IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 16, 2018, accepted October 3, 2018, date of publication October 9, 2018, date of current version November 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875074

Short-Term Scheduling of Vehicle Testing System Using Object Petri Net

YISHENG AN[1], (Member, IEEE), NAIQI WU[2,3], (Senior Member, IEEE), AND PEI CHEN[1], (Student Member, IEEE)

[1]School of Information Engineering, Chang'an University, Xi'an 710064, China
[2]Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China
[3]State Key Laboratory of Precise Electronic Manufacturing Technology and Equipment, Guangdong University of Technology, Guangzhou 510006, China

Corresponding authors: Yisheng An (aysm@chd.edu.cn) and Naiqi Wu (nqwu@must.edu.mo)

This work was supported in part by the National Natural Science Foundation of China under Grant 51278058 and Grant 61703053, in part by the Natural Science Foundation of Shaanxi Province under Grant 2014JZ019, in part by the China Postdoctoral Science Foundation under Grant 2012M521729, in part by the Fundamental Research Funds for the Central Universities under Grant 310824162021 and Grant 300102248504, and in part by the Science and Technology Development Fund (FDCT) of Macau under Grant 106/2016/A3.

ABSTRACT This paper addresses the modeling, analysis, and scheduling optimization of a vehicle testing system, a widely used facility in China's transportation management and automotive manufactures. To do so, object Petri net models are built to describe its resources (including testing equipment and field) and operations. Based on these models, a short-term scheduling framework and an algorithm for optimally generating a vehicle's testing sequence are presented. By this framework, a waiting area with adjustable capacity is set for vehicles to be inspected in the system. The complete permutations of vehicles in the waiting area are considered as tokens in an object Petri net. In this way, with the models, the total testing turnaround time of a vehicle sequence can be calculated and the optimal one can be chosen. Experimental results show that the proposed method can reduce the average testing turnaround time by about 31% compared with the traditional first-come-first-served-based one. In addition, the proposed method can obtain a shorter average testing turnaround time when the re-inspection rates are 25%, 50%, 75%, and 100%. Thus, it is applicable and effective.

INDEX TERMS Vehicle testing system, object Petri net, short-term scheduling.

I. INTRODUCTION

Vehicle testing system (VTS) is a comprehensive and specially purposed system. It is designed to inspect in-used or newly produced vehicles for the performance of multiple indicators. These indicators include safety, reliability, fuel economy, and emissions, etc. It is a widely used testing system in China's automotive manufactures and transportation management to guarantee that the in-used vehicles are in good status. Practical experience shows that VTS can significantly improve driving safety, reduce carrying costs and air pollution, and incidence of traffic accidents.

For the design and implementation of a VTS, the following three issues are critical: 1) accuracy and reliability of data acquisition and transmission; 2) the operational architecture based on networked measurement and control; and 3) methods for scheduling, optimization and performance analysis of the vehicle testing process. Extensive studies have been conducted on the first two issues and significant advancements have been achieved [1]–[6]. However, the scheduling optimization problem of a VTS receives less attention and is challenging. Also, in the literature, it lacks studies on modeling and quantitative performance analysis for the operations of a VTS. Thus, up to now, a first-come-first-served-(FCFS)-based dispatching strategy is still applied to schedule vehicles to be tested for almost all existing VTSs. Despite the simplicity of such a dispatching strategy, practical experiences show that it is not effective in terms of performance due to a long average turnaround time.

Given a group of vehicles to be tested, the short-term scheduling problem of a VTS is to obtain a reasonable permutation of the vehicles to complete the given testing tasks for each vehicle. According to the characteristics of a VTS, Chen *et al.* [7] classify it as a multi-purpose job shop scheduling process. For a multi-purpose job shop scheduling problem, Chaudhry and Khan [8] present a comprehensive review of its modeling and optimization techniques. Mathematical programming methods, such as mixed integer linear programming (MILP) models or mixed integer nonlinear

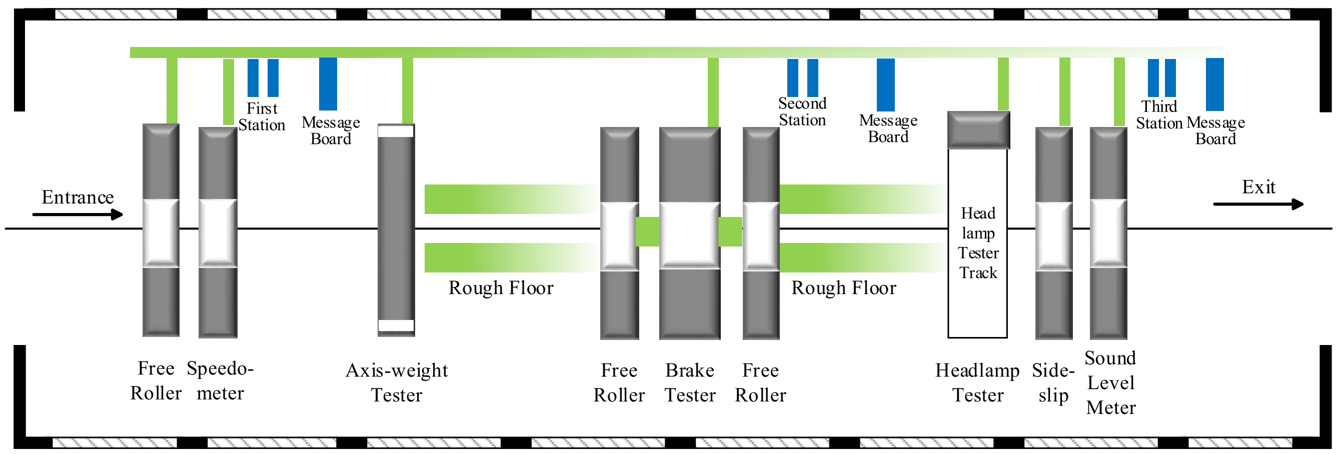


FIGURE 1. A layout of a vehicle testing system involving three testing stations.

programing (MINLP) models are widely applied for finding an optimal solution of such scheduling problems [9]–[14]. An MILP formulation for short-term scheduling of a VTS is developed by Chen *et al.* [7]. Nait-Sidi-Moh and El-Amraoui [13] adopt an MILP model to optimize the configuration of a production line and calculate the optimal cycle time. Liao and Liao [14] propose an MILP-based method for the flow shop scheduling problem with unlimited or zero intermediate storage. As pointed out in [15] and [16], such a scheduling problem has an NP-hard complexity in nature. Moreover, to formulate an MILP or MINLP model for such problems with acceptable accuracy, the time should be divided into uniform slots that are small enough, leading to a large number of binary variables such that the problem is very difficult to solve. In order to simplify the solution process for solving the problems modeled by an MILP or MINLP, some assumptions may be made, which can affect the solution accuracy and limit the applications of an obtained solution.

The operations of a VTS are driven by a series of events, i.e., a vehicle enters a testing station, a vehicle leaves a testing station, a piece of testing equipment starts or stops its operation, data are collected and transmitted, etc. The scheduling problem of a VTS is a typical discrete event system (DES), which is characterized by critical properties such as parallelism, concurrency, and synchronization. As Petri net (PN) is an effective tool for handling concurrent and/or distributed activities and sequences [17], [18], it is widely used for modeling, analysis, scheduling, and optimization in DESs and hybrid systems [19]–[35]. With Petri nets, the dynamic behavior of a system can be clearly described in both mathematical and graphic ways. In [36], a detailed survey on modeling, analysis, planning and scheduling, supervision and coordination control, and design of production systems by using Petri nets is given. In [37], a cluster tool configured with several wafer processing modules and a robot is successfully modeled and scheduled by using a kind of colored-timed-resource-oriented Petri nets. In [38], the qualitative behavior of batch production plants is modeled in a hierarchical way by using Petri nets with dynamic tokens. By describing the symbolic behavior at the lower level, it presents a three-level scheduling optimization strategy.

In this paper, we describe the dynamic behavior of a VTS by using an object Petri net (OPN) such that the space and time constraints can be well modeled. Based on this model, a scheduling framework for the operations in a VTS is presented to find a reasonable and effective vehicle testing sequence with the considered constraints being satisfied.

The remainder of this paper is organized as follows. Section II briefly describes the specific scheduling problem and explains the motivation with an example. After Section III develops the OPN model, the OPN-based scheduling method is proposed in Section IV. Section V illustrates the advantages of the proposed method by using experiments and analysis. Conclusions are given in Section VI.

II. PROBLEM DESCRIPTION AND MOTIVATION

The general layout of a VTS is illustrated in Fig. 1. In this VTS, there are three testing stations. Station 1 is for speedometer testing, Station 2 inspects the axle weight and checks the performance of the brake, and Station 3 tests the performance of the headlamp, sideslip, and sound level. The equipment for different testing operations and its functions are given as follows.

- Speedometer platform. Its function is to calibrate a vehicle's speedometer.
- Axle weight platform. It is used to measure the axle weight of a vehicle.
- Brake test platform. It is used to inspect the braking force of a vehicle.
- Headlamp tester. It is used to test the luminosity, up and down, and left and right deviation of the head light.
- Sideslip platform. It is used to measure the inward and outward inclination of the wheels.
- Sound level meter. It is used to test the volume of a vehicle's horn.

