IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 16, 2018, accepted July 20, 2018, date of publication August 2, 2018, date of current version August 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2862390

Exploring Auto-Generation of Network Models With Performance Evaluation Process Algebra

JIE DING[1,2], RUI WANG[2], XIAO CHEN[3,4], (Member, IEEE), AND YING-EN GE[5]
[1]China Institute of FTZ Supply Chain, Shanghai Maritime University, Shanghai 201306, China
[2]School of Information Engineering, Yangzhou University, Yangzhou 225127, China
[3]School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China
[4]State Key Laboratory of Software Development Environment, Beihang University, Beijing 100083, China
[5]College of Transport & Communications, Shanghai Maritime University, Shanghai 201306, China

Corresponding authors: Xiao Chen (xiaochen@ujs.edu.cn) and Jie Ding (jieding78@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 61472343, 61702233, and 71431003, in part by the Natural Science Foundation of Jiangsu Province (China) under Grant BK20160543, in part by the State Key Laboratory of Software Development Environment under Grant SKLSDE-2017KF-03, and in part by the Blue Project for Young Research Leader of Jiangsu (China).

ABSTRACT Formal method plays an important role in modeling large scale concurrent networks through its efficient model construction and analysis. Taking urban road networks and public transportation systems as examples, such models can be defined in a formal method in order to investigate the performance of current bus-line deployment based on a given road network. This paper considers how to efficiently build a formal model based on such an original prototype of the specified road network and transportation system. As the model prototype can be represented by a directed graph that is then transformed into a numerical incidence matrix, we proposed an algorithm, in this paper, to assist the construction of formal models by sorting all potential compositional structures and components based on the previously obtained numerical incidence matrix. Thereafter, a performance evaluation process algebra-based formal model can be automatically generated on the basis of sorted compositional structures, which extends the use of formal method for large-scale and comprehensive network, and system modeling. The findings reveal that the proposed algorithm can efficiently find all potential compositional structures that include all potential components and related activity flows in models.

INDEX TERMS Formal method, PEPA, incidence matrix, compositional structures, sorting algorithm.

I. INTRODUCTION

Stochastic Process Algebra (SPA), as a formal mathematical modeling language, has been widely used for modeling and evaluating performance of parallel systems, which allows Continuous Time Markov Chains (CTMCs) [1] and composition property. Since SPA were proposed in the early 1990s, various stochastic process algebras, such as PEPA [2]–[5], Interactive Markov Chains (IMC) [6] and Modeling and Description language for Stochastic Timed systems (MoDest) [7], have enjoyed great success in practical applications.

Apart from this, the numerical representation of SPA has also been widely applied. In [8] and [9], researchers have used algebraic tools to represent PEPA models in numerical form and to express the state space generated from PEPA models in the form of numerical vectors. In addition, [8] has established a link between the syntax and numerical representation of PEPA. Moreover, Ding has proposed an algorithm to represent a PEPA model through the form of numerical matrix (i.e. activity matrix)), and proved that this algebraic form of numerical representation can effectively alleviate state space explosion in order to model large and complex concurrent systems. Finally, a relatively simple and intuitive method was developed to check potential deadlocks of the state space by representing it with numerical vectors.

Concurrent systems are easy to be depicted by directed graphs. Alternatively, numerical matrices (i.e. incidence matrix) can be used to represent the architecture of concurrent systems. According to the numerical description method(i.e. incidence matrix), we can obtain the structure information of a concurrent system. However, the incidence matrix just specifically describes the relationship between each state in the system. Moreover, it cannot give the detailed description on other properties of a system (e.g. the composition of a concurrent system). Furthermore, when the system scale and structure become large and complex, it is difficult to

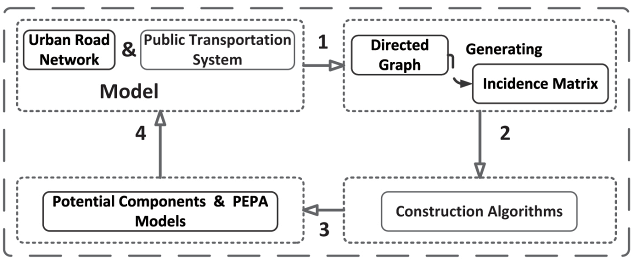


FIGURE 1. Algorithm construction process based on the urban transportation network model.

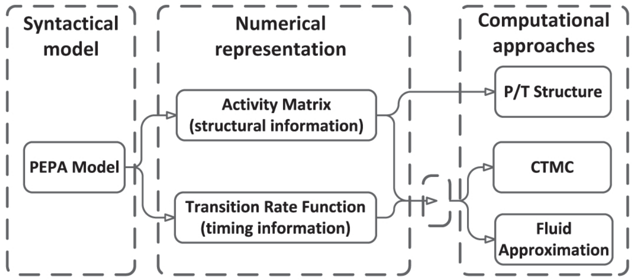


FIGURE 2. A three-levels PEPA modeling framework [8].

investigate its compositional structure using incidence matrix. Furthermore, PEPA modeling language can describe the concurrent structure of a system well (i.e. on the premise of ensuring the overall performance of a system, divide the system into multiple subsystems that are connected with shared activities) as it inherits advantages of a composition property from the stochastic process algebra (SPA). Furthermore, PEPA language describes and represents a system through the syntactical representation. Regarding the incidence matrix, PEPA language is more intuitive and convenient to carry out deep research on the structure of large scale and complex systems.

In this situation, based on the incidence matrix, a specific construction algorithm is proposed on the composition property of PEPA and incidence matrix. This algorithm is mainly to sort out all potential compositional structures and components in line with syntactic properties of PEPA language. That is to say, through the proposed algorithm, we can construct all potential PEPA models corresponding to the systems. Therefore, according to the composition property, a complete and complex concurrent system can be divided into multiple sub-models based on PEPA that consist of different components. After analysing and researching the structure of each component based on the generated PEPA models, it is easier and faster to construct and build a complete PEPA model of the given concurrent system. In addition, as this paper focuses on qualitative analysis of PEPA (i.e. take advantage of the structure of PEPA to investigate the concurrent systems), the corresponding rate of each labelled activity will be considered on reasonable assumptions.

In order to describe the sorting algorithm proposed this paper directly and effectively, a real world sample of concurrent system (i.e. the urban road network and public transportation system) is used for intuitive illustration shown in Fig.1. Nowadays, the growth of vehicles exceeds the improvement of transportation infrastructures, which results in the emergence of all kinds of traffic problems (e.g. traffic congestion, traffic infrastructure management) [10], [11]. For most cities, traffic congestion and the unbalanced road utilization have always been a more prominent and serious problem [12]. Hence, for this traffic and environment problems that almost all cities are meeting and many large cities choose to implement the vehicle limit line policy. Apart from this, carpool and public transportation are becoming a more popular traffic mode. For the carpool travel in Megalingam *et al.* [13], Arnould *et al.* [14], Huang *et al.* [15], and Lin *et al.* [16] present an intelligent carpool system, which consists of the mobile client module and the cloud global carpool system. Meanwhile, authors also proposed an intelligent carpool matching algorithm to analyse the specific performance of the system. Moreover, in [17], Lalos *et al.* study how to use positioning system to implement a dynamic network which can support car and taxi pool and maximize the utilization of the spare seats of cars and taxis. Furthermore, public transportation system also plays an important role in alleviating these traffic and environment problems. Therefore, a well designed bus line deployment and a plurality of alternative public transportation deployment based on the urban road network are also of great significance. Hence, according to the directed graph, our research first give the corresponding incidence matrix, and then sort the incidence matrix in order to construct all potential PEPA models (i.e. all potential and optional concurrent deployment of bus lines) of the system by the proposed algorithm. Finally, a simple analysis is generated for performance evaluation on the obtained results (i.e. the multiple public transportation deployments) of the algorithm.

Section II illustrates a brief introduction to PEPA, labelled activities, pre and post sets of labelled activities and incidence matrix. Section III provides the public transportation road network and the corresponding incidence matrix. Section IV gives the specific model construction algorithm. Section V demonstrates the algorithm details about how to construct the incidence matrix which is corresponding to the model of a simplified urban public transportation road network in Section III and make analyses based on the observed results from the construction algorithm. Section VI concludes the whole research.

II. BRIEF INTRODUCTION TO STOCHASTIC PROCESS ALGEBRA

A. INTRODUCTION TO PEPA

As a stochastic process algebra, PEPA can be applied to describe concurrent systems underlying stochastic process in the form of interactive components which engage in activities.

As shown in Fig.2, a three-level modeling process with PEPA is designed. It is obvious that each PEPA model implies an underlying Place/Transition (P/T) structure. From a perspective of graph, a PEPA model can be represented by an especial class of stochastic Petri nets (SPNs) [18]. From the algebraic point of view, activity matrix can be used to represent the P/T structure of PEPA model and depict the implementation process of each place of PEPA models through changes in the value of incidence matrix elements.

