments are carried out for the selected four target paths in TABLE I. Here, the total number of individuals, the maximum NOI and the variable dimension of the MPGA are set as 100, 30, and 14, respectively.

##### a) Comparison of Number of Subpopulation

We set a fixed migration interval and migration rate, and run each condition 30 times to solve the average value and standard deviation of NOI, respectively. The statistical results are presented in TABLE II. The same conclusion is obtained for the selected four target paths. When the number of subpopulation is equal to 2, the average value and standard deviation of NOI are both minimum, indicating that the generation efficiency and stability of the current condition are the best.

##### b) Comparison of Migration Interval

We set a fixed number of subpopulation and migration rate, and run each condition 30 times to solve the average value and standard deviation of NOI, respectively. The statistical results are presented in TABLE III. The same conclusion is obtained for the selected four target paths. When the migration interval is equal to 2, the average value and standard deviation of NOI are both minimum, indicating that the generation efficiency and stability of the current condition are the best.

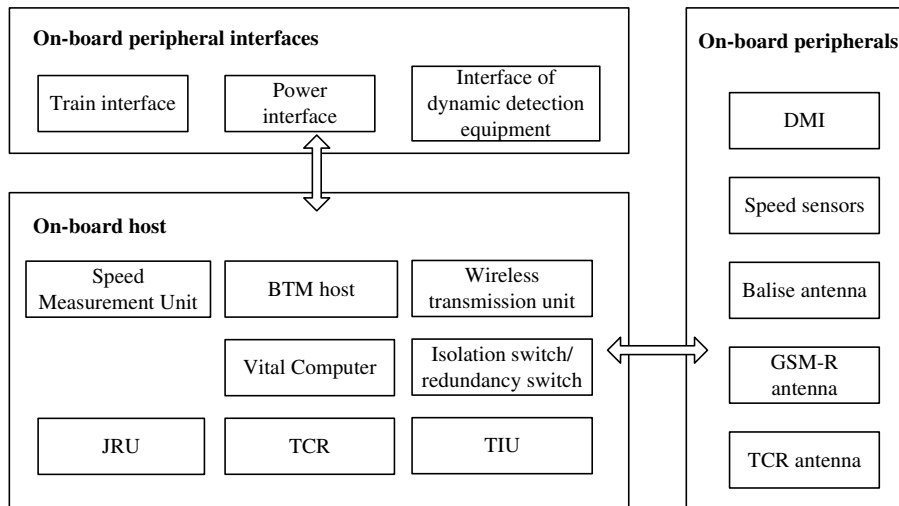


Fig. 3. Block diagram of the ATP.