TABLE 1. Test items and time taken by each equipment.

Station	Test Item	Time consuming (minutes)	Total time (minutes)
First Station (FS)	speed	2	2
Second Station (SS)	axle weight	3	6
	brake	3	
Third Station (TS)	headlamp	2	4
	slide	1	
	sound	1	

TABLE 2. Combinations of vehicle testing requirements.

Test Requirement	Stations and testing sequence	Testing time (minutes)
TR1	FS → SS → TS	12
TR2	FS	2
TR3	FS → SS	8
TR4	SS	6
TR5	SS → TS	10
TR6	TS	4
TR7	FS → TS	6

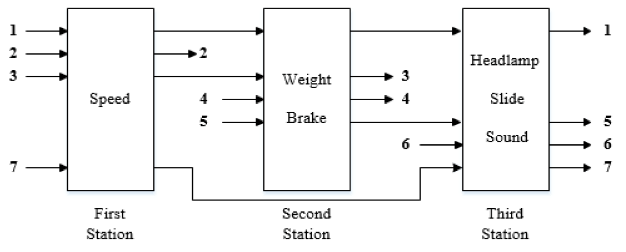


FIGURE 2. The multi-purpose vehicle testing requirements.

The average testing time of each item is listed in Table 1. Due to the large population of vehicles, there are many vehicles to be inspected in a VTS every day. In the present practice, the vehicles to be inspected for the first time are called non-inspected vehicles. When vehicles arrive, they are arranged in a queue and a dispatcher selects a vehicle to be inspected according to the rule of FCFS. For a non-inspected vehicle, it is compulsory to have all items examined via all three testing stations. After a vehicle completes the test for the required items, often some of items do not pass the test. Then, such vehicles must be repaired such that they can pass the test. A repaired vehicle then goes to a VTS for re-inspection again and it is called an inspected vehicle.

Thus, in a VTS, there are both non-inspected and inspected vehicles for testing. Note that an inspected vehicle does not need to test all the items, but only the ones that are not passed for the previous inspection and the required test items are different for different inspected vehicles. Thus, with both non-inspected and inspected vehicles for testing, an FCFS-based dispatching strategy must not be effective due to that the vehicles have different inspection items. Based on the layout of a VTS shown in Fig. 1 and the above discussion, there are seven combinations of inspection tasks as shown in Fig. 2. We introduce a new term called testing requirement (TR) to represent these seven possibilities and list them in Table 2. TR1 indicates that a vehicle is sequentially tested through the first, second and third stations; TR2, TR4, and TR6 indicate that a vehicle is tested by a single station only; while TR3, TR5, and TR7 indicate that a vehicle is tested by two stations. The symbol "→" indicates the order in which the vehicle enters the adjacent station.

In addition, a piece of testing equipment at each station can inspect one vehicle parked at its testing field only at a time. When a vehicle completes its inspection at any station, whether it can enter the next station or not depends on if the testing field of the next station is emptied. If it is emptied, then the vehicle can be driven into it; otherwise, the vehicle has to wait until the next station's testing field is available. In this way, a vehicle needs to wait unnecessarily at some stations and such wait can propagate from station to station and from vehicle to vehicle such that the waiting time accumulates, which greatly affect the productivity and utilization of equipment. For example, suppose that two vehicles A and B arrive at a VTS sequentially, vehicle A is a non-inspected one, whereas B is an inspected one with testing items that are performed in the third station only. Therefore, the total testing turnaround time for these two vehicles is recorded as the interval between the A's entering the VTS and B's leaving the VTS, that is $(2 + 6 + 4) + 4$ (refer to Table 1), totally 16 minutes. The average testing turnaround time for each vehicle is eight minutes. However, if we adjust the testing sequence of these two vehicles, that is, dispatching B first and then A, although B arrives slightly later than A. Then, the total testing turnaround time is 12 minutes, and the average testing turnaround time for them drops to six minutes.

By analyzing the vehicle testing process of a large number of existing VTSs, we notice that such a phenomenon occurs frequently during the vehicle testing process and this inevitably affects the overall throughput and inspection efficiency of a VTS. The underlying reason lies in that the scheduling performance is completely dependent on the arriving order of vehicles, which may result in unnecessary waiting for many vehicles that have testing tasks at only one or two stations. Therefore, in operating a VTS, the key is to make the system operate in a paced way to avoid vehicles' unnecessary long waiting such that the idle time of each piece of equipment is as short as possible.

It is a great challenge to find a short-term schedule for a VTS such that a VTS can operate in a paced way and, up to now, there are no efficient technique and tool for this purpose [7]. In fact, in practice, the job for scheduling a VTS is still done manually by a planner. The primary requirement for its short-term scheduling is its feasibility and there is no conflict when a schedule is executed. In addition, efficiency is another major issue to be taken into consideration. Because of the complexity of this scheduling problem, given a schedule, no one can predict what states would be reached.

The state space observed by an operator is very limited and a decision made by an operator is generally not optimal. Instead, an operator makes a decision based on the current state at a time and the entire schedule is obtained in a step-by-step way. Thus, it is desired to have an efficient tool that helps a planner to check the feasibility and effectiveness of a partial schedule. However, at present, there is no such a tool available, and a staff checks its feasibility approximately, which may lead to an infeasible and inefficient schedule. This motivates us to conduct this study. To solve the problem, an object Petri net-based method is proposed. By this method, based on the FCFS rule, each time a group of vehicles are dispatched to a waiting area and an optimal schedule is then efficiently found for this group of vehicles by controlling the number of vehicles in the group. In this way, the performance of a VTS is significantly improved.

The contributions of this work include: 1) an object Petri net model for the short-term scheduling problem of a VTS is developed such that the dynamic behavior can be well described; 2) we propose the concept of a waiting area with adjustable capacity for vehicles to be grouped and inspected; and 3) an algorithm based on OPN model simulation is presented to find an optimal short-term schedule for a group of vehicles in the waiting area. Hence, it is meaningful to adopt OPN to describe the operational process of a VTS and the OPN-based scheduling framework can be used as a tool for the operations of a VTS.

III. OBJECT PETRI NET

In this section, we present the OPN model for describing the operations of a VTS. We assume that the readers are familiar with the basic concept of Petri nets (for their definition and theory, the readers can refer as to [17] and [18]). From the viewpoint of object oriented programming, a collection of places is used to store the data structure of systems and transitions are used to represent events that drive the system state evolution. The capabilities of a high-level programming language and module cohesiveness are associated with PNs, leading to the concepts of object Petri net [39], [40].

An OPN model is composed of a system net, a number of object nets, and the interaction relation between them, we briefly explain these terms as follows by presenting the basic definitions for it. In addition, in order to clearly distinguish the symbols of system net and object net, an extra mark "$\sim$" is placed on top of each symbol of the system net, such as: $\tilde{t}, \tilde{p}, \tilde{T}, \tilde{P}, \cdots$ etc. Then, we first present the definition of a system net.

Definition 1: A system net is denoted as $SN = (\tilde{P}, \tilde{T}, \tilde{A}, \tilde{E})$, where

1) $\tilde{P} = P_{ob} \cup P_{bt}$ is a finite set of places, where P_{ob} and P_{bt} are finite sets of object and ordinary places, respectively;
2) $\tilde{T} = T_{or} \cup T_{uc}$ is a finite set of transitions, where T_{or} and T_{uc} are finite sets of ordinary and object interactive transitions, respectively;
3) $\tilde{A}$ is a finite set of directed arcs such that $\tilde{P} \cap \tilde{T} = \tilde{P} \cap \tilde{A} = \tilde{T} \cap \tilde{A} = \varnothing$;
4) $\tilde{E}$ is an arc expression function that defines expressions on directed arcs.

Besides a system net, in an object Petri net, there are a set of object nets. The i-th object net is defined as follows.

Definition 2: The i-th object net is denoted as $ON_i = (P_i, T_i, A_i, E_i)$, where

1) P_i is a finite set of places in the i-th object net;
2) T_i is a finite set of transitions in the i-th object net;
3) A_i is a finite set of directed arcs;
4) E_i is an arc expression function.

With Definitions 1 and 2 for a system net and an object net, we can present an object Petri net as follows.

Definition 3: An object Petri net is a four-tuple $OPN = (SN, \{ON_i\}_{i \in \mathbb{N}}, S, \Re_0)$, where

1) SN is a system net;
2) $\{ON_i\}_{i \in \mathbb{N}}$ is a finite set of object nets, $\mathbb{N}$ is a set of natural numbers for the indices of the object nets;
3) S is the interaction relation;
4) $\Re_0$ is the initial making of an object Petri net.

Definition 3 presents the structure of an object Petri net. The following definition presents the marking for describing the state of an object Petri net.

Definition 4: $\Re_0$ specifies the initial token distribution, where $\Re_0 : \tilde{P} \rightarrow \mathbb{N} \cup Bag(ON)$ with $ON := \{ON_1, ON_2, \cdots, ON_k\}$, where Bag denotes a multi-set.

In an object Petri net, it is important to give the interaction relation between the system net and the object nets. The following definition clarify this issue.