However, other than traditional process algebras, PEPA adopts a pair (α, r) to represent each activity in which α is the type of activity and r is the rate of activity. In addition, the PEPA language consists of several combinators and the semantics of structured operations of PEPA is introduced in [2], [8], [9], and [19]. The syntax of PEPA is defined as:

$$\begin{aligned} S &::= (\alpha, r).S | S + S | C_S \\ P &::= P \underset{L}{\bowtie} P | P/L | C \end{aligned} \tag{1}$$

In the above definition, S represents a sequential component, and P represents a parallel model component. Meanwhile, C is a constant which stands for either S or P. C_S denotes a constant which is applied to denote sequential components. Moreover, to ensure each component of a PEPA model cooperated in sequential process, it is necessary to separate all of these constants from syntactic level.

Prefix: Prefix manifests in the form of $(\alpha, r).S$. It is applied to depict the execution of a component which has a definite pre-activity (α, r) and generates the following component S.

Choice: Choice can be described as $S + Q$ and the components both belong to the same system. Moreover, the two components are both enabled to be executed. However, the component with the faster rate wins in racing.

Hiding: Hiding can be described as P/L. It means that all activities in L can be seen as private activities of component P. In other words, component P remains unchanged after executing activities in L.

Cooperation: Cooperation can be abstracted as $P \underset{L}{\bowtie} Q$. It means that both components P and Q must execute the activities of L cooperatively. When $L = \emptyset$, it indicates that components P and Q are both individual and can manifest in the form of $P \| Q$.

Constant: Constants can be described as an equation in the form of $S \stackrel{\text{def}}{=} T$. It means that the component S is similar to component T after being executed.

According to the operational semantic rules referring to [2], [3], [20], and [21] for details, a PEPA model can be considered as a system with multiple labelled transitions:

$$(C, Act, \{\xrightarrow{(\alpha,r)} | (\alpha, r) \in Act\}) \tag{2}$$

where C is the set of all components, Act can be regarded as the set of activities and $\xrightarrow{(\alpha,r)}$ is used to denote the multi-relation.

Then, the specific operational semantics of PEPA are shown as follows (see details in [2]):

$$\text{Prefix:} \quad \frac{}{(\alpha, r).E \xrightarrow{(\alpha,r)} E}$$

$$\text{Choice:} \quad \frac{E \xrightarrow{(\alpha,r)} E'}{E + F \xrightarrow{(\alpha,r)} E'}, \qquad \frac{F \xrightarrow{(\alpha,r)} F'}{E + F \xrightarrow{(\alpha,r)} F'}$$

$$\text{Cooperation:} \quad \frac{E \xrightarrow{(\alpha,r)} E'}{E \underset{\{L\}}{\bowtie} F \xrightarrow{(\alpha,r)} E' \underset{\{L\}}{\bowtie} F} (\alpha \notin L),$$

$$\frac{F \xrightarrow{(\alpha,r)} F'}{E \underset{\{L\}}{\bowtie} F \xrightarrow{(\alpha,r)} E \underset{\{L\}}{\bowtie} F'} (\alpha \notin L)$$

$$\frac{E \xrightarrow{(\alpha,r_1)} E' \quad F \xrightarrow{(\alpha,r_2)} F'}{E \underset{\{L\}}{\bowtie} F \xrightarrow{(\alpha,R)} E' \underset{\{L\}}{\bowtie} F'} (\alpha \in L),$$

where

$$R = \frac{r_1}{r_\alpha(E)} \frac{r_2}{r_\alpha(F)} min(r_\alpha(E), r_\alpha(F)),$$

where $r_\alpha(E)$, $r_\alpha(F)$ are the apparent rates of action of type α in the component E and F respectively.

$$\text{Hiding:} \quad \frac{E \xrightarrow{(\alpha,r)} E'}{E/L \xrightarrow{(\alpha,r)} E'/L} (\alpha \notin L), \qquad \frac{E \xrightarrow{(\alpha,r)} E'}{E/L \xrightarrow{(\tau,r)} E'/L} (\alpha \in L)$$

$$\text{Constant:} \quad \frac{E \xrightarrow{(\alpha,r)} E'}{A \xrightarrow{(\alpha,r)} E'} (A \stackrel{\text{def}}{=} E)$$

B. INTRODUCTION TO NUMERICAL REPRESENTATION OF PEPA

From Section II-A, there are two types of activities in PEPA, which are individual activities and shared activities. Here, definition of labelled activities is given based on these two types of activities. Furthermore, three other definitions are specified about pre set, post set and incidence matrix.

Definition 1 (Individual Labelled Activity): If α is an individual activity, and

$$P \in pre(\alpha); \quad Q \in post(P, \alpha) \tag{3}$$

then α can be labelled as $\alpha^{P \rightarrow Q}$.

Definition 2 (Shared Labelled Activity): If α is a shared activity, and

$$\begin{aligned} &Q_1, Q_2, \cdots, Q_n \in post(pre(\alpha)[1], \alpha) \\ &\quad \times post(pre(\alpha)[2], \alpha) \times \cdots \times post(pre(\alpha)[n], \alpha) \end{aligned} \tag{4}$$

then α can be labelled as α^{ω}, where

$$\begin{aligned} \omega = (&pre(\alpha)[1] \rightarrow Q_1, \ pre(\alpha)[2] \rightarrow Q_2, \\ &\cdots, \ pre(\alpha)[n] \rightarrow Q_n) \end{aligned} \tag{5}$$

Definition 3 (Pre Set): If a labelled activity α can be executed by the place P, i.e. $P \xrightarrow{\alpha} \cdot$, then P is named as

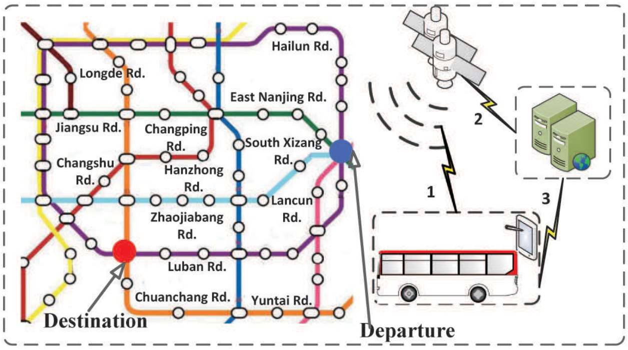


FIGURE 3. Intelligent transportation system based on a local traffic network [22].

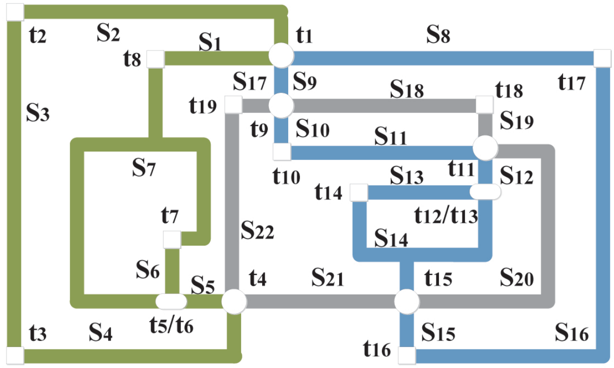


FIGURE 4. Abstracted traffic network model based on Fig.3.

a pre place of α. The set pre(α) is used to store all pre places of α.

Definition 4 (Post Set): If the place T can be obtained by performing the labelled activity α, that is $\cdot \xrightarrow{\alpha} T$, then T is a post place of α. The set post(α) is used to store all post places of α.

Definition 5 (Activity Matrix): For a certain PEPA model, if the number of labelled activities is $N_{A_{label}}$ and the number of distinct local derivatives is N_D. Then, it is easy to know the corresponding activity matrix can be described as $C_{N_D \times N_{A_{label}}}$ and the elements of C can be defined as follows:

$$C_{(P_i,\alpha_j)} = \begin{cases} +1, & \text{if } P_i \in \text{post}(\alpha_j) \\ -1, & \text{if } P_i \in \text{pre}(\alpha_j) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

III. AN URBAN ROAD NETWORK SYSTEM AND RELATED INCIDENCE MATRIX

This section first shows a map of the urban road network displayed in Fig. 3 of Section III-A. According to the map, we further generate an abstracted road network model (seeing Fig. 4) that will be applied in the following investigation. Next, Section III-B presents specific Petri-net modeling rules of the road network model shown in Fig. 4 through summarizing the characteristics of all types of road intersections and road sections. Finally, the Section III-C demonstrates the Petri-net-based traffic model and the corresponding incidence matrix of the given urban road network shown in Fig. 4.

A. URBAN ROAD NETWORK

In this section, the real-world scenario is simplified and abstracted by giving the corresponding model scheme of the urban road network and public transportation system. As shown in Fig.3, the white circles and ovals represent various road junctions (such as crossing intersections, three-way intersections and curve intersections). The lines between intersections represent each road section. Furthermore, Fig. 3 also describes the general architecture of the public transportation system.

In order to facilitate the modeling and analysis, Fig.3 depicts the road network model and sets labels on the sections (e.g., $S_1, S_2, \cdots, S_{22}$) and intersections (e.g., $t_1, t_2, \cdots, t_{19}$).

