

Article

Hierarchical Colored Petri Nets for Modeling and Analysis of Transit Signal Priority Control Systems

Yisheng An ^{1,*} , Naiqi Wu ² , Xiangmo Zhao ¹, Xuan Li ¹ and Pei Chen ¹

¹ School of Information Engineering, Chang'an University, Xi'an 710064, China; xmzhao@chd.edu.cn (X.Z.); lixuan091708@gmail.com (X.L.); cpei0808@gmail.com (P.C.)

² Institute of Systems Engineering, Macau University of Science and Technology, Taipa 999078, Macau; nqwu@must.edu.mo

* Correspondence: aysm@chd.edu.cn; Tel.: +86-29-8233-4359

Received: 13 December 2017; Accepted: 17 January 2018; Published: 19 January 2018

Abstract: In this paper, we consider the problem of developing a model for traffic signal control with transit priority using Hierarchical Colored Petri nets (HCPN). Petri nets (PN) are useful for state analysis of discrete event systems due to their powerful modeling capability and mathematical formalism. This paper focuses on their use to formalize the transit signal priority (TSP) control model. In a four-phase traffic signal control model, the transit detection and two kinds of transit priority strategies are integrated to obtain the HCPN-based TSP control models. One of the advantages to use these models is the clear presentation of traffic light behaviors in terms of conditions and events that cause the detection of a priority request by a transit vehicle. Another advantage of the resulting models is that the correctness and reliability of the proposed strategies are easily analyzed. After their full reachable states are generated, the boundness, liveness, and fairness of the proposed models are verified. Experimental results show that the proposed control model provides transit vehicles with better effectiveness at intersections. This work helps advance the state of the art in the design of signal control models related to the intersection of roadways.

Keywords: Petri nets; modeling; traffic signal control; transit priority; state spaces

1. Introduction

Public transit systems have been extensively studied in the literature [1–3] to improve the safety and efficiency of transportation in urban traffic networks. The quality of the transit service depends heavily on how punctually the transit arrives at each bus stop and picks up passengers. Especially, transit signal priority control and signal switching behaviors are becoming necessary functions for any transit system to improve the safety and efficiency of public transportation.

Transit signal priority (TSP) control operates under the logic explained as follows. It extends the originally planned green light phase or starts the phase early to make the green light time longer if some conditions are met such that some transit vehicles can go through an intersection without waiting there. To decide which logic should be used, a quick calculation is performed on site using a bus arrival time prediction model based on historical data [4–6]. There are many classical TSP methods being employed, including offset modification [7], green wave [8], cycle extension [9], and phase recall [10]. Wahlstedt [11] presented a simulation-based model for analyzing partial dynamic signal timings as well as fully adaptive signal control systems. Wolput et al. [12] developed a logic-based adaptive TSP control system with double detection and they make comparison with full priority for mixed lane conditions. Ahmed et al. [13] presented a traffic control system that can work standalone to handle various boundary conditions of the recurrent, non-recurrent congestion, TSP and downstream blockage conditions to improve the overall traffic network vehicular productivity and efficiency. In addition,

there are some studies, such as [11,14], based on adaptive traffic signal control such as SCOOT and PRIBUSS to explore new priority strategies. Although these approaches promoted the transit signal priority control research to some extent, it is worth noting that all these approaches are mainly based on analytical methods and many strict mathematical assumptions. Furthermore, the correctness and reliability of the TSP control systems are unproven, and potential design errors can lead to conflicting traffic at the intersection, which in turn affects the actual control effect.

Petri nets (PN) have shown to be powerful in modeling, analyzing, scheduling, and controlling for various kinds of information and production systems that are characterized as being concurrent, asynchronous, distributed and stochastic [15–24]. Their graphical representation provides a friendly man-machine interface for simulation and logic control. To the best of the authors' knowledge, the work in [25] may be the earliest attempt to apply PNs for the modeling and analysis of urban traffic networks. Since then, a lot of researchers have focused their attention on this specific area. Ng et al. [26] presented a comprehensive review on the applications of PNs in modeling, analysis, and control of traffic networks. A variety of high-level PNs such as continuous PNs with variable speed, first-order Hybrid PNs and Synchronized Timed PNs have been used for modeling and analysis of traffic flow [27,28], and urban traffic networks [29,30]. Unlike the continuous characteristics of vehicle flow, the behavior of a signal control system at the intersection shows a discrete feature. List et al. [31] used P-invariant and a reachability tree to confirm that a PN-based eight-phase traffic signal control model is conflict- and deadlock-free, respectively. Huang et al. [29] developed modularized eight-phase, six-phase, and two-phase traffic-light control systems, respectively. The advantage of these models is that the size of the urban traffic network control systems can be easily extended with their proposed modular technique. It is worth noting that the common feature of the abovementioned research is that PNs are used for regular adjustment of phase and timing of traffic signal systems. Meanwhile, we also notice that some scholars aim at reducing delay for transit or emergency vehicles at signalized intersections by temporarily adjusting the traffic signal timing to benefit transit vehicles [32,33].

This paper proposes a new methodology to design a control model for the TSP systems based on Hierarchical-Colored PNs (HCPN) and adopts the state space analysis to verify the correctness and reliability of the control model before the deployment stage. The resulting system is based on the network of the on-board unit and roadside unit (RSU) for TSP applications that provide transit vehicles with safety and convenience. Once an approaching transit vehicle is detected by an RSU, signal control systems determine whether the current state meets the priority control conditions. If it is satisfied, the transit vehicle and its predecessor or successor vehicles with the same direction are allowed to pass through the intersection. If it is not satisfied, we might prefer to maintain the original control strategy without conducting priority control ones. As a result, one can infer that the proposed priority strategies are able to reduce the travel time of public transit vehicles moderately and economically.

The remainder of this paper is organized as follows. Section 2 demonstrates the model of the TSP intersection. Section 3 develops the HCPN-based TSP control models. Then, the analysis of the results is presented by using full reachable states in Section 4. Concluding remarks are given in Section 5.

2. Transit Priority Signalized Intersection

An example of intersection applying the transit signal priority control system is depicted in Figure 1. The TSP intersection can be divided into three parts: signalized intersection, transit detection, and signal priority controller. The traffic flows are ruled by four-phase signal control logic presented in Figure 2.

2.1. Signalized Intersection

A signalized intersection connects two or more roads. It provides a conflicting area for vehicles to occupy at different times. We consider a four-way intersection (as shown in Figure 1), which is operated by a four-phase traffic signal control logic (as shown in Figure 2). This intersection connects eight roads expressed as R_{Df} , $D \in \{E, W, N, S\}$ and $f \in \{i, o\}$, where D models the directions of

traffic flows, including eastbound, westbound, northbound, and southbound, f models a vehicle entering or leaving the specific road, respectively. Presently, a four-phase signal control system with a deterministic phase transition structure at intersections is widely used, although there are studies on applying non-deterministic transition structure to improve the performance [10]. The aim of this work is to test the TSP control strategies and a deterministic phase transition structure is applied.