Definition 5: Interaction relation is denoted as: $S : \tilde{T} \times T' \rightarrow \{0, 1\}$, $T' = \bigcup_{i \in \mathbb{N}} T_i$, where 1 indicates that there is an interaction relation between a system net and an object net, while 0 indicates that there is no interaction relation.

The introduction of interaction relation into an OPN greatly enriches the semantics of the places. Petri net with the enhanced modeling capabilities is well suited for modeling concurrency with multiple sets of activities. In this paper, the activities with inherent logic can be represented by an object net, and multiple object nets are connected with the system net.

Similar to the initial marking for a system net and object net, any reachable marking in an OPN also adds a description of the token distribution within the object nets. Therefore, a reachable marking can be represented by

1) A distribution of object nets or black tokens $\Re : \tilde{P} \rightarrow \mathbb{N} \cup Bag(ON)$, and
2) A vector $\mathrm{M} = (m_1, m_2, \ldots, m_k)$ with m_i being the current marking of ON_i $(1 \leq i \leq k)$.

Based on the above definitions, the following definition presents the evolution rules for an object Petri net.

Definition 6: Let $(\Re, \mathrm{M})$ be a marking of an object Petri net $OPN = (SN, \{ON_i\}_{i \in \mathbb{N}}, S, \Re_0)$, $\tilde{t}$ is a transition of a system net and $\tilde{t} \in \tilde{T}$, $t \in T_i$, is a transition of the i-th object net $ON_i \in ON$ such that $(\tilde{t}, t) \in S$ forms an interaction relation. Then, $(\tilde{t}, t)$is activated in $(\Re, \mathrm{M})$ if:

a) $\Re(\tilde{p}) \geq \tilde{E}(\tilde{p}, \tilde{t})'ON_i$ for all $\tilde{p} \in {}^{\bullet}\tilde{t} \cap P_{ob}$,
b) $\Re(\tilde{p}) \geq \tilde{E}(\tilde{p}, \tilde{t})$ for all $\tilde{p} \in {}^{\bullet}\tilde{t} \cap P_{bt}$ and
c) t is activated at marking $m = pr_i(\mathrm{M})$, where $pr_i(\mathrm{M})$ denotes the i-th component m_i in M.

This is denoted by $(\Re, \mathrm{M})[(\tilde{t}, t)\rangle$. In this case the successor marking $(\Re', \mathrm{M}')$ of an OPN can be expressed as

a) $\Re'(\tilde{p}) = \Re(\tilde{p}) - \tilde{E}(\tilde{p}, \tilde{t})'ON_i + \tilde{E}(\tilde{t}, \tilde{p})'ON_i$ for all $\tilde{p} \in P_{ob}$.
b) $\Re'(\tilde{p}) = \Re(\tilde{p}) - E(\tilde{p}, \tilde{t})_i + E(\tilde{t}, \tilde{p})$ for all $\tilde{p} \in P_{bt}$.
c) $\mathrm{M}' = \mathrm{M}_{i \to m'}$, i.e., the i-th component of M is substituted by m'.

In Definition 6, $\tilde{E}(\tilde{p}, \tilde{t})'ON_\mathrm{i}$ represents the multiset containing one element ON_i with multiplicity $\tilde{E}(\tilde{p}, \tilde{t})$.

In general, an object Petri net provides a paradigm such that both data types and system behavior in the paradigm are encapsulated into a Petri net. Note that tokens in a traditional Petri net or colored Petri net can represent numbers or color types only, but cannot describe the system interaction behavior. However, the semantic of a token in an OPN is greatly expanded to represent numbers as well as object nets. Interactions can be established between different object nets or between the system net and object nets to indicate the change of system status. This is why we choose object Petri net to model a VTS for its short-term scheduling problem.

In addition, in a queue of vehicles to be inspected, non-inspected and inspected vehicles are mixed, indicating that the testing requirements are diverse and thus the testing process is not a unified process. It is also worthy to notice that the testing process is composed of two levels. At the upper level, a scheduler dispatches vehicles into the three testing stations according to the status of the stations. At the lower level, the vehicles are examined for specific testing items within each station. Therefore, the software requirement specification of the scheduler makes an OPN suitable to be used in modeling and analyzing the operations of a VTS.

IV. OPN MODEL FOR SHORT-TERM SCHEDULING

The modeling and analysis of the logic layer for scheduling a VTS inevitably involves the processing time at the physical layer. The exact timing of an event and the correctness of the system behavior depend not only on the logical outcome of the event but also on the timing of the events. In this section, we first present the constraints imposed on the system and then present the detailed OPN-based model for testing equipment, testing field, and testing process. Last, we discuss how to effectively schedule the system by using this model.

A. PROBLEM DESCRIPTION

For the operations of a VTS, one needs to optimize an objective and at the same time make the relevant requirements satisfied. Below, we present the objective function and constraints for the short-term scheduling problem in a VTS:

1) OBJECTIVE FUNCTION

Given a number of vehicles to be inspected, the goal of the short-term scheduling problem is to minimize the makespan (MS). Thus, its objective function is:

$$\min \ \mathrm{MS} = T_{e_{\tilde{k}\tilde{j}}} \tag{1}$$

where $\tilde{k}$ represents the last vehicle, $\tilde{j}$ represents the last station, T_e represents the time when a vehicle leaves a testing station.

2) TIME CONSTRAINT AT A TESTING STATION

The inspection of a vehicle at any testing station must satisfy the following time constraint:

$$T_{e_{k,j}} - T_{s_{k,j}} = \sum_{i \in I} X_{ik} P_j, \quad \forall k \in K, j \in J \tag{2}$$

where $T_{e_{k,j}}$ represents the time when the k-th vehicle finishes its testing at the j-th testing station; $T_{s_{k,j}}$ represents the time when the k-th vehicle starts its testing at the j-th testing station; X_{ik} indicates whether the k-th vehicle is tested on the i-th equipment or not, if yes, it is 1; otherwise, it is 0; and P_j is the time required for a vehicle to be inspected at the j-th station. I, J, and K represent the set of equipment, testing stations, and the vehicles to be tested.

3) TIME CONSTRAINT AT ADJACENT TESTING STATIONS

The vehicle must meet the following time constraint when testing at two adjacent testing stations

$$T_{s_{k,j+1}} - T_{e_{k,j}} \geq 0, \quad \forall k \in K, j \in J \backslash | \tilde{j} | \tag{3}$$

where $T_{s_{k,j+1}}$ represents the time when the k-th vehicle starts its testing at the $(j+1)$-th testing station; and $T_{e_{k,j}}$ represents the time when the k-th vehicle finishes its testing at the j-th testing station. This constraint indicates that the k-th vehicle must complete the inspection at the j-th station earlier than the time at which it starts the inspection at the $(j+1)$-th station.

In addition, the following time constraint should be satisfied when two vehicles are inspected simultaneously at two adjacent testing stations:

$$T_{s_{k+1,j}} - T_{s_{k,j+1}} \geq 0, \forall k \in K \backslash \{\tilde{k}\}, j \in J \backslash | \tilde{j} | \tag{4}$$

where $T_{s_{k+1,j}}$ represents the time when the $(k+1)$-th vehicle starts its testing at the j-th testing station. This constraint indicates that the $(k+1)$-th vehicle starts its testing at the j-th testing station, if and only if the k-th vehicle enters the $(j+1)$-th testing station and starts testing. Furthermore, the following inequality constraint should also be satisfied:

$$T_{s_{k+1,j}} - T_{e_{k,j}} \geq T_{s_{k+1,j}} - T_{s_{k,j+1}} \geq 0, \quad \forall k \in K \backslash \{\bar{k}\}, j \in J \backslash \{\bar{j}\} \tag{5}$$

Formula (5) indicates that, at the j-th testing station, the time to start testing the $(k+1)$-th vehicle should be later than the time when the k-th vehicle leaves the testing station. With Objective (1) and Constraints (2)-(5), an integer programming problem is formulated to solve the short-term scheduling problem of a VTS. It should be pointed out that

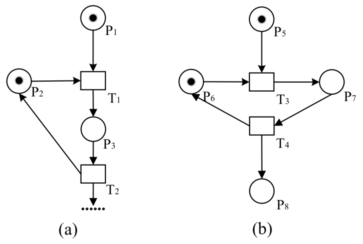


FIGURE 3. OPN models for a piece of equipment and a testing field.

TABLE 3. The Physical Meaning of Places and Transitions in Fig. 3.

Element	Type	Meaning
P_1, P_5, P_8	Place	Vehicle to be tested
P_2	Place	Equipment used in the testing process
P_3	Place	Vehicle in the testing process
P_6	Place	Photoelectric switch is connected
P_7	Place	Photoelectric switch is blocked
T_1/T_2	Transition	Start/end of the testing process
T_3/T_4	Transition	Block/restore photoelectric switch

the above integer programming problem cannot be efficiently solved unless the number of vehicles to be inspected is very small.

B. OPN MODEL FOR THE SHORT-TERM SCHEDULING

With the definition of an OPN presented in Section III and constraints described above, we can now model the components and the system behavior of operating a VTS.