B. DESCRIPTION OF PETRI NET MODELING RULES

Based on the abstracted urban road network in Fig.4, we will next give the specific Petri net modeling rules for all types of traffic elements (i.e. intersections, T-intersections, Bend-intersections and sections). The specific modeling rules and their descriptions are shown as follows:

1. Descriptions of modeling rules for intersections in urban road network:

- First, we abstract the intersections as concurrent transitions in Petri net models.
- Second, we abstract the corresponding four sections connected to intersection as input and output places of the concurrent transition under Petri net modeling.

As shown in Fig.5, taking the intersection t_1 as an example. The four sections connected to t_1 are S_1, S_2, S_8 and S_9, respectively. Then, in the corresponding Petri net modeling rules, S_1 and S_2 are used as input places of concurrent transition t_1. Similarly, S_8 and S_9 are used as output places of t_1. The specific modeling diagram of the intersection modeling effect is shown in Fig.5.

2. Descriptions of modeling rules for bend intersections in urban road network:

- First, we abstract the Bend-intersections as individual transitions in Petri net.
- Second, we abstract the corresponding two sections connected to Bend-intersection as input and output places of the individual transition in Petri net.

As shown in Fig.5, taking the Bend-intersection t_7 as an example. The two sections connected to t_7 are S_6 and S_7. Then, in the corresponding Petri net modeling rules, S_6 is used as input place of individual transition t_7. Similarly, s_7 is used as output place of t_7. The specific modeling diagram of the Bend-intersection modeling effect is shown in Fig.5.

3. Descriptions of modeling rules for T-intersections in urban road network:

- First, we abstract the T-intersections as choice transitions in Petri net.

TABLE 1. Incidence matrix of the road network model in Fig.4.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}
S_1	−1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
S_2	1	−1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S_3	0	1	−1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S_4	0	0	1	−1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S_5	0	0	0	1	−1	−1	0	0	0	0	0	0	0	0	0	0	0	0	0
S_6	0	0	0	0	1	0	−1	0	0	0	0	0	0	0	0	0	0	0	0
S_7	0	0	0	0	0	1	1	−1	0	0	0	0	0	0	0	0	0	0	0
S_8	−1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
S_9	1	0	0	0	0	0	0	0	−1	0	0	0	0	0	0	0	0	0	0
S_{10}	0	0	0	0	0	0	0	0	1	−1	0	0	0	0	0	0	0	0	0
S_{11}	0	0	0	0	0	0	0	0	0	1	−1	0	0	0	0	0	0	0	0
S_{12}	0	0	0	0	0	0	0	0	0	0	1	−1	−1	0	0	0	0	0	0
S_{13}	0	0	0	0	0	0	0	0	0	0	0	1	0	−1	0	0	0	0	0
S_{14}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	−1	0	0	0	0
S_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−1	0	0	0
S_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−1	0	0
S_{17}	0	0	0	0	0	0	0	0	−1	0	0	0	0	0	0	0	0	0	1
S_{18}	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	−1	0
S_{19}	0	0	0	0	0	0	0	0	0	0	−1	0	0	0	0	0	0	1	0
S_{20}	0	0	0	0	0	0	0	0	0	0	1	0	0	0	−1	0	0	0	0
S_{21}	0	0	0	−1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
S_{22}	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−1

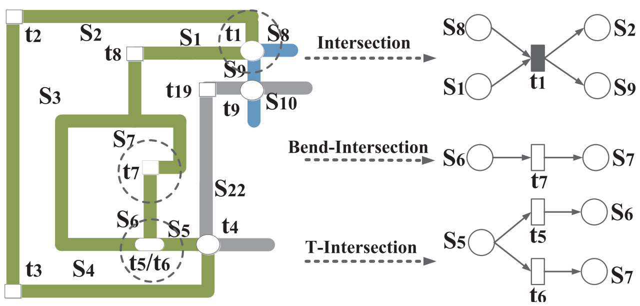


FIGURE 5. Specific Petri net modeling rules for all intersections based on the simplified urban road network of Fig.3.

• Second, we abstract the corresponding three sections connected to T-intersection as input and output places of the choice transition in Petri net.

As shown in Fig.5, taking the T-intersection t_5/t_6 as an example. Because it divides the choice transitions into two columns in the following incidence matrix, t_5/t_6 is adopted to represent the corresponding T-intersection uniformly. The three sections connected to t_7 are S_5, S_6, and S_7. Then, in the corresponding Petri net modeling rules, S_5 is used as input place of choice transition t_5/t_6. Similarly, S_6 and S_7 are used as output places of t_5/t_6. The specific diagram of the T-intersection modeling effect is shown in Fig.5.

C. PETRI NET MODEL AND INCIDENCE MATRIX

This section defines the Petri net model (i.e., Fig. 6) of the given urban road network model according to the simplified urban road network sketch showing in Fig.4, and also generates the corresponding Petri net modeling rules in Section III-B. Furthermore, based on the numerical representation (i.e. incidence matrix) rules of PEPA, the corresponding incidence matrix can be obtained (as shown in Table 1).

From the elements of incidence matrix, it is easy to obtain and describe the overall structure information of the road network in Fig.4. Here, we give some brief descriptions on the incidence matrix. From Section II-B, it is easy to know the meaning of the elements in the incidence matrix. In addition, columns and rows of the incidence matrix represent intersections and sections of the road network respectively. Then, we use the first column of the incidence matrix as an example, it means through the intersection t_1, section S_1 (i.e. the element "−1") will reach section S_2 (i.e. the element "1") and section S_8 (i.e. the element "−1") can reach section S_9 (i.e. the element "1").

From the incidence matrix (Table 1), it is difficult to investigate the road network structure. Meanwhile, it is also difficult to obtain the public transportation lines deployment. Therefore, based on the incidence matrix, an algorithm is proposed in Section IV to construct all qualified PEPA models of the urban road network (i.e. bus-line deployment). Finally, based on the generated PEPA models, experiments will be conducted for analysing the road network and public transportation system.

IV. INCIDENCE MATRIX BASED PEPA MODEL CONSTRUCTION ALGORITHM

In this section, the specific algorithm is demonstrated and used to construct all qualified PEPA models from the given numerical representation of a model (i.e. incidence matrix). The proposed algorithm aims to sort out the potential structures of a given model by analysing its corresponding incidence matrix. This sorting idea has been initially and successfully used to extract the compositional structure of Petri net [23]. At the same time, in order to achieve all potential PEPA models that meet the syntax of PEPA language, a series of operations are taken on elements of incidence

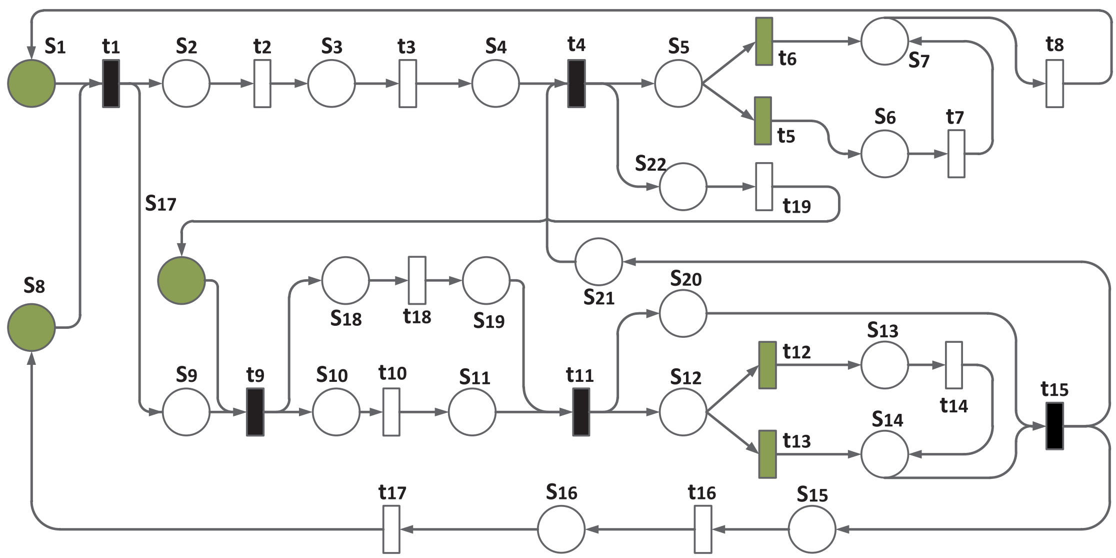


FIGURE 6. The corresponding Petri net model of the simplified urban road network of Fig.3.

matrix depending on composition of PEPA, labelled activities and the underlying P/T structures.

To facilitate the implementation of algorithm, this operation process is abstracted into two sub-algorithms. Algorithm 1 mainly aims to carry out Cartesian products, permutations and combinations on shared activities and preliminarily generate all possible classification without further detailed judgements. The algorithm 2 is based on the results of Algorithm 1. It conducts a specific and detailed judgements on the results, removes the case which does not meet the syntax of PEPA and generates all qualified PEPA models. Moreover, as shown in Table 2, a brief introduction is conducted to some acronyms used in Algorithms 1 and 2.