Before further describing the control logic, the movement of traffic flow and the phase of signal light should be defined. A movement specifies the vehicle's trajectory at the intersection. In Figure 1, by taking road section R_{Wi} as an example, there are three movements including $R_{Wi} \rightarrow R_{No}$, $R_{Wi} \rightarrow R_{Wo}$ and $R_{Wi} \rightarrow R_{So}$. Therefore, there are totally 12 movements at the intersection and, among them, the four turn-right movements can be combined into the four go-straight movements. Movements are divided into many groups, and these groups can be named as a phase. When a phase is selected, related movements receive a permission signal (e.g., green light), while all other movements receive a prohibit signal (e.g., red light).

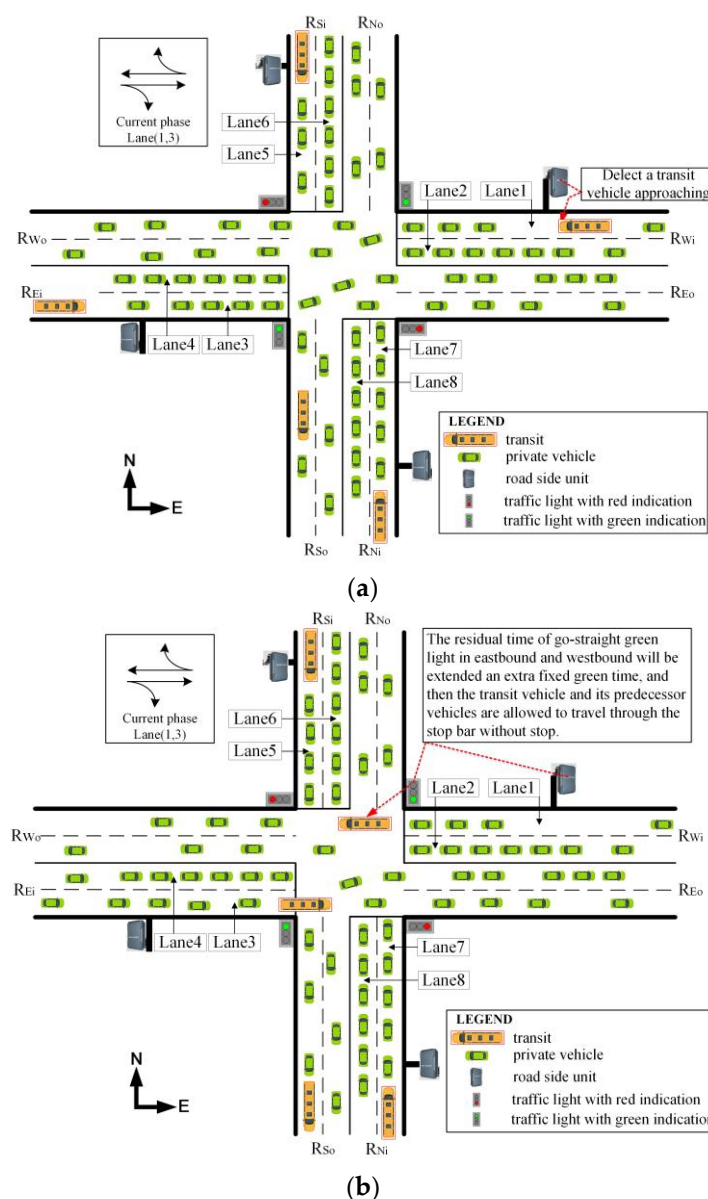


Figure 1. An example of intersection with the transit signal priority (TSP) control systems. (a) Before the priority strategy is applied; (b) After the priority strategy is applied.

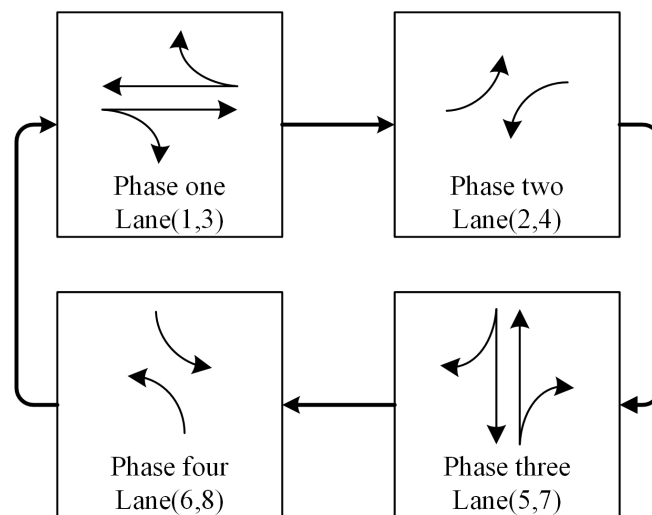


Figure 2. Four-phase transitions of the traffic signal control.

For a typical intersection, four traffic light sets are deployed at the intersection. Each set has four types of lights involving a red one (R), a yellow one (Y), a green one with turn-left arrow (TL), and a green one with go-straight and turn-right arrow combined (GS). In phase one, GS signals turn on in the eastbound and westbound traffic directions; in other words, the movements such as $R_{Ei} \rightarrow R_{Eo}$, $R_{Ei} \rightarrow R_{So}$, $R_{Wi} \rightarrow R_{Wo}$, $R_{Wi} \rightarrow R_{No}$ get the permission of passing, and R signals are displayed for all other movements. In phase two, TL signals turn on in the eastbound and westbound traffic directions, which means the movements $R_{Wi} \rightarrow R_{So}$ and $R_{Ei} \rightarrow R_{No}$ are allowed, and the R signals turn on for all other movements. In phase three, GS signals turn on in the northbound and southbound traffic directions, which indicates that the movements such as $R_{Ni} \rightarrow R_{Eo}$, $R_{Ni} \rightarrow R_{No}$, $R_{Si} \rightarrow R_{Wo}$, and $R_{Si} \rightarrow R_{So}$ get the permission of passing, and R signals are displayed for all other movements. Lastly, in phase four, TL signals turn on in the northbound and southbound traffic directions, and the movements $R_{Si} \rightarrow R_{Eo}$ and $R_{Ni} \rightarrow R_{Wo}$ are allowed, and R signals are displayed for all other movements.