1) MODELING OF TESTING EQUIPMENT

In a VTS studied in this article, there are mainly six kinds of testing equipment as shown in Fig. 1. From the perspective of Petri net modeling, the testing equipment can be represented by a place with or without a token in it, indicating whether the equipment is available or not. In addition, a piece of testing equipment is reusable. Hence, the place representing a piece of equipment is both the input place of the transition for starting a task and the output place of the transition ending a task. Place p_2 in Fig. 3(a) illustrates such a model. The physical meaning of places and transitions is given in Table 3.

2) MODELING OF TESTING FIELD

In each testing station, in addition to the necessary testing equipment, there is also an area for a vehicle to park. We call it the testing field. The testing field is a critical resource that is mutually exclusive, as each testing field can accommodate one vehicle only at a time. Usually, two pairs of photoelectric switches are installed at the entrance and exit of the testing field. We can use the status of photoelectric switch to indicate whether the testing field is available or not, as depicted in Fig. 3(b). Photoelectric switches can be modeled by two places and two transitions. The places present the status of the photoelectric switches, i.e., if it is on or off, indicating that the testing field is occupied or empty. The transitions indicate that the behavior of a testing vehicle, such as entering or leaving the testing field.

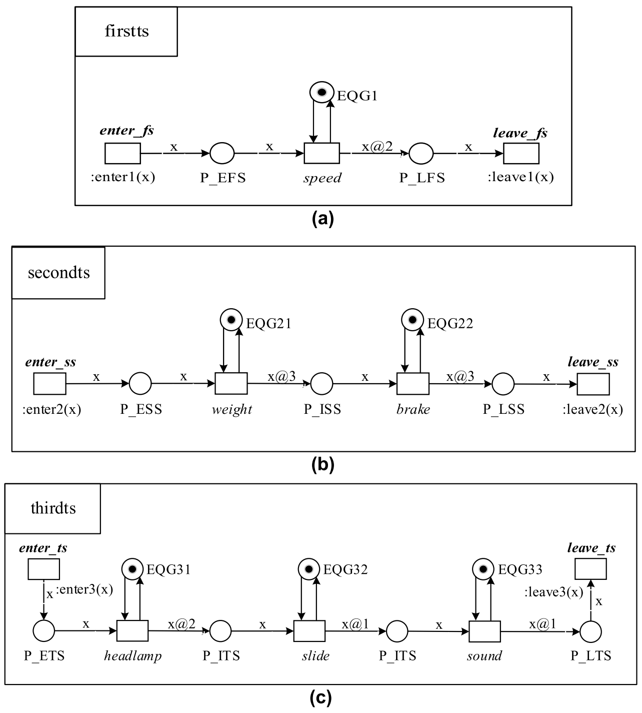


FIGURE 4. OPN models of the three testing stations. (a) Object net of first station. (b) Object net of second station. (c) Object net of third station.

3) MODELING OF TESTING STATIONS

The diagrams depicted in Fig. 4 (a)-(c) show the OPN-based testing station models. The physical meanings of places and transitions are presented in Table 4. According to the definition of an OPN, these three models belong to object nets. Fig. 4(a) presents the model for the first testing station, where transition *speed* represents the measurement of a vehicle's speedometer and the arc expression between transition *speed* and place P_LFS indicates that it takes two minutes to complete the corresponding testing task. Fig. 4(b) presents the model for the second testing station, where transitions *weight* and *brake* represent measuring a vehicle's axle weight and breaking force, respectively. The two arc expressions x@3 give the time taken for tasks of axle weight measuring and brake testing, both of them take three minutes. Similarly, the model of third testing station is shown in Fig. 4(c), the three inspection items such as headlight, sideslip, and sound level are modeled. The three arc expressions x@2, x@1, and x@1 in Fig. 4(c) mean that the time taken for headlight, sideslip, and sound level testing is two minutes, one minute, and one minute, respectively. In total, it takes 12 minutes for a vehicle to complete all inspection tasks at Station 3.

4) MODELING OF THE OPERATIONS

The diagram in Fig. 5 shows an OPN-based model describing the operations of a VTS, including a model describing the complete inspection process represented by a system net and three instances of object net models for three testing stations. Places ENT and OUT represent two critical events, namely

TABLE 4. The Physical Meaning of Places and Transitions in Fig. 4 and Fig. 5.

Element	Type	Meaning
ENT/OUT	Place	Enter/Exit the testing system
RDT1/2/3	Place	Prepare to enter the first-/second-/third-station
FS/SS/TS	Place	Represent the first-/second-/third- station
RDL1/2/3	Place	Prepare to leave the first-/second-/third-station
TFD1/2/3	Place	Represent the different testing fields in the first-/second-/third- station
P_EFS/P_LFS	Place	Represent entering or leaving the first-station
P_ESS/P_ISS/ P_LSS	Place	Represent entering or leaving or being in the second-station
P_ETS/P_LTS/ P_ITS1/P_ITS2	Place	Represent entering or leaving or being in the third-station
EQG1/21/22/31/ 32/33	Place	Represent the testing equipment used at each station
ARR1/2/3	Transition	Arrive at different testing stations
LEA	Transition	Leave the third-station
speed, *weight*, *brake*, *headlamp*, *slide*, *sound*	Transition	Represent different testing items
(*TS1,enter_fs*), (*TSD1,leave_fs*), (*TS2,enter_ss*), (*TSD2,leave_ss*), (*TS3,enter_ts*), (*TSD3,leave_ts*)	Interaction relation	Six pairs of interaction relation between the system net and three object nets, such as firstts, secondts and thirdts

vehicle entering the first-station and leaving the third-station. In this proposed model, logical monitor points are set up in these two specific places to record the event and time when a token is deposited into or removed from it.

According to the definition of an OPN, the individual instances of three object nets shown in Fig. 4 can be generated based on Renew simulation platform in the form of tokens deposited in the three specific places FS, SS, and TS. An inscription, such as "fs:enter1(x)", is related to a corresponding inscription ":enter1(x)" in the object net which is referenced by fs. The two pairs (TS1, enter_fs) and (TSD1, leave_fs) implement the interaction relation between the inspection process of a vehicle and the first object net: firstts.

When Transition *TS1* fires, by the inscription "fs:new firstts", an instance of the object net firstts (see Fig. 4 (a)) is created with a new identifier fs. Then, a reference to this instance is deposited in object place *FS* in the form of a token. Meanwhile, Transition *enter_fs* in this instance fires and a token is deposited into Place P_EFS. This token represents that a vehicle is ready to start its speedometer test. After Transition *speed* fires, a token is deposited in Place P_LFS with time stamp 2. When Transition *leave_fs* in this instance fires, the associated Transition *TSD1* in the system net also fires and one black token is deposited into Place RDL1 and then Transition ARR2 is enabled.

Note that the interaction between the system net and the object nets for the second-station and third-station is similar to the above process and thus the detailed description for them is omitted.

With this model, given a vehicle testing sequence, at any time, one can know the state of the system, including the vehicles that are being inspected in the stations and when a vehicle enters and leaves the system, presenting the performance for a given schedule in a dynamic way. Thus, the model provides a tool for scheduling the system.

C. SCHEDULING FRAMEWORK AND OPTIMIZATION

With the OPN models for the operations of a VTS, it is now ready to present a method for helping the VTS operator to obtain an efficient schedule, i.e., a vehicle sequence for inspection. Hence, a short-term scheduling framework and OPN-based scheduling strategy are described in this section.

First of all, an OPN-based short-term scheduling framework is proposed in Fig. 6. It consists of an OPN model simulation platform, scheduling time constraints, testing field constraint, scheduling strategy, and scheduling execution module. The objective of the scheduling framework is to provide a tool to find a reasonable and effective vehicle sequence with shortest testing turnaround time. This scheduling framework can be virtually viewed as a parallel system corresponding to the dispatcher in a VTS. In this framework, the OPN model can be simulated many times with different input tokens and each token corresponds to a possibility of a full permutation of vehicles in the waiting area. Hence, the best vehicle sequence corresponding to the shortest testing turnaround time can be screened out and submitted to the VTS scheduler for dispatching the vehicles.

Based on the developed model, to obtain an optimal schedule, theoretically one need to enumerate all the permutations and choose the best one by using the model. However, this is impossible and not practicable. First, there are hundreds of vehicles to be inspected in a VTS every day. It is computationally infeasible to enumerate all the permutations for so many vehicles. Also, it is not practicable since the vehicles to be tested in a day are not all available at the beginning. To effectively solve this problem, we propose a scheduling method as follows.

In practice, at any time, there are tens of vehicles waiting for testing. To schedule the vehicles, we set a waiting area such that the vehicles in this area are tested next. With a waiting area, from the vehicles waiting for testing, each time we dispatch a group of vehicles into the waiting area by using the FCFS rule and then enumerate all the permutations of the vehicles in the waiting area. Then, with the PN model, the best one is chosen. The key is to decide the number of vehicles in the group such that a solution can be found efficiently, which will be discussed later. In this way, an efficient and effective method is proposed.