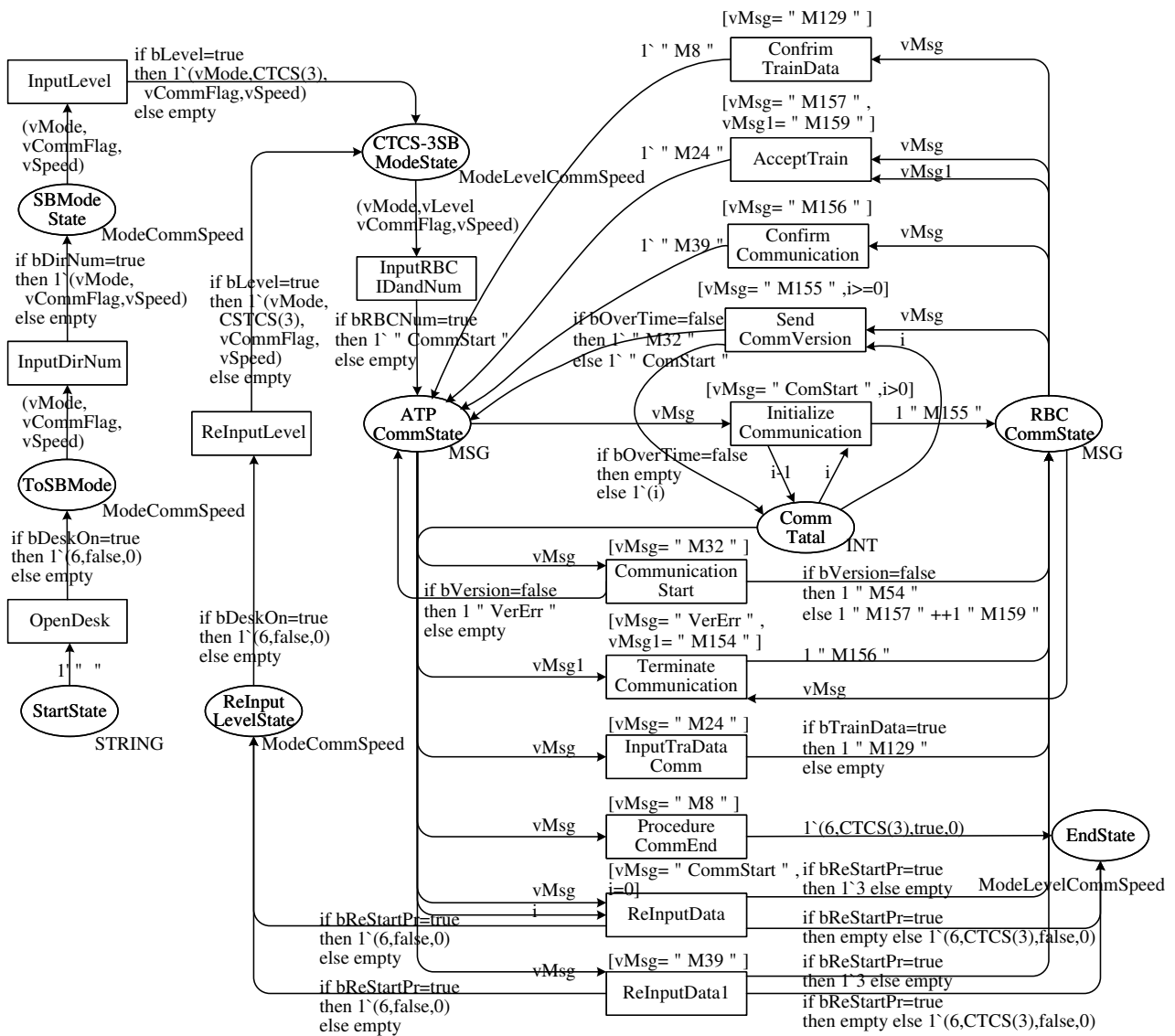


Fig. 4. The startup procedure model of the ATP.

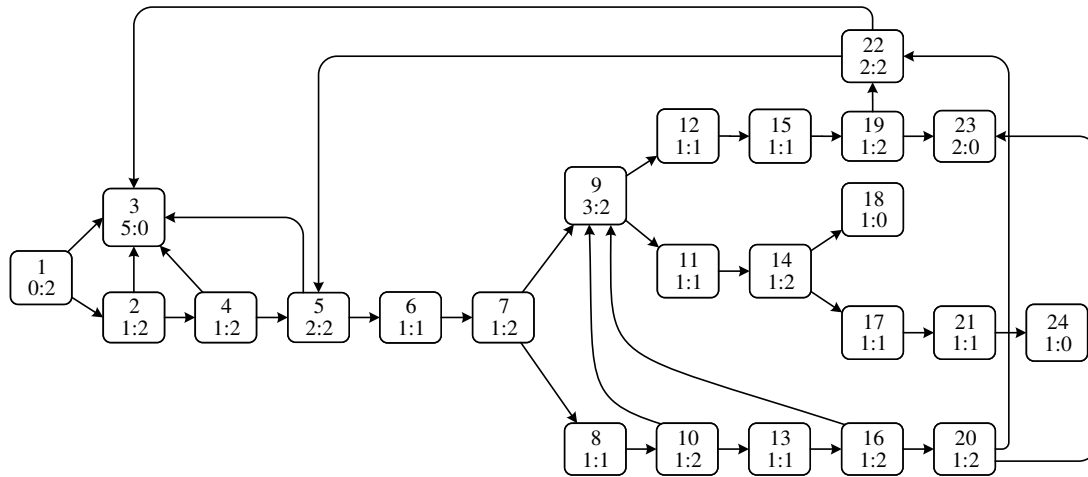


Fig. 5. State space of the startup procedure model.