2.2. Transit Detection

Consider the intersection depicted in Figure 1. We deploy an RSU including sensors, controllers and processors at the roadside of each road to help us to collect the traffic data. Generally, sensors include a weather sensor, traffic flow sensor, probe vehicle detection module, and dedicated short-range communication module. Hence, we can use an RSU to detect the approaching of a transit vehicle, and send a message to the traffic signal controller at the intersection through the communication module. Once the controller receives a message, the travel time from current location to stop bar is estimated. This value and the residual time of the current phase are compared and employed to determine whether the transit priority strategy is triggered or not.

2.3. Transit Signal Priority Control

When a transit vehicle is approaching the crossing section, some specific scenarios could emerge immediately. For example, at phase one, eastbound and westbound GS green lights turn on and the remaining green time is six seconds. At this moment, an approaching transit vehicle is detected by the RSU installed on Lane 1 in Figure 1. Suppose that it takes 10 s for the transit vehicle to travel from the current location to the stop bar. This means that when the transit vehicle arrives at the stop bar, phase one will change to phase two and it has to wait here for almost a complete red light period. An obvious solution is to increase an extra green time for phase one temporarily and give the transit vehicle and other queued vehicles in front of it enough time to pass through the stop bar. This can

effectively avoid the unnecessary waiting of a large number of passengers on the transit vehicle, and thus reduces the transit intersection delay time and travel time, as well as improves transit service reliability and quality of service.

As noted in [26], a traffic signal control system is a typical distributed, concurrent, and non-deterministic system. For such a complex system, if there is no effective method or tool for verification, one may design a system such that some important interaction patterns are missed, leading to gaps or malfunctions in the system. Therefore, how to guarantee the correctness and reliability of the TSP control strategies is an important issue. Beyond the reachability graph used in [33], this paper presents an executable model and its full state spaces to handle the dynamic alternation of traffic signal lights when one or more transit vehicles are detected by RSUs concurrently or asynchronously.

3. Petri Net Representation

In this section, the fundamental concepts of HCPN are reviewed briefly. Then, we present several assumptions for modeling and analysis. After that, HCPN-based executable models involving the adjustment of light time and conflict resolution are illustrated for the design of TSP control systems.

3.1. HCPN Concepts

It is well known that places, transitions, and directed arcs are three main constituent elements in PNs, a particular kind of bipartite-directed graphs. From the viewpoint of system engineering modeling, a collection of places is used to express the state of systems, and transitions are used to represent events that cause the system state change. The capabilities of a high-level programming language and module hierarchy are associated with PNs, leading to the concepts of Hierarchical-Colored PNs [34,35].

Definition 1. A Colored Petri net is a 9-tuple $\Sigma = (P, T, A, C, V, W, G, E, I)$, where:

1. P is a finite set of places;
2. T is a finite set of transitions and $P \cap T = \emptyset$;
3. A is a finite set of directed arcs, $A \subseteq P \times T \cup T \times P$;
4. C is a finite set of non-empty color sets;
5. V is a set of typed variables, such that $\forall v \in V, \text{Type}[v] \in C$;
6. $W : P \rightarrow C$ is a color set function that assigns a color set to each place;
7. $G : T \rightarrow \text{EXPR}_V$ is a guard function that assigns a guard to each transition t such that $\text{Type}[G(t)] = \text{Bool}$;
8. $E : A \rightarrow \text{EXPR}_V$ is an arc expression function that assigns an arc expression to each arc a such that $\text{Type}[E(a)] = C(p)_{\text{MS}}$, here, MS means multiset;
9. $I : P \rightarrow \text{EXPR}_0$ is an initialization function that assigns an initialization expression to each place p such that $\text{Type}[I(p)] = C(p)_{\text{MS}}$.

Definition 2. A Colored Petri net module can be expressed as a 4-tuple $\Sigma_M = (\Sigma, T_{\text{sub}}, P_{\text{port}}, PT)$, where:

1. $\Sigma = (P, T, A, C, V, W, G, E, I)$ is a non-hierarchical colored Petri net;
2. $T_{\text{sub}} \subseteq T$ is a set of substitution transitions;
3. $P_{\text{port}} \subseteq P$ is a set of port places;
4. $PT : P_{\text{port}} \rightarrow \{IN, OUT, I/O\}$ is a port type function that assigns a port type to each port place.

Definition 3. A Hierarchical-Colored Petri net can be expressed as a 4-tuple $\Sigma_H = (S, SM, PS, FS)$, where:

1. S is a finite set of modules. Each module is a Colored Petri net module $s = ((P^s, T^s, A^s, C^s, V^s, W^s, G^s, E^s, I^s), T_{sub}^s, P_{port}^s, PT^s)$, it is required that $(P^{s_1} \cup T^{s_1}) \cap (P^{s_2} \cup T^{s_2}) = \emptyset$ for all $s_1, s_2 \in S$ such that $s_1 \neq s_2$;
2. $SM : T_{sub} \rightarrow S$ is a submodule function that assigns a sub module to each substitution transition. It is required that the module hierarchy is acyclic;
3. PS is a port-socket relation function that assigns a port-socket relation $PS(t) \subseteq P_{sock}(t) \times P_{port}^{SM(t)}$ to each substitution transition t , it is required that $ST(p) = PT(p'), C(p) = C(p')$, and $I(p)\langle \rangle = I(p')\langle \rangle$ for all $(p, p') \in PS(t)$ and all $t \in T_{sub}$;
4. $FS \subseteq 2^P$ is a set of non-empty fusion sets such that $C(p) = C(p')$ and $I(p)\langle \rangle = I(p')\langle \rangle$ for all $p, p' \in fs$ and all $fs \in FS$.

In practical applications of PNs for system engineering modeling, complex business logic requirements may lead to the state space of a net system being extremely large such that it is very difficult to perform simulation and performance optimization. Fortunately, HCPN formalisms using fusion place, port/socket place and substitution transition (also known as aggregated transition) provide a hierarchical formulation to solve this issue, reducing the dimension of the state space. In other words, a complete large-scale CPN model can be divided into a number of sub-nets, which is easily represented and investigated.

When an HCPN executable model is built, according to the firing rules defined in [34], full state spaces representing all possible executions of the model can be generated. The full state space is described by a directed graph where the nodes correspond to the set of reachable markings, and the arcs correspond to the occurrence-binding elements. Therefore, in full state spaces, we can observe not only all reachable states, but also all state changes. On this basis, some critical system properties such as liveness, boundness, and fairness can be validated and guaranteed.

3.2. Assumptions and Rules

In this section, we make the following assumptions such that the obtained executable model is easily simulated and dynamic behavior analysis can be done.