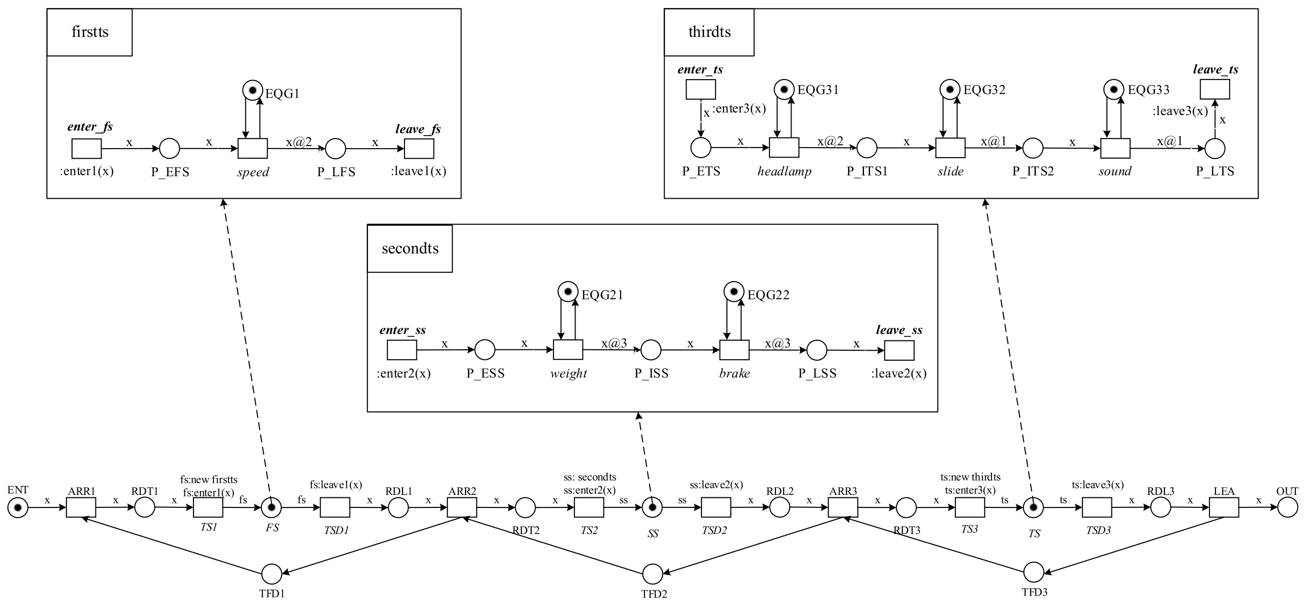


FIGURE 5. Model of testing operations in a VTS.

Next, we describe in detail the algorithm for generating optimal vehicle sequences based on the OPN model and simulation as follows. In the algorithm, the input parameters m, n, and D_n indicate the capacity of the waiting area, the number of vehicles waiting to enter the VTS, and the list of testing requirements of these vehicles. The output parameters D_s and *minTime*represent the optimal list of sequence of vehicles and the corresponding testing turnaround time. Parameters s and T_s represent the index of a sequence of vehicles and the corresponding testing turnaround time.

Algorithm 1 OPN-Based Vehicle Sequence Optimization

```
Input: $m$, $n$, $D_n$
Output: $D_s$, $minTime$
1:  function Sequence Optimization($m$, $n$)
2:      $T_u \leftarrow 0$
3:      $batch \leftarrow ROUNDUP(D_n/m)$
4:      for each $i \leftarrow 1{:}batch$
5:          generate m! //Complete permutation of a batch
6:          for each $u \leftarrow 1{:}m!$
7:              $T_f \leftarrow$ Time stamp of first vehicle enters VTS;
8:              $T_l \leftarrow$ Time stamp of last vehicle leaves VTS;
9:              $T_u \leftarrow T_l - T_f$
10:         end for
11:         $minTime \leftarrow 1000,\ index \leftarrow 1$
12:         for each $s \leftarrow 1{:}m!$
13:             if $minTime > T_s$ then
14:                 $minTime \leftarrow T_s,\ index \leftarrow s$
15:             end if
16:         end for
17:         $D_s \leftarrow$ optimal sequence of each batch
18:         $minTime += minTime$
19:         $D_s += D_s$
20:     end for
21:     output Ds, minTime
22: end function
```

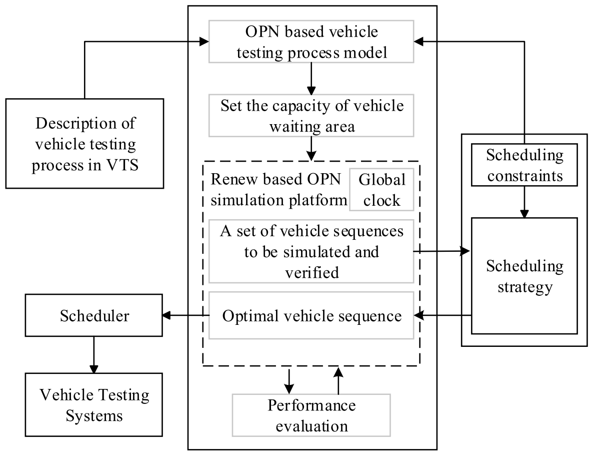
V. CASE STUDY AND PREFERENCE ANALYSIS

In this section, to illustrate the merits of the proposed method, based on Renew 2.5 on a PC with a core i7-6700 3.41GHz processor and 4 Gb memory, experiments and simulation are done by using the data sets taken from the log files of a universal vehicle testing system belonging to Shaanxi Automobile Group.

A. AN ILLUSTRATIVE CASE

In this part, we briefly introduce an industrial case which is taken from an existing VTS to illustrate the necessity to explore an optimal vehicle sequence for improving the testing efficiency. Suppose that there are exactly three vehicles waiting to be inspected. The testing requirements of these three vehicles are different. One vehicle declares for full testing, while the other two require the first station testing as well as the first and second station testing. According to the principle shown in Table 2, they are TR1, TR2, and TR3, respectively. Then, the complete permutation of these three testing requirements can be generated as shown in Fig. 7.

Take the first testing sequence: TR1 $\rightarrow$ TR2 $\rightarrow$ TR3 as an example, the vehicle that declares the full testing enters the VTS first, and then the vehicle that declares the first station testing, and the vehicle that enters the VTS last is the vehicle that declares the testing at the first and second stations. In the proposed OPN-based model, the corresponding tokens are generated for each of the three testing requirements. These tokens are deposited in place ENT in turn, and the time when



OPN based vehicle testing process model
Set the capacity of vehicle waiting area
Description of vehicle testing process in VTS
Renew based OPN simulation platform
Global clock
A set of vehicle sequences to be simulated and verified
Optimal vehicle sequence
Scheduling constraints
Scheduling strategy
Scheduler
Vehicle Testing Systems
Performance evaluation

FIGURE 6. Short-term scheduling framework.

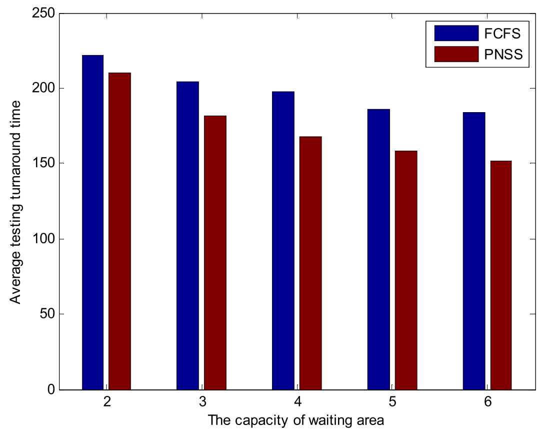
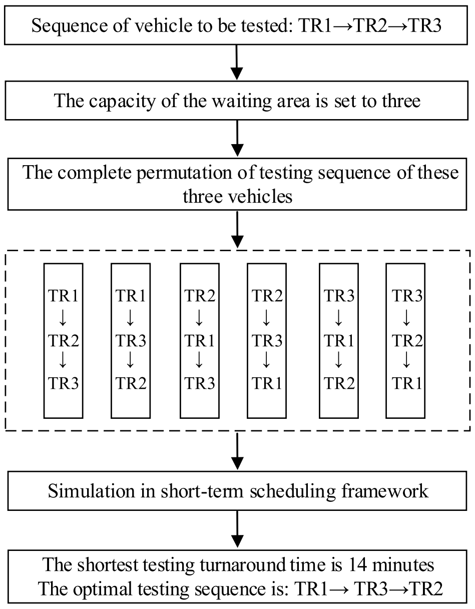


FIGURE 8. The average testing turnaround time of two strategies with different waiting area capacity.



Sequence of vehicle to be tested: TR1→TR2→TR3
The capacity of the waiting area is set to three
The complete permutation of testing sequence of these three vehicles
TR1 ↓ TR2 ↓ TR3
TR1 ↓ TR3 ↓ TR2
TR2 ↓ TR1 ↓ TR3
TR2 ↓ TR3 ↓ TR1
TR3 ↓ TR1 ↓ TR2
TR3 ↓ TR2 ↓ TR1
Simulation in short-term scheduling framework
The shortest testing turnaround time is 14 minutes
The optimal testing sequence is: TR1→ TR3→TR2

FIGURE 7. Process of generating the optimal vehicle sequence for three vehicles.

the first token is deposited is recorded as $T_{s_{1,1}}$ and the initial value is zero. According to the testing process in Fig. 5, these three tokens are also deposited in place OUT, and the time when the third token is deposited is recorded in the variable $T_{e_{3,3}}$ and the final value is 18. Based on the above analysis, we can conclude that the total testing turnaround time of this testing sequence is 18 minutes. Similarly, the testing turnaround time of other five sequences are 14, 16, 20, 18, and 20 minutes, respectively. Obviously, among the six optional sequences, the shortest testing turnaround time is 14 minutes, and the corresponding sequence is TR1 $\rightarrow$ TR3 $\rightarrow$ TR2.