A. PARAMETER DESCRIPTIONS

In this subsection, in order to facilitate the description of proposed algorithms, all related parameters and their descriptions are summarized in Table 2.

B. ALGORITHM SCENARIO

This subsection gives the specific flow of PEPA model construction algorithm. In order to construct all potential and qualified PEPA models of a given Petri net, we design the corresponding operation steps, constraint conditions and judgement conditions. Furthermore, for this purpose, we also take incidence matrix as the processing object and combine PEPA semantics and features of components in PEPA models. The specific algorithm flow is introduced as:

(1) Label all rows and columns of the given incidence matrix with: $S_1, S_2, \cdots, S_n$ and $t_1, t_2, \cdots, t_m$.

(2) Traverse the incidence matrix based on each column:

(2.1) Give the judgement of all elements c_{ij} in the column: If $c_{ij} = -1/c_{ij} = 1$, then deposit c_{ij} into pre/post set of the labelled activity t_j.

TABLE 2. Descriptions of parameters in Algorithm 1 and 2.

Varibles	Explanation
C	Incidence matrix.
$\#P$; $\#T$	The number of rows (places) and columns (labeled activities) of C respectively.
$negCount$	The number of "-1" (pre($\cdot$)) of every column of C.
pre($\cdot$); post($\cdot$)	The pre set and post set of every labeled activity.
I_{Act}	The set of pairs (pre(t_i),t_i,post(t_i)) of every individual labeled activity.
TC_{Act}	Store pairs (pre(t_j),t_j,post(t_j)) of every shared labelled activity that after $CPPC$.
C_{Act}	The set of TC_{Act} of all shared labeled activities.
$\#C_{Act}$	The number of objects in set C_{Act}.
CP_{Act}	Store results of all objects of C_{Act} after $CPPC$.
RS_{temp}	Store all possible classifications without specific judgment.
S_{Act}	Store all shared labeled activities.
$\#RS_{temp}$	The number of all objects in set RS_{temp}.
S_{com}	Store every possible component.
$\#S_{com}$	The number of all objects in S_{com}.
$\#T_{numShared}$	The number of one shared labeled activity in a component.
RS_{init}	Store components of qualified PEPA models.
RS	Store all qualifying PEPA models.

(2.2) After traversing a column, use parameter $negCount$ denote the number of $c_{ij} = -1$. (2.2.1) A decision will be made through counting the value of $negCount$ once finish traversing one column. If $negCount = 1$, the labelled activity t_j is an individual activity, and deposit the triple (S_i, t_j, S_k) into set I_{Act} (assume $S_i \in$ pre(t_j) and $S_k \in$ post(S_i, t_j)).

(2.2.2) It means that t_j is a shared labelled activity, if $negCount \geq 2$. Then, conduct $CPPC$ on elements of pre(t_j) and post(t_j). Thereafter, deposit the results into set C_{Act}. (e.g. If pre(t_j) $= \{S_i, S_l\}$ and post(t_j) $= \{S_m, S_n\}$, the results can be obtained in the form of: $\{\{(S_i, t_j, S_m)(S_l, t_j, S_n)\}\{(S_i, t_j, S_n)(S_l, t_j, S_m)\}\}$).

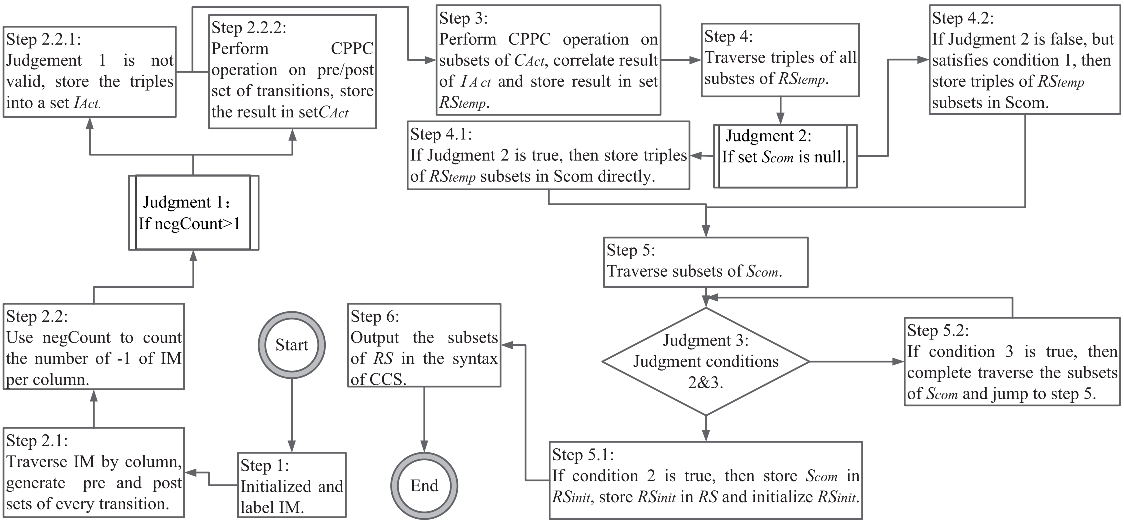


FIGURE 7. The proposed PEPA models construction algorithm flow chart.

(3) Conduct $CPPC$ on each subset of C_{Act}, combine each result with I_{Act} and deposit them into the set RS_{temp}.

(4) Complete traversal all combination triples (S_i, t_j, S_k) in the subset of RS_{temp}:

(4.1) If S_{com} is null, then deposit (S_i, t_j, S_k) into S_{com}.

(4.2) If S_{com} is not null and S_i/S_k of (S_i, t_j, S_k) exists in the combination pairs of S_{com}, deposit (S_i, t_j, S_k) into S_{com}.

(5) Traverse subsets of S_{com} and judge:

(5.1) Deposit S_{com} into the set RS_{init}, if the number of the same shared activity is less than 2 and all combination pairs can form a closed loop.

(5.2) If the number of the same shared activity is more than 2 or all combination pairs cannot form a closed loop, the traversal of subsets of RS_{temp} will be finished and then jump to (6).

(6) Deposit RS_{init} into RS and output all eligible PEPA models according to RS.

Based on above algorithm descriptions, implementation of algorithm flows is summarized intuitively in the from of a flow chart represented in Fig. 7. Furthermore, in Fig.7, the conditions 1, 2 and 3 correspond to the specific operations in the step (4.2), (5.1) and (5.2) of algorithm flow respectively.

C. KEY STEP DEFINITION

Based on the description of previous algorithm flow, this section summarizes the core steps of the algorithm flow and gives the detailed formalizing definitions.

Definition 6 (Total Permutation and Double Cartesian Product on Concurrent Transition): If A and B represent the pre and post sets of transition t, respectively. $\#A$ and $\#B$ represent the number of different places of sets A and B. $CPPC(A, B)$ is defined as a set of operations shown as:

$$(1).\ f_1(a_i, t, B) = (a_i, t) \times B = \{< a_i, t > \wedge b\}$$
$$(b \in B, a_i \in A, i \in [1, \#A])$$

$$(2).\ f_2(i) = f_2(f_1(a_i, t, B)) = \vee f_1(a_i, t, B) = \{f_1(a_1, t, B), f_1(a_2, t, B), \cdots, f_1(a_{\#A}, t, B)\}$$

$$(3).\ CPPC(A, t, B) = Xf_2(i) = \{< C^1_{f(2)}(1), C^1_{f(2)}(2), \cdots, C^1_{f(2)}(\#A) > | \ f_2(1) \wedge f_2(2) \wedge \cdots \wedge f_2(\#A)\ \&\ f_2(1)(post) \neq f_2(2)(post) \neq \cdots \neq f_2(\#A)(post)\} \quad (7)$$

Definition 7 (Judgement on End-to-End): For some given triples: (t_1, p_a, p_b), (t_j, p_b, p_c), (t_m, p_d, p_a), (t_n, p_e, p_f), …. If $pre(t_i) = post(t_m)$(e.g., place p_a) and $pre(t_i) = post(t_j)$(e.g., place p_b). Then, the set L, including triples (t_1, p_a, p_b), (t_j, p_b, p_c) and (t_m, p_d, p_a), meets the request of ETE. ETE_L is adopted to represent it:

$$ETE_L = \{(t_1, p_a, p_b), (t_j, p_b, p_c), (t_m, p_d, p_a)\} \quad (8)$$

Definition 8 (Combination of End-to-End): For a set L and

$$ETE_L = \{(t_1, p_a, p_b), (t_j, p_b, p_c), (t_m, p_d, p_a)\}.$$

If L meets the condition of ETE, then we can combine all triples of L and use $Comb_{ETE}(L)$ to represent this process:

$$Comb_{ETE}(L) = (p_d, p_a, p_b, p_c) \quad (9)$$

Definition 9 (Place Activity Formation): For a given Petri-net model and $\forall p_i \in P$. If p_i can reach p_j by executing different Place-Activity-Formation, such as:

$$p_i \rightarrow post(p_i) \rightarrow pre(post(p_i)) \rightarrow \cdots \rightarrow pre(p_j) \rightarrow p_j$$

Then, $PAF(i)$ is used to represent this PAF, and $PAF(p_i, p_j)$ describes this process.