1. In the urban traffic network, when one vehicle is approaching to an intersection, usually, the driver prefers to decelerate the vehicle and ensures that the vehicle safely passes through the intersection or waits in a queue. Hence, we set the nominal travel speed at V_{tv} km/h, and the RSU sensor for transit detection is deployed at a distance of D_{RSU} from the stop bar;
2. In each phase, the traffic lights turn on in the following order: green lights for T_G seconds, yellow lights for T_Y seconds, and red lights for T_R seconds. In order to ensure safety, when a cycle is completed, all-in-red lights turn on for T_{AR} seconds. Therefore, a complete signal cycle is $T_G + T_Y + T_R + T_{AR}$ seconds;
3. The event that the RSU sensor detects an approaching transit vehicle is stochastic and uncertain, but the traffic signal controller's response to the event is deterministic. Therefore, to cope with the complexity of the TSP control model, it is crucial to reduce some redundant states while a full state space is constructed and analyzed.

Different RSU sensors may detect transit vehicles and send conflicting requests to the traffic signal controller simultaneously. In order to handle this dilemma, we have the following rules:

1. If the traffic signal controller receives more than two green extension requests in each phase, these requests are handled in accordance to the principle that only the first one is served and the others are neglected;
2. The same principle can be used to the two or more red truncation requests;
3. If the controller receives two different requests at the same time, the response to the green extension request is superior to the response to the red truncation request.

For example, in Figure 1a, four approaching transit vehicles are detected in four different road sections, such as Lanes 1, 3, 5, and 7, and they send priority requests in sequence. Obviously, transit vehicles on Lanes 1 and 3 request green extension, while transit vehicles on Lanes 5 and 7 request red truncation. With the help of above rules, the requests made by transit vehicles on Lanes 5 and 7 are ignored first, and then the request made by transit vehicle on Lane 3 is ignored too, only the request issued on Lane 1 can get a response.

3.3. Green Extension Strategy

The conventional green extension strategy begins with an initial fixed green time; each time a transit vehicle is detected by the magnetic loops or cameras, it will add an extra green period until the maximum green time is reached [6]. In the proposed strategy, an extra green time will be appended to the green time, while the following two conditions are met simultaneously. The first one is that an approaching transit vehicle is detected by the RSU sensor; the second one is that the time when the transit vehicle is detected is at the specific green period. Different from the conventional strategy, this strategy conditionally decides whether an extra green time is appended based on the residual green time when the transit vehicle arrives at the stop bar.

The new green extension strategy that we propose is the following:

1. The residual green time when a transit vehicle is detected is expressed as T_{r_gt} . The additional green time is set to T_{δ_gt} ;
2. The distance between the RSU sensor installed on the road side and the stop bar is expressed as D_{RSU} . The nominal speed of the transit vehicle is expressed as V_{tv} , when it is detected by the RSU sensor. The estimated time to travel to the stop bar is expressed as $T_e = D_{RSU}/V_{tv}$;
3. When a TSP request is received by the controller in the current phase, it checks whether the detected transit vehicle has enough green time to travel to the stop bar or not, there are four cases should be considered:

- Case 1. If the condition $(T_e < T_{r_gt})$ holds, it means that the transit vehicle will travel through the intersection without delay.
- Case 2. If the condition $(T_{r_gt} \leq T_e)$ holds, it means that the yellow light will turn on and the transit vehicle has to wait for the green light in next cycle before the stop bar.
- Case 3. For Case 2, if the formula $(T_e - T_{\delta_gt} < T_{r_gt})$ is also satisfied, then an additional green time (T_{δ_gt}) can be appended to the residual green time such that the new remaining green time ($T_{r_gt} + T_{\delta_gt}$) is long enough to serve the transit vehicle during this phase.
- Case 4. If the condition $((0 < T_{r_gt}) \text{ and } (T_{r_gt} \leq (T_e - T_{\delta_gt})))$ holds, it means that the original residual green is between one to $(T_e - T_{\delta_gt})$ and is not long enough to serve this vehicle during this phase either. In this situation, even if an additional green time is appended, the transit vehicle has to wait until the next cycle.

3.4. Red Truncation Strategy

The conventional red truncation strategy also begins with an initial fixed red time; each time a transit vehicle is detected, a fixed period of non-transit phase is truncated from the scheduled time until the minimal red time is reached. In the proposed red truncation strategy, a fixed red time will be truncated from the scheduled red time, if and only if the transit vehicle is detected by the RSU sensor at the specific red light period. The difference from the conventional red truncation strategy is whether a red truncation operation is triggered depends on the remaining red time when the transit vehicle arrives at the stop bar.

The proposed new red truncation strategy is the following:

1. The residual red time when a transit vehicle is detected is expressed as T_{r_rt} . The threshold of red time to be truncated is set to T_{δ_rt} ;

2. The three parameters D_{RSU} , V_{tv} , T_e have the same meaning as in Section 3.3;
3. When a TSP request is received by the controller in the current phase, it checks if the detected transit vehicle needs to stop at least once. The more detailed information is summarized into four cases as follows.

- Case 1. If the condition $(T_{r_rt} \leq T_e)$ is satisfied, it means that the traffic signal will turn green before the transit vehicle travels to the stop bar.
- Case 2. If the condition $(T_{r_rt} > T_e)$ holds, it means that the detected transit vehicle need to stop at least once.
- Case 3. For Case 2, if the formula $(T_e \geq T_{r_rt} - T_{\delta_rt})$ is also satisfied, then a fixed period of red time (T_{δ_rt}) can be truncated from the current phase. Then, the new remaining red time $(T_{r_rt} - T_{\delta_rt})$ is guaranteed such that this transit vehicle can travel through the stop bar without any stop.
- Case 4. If the condition $(T_{r_rt} - T_e > T_{\delta_rt})$ holds, it means that the residual red time is quite long, beyond the adjustment range. Hence, the transit vehicle has to wait before the stop bar until the red light turns green.

3.5. General Modeling Methodology

This work proposes a general methodology for designing the HCPN-based TSP control model based on the idea of a Hierarchical design.

Inputs: An intersection with TSP requirements, RSUs.

Output: A HCPN-based TSP control model.

- Steps:*
- (1) Identify the main states of the signal control system and its transitions.
 - (2) Define the colorset, variable, and functions used in the potential CPN model.
 - (3) Design the four-phase (or two-phase, six-phase) traffic signal control CPN model.
 - (4) Using the hierarchical method, transform the CPN model into the HCPN model.
 - (5) Construct two HCPN submodules based on two proposed TSP strategies, and integrated into the HCPN model.
 - (6) According to the application requirements, set the following parameters: T_G , T_Y , T_R , T_{AR} , D_{RSU} , V_{tv} , T_{δ_gt} , T_{δ_rt} .
 - (7) Simulate and refine the HCPN-based TSP control model.
 - (8) Analyze the HCPN-based TSP control model.