TABLE I  
SELECTED TARGET PATHS

The selected target paths	The covered nodes	The path length
The selected target path <sub>1</sub>	$n_1 \rightarrow n_2 \rightarrow n_4 \rightarrow n_5 \rightarrow n_6 \rightarrow n_7 \rightarrow n_8 \rightarrow n_{10} \rightarrow n_{13} \rightarrow n_{16} \rightarrow n_9 \rightarrow n_{11} \rightarrow n_{14} \rightarrow n_{18}$	13
The selected target path <sub>2</sub>	$n_1 \rightarrow n_2 \rightarrow n_4 \rightarrow n_5 \rightarrow n_6 \rightarrow n_7 \rightarrow n_8 \rightarrow n_{10} \rightarrow n_{13} \rightarrow n_{16} \rightarrow n_9 \rightarrow n_{12} \rightarrow n_{15} \rightarrow n_{19}$ $\rightarrow n_{23}$	14
The selected target path <sub>3</sub>	$n_1 \rightarrow n_2 \rightarrow n_4 \rightarrow n_5 \rightarrow n_6 \rightarrow n_7 \rightarrow n_8 \rightarrow n_{10} \rightarrow n_9 \rightarrow n_{12} \rightarrow n_{15} \rightarrow n_{19} \rightarrow n_{22} \rightarrow n_5$ $\rightarrow n_3$	14
The selected target path <sub>4</sub>	$n_1 \rightarrow n_2 \rightarrow n_4 \rightarrow n_5 \rightarrow n_6 \rightarrow n_7 \rightarrow n_8 \rightarrow n_{10} \rightarrow n_{13} \rightarrow n_{16} \rightarrow n_{20} \rightarrow n_{22} \rightarrow n_5 \rightarrow n_6$ $\rightarrow n_7 \rightarrow n_9 \rightarrow n_{11} \rightarrow n_{14} \rightarrow n_{17} \rightarrow n_{21} \rightarrow n_{24}$	20

TABLE II  
STATISTICAL TABLE OF NOI UNDER THE DIFFERENT NUMBERS OF SUBPOPULATION

The selected target paths	The average value of NOI			The standard deviation of NOI		
	$h * h_{cp}=50*2$	$h * h_{cp}=20*5$	$h * h_{cp}=10*10$	$h * h_{cp}=50*2$	$h * h_{cp}=20*5$	$h * h_{cp}=10*10$
The selected target path <sub>1</sub>	2.83	4.17	4.60	1.83	2.49	2.44
The selected target path <sub>2</sub>	3.07	4.07	5.27	1.67	1.71	2.10
The selected target path <sub>3</sub>	4.53	4.97	7.00	1.89	2.48	3.67
The selected target path <sub>4</sub>	9.03	10.43	12.23	4.00	4.27	5.49

TABLE III  
STATISTICAL TABLE OF NOI UNDER THE DIFFERENT MIGRATION INTERVALS

The selected target paths	The average value of NOI			The standard deviation of NOI		
	$T_{interval} = 2$	$T_{interval} = 5$	$T_{interval} = 10$	$T_{interval} = 2$	$T_{interval} = 5$	$T_{interval} = 10$
The selected target path <sub>1</sub>	2.73	3.40	3.73	1.55	1.93	1.67
The selected target path <sub>2</sub>	3.27	3.70	4.20	1.73	2.35	2.30
The selected target path <sub>3</sub>	3.73	4.97	5.13	1.69	2.44	2.23
The selected target path <sub>4</sub>	8.67	10.53	11.37	3.62	4.50	4.80

c) Comparison of Migration Rate

We set a fixed number of subpopulation and migration interval, and run each condition 30 times to solve the average value and standard deviation of NOI, respectively. The statistical results are presented in TABLE IV. The same conclusion is obtained for the selected four target paths. When the migration rate is equal to 0.8, the average value and standard deviation of NOI are both minimum, indicating that the generation efficiency and stability of the current condition are the best.

According to the above experiments, when  $h_{cp}$ ,  $T_{interval}$  and Rate of the MPGA are set to 2, 2, and 0.8, respectively, the average value and standard deviation of NOI are both least, indicating that the generation efficiency and stability of the current condition are the best.