Therefore, we can infer that a good vehicle testing sequence can be selected depending on the number of vehicles in the group for dispatching into the waiting area, which can greatly improve the overall efficiency of a VTS. In the follows, we discuss the rationality of the proposed method from four aspects.

B. CAPACITY ANALYSIS OF THE WAITING AREA

The number of vehicles that are dispatched into the waiting area in a group is dependent on the capacity of the waiting area. We can assume that this number is equal to the capacity of the waiting area. In order to analyze the impact of waiting area capacity on the performance of OPN-based scheduling strategy (PNSS), we randomly selected 120 data sets from the log files of Shaanxi Automobile Group's VTSs for experiments. Each data set contains 30 vehicles' testing requirements as given below:

TR1, TR1, TR1, TR1, TR3, TR7, TR1, TR5, TR1, TR2, TR1, TR1, TR3, TR4, TR1, TR7, TR6, TR1, TR5, TR1, TR1, TR2, TR4, TR1, TR6, TR5, TR1, TR3, TR1, TR7.

The waiting area capacity is set to 2, 3, 4, 5, and 6 such that data set can be divided into several groups and vehicles in each group are dispatched by both the FCFS and the proposed PNSS scheduling strategies. The average testing turnaround time of these 120 data sets under different waiting area capacity is shown in Fig. 8. Obviously, the average testing turnaround time of these 30 vehicles under the PNSS strategy is always less than FCFS-based strategy. At the same time, as the capacity of waiting area increases, the overall testing turnaround time shows a decreasing trend. This is mainly due to the fact that more vehicles in the waiting area can adjust the testing sequence, thus reducing unnecessary waiting time. As a proof, Table 6 shows the testing results of these 30 vehicles under different capacity for the PNSS.

C. COMPUTATIONAL COMPLEXITY FOR SOLVING OPTIMAL VEHICLE SEQUENCE

Due to unilateral emphasis on fairness and no extra computational time, a large number of existing VTSs still use a FCFS-based scheduling strategy. This is mainly because that, if a VTS employs the PNSS to solve the optimal vehicle sequence, which involves not only the computation

TABLE 5. The inspection sequence and time-consume of 30 vehicles in the PNSS.

Capacity of waiting area	2	3	4	5	6
Inspection sequence (Corresponding to the shortest testing time）	11 (18)	111 (24)	1111 (30)	11113 (32)	711113 (34)
	11 (18)	137 (18)	7513 (20)	75112 (24)	511112 (34)
	73 (10)	511 (22)	1112 (24)	41113 (30)	611374 (26)
	51（16）	112 (18)	1374 (20)	65711 (26)	511142 (30)
	12（12）	413 (18)	6511 (22)	61142 (20)	675113 (28)
	11 (18)	671 (14)	1142 (20)	75113 (26)	
	43 (12)	511 (22)	6513 (18)		
	71 (14)	142 (14)	71 (14)		
	61 (12)	651 (16)			
	51 (16)	713 (16)			
	12 (12)				
	14 (14)				
	65 (10)				
	13 (14)				
	71 (14)				
Total turnaround time	222	182	168	158	152

of the complete permutation of the vehicles in the waiting area but also the simulation of the OPN-based models, such that the computational complexity must be greater than the FCFS-based scheduling strategy. Therefore, it is necessary to analyze the computational complexity of the PNSS, and calculate the actual time consumption for solving the optimal vehicle sequences.

According to the algorithm developed in Section IV, the number of iterations of the two nested 'for' loops, such as **for each** $i \leftarrow 1$: *batch* and **for each** $u \leftarrow$ ***1***:$m!$, is determined by the length of all queued vehicles to be inspected and the capacity of the waiting area. If the capacity of the waiting area m is equal to D_n such that two nested 'for' loops are reduced to a single 'for' loop, whereas, the time complexity of the algorithm may be increased. Hence, the number of the executions of the loop body in the algorithm is

$$\frac{D_n}{m} \times 3 \times m! \tag{6}$$

Since D_n is a natural number with smaller value, the computational complexity of the algorithm is O((m − 1)!). Note that if m is not large, the problem can be efficiently solved. Thus, the key is to determine m such that it is computational feasible and at the same time the performance can be significantly improved. This is discussed next.

We randomly select a set of historical testing data from the log file of Shaanxi Automobile Group's VTSs to analyze the actual time consumption in solving the optimal vehicle sequences. This data set includes ten vehicles' testing requirements such as TR1, TR2, TR4, TR1, TR6, TR5, TR1, TR3, TR1, and TR7. Among them, four vehicles are required to perform a full testing; three vehicles need to be inspected by only one testing station; and the other three vehicles need to be inspected by two testing stations. In the experiment, we employ the first three vehicles, the first four vehicles, . . ., and total ten vehicles as eight groups of testing requirements and used as input data. Grouped testing requirements, testing turnaround time with two different scheduling strategies and actual time consumption required to obtain an optimal vehicle sequence are illustrated in Table 5.

Obviously, when the capacity of the waiting area is less than or equal to seven, the actual time consumption by using the PNSS is less than one second, meanwhile the PNSS greatly outperforms the FCFS-based scheduling strategy, it reduces the testing turnaround time by about 31%. In the case that the capacity of the waiting area is set to eight, nine, and ten, the time consumption for solving the optimal vehicle sequence is increased. Even so, for a group with eight to ten testing requirements, the computation time for obtaining the optimal vehicle sequence by the PNSS is still acceptable.

The key factor for this performance improvement lies in the operations of a VTS in a paced way by using the PNSS, that is, the PNSS avoids unnecessary waiting just by adjusting the vehicle inspection sequence, which not only speeds up the inspection progress, but also improves the equipment utilization. In Table 6, when the capacity of waiting area is six or seven, the actual time consumption for solving the optimal vehicle sequence problem in both cases is less than one second. Thus, the capacity of the waiting area can be designed to be six or seven. Note that, if the capacity is designed to be seven, the performance must be better than that obtained by capacity six in terms of testing turnaround time. Thus, it is reasonable to test the proposed method by using capacity six for the waiting area. Moreover, the construction cost of a VTS with capacity six is less than that with capacity seven. Hence, we evaluate the performance of the

TABLE 6. **The time consumption for solving each optimal vehicle sequence.**

Groups of testing requirements	Testing turnaround time (m)		Actual time consumption for each optimal vehicle sequence (s)
	PNSS	FCFS	
TR1, TR2, TR4	14	18	0.006
TR1, TR2, TR4, TR1	20	28	0.012
TR1, TR2, TR4, TR1, TR6	20	32	0.06
TR1, TR2, TR4, TR1, TR6, TR5	24	34	0.36
TR1, TR2, TR4, TR1, TR6, TR5, TR1	30	44	0.84
TR1, TR2, TR4, TR1, TR6, TR5, TR1, TR3	34	46	6.72
TR1, TR2, TR4, TR1, TR6, TR5, TR1, TR3, TR1	36	56	15.12
TR1, TR2, TR4, TR1, TR6, TR5, TR1, TR3, TR1, TR7	40	60	151.2

proposed method by setting the waiting area capacity to be six in the follows.

D. ANALYSIS OF RE-INSPECTION RATE

We further discuss the impact of different re-inspection rates on the performance of the PNSS strategy when the waiting area capacity is set to six. Re-inspection rate refers to the proportion of vehicles that need to be re-inspected in all the testing vehicles (including the full testing ones and the re-inspected ones). In this section, performance test is done for the re-inspection rate 0, 25%, 50%, 75%, and 100%, respectively. Take 30 vehicles' testing requirement data as a data set, and 120 sets of such data set are randomly extracted from the above mentioned testing system log files. Among them, the number of data sets with a re-inspection rate being zero is 19, the number of data sets with a re-inspection rate being 25% is 61, and the number of data sets with the re-inspection rate being 50%, 75%, and 100% is 30, seven, and three, respectively. According to the suggestion that the capacity is set to six, 30 vehicles in each data set can be divided into five groups for testing.

1) THE CASE WITH RE-INSPECTION RATE BEING ZERO

In this situation, the 19 selected data sets with re-inspection rate being zero are required to conduct the full testing, that is, every vehicle has to be tested at the first-, second- and third-stations sequentially. Experimental results show that the average test turnaround time is the same for both PNSS and FCFS, i.e., 210 minutes.

2) THE CASE WITH RE-INSPECTION RATE BEING 25%

In this situation, 75% of the selected vehicles are required to conduct the full testing, and re-check the other 25% of vehicles. Experiments are performed using the FCFS and PNSS strategies on the 61 data sets, the average testing turnaround time for these vehicles can be obtained as 192 and 172 minutes, respectively.