Definition 10 (Closed Loop): For a given Petri net component PN and $\forall p_i \in P$, if we can always find a $PAF(i)$ to make $PAF(p_i, p_i)$ reliable. Then, PN can form a closed loop.

D. PEPA MODEL CONSTRUCTION ALGORITHM

Based on the specific description of the algorithm flow in Section IV-B, a PEPA model construction algorithm is proposed. The algorithm takes the corresponding incidence matrix as the processing object and takes the PEPA semantics as judgement and constraint conditions. Furthermore, in order to facilitate the description of the algorithm, the algorithm is devided into two sub-algorithms in terms of the entire construction process of PEPA models. The specific two sub-algorithms are introduced in Sections IV-D1 and IV-D2.

1) ALGORITHM 1: INITIAL OPERATIONS ON INCIDENCE MATRIX

From the scenario and definition of the algorithm, it can be learnt that the main function of Algorithm 1 is responsible for the intermediate operations of the entire process that constructs PEPA models from its numerical representation. In addition, Model 1 is used as an example to illustrate the operations of Steps 19 and 25 in Algorithm 1.

The Step 19 is used to conduct *CPPC* on pre and post sets of the same shared labelled activity. Here, $\eta^{(P_1 \rightarrow P_2, Q_1 \rightarrow Q_2)}$ is used as an example (η is a shared labelled activity). Then, we can obtain the sets: $\text{pre}(\eta) = \{P_1, Q_1\}$ and $\text{post}(\eta) = \{P_2, Q_2\}$.

After the operation of Step 19, the results are obtained: $\{(P_1, \eta, P_2)(Q_1, \eta, Q_2)\}$ and $\{(P_1, \eta, Q_2)(Q_1, \eta, P_2)\}$, then the results must be deposited into the set TC_{Act}.

The Step 25 is used to conduct *CPPC* on all objects of $C_{Act}(i)$ and to store results into the set CP_{Act}. Here, $\eta^{(P_1 \rightarrow P_2, Q_1 \rightarrow Q_2)}$ and $\xi^{(P_1 \rightarrow P_3, Q_1 \rightarrow Q_3)}$ are used as examples. From Step 19, we can obtain:

$$C_{Act} = \left\{ \begin{array}{l} \left\{ \begin{array}{l} \{(P_1, \eta, P_2)(Q_1, \eta, Q_2)\}, \\ \{(P_1, \eta, Q_2)(Q_1, \eta, P_2)\} \end{array} \right\}, \\ \left\{ \begin{array}{l} \{(P_1, \xi, P_3)(Q_1, \xi, Q_3)\}, \\ \{(P_1, \xi, Q_3)(Q_1, \xi, P_3)\} \end{array} \right\} \end{array} \right\}.$$

Then, according to the operation of Step 25, we can obtain:

$$CP_{Act} = \left\{ \begin{array}{l} \left\{ \begin{array}{l} (P_1, \eta, P_2)\,(Q_1, \eta, Q_2) \\ (P_1, \xi, P_3)\,(Q_1, \xi, Q_3) \end{array} \right\}, \\ \left\{ \begin{array}{l} (P_1, \eta, P_2)\,(Q_1, \eta, Q_2) \\ (P_1, \xi, Q_3)\,(Q_1, \xi, P_3) \end{array} \right\}, \\ \left\{ \begin{array}{l} (P_1, \eta, Q_2)\,(Q_1, \eta, P_2) \\ (P_1, \xi, P_3)\,(Q_1, \xi, Q_3) \end{array} \right\}, \\ \left\{ \begin{array}{l} (P_1, \eta, Q_2)\,(Q_1, \eta, P_2) \\ (P_1, \xi, Q_3)\,(Q_1, \xi, P_3) \end{array} \right\} \end{array} \right\}.$$

Finally, the detailed algorithm definition is given in Algorithm 1.

Algorithm 1 Initial Operations on the Given Incidence Matrix

Input: C, $\#P$, $\#T$
Output: RS_{temp}
1: **for** $j \in [1, \#T]$ **do**
2: $negCount = 0$
3: **for** $i \in [1, \#P]$ **do**
4: **if** $C_{ij} = -1$ **then**
5: $\text{pre}(t_j)=\text{post}(t_j) \bigcup \{P_i\}$
6: $negCount = negCount + 1$
7: **end if**
8: **if** $C_{ij} = 1$ **then**
9: $\text{pre}(t_j)=\text{post}(t_j) \bigcup \{P_i\}$
10: **end if**
11: **end for**
12: **if** $negCount = 1$ **then**
13: **for** each pair $(P, Q) \in pre(t_j) \times post(t_j)$ **do**
14: $I_{Act} = I_{Act} \bigcup \{(P, t, Q)\}$
15: **end for**
16: **end if**
17: **if** $negCount \geq 2$ **then**
18: **for** $m \in [1, \#pre(t_j)]$; $n \in [1, \#post(t_j)]$ **do**
19: Do $CPPC(pre(t_j)_m, post(t_j)_n)$, deposit the results into TC_{Act}
20: $C_{Act} = C_{Act} \bigcup TC_{Act}$, then initialize TC_{Act}
21: **end for**
22: **end if**
23: **end for**
24: **for** $i \in [1, \#C_{Act}]$ **do**
25: Do $CPPC$ on $C_{Act}(i)$, then deposit the results into CP_{Act}
26: $CP_{Act}(i) = CP_{Act}(i) \bigcup I_{Act}$
27: $RS_{temp} = RS_{temp} \bigcup CP_{Act}$, then initialize CP_{Act}
28: **end for**

2) ALGORITHM 2: GENERATING ALL PEPA MODELS

From the preceding description, the main function of algorithm 2 is to do further specific judgement on results of Algorithm 1, generate all components and cooperation of each qualified model and finally recover all qualified potential PEPA models.

In addition, in Algorithm 2, Steps 3-14 are used to classify all objects of each subset of RS_{temp} and initially generate all potential components. Steps 15-22 are mainly responsible for screening all components obtained through Steps 3-14. Then, the algorithm generates all components of the potential and eligible PEPA models, which is defined in the subsequent Algorithm 2.

V. CONSTRUCTING PEPA MODELS FOR THE PUBLIC TRANSPORTATION ROAD NETWORK SYSTEM USING ALGORITHM 1 AND 2

This section mainly achieves two goals based on the proposed PEPA models construction algorithm:

Algorithm 2 Generating All Qualified PEPA Models

Input: RS_{temp}, S_{Act}
Output: RS

for $i \in [1, \#RS_{temp}]$ **do**
  while $RS_{temp}(i) \neq \emptyset$ **do**
    for $j \in [1, \#RS_{temp}(i)]$ **do**
      if $S_{com} = \emptyset$ **then**
        $S_{com} = S_{com} \bigcup \{RS_{temp}(ij)\}$, then delete $RS_{temp}(ij)$ from $RS_{temp}(i)$
      end if
      if $S_{com} \neq \emptyset$ **then**
        for $m \in [1, \#S_{com}]$ **do**
          if $RS_{temp}(ij)[1] = S_{com}(m)[1]$ or $RS_{temp}(ij)[1] = S_{com}(m)[3]$ **then**
            $S_{com} = S_{com} \bigcup \{RS_{temp}(ij)\}$, then delete $RS_{temp}(ij)$ from $RS_{temp}(i)$
          end if
        end for
      end if
    end for
    for $n \in [1, \#S_{com}]$ **do**
      if $\exists k, l \in [1, \#S_{com}]; k, l \neq n : S_{com}(n)[1] = S_{com}(k)[3]$ and $S_{com}(n)[3] = S_{com}(l)[1]$ **then**
        $RS_{init} = RS_{init} \bigcup \{S_{com}(n)\}$
      end if
      if $\forall k, l \in [1, \#S_{com}]; k, l \neq n : S_{com}(n)[1] \neq S_{com}(k)[3]$ or $S_{com}(n)[3] \neq S_{com}(n)[3] \neq S_{com}(l)[1]$ or $\#T_{numShared} \geq 2$ **then**
        Initialize S_{com}
      end if
    end for
  end while
end for

• Section V-A constructs the corresponding PEPA models for the given urban road network.

• Section V-B generates performance evaluation on the PEPA models of the urban road network.

A. PROCESS OF PEPA CONSTRUCTION

According to the proposed construction algorithms, operations will be conducted on the incidence matrix (i.e. Table 1) and finally output all potential PEPA models.