3) Comparison of Algorithms:

The test data of the selected target paths in TABLE I are generated by the novel method and the traditional GA, respectively. The statistical table of NOI under the above different methods is shown as TABLE V. The above two

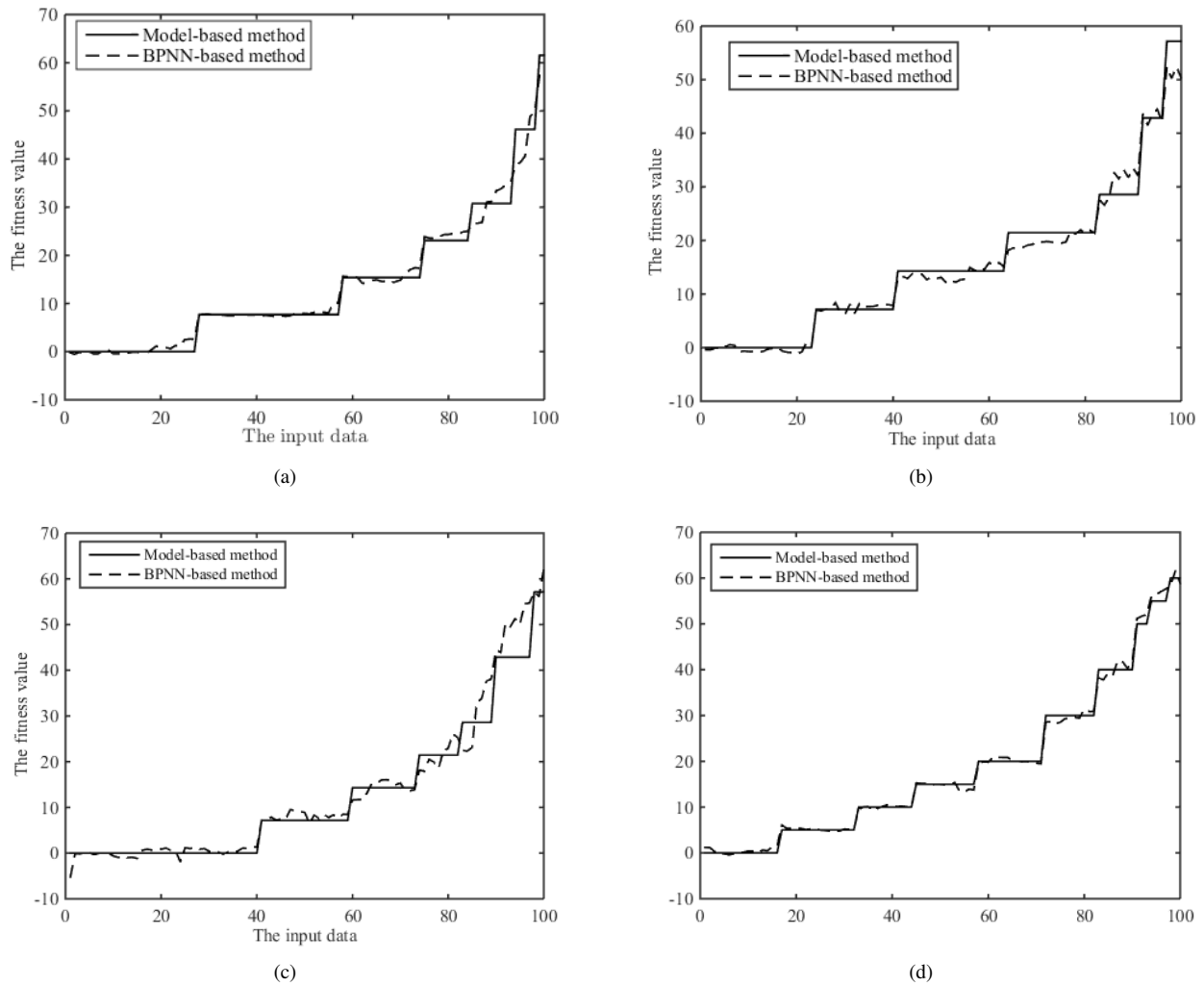


Fig. 6. Comparison of the fitness value solving methods under the selected target paths. (a) The selected target path<sub>1</sub>. (b) The selected target path<sub>2</sub>. (c) The selected target path<sub>3</sub>. (d) The selected target path<sub>4</sub>.