3.6. Transit Signal Priority Control Model

In this section, the intersection given in Figure 1 is taken as an example. This intersection is structured for a TSP control system that includes four RSU sensors for detecting transit vehicles in each direction. An interesting feature of the TSP control model is that the event representing that an approaching transit vehicle is detected has a stochastic characteristic. Intuitively speaking, we cannot predetermine when, in which road section, and at which phase, there will be an approaching transit vehicle.

In HCPN-based TSP model, while a transit vehicle is traveling through the crossing zone, three spectacle situations could happen in the TSP control systems. We describe in detail how they operate by considering both the trajectory of transit vehicle and traffic light state while a transit vehicle is detected. The HCPNs in Figure 3 model the complete operation of priority strategies handled by the traffic signal controller while the transit vehicle goes straight through the crossing zone. In the proposed model, the colors of the potential tokens must satisfy the expressions associated with the

colors in the arc expressions. The colors and variables used to describe the information required in the relevant places and arcs are summarized in Table 1. Furthermore, in order to improve the readability, we denote EW and NS as directions, including east-westbound and north-southbound, and GS and TL as the behavior of a transit vehicle at the intersection such as go-straight (including turn-right) and turn-left, as well as AR, R, G, Y as traffic lights, such as all-in-red, red, green, and yellow. The physical meanings of places and transitions are explained in Table 2. The parameters used in the HCPNs are set as follows: $T_G = 27$ s, $T_Y = 3$ s, $T_R = 90$ s, $T_{AR} = 2$ s, $D_{RSU} = 100$ m, $V_{tv} = 10$ m/s, $T_{\delta_{gt}} = 5$ s, and $T_{\delta_{rt}} = 5$ s.

In order to reduce the complexity of the proposed model further, we model the TSP control in two levels with one module at the upper level and six sub-modules at the lower level, which represent the detailed process of traffic lights in each phase, respectively. In Figure 3a, there are six aggregated transitions (phase1 to phase4, green extension, and red truncation), which are represented by boxes with double-line borders. Each aggregated transition has a submodule related to it. Essentially, a submodule presents a more detailed view of the behavior represented by the aggregated transition. Here, these six aggregated transitions correspond to the submodules shown in Figure 3b–g, respectively.

Two transitions in Figures 3f and 4g, green extension and red truncation, model the TSP control system's interfaces. Therefore, a transit vehicle's priority request can be transferred to the signal controller through them. Technically, with the help of these transitions and two groups of places {EW_GS_G, EW_TL_R, NS_GS_R, NS_TL_R} and {EW_GS_R, EW_TL_R, NS_GS_R, NS_TL_G}, two priority strategy models can be incorporated into the normal traffic signal control model to handle the priority request initiated by one or more approaching transit vehicles.

Table 1. Color specification in Figure 3.

Name	Type	Descriptions	
		Definition	Meaning
TIME	Colorset	Int 1..N	Time stamp
SENSOR	Colorset	String	Road side sensor
COLOR	Colorset	Enumeration with G Y R	Traffic light color
SL	Colorset	Production COLOR * TIME	Traffic light
randnum	Colorset	Int with 1..4	Random event of transit vehicle is detected
ELtick	Variable	ELtick: TIME	Time stamp of turn-left lights in eastbound and westbound traffic directions
EGtick	Variable	EGtick: TIME	Time stamp of go-straight lights in eastbound and westbound traffic directions
NLtick	Variable	NLtick: TIME	Time stamp of turn-left lights in northbound and southbound traffic directions
NGtick	Variable	NGtick: TIME	Time stamp of go-straight left lights in northbound and southbound traffic directions
t	Variable	t: TIME	Value of time stamp
s	Variable	s: SENSOR	
c	Variable	c: Color	
n	Variable	n: randnum	

In Figure 3f, we depict the proposed green extension strategy in the CPN model. The physical meanings of places and transitions are illustrated in Table 2. For example, the northbound and southbound go-straight and turn-left lights are red, the eastbound and westbound turn-left lights are red, only the eastbound and westbound go-straight lights are green and the residual time is between six to 10 s. At this moment, if an approaching transit vehicle is detected, then the green extension strategy can be considered, as shown in arc expressions in Figure 3f:

if ($0 \leq t - EGtick$) *andalso* ($t - EGtick < 5$) *then* $1'(glt, EGtick + 5)$ *else* $1'(glt, EGtick)$