(1) First, according to Steps 3-11 of Algorithm 1, all pre and post sets of all labelled activities (i.e. intersections) are obtained as follows (here we just list three pre and post sets of labelled activities):

$$\begin{aligned} \mathrm{pre}(t_1) &= \{S_1, S_8\}, & \mathrm{post}(t_1) &= \{S_2, S_9\}; \\ \mathrm{pre}(t_2) &= \{S_2\}, & \mathrm{post}(t_2) &= \{S_3\}; \\ \mathrm{pre}(t_3) &= \{S_3\}, & \mathrm{post}(t_3) &= \{S_4\}; \end{aligned}$$

Then, according to Definition 3 and 4, it is easy for readers to obtain the other pre and post sets of other labelled activities (e.g., $t_4, t_5, \cdots, t_{19}$).

(2) Second, according to (1) and Steps 12-16 of Algorithm 1, the set I_{Act} can be obtained:

$$I_{Act} = \left\{ \begin{array}{c} (S_2, t_2, S_3)(S_3, t_3, S_4)(S_5, t_5, S_6) \\ (S_5, t_6, S_7)(S_6, t_7, S_7)(S_7, t_8, S_1) \\ (S_{10}, t_{10}, S_{11})(S_{12}, t_{12}, S_{13})(S_{12}, t_{13}, S_{14}) \\ (S_{13}, t_{14}, S_{14})(S_{16}, t_{17}, S_8)(S_{15}, t_{16}, S_{16}) \\ (S_{18}, t_{18}, S_{19})(S_{22}, t_{19}, S_{17}) \end{array} \right\}.$$

In addition, according to (1) and Steps 17-22 of Algorithm 1, conduct Cartesian product, permutations and combinations on the pre and post sets of shared labelled activities. Then, the set C_{Act} can be obtained (here just give one subset of C_{Act}):

$$C_{Act} = \left\{ \begin{array}{c} \left\{ \begin{array}{c} (S_{14}, t_{15}, S_{15})(S_{20}, t_{15}, S_{21})(S_1, t_1, S_2) \\ (S_8, t_1, S_9)(S_{11}, t_{11}, S_{12})(S_{19}, t_{11}, S_{20}) \\ (S_4, t_4, S_5)(S_{21}, t_4, S_{22})(S_9, t_9, S_{10}) \\ (S_{17}, t_9, S_{18}) \end{array} \right\} \\ \cdots \end{array} \right\}.$$

(3) Third, according to Algorithm 2 and the results of Algorithm 1, it is easy to obtain eight potential and eligible PEPA models. Then, the set RS will be obtained as follows (here just give one PEPA model):

$$RS = \left\{ \begin{array}{c} \left\{ \begin{array}{c} \left\{ \begin{array}{c} (S_1, t_1, S_2)(S_2, t_2, S_3)(S_3, t_3, S_4) \\ (S_4, t_4, S_5)(S_5, t_5, S_6)(S_5, t_6, S_7) \\ (S_6, t_7, S_7)(S_7, t_8, S_1) \end{array} \right\} \\ \left\{ \begin{array}{c} (S_8, t_1, S_9)(S_9, t_9, S_{10}) \\ (S_{10}, t_{10}, S_{11})(S_{11}, t_{11}, S_{20}) \\ (S_{20}, t_{15}, S_{15})(S_{15}, t_{16}, S_{16}) \\ (S_{16}, t_{17}, S_8) \end{array} \right\} \\ \left\{ \begin{array}{c} (S_{12}, t_{12}, S_{13})(S_{12}, t_{13}, S_{14}) \\ (S_{13}, t_{14}, S_{14})(S_{14}, t_{15}, S_{21}) \\ (S_{21}, t_4, S_{22})(S_{22}, t_{19}, S_{17}) \\ (S_{17}, t_9, S_{18})(S_{18}, t_{18}, S_{19}) \\ (S_{19}, t_{11}, S_{12}) \end{array} \right\} \end{array} \right\}; \\ \cdots \end{array} \right\}.$$

(4) Through Sections IV and V-A, it can be easily found that the construction algorithms successfully sort out all eight potential PEPA models of the public road network (shown in Table 3). That is to say, based on the urban road network, the eight public traffic-line deployments can be found. From Table 3, it is easy to find the Model (I) that is the original bus lines deployment, and then the other seven options are the alternative deployments. As shown in Figs. 5-7, we provide three deployments (model (III), model (VI) and model ($VIII$)) of the eight models. With these generated deployments and the actual situation (such as changes in residents distribution density and road traffic flow), then, it will be more convenient to find the most suitable public traffic-line deployment.

TABLE 3. The component combination of all potential PEPA models of Table 1.

Model	Components of Model		
Model (I)	$\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$	$\{S_8, S_9, S_{10}, S_{11}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}\}$	$\{S_{17}, S_{18}, S_{19}, S_{20}, S_{21}, S_{22}\}$
Model (II)	$\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$	$\{S_8, S_9, S_{18}, S_{19}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}\}$	$\{S_{10}, S_{11}, S_{20}, S_{21}, S_{22}, S_{17}\}$
Model (III)	$\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$	$\{S_8, S_9, S_{10}, S_{11}, S_{20}, S_{15}, S_{16}\}$	$\{S_{12}, S_{13}, S_{14}, S_{21}, S_{22}, S_{17}, S_{18}, S_{19}\}$
Model (IV)	$\{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$	$\{S_8, S_9, S_{18}, S_{19}, S_{20}, S_{15}, S_{16}\}$	$\{S_{10}, S_{11}, S_{12}, S_{13}, S_{14}, S_{21}, S_{22}, S_{17}\}$
Model (V)	$\{S_1, S_9, S_{10}, S_{11}, S_{12}, S_{13}, S_{14}, S_{21}, S_5, S_6, S_7\}$	$\{S_2, S_3, S_4, S_{22}, S_{17}, S_{18}, S_{19}, S_{20}, S_{15}, S_{16}, S_8\}$	
Model (VI)	$\{S_1, S_9, S_{18}, S_{19}, S_{12}, S_{13}, S_{14}, S_{21}, S_5, S_6, S_7\}$	$\{S_2, S_3, S_4, S_{22}, S_{17}, S_{10}, S_{11}, S_{20}, S_{15}, S_{16}, S_8\}$	
Model (VII)	$\{S_1, S_9, S_{10}, S_{11}, S_{20}, S_{21}, S_5, S_6, S_7\}$	$\{S_2, S_3, S_4, S_{22}, S_{17}, S_{18}, S_{19}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}, S_8\}$	
Model $(VIII)$	$\{S_1, S_9, S_{18}, S_{19}, S_{20}, S_{21}, S_5, S_6, S_7\}$	$\{S_2, S_3, S_4, S_{22}, S_{17}, S_{10}, S_{11}, S_{12}, S_{13}, S_{14}, S_{15}, S_{16}, S_8\}$	

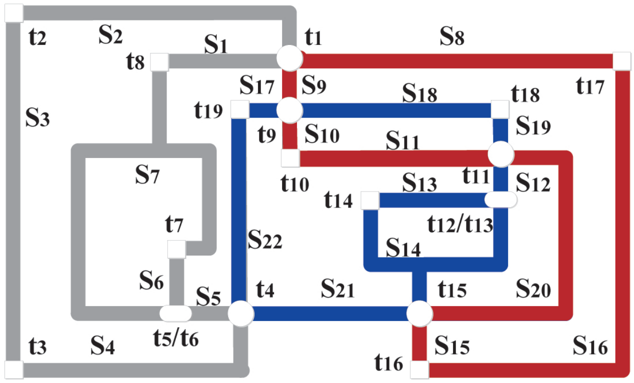


FIGURE 8. The potential bus lines deployment of model (*III*) in Table 3.

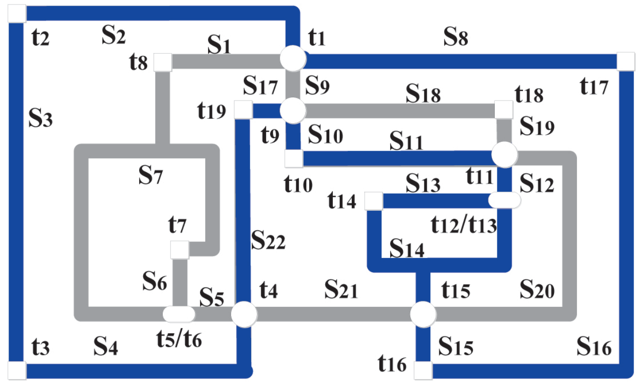


FIGURE 10. The potential bus lines deployment of model (*VIII*) in Table 3.

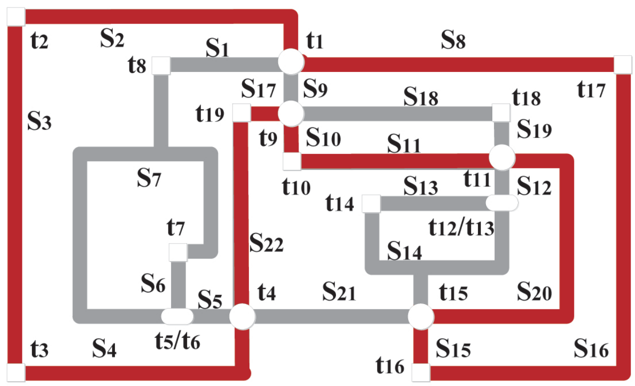


FIGURE 9. The potential bus lines deployment of model (*VI*) in Table 3.