TABLE IV  
STATISTICAL TABLE OF NOI UNDER THE DIFFERENT MIGRATION RATES

The selected target paths	The average value of NOI			The standard deviation of NOI		
	Rate = 0.2	Rate = 0.5	Rate = 0.8	Rate = 0.2	Rate = 0.5	Rate = 0.8
The selected target path <sub>1</sub>	3.90	3.53	2.83	2.48	2.16	1.98
The selected target path <sub>2</sub>	5.00	3.63	3.27	2.52	2.02	1.75
The selected target path <sub>3</sub>	6.03	5.50	3.70	2.12	2.13	1.49
The selected target path <sub>4</sub>	11.20	9.80	7.97	3.63	4.11	3.36

TABLE V  
STATISTICAL TABLE OF NOI UNDER THE DIFFERENT METHODS

The selected target paths	The average value of NOI		The standard deviation of NOI	
	The novel method	The traditional GA	The novel method	The traditional GA
The selected target path <sub>1</sub>	3.23	4.23	1.31	1.99
The selected target path <sub>2</sub>	2.63	4.63	1.87	2.82
The selected target path <sub>3</sub>	4.70	7.03	2.12	2.94
The selected target path <sub>4</sub>	6.67	13.63	2.36	4.29

methods are run 30 times each time, and  $h_{cp}$ ,  $T_{interval}$  and  $Rate$  of the MPGA are set to 2, 2 and 0.8, respectively. As can be seen, (1) the average value of NOI required by the proposed method is 3.23, 2.63, 4.7, and 6.67, respectively, which are all smaller than that by the traditional GA. It shows

that the novel method has better efficiency. (2) The standard deviation of NOI required by the proposed method is 1.31, 1.87, 2.12, and 2.36, respectively, which are smaller than that by the traditional GA. It shows that the novel method has greater stability. (3) The more complex the selected target



path, the more superiority the novel method.

## V. CONCLUSIONS

A novel method on the basis of BPNN and MPGA is put forward in the paper, so as to enhance the test efficiency of the on-board subsystem. Here, we design the CPN models according to the relevant requirements specifications, and transform the generation problem with test data into a mathematical function optimization problem. In addition, the BPNN method is employed to replace the solving process of individual fitness value and the MPGA is adopted to make up for the shortcomings of the traditional GA. The experiments show that it is possible and effective to obtain the fitness value by the BPNN method, and the novel method has better efficiency and greater stability than the traditional GA. The proposed method introduces a novel idea and solution to automatically generate test data for the on-board subsystem, which greatly shortens the preparation time of test data.

Due to our results, we will continue to follow this research direction to extend further work. (1) One direction is the MBT-based test data generation method to solve the multi-path coverage problem. (2) The other direction is to perform more case studies of the on-board subsystem.