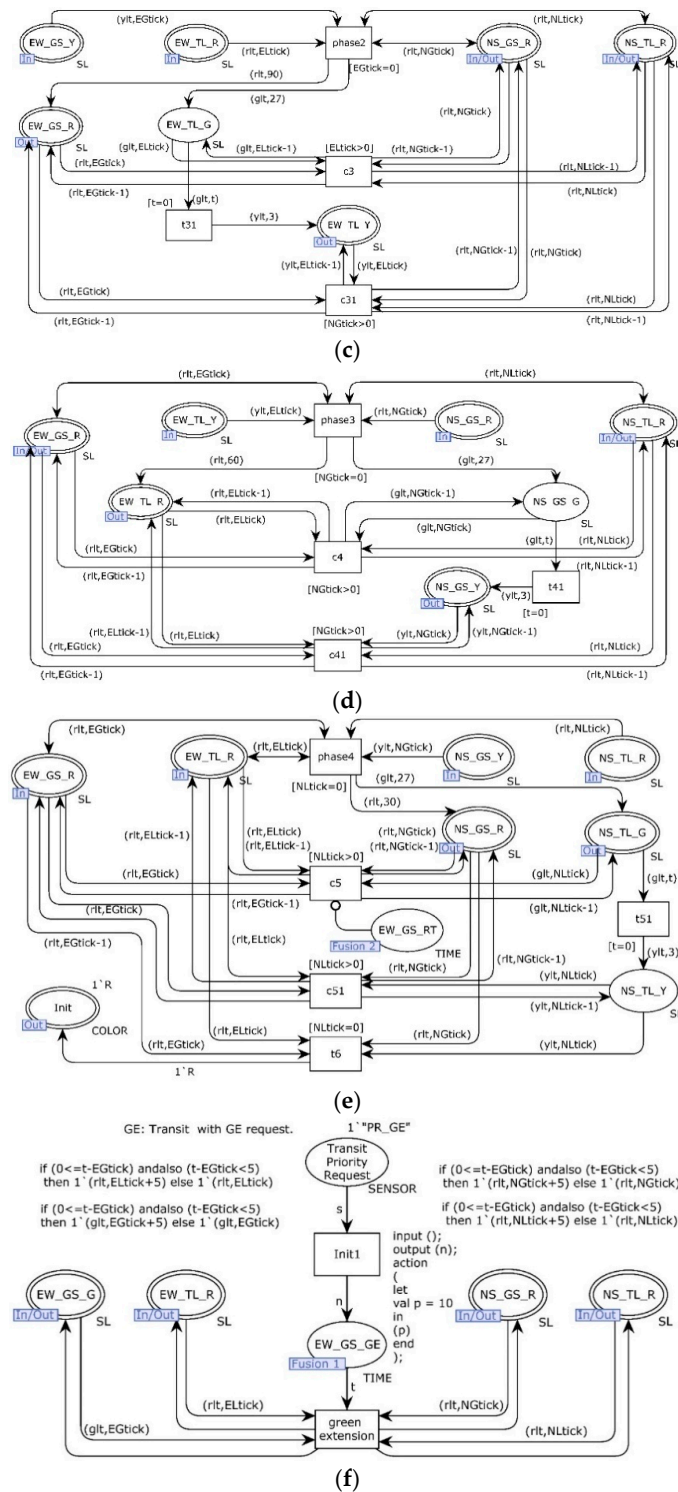


Figure 3. Cont.

are six transitions: enter_all_in_red, c1, phase1, c2, t21 and c21. Transition enter_all_in_red is used to represent that the status of the red signals are in all-in-red state, which means that the red lights in eastbound, westbound, northbound and southbound should be turn on to forbid the incoming vehicles entering the crossing zone. A detailed description of the simulation process of HCPN-based TSP model can be found in Appendix A.

In summary, the markings of the proposed model can be observed, and we can conclude that no place has an unbound number of tokens. Similar analysis can be applied to the scenario of a single transit line turning left through the intersection.

4.2. State Space Analysis

In this section, we describe the properties of the proposed HCPN models in detail. Two full state spaces are generated based on the current initial marking, and the statistics obtained by full state space analysis are illustrated in Table 3, which can be used to analyze the boundness and liveness of the TSP control model with transit vehicle go-straight or turn-left through the crossing zone. In order to verify the correctness and validation of these two strategies, and to save space, the partial state spaces are employed in Figure 4. Arcs in Figure 4 are listed in Appendix B. Compared with the state space of the normal four-phase traffic light control model, the number of nodes and arcs in the state spaces and the Scc graph both increase greatly because the proposed model represents the interactions between transit and traffic controller, and enumerates all the possible responses made by the traffic controller. These functions are exactly what traditional TSP control strategies cannot provide.

Table 3. Statistics of simulations of Hierarchical Colored Petri nets (HCPN) model. TSP: transit signal priority.

Model	State Spaces		Scc Graph		Boundness		Dead Marking	Live Transition	Fairness
	Nodes	Arcs	Nodes	Arcs	Upper	Lower			
Normal	132	132	1	0	1	0	None	All	All
TSP	1188	1986	664	1458	1	0	None	Part	Part

Part: transitions in phase1, phase2, phase3, and phase4 submodules.

In Figure 4a, node1 represents the initial state of the TSP control system, we see that node1 has three enabled binding elements:

Arc1:1→2 (phase1'enter_all_in_red:{c=R})

Arc2:1→3 (green_extension'Init1:{s="Bus Request GE", n=10})

Arc3:1→4 (red_truncation'Init2:{s="Bus Request RT", n=10})

These arcs lead to the markings represented by nodes2, node3, and node4, respectively. Node3 and node4 represent that an approaching transit vehicle is detected at the initial state; actually, the transit priority request issued at this time will be neglected. Node2 represents the model entering the all-in-red state. Nodes2, node5, and node9 have two enabled binding elements:

Arc4:2→5 (phase1'c1:{EGtick=2, ELtick=2, NGtick=2, NLtick=2})

Arc11:5→9 (phase1'c1:{EGtick=1, ELtick=1, NGtick=1, NLtick=1})

We illustrate the countdown of the time stamp in the all-in-red state. The markings represented by nodes7, node11, and node15 are identical to the node4 so that the transition Init2 in red_truncation fires. Similarly, the markings represented by nodes6, node10, and node14 are identical to the node3 so that the transition Init1 in green_extension fires. We found that the sequence of states that really make sense is {node1, node2, node5, node9, node13, ... }.

Figure 4b shows the partial state spaces concerning the green extension strategy. Note that in a real transit signal priority control system, no one knows when an approaching transit vehicle is detected. Meanwhile, we know that once a transit vehicle is detected, the response of the TSP control

system to the priority request is a deterministic process. There are five groups of nodes and arcs ,i.e., {node110, node117, node95, Arc208, Arc221}, {node116, node123, node101, Arc219, Arc232}, {node122, node129, node107, Arc230, Arc243}, {node128, node135, node113, Arc241, Arc254}, and {node134, node141, node119, Arc252, Arc265} that indicate the state change of the TSP control system when the priority request is received in the east-westward go-straight green time of 10 to 15 s. In Figure 4b, node98 is reachable from node13, but is not reachable from node14 and node15.

Figure 4c demonstrates the partial state spaces concerning the red truncation strategy. Node725 is reachable from node132, node147 and node146. There exist five sets of nodes and arcs , i.e., {node735, node747, node759, Arc1336, Arc1355}, {node746, node758, node771, Arc1354, Arc1373}, {node757, node770, node783, Arc1372, Arc1392}, {node769, node782, node794, Arc1391, Arc1411}, {node781, node793, node804, Arc1410, Arc1428} that exhibit the red truncation process when the go-straight red light time is between six to 10 seconds in east-westward traffic direction, respectively. Finally, node823 and node848 will lead to node868, indicating that a specific state of the TSP control system, that is, the transition phase4't6 is enabled. After it fires, it will lead to a state loop consisting of 132 nodes, such as {node868 → node877 → node885 →, ... , → node859 → node868}.

Based on the above analysis, the trajectory of the state change of the TSP control system can be divided into five scenarios: no transit vehicle requests priority (Scenario 1), two different transit vehicles request green extension and red truncation in the same cycle (Scenario 2), only one transit vehicle requests green extension (Scenario 3), only one transit vehicle requests red truncation (Scenario 4), one or more transit vehicles request priority but the TSP control system does not respond (Scenario 5). We list the state sequences for each scenario in Table 4. These scenarios show that a reasonable sequence of states can be found in the full state space as long as a transit vehicle is detected at a specified period of a red or green signal, indicating a response to the transit's priority request. Conversely, even if the model does not process the transit priority request, it can still operate according to the normal four-phase traffic signal controller.

After the examination, we can infer that, if there is no limit on the token number of transit priority requests, which represents the number of transit vehicles to be considered in the simulation, the proposed TSP control models are live and reversible. Their state space analysis proves that the liveness and reversibility are guaranteed. Therefore, the practical TSP control system implemented according to this model will not cause man-made conflicts between vehicles in different movements.