3) THE CASE WITH RE-INSPECTION RATE BEING 50%, 75%, AND 100%

Similarly, when the re-inspection rate is 50%, 75%, and 100%, we can obtain the average testing turnaround time for the thirty, seven, and three data sets, respectively. As shown in Fig. 9, the average testing turnaround time under the PNSS strategy is always less than or equal to the one of the FCFS. When the re-inspection rate is 75%, the total time difference is the largest between the two strategies.

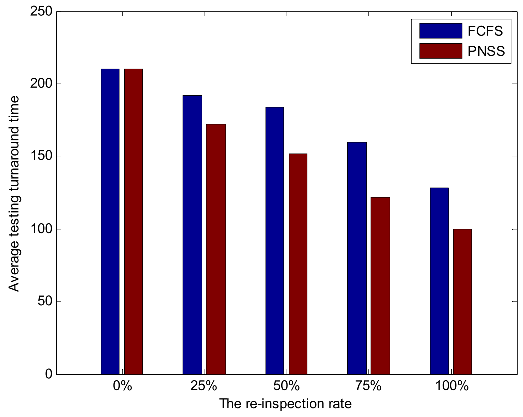


FIGURE 9. **Statistics at different re-inspection rates.**

E. ANALYSIS OF INDIVIDUAL VEHICLES THAT ARE ADVANCED OR POSTPONED

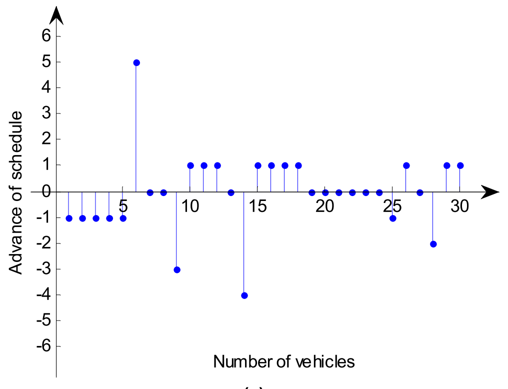
Although the FCFS-based scheduling strategy results in unnecessary wait, it is still the main scheduling method used by existing VTSs. The main reason is that the FCFS-based scheduling strategy has fairness from the perspective of individual vehicles. By the PNSS, the inspection of some early arrived vehicles is postponed, while the inspection of some late arrived vehicles is advanced. We discuss this issue as follows. Assume that the capacity of the waiting area is set to six. Then, Fig. 10 gives an intuitive description of this situation with an example.

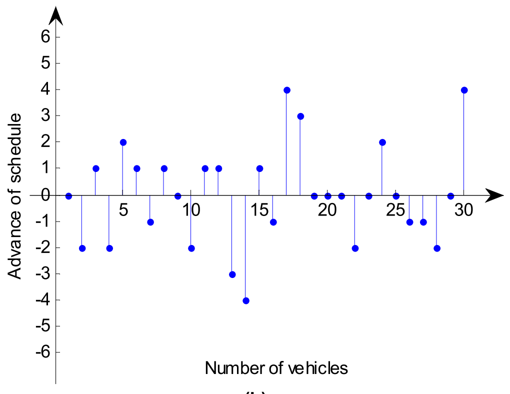
Fig. 10(a) is a case where the re-inspection rate is 25%. The corresponding vehicle arrival sequence and the optimal vehicle sequence obtained by the PNSS are as follows.

FCFS: TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR3 $\rightarrow$ TR7 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR2 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR5 $\rightarrow$ TR4 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR1 $\rightarrow$ TR6 $\rightarrow$ TR1 $\rightarrow$ TR4 $\rightarrow$ TR1 $\rightarrow$ TR1

TABLE 7. Proportion of vehicle testing being advanced or postponed.

Re-inspection rate (Number of data sets)	Positions of vehicle testing being advanced or postponed										
	-5	-4	-3	-2	-1	0	1	2	3	4	5
25% (61)	3.3%	1.1%	4.4%	2.3%	10%	40%	31.1%	3.3%	1.1%	1.1%	2.3%
50% (30)	4.4%	1.1%	4.4%	7.8%	13.3%	22.2%	33.4%	5.6%	2.3%	4.4%	1.1%
75% (7)	1.1%	5.6%	6.7%	12.2%	13.3%	10%	20%	6.7%	10%	13.3%	1.1%





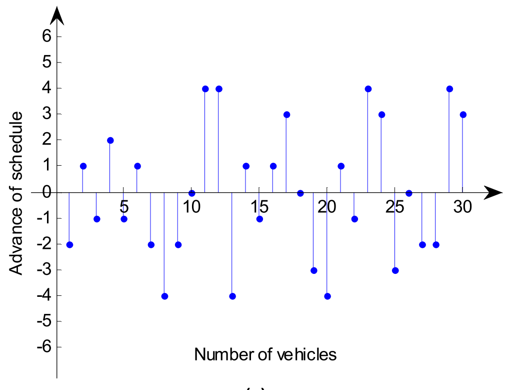


FIGURE 10. Statistics of vehicles being advanced or postponed. (a) Re-inspection rate is about 25%. (b) Re-inspection rate is about 50%. (c) Re-inspection rate is about 75%.

PNSS: TR7 → TR1 → TR1 → TR1 → TR1 → TR3 → TR1 → TR1 → TR1 → TR1 → TR1 → TR2 → TR5 → TR1 → TR1 → TR1 → TR1 → TR4 → TR1 → TR1 → TR1 → TR1 → TR1 → TR1 → TR6 → TR1 → TR1 → TR1 → TR1 → TR4

Compared with the FCFS-based scheduling strategy, the PNSS advances the 6th vehicle in the first group by five positions, while the rest of the vehicles are postponed by one position. In the second group, the first two vehicles remains in the original position, the third vehicle is postponed by three positions and the remaining three vehicles are all advanced by one positon. On the whole, 11 vehicles are dispatched in advance during the entire testing process, and 10 vehicles are dispatched in their original sequences. Meanwhile, nine vehicles are postponed, but only three vehicles are intentionally delayed by more than two positions. In addition, another issue worth noting in Fig. 10 is that as the re-inspection rate increases, the number of vehicles scheduled to be advanced or postponed in each data sets also shows an increasing trend. At the same time, the testing turnaround time shows a downward trend as the re-inspection rate is increased, i.e., decreasing from 198 minutes to 180 minutes, from 182 minutes to 144 minutes, and from 160 minutes to 122 minutes, respectively.

To further support the above conclusions, we perform experiments on 61 data sets with a re-inspection rate of 25%, 30 data sets with a re-inspection rate of 50%, and seven data sets with a re-inspection rate of 75%. We obtain the statistics on the data that the testing is advanced or postponed, as shown in Table 7. In the cases of three re-inspection rates, the proportion of vehicles being scheduled ahead of time or maintaining the original position reach 78.9%, 69% and 61.1%, respectively; while only 8.8%, 9.9% and 13.4% of the vehicles are postponed more than two positions. This must be the only cost of reducing the testing turnaround time for a group of vehicles to be inspected.

In summary, given a certain proportion of re-inspection vehicles in the queue to be inspected, the PNSS strategy can achieve shorter testing turnaround time and better equipment utilization as a whole.

VI. CONCLUSION

This paper is motivated by tackling the inefficiency of FCFS-based scheduling strategy in existing vehicle testing systems and optimizing the operations of a VTS by finding an optimal vehicle testing sequence. The short-term scheduling of vehicles in a VTS is a real industrial problem. We study the feasibility of an optimal scheduling method where a group of vehicles with different testing requirements continuously enter the testing system. We aim to achieve a minimum average testing turnaround time.

Instead of mathematical programming models, this paper studies the short-term scheduling problem of a VTS by employing object Petri net models. A scheduling framework, which basically consists of an object Petri net-based models for the operations of a VTS and a related scheduling strategy, is proposed. By the proposed method, it proposes the concept of a waiting area with adjustable capacity for grouping the vehicles to be inspected such that a solution can be found in an acceptable time. Then, we analyze the performance of the proposed scheduling strategy based on a key indicator, that is, the testing turnaround time. Experiment results show that the OPN-based scheduling strategy is a promising tool for the effective operations of a VTS.

REFERENCES

[1] X. M. Zhao, C. L. Nan, J. Ma, and X. F. Guo, "Study of braking data quasi-harmony and optimization method in auto brake function test," *China J. Highway Transp.*, vol. 16, no. 3, pp. 100–104, 2003.

[2] Z. D. Liu, S. Z. Yao, G. Y. Wang, and W. He, "A study on the roller specifications of heavy-duty chassis dynamometer," *Automot. Eng.*, vol. 29, no. 1, pp. 79–82, 2007.

[3] L.-Y. Guo, B. Liang, X.-M. Zhao, and A. G. Dong, "Application of empirical mode decomposition(EMD) in data smoothing for vehicle braking," *J. Chang'an Univ. Natural Sci. Ed.*, vol. 29, no. 4, pp. 97–100, 2009.

[4] Q.-J. Ma, "Research of key technique in chassis dynamometer for registered vehicles," *China Meas. Test*, vol. 35, no. 3, pp. 93–96, 2009.

[5] C. H. Qian, "Research on measuring and control system for automotive detection system based on embedded real-time system," Ph.D. dissertation, Colleges Automobile Eng., Jilin Univ., Changchun, China, 2008.