## REFERENCES

- [1] Y. Papadopoulos and J.A. Mcdermid, "The potential for a generic approach to certification of safety critical systems in the transportation sector," *Reliability Engineering & System Safety*, vol. 63, no. 1, pp. 47-66, 1999.
- [2] S. Reinach and A. Viale, "Application of a human error framework to conduct train accident/incident investigations," *Accident Analysis & Prevention*, vol. 38, no. 2, pp. 396-406, 2006.
- [3] The Ministry of Railways of The People's Republic of China, "Functional requirements specification of the CTCS-3 Train Control System (v1.0)," Beijing: China Railway Publishing House, 2008.
- [4] The Ministry of Railways of The People's Republic of China, "System requirements specification of the CTCS-3 Train Control System (v1.0)," Beijing: China Railway Publishing House, 2008.
- [5] The Ministry of Railways of The People's Republic of China, "Test cases of the CTCS-3 Train Control System (V2.0)," Beijing: China Railway Publishing House, 2008.
- [6] S.R. Dalal, A. Jain, N. Karunanithi, J.M. Leaton, C.M. Lott, G.C. Patton, et al., "Model-based testing in practice," in *Proc. of the 21th Int. Conference on Software Engineering 1999*, pp.285-294.
- [7] M. Broy, B. Jonsson, J.P. Katoen, M. Leucker, and A. Pretschner, "Model-Based Testing of Reactive Systems," New York: Springer Berlin Heidelberg, 2005, ch. 14.
- [8] S. Xanthakis, C. Ellis, C. Skourlas, A.L. Gal, S. Katsikas, and K. Karapoliou, "Application of genetic algorithms to software testing," in *Proc. of the 1992 Int. Conference on Software Engineering & Its Applications 1992*.
- [9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of the 1995 Int. Conference on Neural Networks 1995*.
- [10] B. Suri and S. Singhal, "Literature survey of Ant Colony Optimization in software testing," in *Proc. of the 2012 CSI 6th Int. Conference on Software Engineering 2012*.
- [11] Q. Liu, "Analysis of functional test method for CTCS-3," *China Safety Science Journal*, vol. 28, no. 2, pp. 79-93, 2018.
- [12] J.-F. Cheng, X.-Y. Zhao, J.-Q. Liu, and Y. Zhang, "Automated Test Generation Based on Colored Petri Net and Improved Depth First Search for Train Control System," in *Proc. of the 2019 Chinese Control Conference 2019*, pp.6761-6765.
- [13] W. Zheng and N.-W. Hu, "Automated Test Sequence Optimization Based on the Maze Algorithm and Ant Colony Algorithm," *International Journal of Computers Communications & Control*, vol. 10, no. 4, pp. 593-606, 2015.
- [14] C. Rao, J. Guo, Y.-D. Zhang, and Z. Cha, "A Combinational Test Suite Generation Approach for CTCS-2 Onboard ATP Equipment," *Journal of the China Railway Society*, vol. 42, no. 11, pp. 110-117, 2020.
- [15] W. Zheng, T. Tang, J.-D. Lv, and G.-D. Wei, "Method of Generating Completeness Test Suit Based on IECP for CTCS-3 Train Control TSM Profile," *Journal of the China Railway Society*, vol. 42, no. 5, pp. 72-83, 2020.
- [16] L. Dou, Y.-D. Zhang, Y. Li, and C. Rao, "Design of Test Cases for Level Transition Function of Train Control System Based on Scene Method," *Railway Standard Design*, vol. 63, no. 7, pp. 141-145, 2019.
- [17] L. Gu, Z.-W. Xu, and M. Mei, "Research on Test Case Generation of Train Control System Based on Operation Profile," *Computer Technology and Development*, vol. 30, no. 5, pp. 1-5, 2020.
- [18] L. Dou, Y. Li, J. Guo, Y. Tong, and H. Lan, "Research on optimization method for safety function test sequence of on-board equipment," *China Safety Science Journal*, vol. 29, no. 5, pp. 73-78, 2019.
- [19] W. Zheng and N.-W. Hu, "Automated Test Sequence Optimization Based on the Maze Algorithm and Ant Colony Algorithm," *International Journal of Computers Communications & Control*, vol. 10, no.4, pp. 593-606, 2015.
- [20] X.-Y. Zhao, Z.-J. Yang, and J.-Y. Lv, "Test Sequence Generation Method of Mode Transition for On-board Equipment Based on Colored Petri Net," *China Railway Science*, vol. 38, no. 4, pp. 115-123, 2017.
- [21] X.-Y. Zhao, "A Multi-objective Test Sequence Set Generation Approach for Train Control System," *Railway Standard Design*, vol. 64, no. 12, pp. 143-149, 2020.