Table 4. Possible state sequences of TSP control system.

Different Scenario	Possible State Sequences (Node#)	Uniqueness
1	1, 2, 5, 9, 13, ... , 98, 104, 110, 116, 122, 128, 134, 140, 146, ... , 194, ... , 388, ... , 580, ... , 872, 1	YES
2	1, 2, 5, 9, 13, ... , 98, 104, 110, ... , 132, ... , 725, 735, ... , 823, ... , 868, 877, 885, ... , 859, 868	NO
3	1, 2, 5, 9, 13, ... , 98, 104, 110, ... , 132, ... , 725, 735, ... , 848, ... , 868, 877, 885, ... , 859, 868	NO
4	1, 2, 5, 9, 13, ... , 98, 104, 110, 116, 122, 128, 134, 140, 146, ... , 725, 735, ... , 823, ... , 868, 877, 885, ... , 859, 868	NO
5	1, 2, 6, 12, 16, 20, 25, 31, ... , 217, ... , 409, ... , 602, ... , 868, 877, 885, ... , 859, 868	NO

4.3. Case Study

This section briefly discusses the application of the HCPN-based TSP control model based on several simulation-based experiments. These data can be used to analyze the capability of providing a method for improving the efficiency of the transit vehicle at intersection. A single intersection (Figure 1) project is generated in VISSIM [36], a commonly used microscopic traffic simulation software. Three kinds of signal control algorithm are implemented in the experiment, such as the fixed time method (FX), conventional TSP algorithm (CTSP) in [6,10], and the proposed HCPN-based TSP control model (PNTSP). The length of each road connecting the intersection is set to 500 m. The traffic flows on each road are basically the same, with values of 900, 1200 and 1500 vehicle per hour (vph), respectively. The cycle length at the intersection is 122 s. What needs to be specially explained is that the indexes

such as the number of transit stops and transit delays are only counted from the road from eastbound to westbound while the total delay of social vehicles is based on statistics on four connecting roads.

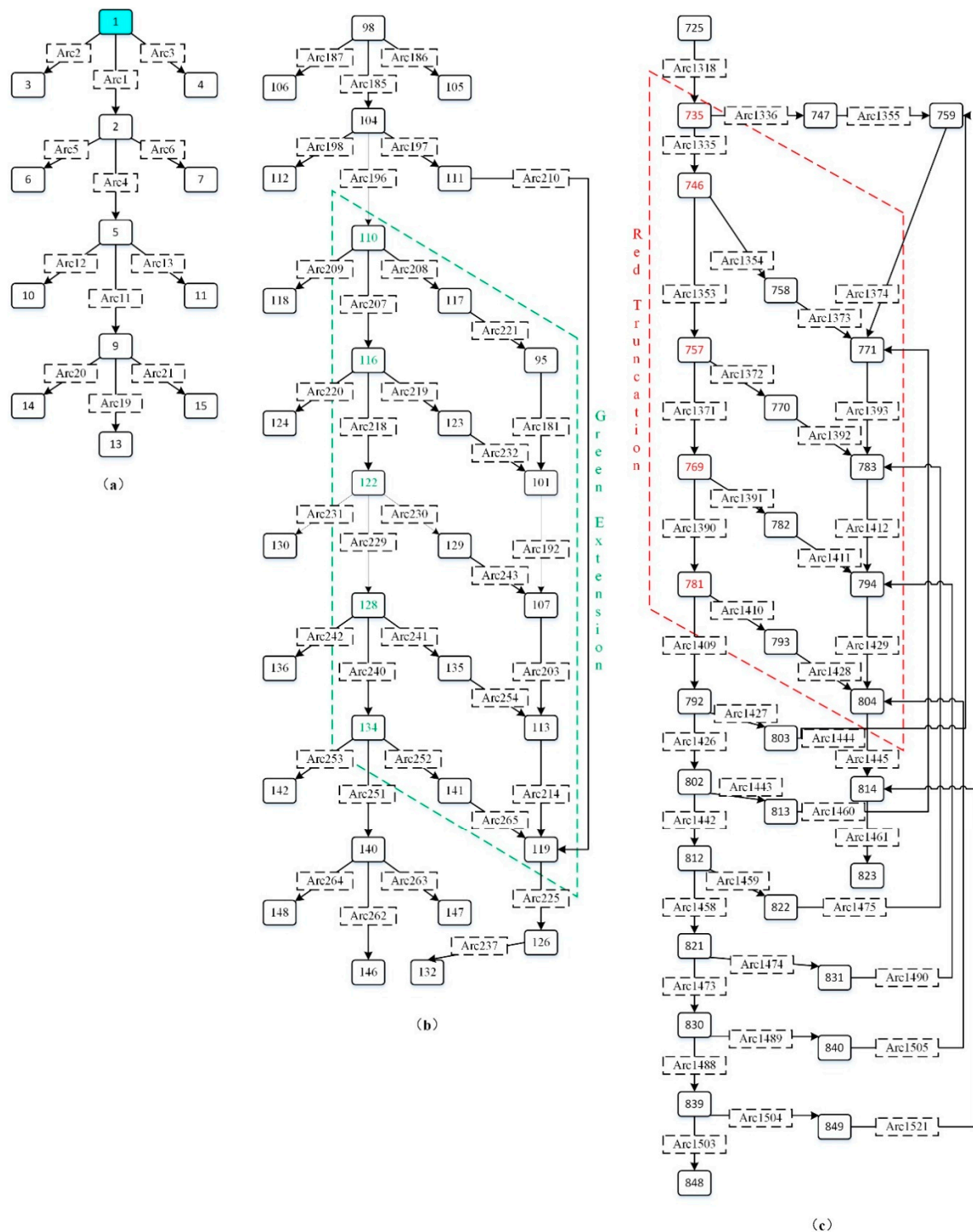


Figure 4. Partial state spaces of the transit signal priority control HCPN model.

Simulation-based results are shown in Table 5. The data in column FX serves as a baseline for comparison. Obviously, no matter what kind of traffic flow conditions and what kind of transit ratio are configured, CTSP and PNTSP methods can achieve a better performance than the FX method for

both the number of transit stops and the transit delay time, although the rate of improvement will decrease as traffic flow increases. In addition, according to the indicator of the total delay of social vehicle, the PNTSP method shows better performance over the CTSP method. That is to say, it keeps the FX method's performance, while the total delay of social vehicles derived from the CTSP method increased by about 20%. The reason for this is that the traditional method mainly predicts traffic flow based on historical data, and then greatly adjusts the phase and duration of signal control, leading to additional delay for vehicles on other connecting roads. The proposed method, by its very nature, is a reactive fixed timing approach. Therefore, when using this method, the transit vehicles have the least impact on social vehicles. At the same time, it also extends the service time for the transit vehicles in a specific green light period and assigns the service time for transit vehicles in a particular red light period. Based on the simulation results, it can be concluded that the proposed HCPN-based TSP control model provides transit vehicles with more accuracy and better effectiveness at the intersection.