(5) According to (3), all qualified PEPA models can be easily generated. And all components of the potential PEPA models and states of every component are listed in Table 3. Here, we just list Model (I). According to Table 3, it is easy for readers to get the other seven PEPA models.Furthermore, we decompose the following system's PEPA models into three components (i.e. *Component*1, *Component*2 and *Component*3). *Component*1 contains states $S_1 - S_7$, *Component*2 contains states $S_8 - S_{16}$ and *Component*3 contains states $S_{17} - S_{22}$. The specific corresponding PEPA models as follows:

$$S_1 \stackrel{\text{def}}{=} (t_1, r_{t_1}).S_2$$
$$S_2 \stackrel{\text{def}}{=} (t_2, r_{t_2}).S_3$$
$$S_3 \stackrel{\text{def}}{=} (t_3, r_{t_3}).S_4$$
$$S_4 \stackrel{\text{def}}{=} (t_4, r_{t_4}).S_5$$
$$S_1 \stackrel{\text{def}}{=} (t_1, r_{t_1}).S_2$$
$$S_5 \stackrel{\text{def}}{=} (t_5, r_{t_5}).S_6 + (t_6, r_{t_6}).S_7$$
$$S_6 \stackrel{\text{def}}{=} (t_7, r_{t_7}).S_7$$
$$S_7 \stackrel{\text{def}}{=} (t_8, r_{t_8}).S_1$$
$$S_8 \stackrel{\text{def}}{=} (t_1, r_{t_1}).S_9$$
$$S_9 \stackrel{\text{def}}{=} (t_9, r_{t_9}).S_{10}$$
$$S_{10} \stackrel{\text{def}}{=} (t_{10}, r_{t_{10}}).S_{11}$$
$$S_{11} \stackrel{\text{def}}{=} (t_{11}, r_{t_{11}}).S_{12}$$
$$S_{12} \stackrel{\text{def}}{=} (t_{12}, r_{t_{12}}).S_{13} + (t_{13}, r_{t_{13}}).S_{14}$$
$$S_{13} \stackrel{\text{def}}{=} (t_{14}, r_{t_{14}}).S_{14}$$
$$S_{14} \stackrel{\text{def}}{=} (t_{15}, r_{t_{15}}).S_{15}$$
$$S_{15} \stackrel{\text{def}}{=} (t_{15}, r_{t_{15}}).S_{16}$$
$$S_{16} \stackrel{\text{def}}{=} (t_{16}, r_{t_{16}}).S_8$$
$$S_{17} \stackrel{\text{def}}{=} (t_9, r_{t_9}).S_{18}$$
$$S_{18} \stackrel{\text{def}}{=} (t_{18}, r_{t_{18}}).S_{19}$$
$$S_{19} \stackrel{\text{def}}{=} (t_{11}, r_{t_{11}}).S_{20}$$
$$Ss_{20} \stackrel{\text{def}}{=} (t_{15}, r_{t_{15}}).S_{21}$$
$$S_{21} \stackrel{\text{def}}{=} (t_4, r_{t_4}).S_{22}$$

TABLE 4. Number of components.

Component	Number
$Component1$	160
$Component2$	160
$Component3$	250

TABLE 5. Duration of all activities.

Action	Duration(s)	Action	Duration (s)
t_1	0.1667	t_2	0.0699
t_3	0.0746	t_4	0.2000
t_5	0.1052	t_6	0.0870
t_7	0.0806	t_8	0.0613
t_9	0.2000	t_{10}	0.1124
t_{11}	0.2500	t_{12}	0.1299
t_{13}	0.1493	t_{14}	0.0769
t_{15}	0.1667	t_{16}	0.0599
t_{17}	0.0741	t_{18}	0.1429
t_{19}	0.1111		

$$S_{22} \stackrel{\text{def}}{=} (t_{19}, r_{t_{19}}).S_{17}$$

$$Component1[a] \underset{\{L_1\}}{\bowtie} (Component2[b] \underset{\{L_2\}}{\bowtie} Component3[c])$$

where

$$L_1 = \{t_1, t_4\}$$
$$L_2 = \{t_9, t_{11}, t_{15}\}$$

B. SYSTEM ANALYSIS BASED ON THE GENERATED PEPA MODELS

System parameter setting, including components and durations of all activities, is specified in Section V-B1, which is based on the PEPA models obtained in Section V-A. Then, Section V-B2 demonstrates detailed performance evaluation of this urban road network.

1) PARAMETERS SETTING

In this section, the specific parameters corresponding to the PEPA models are specified in Table 4, in which the specific initial numbers of all components (i.e. $Component1$, $Component2$ and $Component3$) are given through reasonable setting of the traffic volume in the actual urban road network. Meanwhile, in Table 5, rational rates are set for vehicles passing through different intersections by accessing information on the Internet and analysing the actual situation. In Table 5, the durations represent the time that one vehicle spends on crossing the corresponding intersection. Furthermore, as the durations of activities shown in Table 5, due to equipment limitation, we have correspondingly shortened the time of vehicles passing through each kinds of intersections.

2) PERFORMANCE EVALUATGION

In this section, we will analyse the response time of the urban road network based on the parameter set in Section V-B1. Based on the actual situation, it is clear that the number of cars in the road network and the number of cars passing in the intersection in a time unit are two important factors affecting the performance of road network. Thereafter, the analysis will discuss the impact of these factors based on the performance of road network.

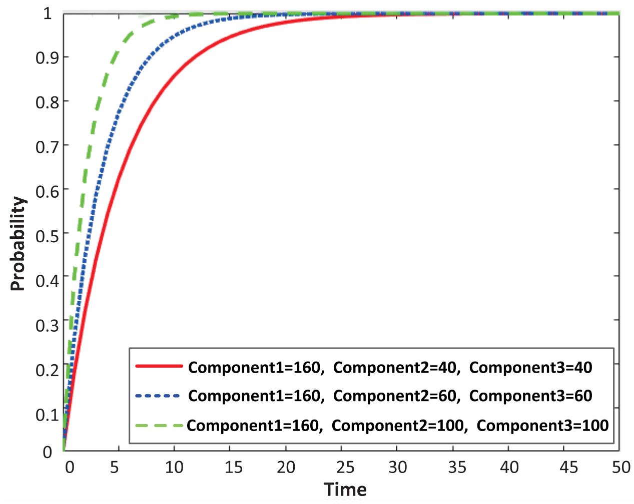


FIGURE 11. Response-time of the corresponding PEPA models.

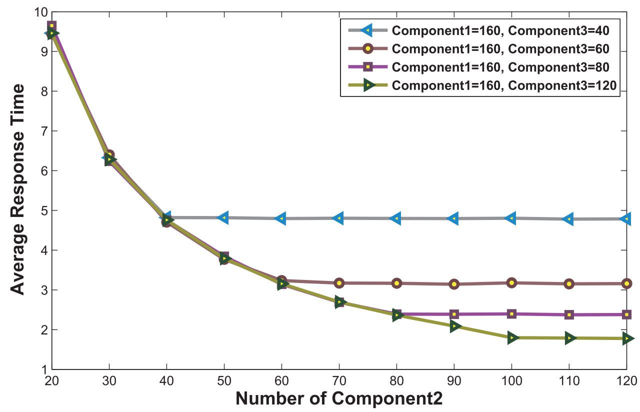


FIGURE 12. Impact of the number of *Component2* on performance.

As shown in Fig.11, by changing the number of cars of $Component2$ and $Component3$, it can be easily found that the road network can ensure more cars pass the intersections faster when the number of cars decreases. So the probability of each car passing the intersection is higher, which means a shorter response time of the cars driving in the urban road network. In other words, it represents a smooth road traffic condition.

In Fig.12, we set the maximum number of cars that are $Component1$ and $Component3$, and discuss the performance of road network by changing the number of cars in $Component2$. From Fig.12, when a greater number of cars are allowed to pass through $Component1$ and $Component2$ of the road network, which is to increase the car capacity of the road network, cars will spend less time on the road. In other words, this generates a smoother road network.

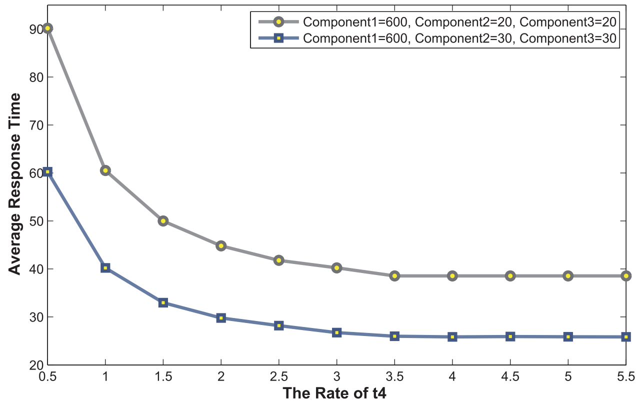


FIGURE 13. Impact of t_4 rate on performance.