- [22] J. Hao, H. Zhang, and P.-L. Lu, "Research on Fault Injection Test Method for Train Control System," *Railway Standard Design*, vol. 64, no. 6, pp. 153-159, 2020.
- [23] B.-Q. Wei, J.-D. Lv, K.-X. Chen, T. Tang, and W. Wang, "Mutation Timed Automata with Input and Output-Based Method of Generating Test Suites for Chinese Train Control System Level 3," *Journal of Southwest Jiaotong University*, vol. 55, no. 5, pp. 927-945, 2020.
- [24] T.S. Chow, "Testing Software Design Modeled by Finite-State Machines," *IEEE Transactions on Software Engineering*, vol. 4, no. 3, pp. 178-187, 1978.
- [25] X. Chen, P. Jiang, Y.-F. Zhang, C. Huang, and Y. Zhou, "Method of Automatic Test Case Generation for Safety-Critical Scenarios in Train Control System," *Journal of Software*, vol. 26, no. 2, pp. 269-278, 2015.
- [26] K. Jensen, "A brief introduction to coloured petri nets," in *Proc. of the 3th International Workshop on Tools and Algorithms for Construction and Analysis of Systems 1997*, pp.203-208.
- [27] K. Jensen and L.M. Kristensen, "Coloured Petri nets: modeling and validation of concurrent systems," New York: Springer Berlin Heidelberg, 2009.
- [28] Y. Zhou and T. He, "Modeling and Analysis of Level Conversion in the Train Control System Based on Activity Diagram and HTCPN," *Computer Applications and Software*, vol. 37, no. 6, pp. 119-125, 2020.
- [29] X.-Y. Wang, D.-C. Dong, D.-X. Ou, and Y.-Q. Ji, "Real-Time Analysis of Block Mode-Communication-Based Train Control System Handover Using Colored Petri Net," *Journal of Tongji University (Natural Science)*, vol. 49, no. 3, pp. 382-390, 2021.
- [30] J.-J. Shi and S.-J. Jiang, "Automatic Test Data Generation Tool of Dynamic Variable Parameters Based on Genetic Algorithm," *Computer Science*, vol. 395, no. 5, pp. 124-127, 2012.
- [31] X. Pan, S. Hao, Z.-G. Yuan, and N.-F. Song, "Evolutionary generation of test data for EFSM based on irrelevant variable separation," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 45, no. 5, pp. 919-929, 2019.
- [32] S.-P. Fan, Y. Zhang, B.-Y. Ma, L. Wan, N.-M. Yao, and Y. Song, "Evolutionary Generation of Test Data for Paths Coverage Based on Balance Optimization Theory," *Acta Electronica Sinica*, vol. 48, no. 7, pp. 303-310, 2020.
- [33] C.-Y. Xia, Y. Zhang, L. Wan, Y. Song, N. Xiao, and B. Guo, "Test Data Generation of Path Coverage Based on Negative Selection Genetic Algorithm," *Acta Electronica Sinica*, vol. 47, no. 12, pp. 2630-2638, 2019.
- [34] T.V. Avdeenok, K.E. Serdyukov, and Z.B. Tsydenov, "Formulation and research of new fitness function in the genetic algorithm for maximum code coverage," *Procedia Computer Science*, vol. 186, no. 5, pp. 713-720, 2021.
- [35] M. Esnaashari and A. H. Damia, "Automation of software test data generation using genetic algorithm and reinforcement learning," *Expert Systems with Applications*, vol. 183, no. 5, pp. 115446, 2021.
- [36] A.M. Mora, G.S. Pablo, J.M. Juan, and A.C. Pedro, "Migration study on a pareto-based island model for MOACOs," in *Proc. of the 15th annual Conference on Genetic and Evolutionary Computation 2013*, pp.57-64.
- [37] R. Yang, Z.-Y. Chen, Z.-Y. Zhang, Z.-C. Liu, and B.-W. Xu, "A new approach of automated test case generation on extended finite state machine," *SCIENTIA SINICA Informations*, vol. 44, no. 5, pp. 588-609, 2014.
- [38] X.-J. Yao, D.-W. Gong, and B. Li, "Evolutional Test Data Generation for Path Coverage by Integrating Neural Network," *Journal of Software*, vol. 27, no. 4, pp. 828-838, 2016.
- [39] X.-F. Zhou, R.-L. Zhao, and Z. Li, "Automatic Generation of Test Data For Extended Finite State Machine Based on Multi-population Genetic

- Algorithm,” *Computer Applications and Software*, vol. 32, no. 11, pp. 1-6, 2015.
- [40] J.C. Potts, T.D. Giddens, and S.B. Yadav, “The development and evaluation of an improved genetic algorithm based on migration and artificial selection,” *IEEE Transactions on Systems, Man & Cybernetics*, vol. 24, no. 1, pp. 73-86, 1994.