[6] X. M. Zhao, J. Ma, K. Guan, and W. Y. Shi, "Distributed computer-net automatic testing system on vehicle general performances," *J. Chang'an Univ. Natural Sci. Ed.*, vol. 23, no. 5, pp. 94–98, 2003.

[7] C. L. Chen, D. C. Yuan, H. H. Shao, and P. Sun, "Modeling and short-term scheduling of synthetic vehicle performance test line," *Control Theory Appl.*, vol. 19, no. 5, pp. 681–688, 2002.

[8] I. A. Chaudhry and A. A. Khan, "A research survey: Review of flexible job shop scheduling techniques," *Int. Trans. Oper. Res.*, vol. 23, no. 3, pp. 551–591, May 2016.

[9] C. A. Méndez, G. P. Henning, and J. Cerdá, "An MILP continuous-time approach to short-term scheduling of resource-constrained multi-stage flowshop batch facilities," *Comput. Chem. Eng.*, vol. 25, nos. 4–6, pp. 701–711, 2001.

[10] C.-W. Hui, H. A. J. van der Meulen, and A. Gupta, "A novel MILP formulation for short-term scheduling of multi-stage multi-product batch plants with sequence-dependent constraints," *Comput. Chem. Eng.*, vol. 24, no. 12, pp. 2705–2717, 2000.

[11] L. F. L. Moro and J. M. Pinto, "Mixed-integer programming approach for short-term crude oil scheduling," *Ind. Eng. Chem. Res.*, vol. 43, no. 1, pp. 85–94, 2004.

[12] S. Moon and A. N. Hrymak, "Mixed-integer linear programming model for short-term scheduling of a special class of multipurpose batch plants," *Ind. Eng. Chem. Res.*, vol. 38, no. 5, pp. 2144–2150, 1999.

[13] A. Nait-Sidi-Moh and A. El-Amraoui, "Modeling and optimization of cyclic hoist schedules in an electroplating line," *J. Syst. Sci. Syst. Eng.*, vol. 25, no. 4, pp. 469–490, 2016.

[14] C.-J. Liao and L.-M. Liao, "Improved MILP models for two-machine flowshop with batch processing machines," *Math. Comput. Model.*, vol. 48, nos. 7–8, pp. 1254–1264, 2008.

[15] N. Wu, C. Chu, F. Chu, and M. Zhou, "Short-term schedulability analysis of crude oil operations in refinery with oil residency time constraint using Petri nets," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 6, pp. 765–778, Nov. 2008.

[16] N. Wu, C. Chu, F. Chu, and M. Zhou, "Schedulability analysis of short-term scheduling for crude oil operations in refinery with oil residency time and charging-tank-switch-overlap constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 190–204, Jan. 2011.

[17] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[18] C. Girault and R. Valk, *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications*. Berlin, Germany: Springer, 2013.

[19] N. Q. Wu and M. Zhou, "Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 446–454, Apr. 2012.

[20] Q. Zhu, M. Zhou, Y. Qiao, and N. Wu, "Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 3, pp. 389–400, Mar. 2018.

[21] S. Wang, D. You, and C. Seatzu, "A novel approach for constraint transformation in Petri nets with uncontrollable transitions," *IEEE Trans. Syst. Man Syst.*, vol. 48, no. 8, pp. 1403–1410, Aug. 2018.

[22] N. Wu, M. Zhu, L. Bai, and Z. Li, "Short-term scheduling of crude oil operations in refinery with high-fusion-point oil and two transportation pipelines," *Enterprise Inf. Syst.*, vol. 10, no. 6, pp. 581–610, 2016.

[23] Y. Chen, Z. W. Li, A. Al-Ahmari, N. Wu, and T. Qu, "Deadlock recovery for flexible manufacturing systems modeled with Petri nets," *Inf. Sci.*, vol. 381, pp. 290–303, Mar. 2017.

[24] F. J. Yang, N. Q. Wu, Y. Qao, M. C. Zhou, and Z. W. Li, "Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 502–516, Mar. 2017.

[25] S. Zhang, N. Wu, Z. Li, T. Qu, and C. Li, "Petri net-based approach to short-term scheduling of crude oil operations with less tank requirement," *Inf. Sci.*, vol. 417, pp. 247–261, Nov. 2017.

[26] Y. An, X. Zhao, and R. Li, "Modeling and simulation of vehicle tesing system based on colored Petri nets," *Comput. Integr. Manuf. Syst.*, vol. 18, no. 9, pp. 1991–2002, 2012.

[27] L. Bai, N. Wu, Z. Li, and M. Zhou, "Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 10, pp. 1456–1467, Oct. 2016.

[28] B. Huang, M. Zhou, P. Zhang, and J. Yang, "Speedup techniques for multiobjective integer programs in designing optimal and structurally simple supervisors of AMS," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 1, pp. 77–88, Jan. 2018.

[29] N. Wu, Z. W. Li, and T. Qu, "Energy efficiency optimization in scheduling crude oil operations of refinery based on linear programming," *J. Cleaner Prod.*, vol. 166, pp. 49–57, Nov. 2017.

[30] Y. An, P. Chen, J. Yang, and L. J. Yang, "A Petri nets based scheduling strategy for vehicle safety performance test system," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Zhuhai, China, Mar. 2018, pp. 1–6.

[31] N. Wu and M. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 203–209, Jan. 2012.

[32] N. Wu, F. Chu, C. Chu, and M. Zhou, "Petri net modeling and cycle-time analysis of dual-arm cluster tools with wafer revisiting," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 1, pp. 196–207, Jan. 2013.

[33] N. Wu, M. Zhou, and Z. W. Li, "Short-term scheduling of crude-oil operations: Enhancement of crude-oil operations scheduling using a Petri net-based control-theoretic approach," *IEEE Robot. Autom. Mag.*, vol. 22, no. 2, pp. 64–76, Jun. 2015.

[34] Y. Hou, N. Wu, M. Zhou, and Z. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 517–530, Mar. 2017.

[35] Y. Qiao, N. Wu, F. Yang, M. Zhou, and Q. Zhu, "Wafer sojourn time fluctuation analysis of time-constrained dual-arm cluster tools with wafer revisiting and activity time variation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 622–636, Apr. 2018.

[36] M. Zhou and B. Hrúz, *Modeling and Control of Discrete-event Dynamic Systems: With Petri Nets and Other Tools*. Berlin, Germany: Springer, 2007.

[37] N. Wu and M. Zhou, "Colored timed Petri nets for modeling and analysis of cluser tools," *Asian J. Control*, vol. 12, no. 3, pp. 253–266, 2010.

[38] E. López-Mellado, N. Villanueva-Paredes, and H. Almeyda-Canepa, "Modelling of batch production systems using Petri nets with dynamic tokens," *Math. Comput. Simul.*, vol. 67, no. 3, pp. 541–558, 2005.

[39] R. Valk, "Object Petri nets: Using the nets-within-nets paradigm," in *Lectures on Concurrency and Petri Nets: Advances in Petri Nets*, vol. 3098. 2004, pp. 819–848.

[40] K.-B. Michael, "A survey of decidability results for elementary object systems," *Fundam. Inform.-Logics, Agents, Mobility*, vol. 130, no. 1, pp. 99–123, 2014.

YISHENG AN (M'16) received the B.S. degree in computer engineering from Shaanxi Normal University, Xi'an, China, in 1995, and the M.S. and Ph.D. degrees in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2007, respectively. In 2010, he was a Visiting Scholar with the Department of Computer Engineering, Eastern Washington University, Spokane, WA, USA. He is currently a Professor with the Department of Computer Science and Engineering, School of Information Engineering, Chang'an University, Xi'an. He has authored or co-authored over 50 peer-reviewed journal papers. His research interests include discrete event systems, Petri net theory and applications, and intelligent transportation systems. He has served as a reviewer for a number of journals.

NAIQI WU (M'04–SM'05) received the B.S. degree in electrical engineering from the Anhui University of Technology, Huainan, China, in 1982, and the M.S. and Ph.D. degrees in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 1985 and 1988, respectively. From 1988 to 1995, he was with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. From 1995 to 1998, he was with Shantou University, Shantou, China. He moved to the Guangdong University of Technology, Guangzhou, China, in 1998. In 2013, he joined the Macau University of Science and Technology, Taipa, Macau, where he is currently a Professor with the Institute of Systems Engineering. His research interests include production planning and scheduling, manufacturing system modeling and control, discrete event systems, Petri net theory and applications, intelligent transportation systems, and energy systems. He has authored or co-authored one book, five book chapters, and over 140 peer-reviewed journal articles. He was an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics—Part C, the IEEE Transactions on Automation Science and Engineering, and the IEEE Transactions on Systems, Man, and Cybernetics: Systems, and the Editor-in-Chief of the *Journal of Industrial Engineering*. He is an Associate Editor of *Information Sciences* and the IEEE/CAA Journal of Automatica Sinica.

PEI CHEN (S'17) received the B.S. degree from the Xi'an University of Posts and Telecommunications, Shaanxi, China, in 2016. She is currently pursuing the M.S. degree with Chang'an University, Xi'an. Her research interests include discrete event systems, Petri net theory and applications, and intelligent transportation systems.