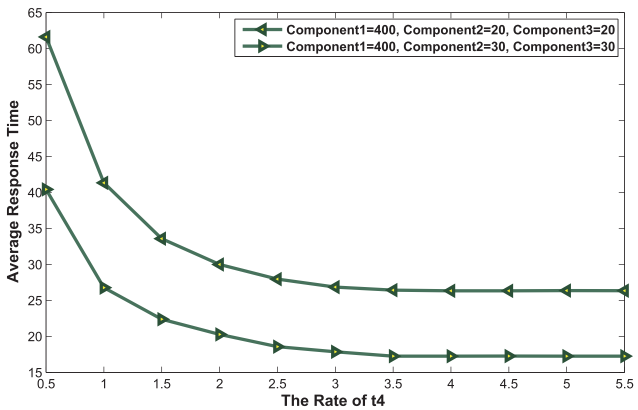


FIGURE 14. Impact of t_4 rate on performance.

In Fig.13 and Fig.14, we first set the number of cars in *Component*1, *Component*2 and *Component*3 as constant. Then, the analysis discusses the performance of the road network by changing the rate of t_4 (i.e. gradually increasing the number of cars passing the intersection t_4 per unit time). From Fig.13 and Fig.14 and with the same parameter, it shows that cars spend less time passing through the road network when the intersections can accept more cars passing in a unit time (i.e. the lower risk of traffic jams on the road network).

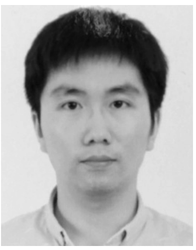
VI. CONCLUSION

This paper aims to explore compositional modeling skills on a large-scale and complex current system by sorting and analyzing the corresponding incidence matrix of a concurrent system (i.e. the model of public transportation road network). According to our research, we found that it is feasible to rebuild the structure of formal models based on their related incidence matrices. Therefore, a sorting algorithm is developed to assist the formal model construction based on PEPA by sorting the related incidence matrix to find all potential compositional structures and components. Then, an example of the model construction is demonstrated by using a traffic network model scenario. Through an example of the traffic-based concurrent system, such as the urban road network and traffic system, the proposed algorithm is useful to facilitate the research of structural analysis of such concurrent systems. Furthermore, the findings reveal the significance of improving the performance of concurrent systems and establishing a connection between numerical representation (i.e. incidence matrix) and formal models (i.e. PEPA model).

REFERENCES

[1] J. R. Norris, *Markov Chains* (Cambridge Series in Statistical and Probabilistic Mathematics). Cambridge, U.K.: Cambridge Univ. Press, 1998.
[2] J. Ding, "Structural and fluid analysis for large scale PEPA models—With applications to content adaptation systems," Ph.D. dissertation, School Inform., Univ. Edinburgh, Scotland, U.K., 2010.
[3] J. Hillston, "Fluid flow approximation of PEPA models," in *Proc. QEST*, vol. 5, 2005, pp. 33–42, doi: 10.1109/QEST.2005.12.
[4] J. Hillston, *A Compositional Approach to Performance Modelling*. New York, NY, USA: Cambridge Univ. Press, 2016.
[5] J. Ding, "A comparison of fluid approximation and stochastic simulation for evaluating content adaptation systems," *Wireless Pers. Commun.*, vol. 84, no. 1, pp. 231–250, 2015.
[6] H. Hermanns, *Interactive Markov Chains* (Lecture Notes in Computer Science), vol. 2428. Berlin, Germany: Springer, 2002.
[7] P. R. D'Argenio, H. Hermanns, J. P. Katoen, and R. Klaren, "MODEST: A modelling language for stochastic timed systems," in *Proc. Joint Int. Workshop Process Algebra Probab. Methods, Perform. Modeling Verification (PAPM-PROBMIV)*, in Lecture Notes in Computer Science, vol. 2165, 2001, pp. 87–104.
[8] J. Ding and J. Hillston, "Numerically representing stochastic process algebra models," *Comput. J.*, vol. 55, no. 11, pp. 1383–1397, 2012.
[9] J. Ding and J. Hillston, "Structural analysis for stochastic process algebra models," in *Proc. AMAST*, Lac-Beauport, QC, Canada, Jun. 2010, pp. 1–27.
[10] M.-K. Jiau and S.-C. Huang, "Services-oriented computing using the compact genetic algorithm for solving the carpool services problem," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2711–2722, Oct. 2015.
[11] B. Singh, A. Gupta, B. Singh, and A. Gupta, "Recent trends in intelligent transportation systems: A review," *J. Transp. Literature*, vol. 9, no. 2, pp. 30–34, 2015.
[12] M.-K. Jiau, S.-C. Huang, and C.-H. Lin, "Optimizing the carpool service problem with genetic algorithm in service-based computing," in *Proc. SCC*, 2013, pp. 478–485.
[13] R. K. Megalingam, R. N. Nair, and V. Radhakrishnan, "Automated wireless carpooling system for an eco-friendly travel," in *Proc. ICECT*, vol. 4, no. 2, 2011, pp. 325–329.
[14] G. Arnould, D. Khadraoui, M. Armendáriz, J. C. Burguillo, and A. Peleteiro, "A transport based clearing system for dynamic carpooling business services," in *Proc. ITST*, 2011, pp. 527–533.
[15] S.-C. Huang, M.-K. Jiau, and C.-H. Lin, "A genetic-algorithm-based approach to solve carpool service problems in cloud computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 352–364, Feb. 2015.
[16] C.-H. Lin, M.-K. Jiau, and S.-C. Huang, "A cloud computing framework for real-time carpooling services," in *Proc. ISSDM*, 2012, pp. 266–271.
[17] P. Lalos, A. Korres, C. K. Datsikas, K. Peppas, and G. S. Tombras, "A framework for dynamic car and taxi pools with the use of positioning systems," in *Proc. COMPUTATIONWORLD*, 2009, pp. 385–391.
[18] M. Silva, "Half a century after Carl Adam Petri's Ph.D. thesis: A perspective on the field," *Annu. Rev. Control*, vol. 37, no. 2, pp. 191–219, 2013.
[19] S. Gilmore, J. Hillston, and M. Ribaudo, "An efficient algorithm for aggregating PEPA models," *IEEE Trans. Softw. Eng.*, vol. 27, no. 5, pp. 449–464, May 2001.
[20] X. Chen and L. Wang, "Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method—PEPA," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 745–748, Apr. 2017.
[21] M. Tribastone, J. Ding, S. Gilmore, and J. Hillston, "Fluid rewards for a stochastic process algebra," *IEEE Trans. Softw. Eng.*, vol. 38, no. 4, pp. 861–874, Jul./Aug. 2012.
[22] *Online Map Website*. Accessed: Apr. 14, 2018. [Online]. Available: http://www.onegreen.net/maps/
[23] J. Ding, X. Chen, and R. Wang, "A compositional analysis method for Petri-net models," *IEEE Access*, vol. 5, pp. 27599–27610, 2017.

JIE DING received the B.S. degree in mathematical education from Yangzhou University, Yangzhou, China, in 2001, the M.S. degree in mathematical statistics from Southeast University, Nanjing, China, in 2004, and the Ph.D. degree in communication from The University of Edinburgh, Edinburgh, U.K., in 2010. He is currently a Professor with Shanghai Maritime University. His research interests include performance modeling for communication and computer systems.



RUI WANG received the B.Sc. degree in software engineering from Yangzhou University, Yangzhou, China, in 2011, where he is currently pursuing the M.Sc. degree with the School of Information Engineering. His research interests include performance modeling and evaluation, IoT, and ITS.

XIAO CHEN received the M.Sc. and Ph.D. degrees in computing science from Newcastle University in 2009 and 2013, respectively. He is currently an Associate Professor with Jiangsu University, China. His research interests include formal modeling and performance analysis for large-scale systems, e.g., smart systems, cloud/fog systems, and blockchain-based applications and systems.

YING-EN GE received the B.Sc. degree from the Dalian University of Technology (DUT), the M.Sc. degree from Southeast University, Nanjing, China, and the Ph.D. degree from Tongji University, Shanghai, in 1999. Before joining DUT in 2010, he was first a Research Assistant with The Hong Kong Polytechnic University, then a Post-Doctoral Researcher with the University of California at Davis from 2000 to 2001, and subsequently a Research Fellow with the University of Ulster from 2001 to 2003, Queens University of Belfast from 2003 to 2006, and Edinburgh Napier University from 2007 to 2008; he was involved in transport consulting arena from 2008 to 2010. He is currently a Professor with the College of Transport & Communications, Shanghai Maritime University. His primary academic interests include transportation network analysis, demand management, transportation policy and the environment and port and shipping operations. He is currently a member of the Editorial Boards of international scholarly journals: *Transport*, *Transport Policy*, *Transportation Research Part D: Transport and Environment*, and *Transportmetrica B: Transport Dynamics*.