Table 5. Statistics of simulations in VISSIM.

Traffic Flows (vph)	Transit Ratio (%)	Number of Transit Stop (Times)			Transit Delay (M)			Total Delay of Social Vehicle (H)		
		FX	CTSP	PNTSP	FX	CTSP	PNTSP	FX	CTSP	PNTSP
900	1	1	1	1	0.36	0.36	0.36	2.87	3.301	2.871
	5	2	2	2	0.63	0.57	0.6	2.859	3.315	2.859
	10	5	4	3	2.42	2.23	1.5	2.877	3.279	2.579
	15	9	7	4	2.46	2.38	1.48	2.796	3.221	2.697
1200	1	2	1	1	0.72	0.63	0.7	4.132	4.743	4.137
	5	3	3	3	0.66	0.62	0.64	4.062	4.858	4.046
	10	7	5	4	3.12	2.81	1.85	4.071	4.865	4.012
	15	11	8	8	4.52	3.91	2.9	4.113	4.944	4.317
1500	1	2	2	2	1.08	1.08	1.06	6.186	7.612	6.495
	5	5	4	4	4.26	3.83	3.72	6.096	7.501	6.584
	10	10	8	9	8.52	7.82	7.96	6.25	7.675	6.825
	15	21	15	19	12.42	11.47	12.1	6.419	7.893	6.908

FX: fixed time method; CTSP: conventional TSP algorithm; PNTSP: the proposed HCPN-based TSP control model.

5. Conclusions

This work for the first time uses HCPN to design a traffic control system for a four-phase intersection dealing with priority to provide public transit passing through the intersection. Two kinds of transit priority strategies, i.e., green extension and red truncation are formalized in this work. Their analysis is performed by the state space method. Their important properties such as boundness, liveness and reversibility are verified. This helps to enhance the state of the art in traffic real-time detection and traffic management of an intersection. Future work is to design a traffic control system that is complex and close to the actual application. For example, the impact of pedestrians on the transit priority should be considered. In addition, if there are multiple transit lines overlapping at the intersection, transits on different lanes may also make a priority request at the same time; hence, how to solve these complex scenes should be considered. Also, in this work, we use a four-phase signal control system with a deterministic phase transition structure. In the future, we will explore the effect of a non-deterministic phase transition structure on TSP control strategies.

Acknowledgments: This paper has been supported by the National Natural Science Foundation of China under Grant No. (51278058, 61703053), the Natural Science Foundation of Shaanxi Province under Grant No. 2014JZ019, China Postdoctoral Science Foundation under Grant No. 2012M521729, and the Fundamental Research Funds for the Central Universities under Grant No. 310824162021.

Author Contributions: This work was conducted by Yisheng An and Xiangmo Zhao with the help of graduate student Xuan Li and Pei Chen. It was mainly drafted by Xuan Li and Pei Chen, and checked and revised by

Yisheng An and Naiqi Wu. Yisheng An and Xuan Li designed and analyzed the proposed model. Pei Chen performed the simulation. Naiqi Wu were responsible for the English polish and proofreading of this work.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

In the initial state, there is a token (1'R) in the Init place:

$$M_0(p) = \begin{cases} 1'R & \text{if } p = \text{Init}_{\text{sock}}^{\text{in}}(\text{phase1}) \\ \emptyset & \text{otherwise} \end{cases}$$

When the transition enter all-in-red fires, a new marking is obtained. That is $M_0[\text{enter all_in_red} > M_1$, this means that a token is removed from Init place, and four tokens are deposited into four all-in-red place respectively,

$$M_1(p) = \begin{cases} 1'(R, 2) & \text{if } p \in \{EW_GS_AR(\text{phase1}), EW_TL_AR(\text{phase1}), \\ & NS_GS_AR(\text{phase1}), NS_TL_AR(\text{phase1})\} \\ \emptyset & \text{otherwise} \end{cases}$$

in M_1 , the duration of all-in-red lights is set to two seconds, then the condition of c1 is meet, when the countdown of all-in-red light reaches zero, that means time stamps in places EW_GS_AR, EW_TL_AR, NS_GS_AR, NS_TL_AR are reduced to 0, then the transition phase1 is enabled, that is $M_1[(c1)^2 \text{phase1} > M_2$. If the duration of each signal light is modified, it can be done by altering the value in the arc expressions involving (glt, 27), (rlt, 30), (rlt, 60), (rlt, 90). Then, the signal light begins its first round of cycle:

$$M_2(p) = \begin{cases} 1'(G, 27) & \text{if } p = EW_GS_G_{\text{sock}}^{\text{out}}(\text{phase1}) \\ 1'(R, 30) & \text{if } p = EW_TL_R_{\text{sock}}^{\text{out}}(\text{phase1}) \\ 1'(R, 60) & \text{if } p = NS_GS_R_{\text{sock}}^{\text{out}}(\text{phase1}) \\ 1'(R, 90) & \text{if } p = NS_TL_R_{\text{sock}}^{\text{out}}(\text{phase1}) \\ 1''PR_GE'' & \text{if } p = \text{Transit Priority Request}(\text{green extension}) \\ 1''PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

M_2 indicates that the go-straight green light in the east-westward traffic direction turns on, hence, the transition c2 is enabled, and starts the countdown of traffic light. Consider a specific scenario, the residual green time in EW_GS_G is reduced to 10 s. At this time, if an approaching transit vehicle is detected, such as in Lane 1 or Lane 3 in Figure 1a, then the transition green extension in Figure 4d is enabled, that is $M_2[(c2)^{17} \text{Init1} > M_3$:

$$M_3(p) = \begin{cases} 1'(G, 10) & \text{if } p = EW_GS_G_{\text{sock}}^{\text{in/out}}(\text{green extension}) \\ 1'(R, 13) & \text{if } p = EW_TL_R_{\text{sock}}^{\text{in/out}}(\text{green extension}) \\ 1'(R, 43) & \text{if } p = NS_GS_R_{\text{sock}}^{\text{in/out}}(\text{green extension}) \\ 1'(R, 73) & \text{if } p = NS_TL_R_{\text{sock}}^{\text{in/out}}(\text{green extension}) \\ 1''PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ 1'10 & \text{if } p = EW_GS_GE(\text{green extension}) \\ \emptyset & \text{otherwise} \end{cases}$$

in M_3 , the conditions for starting the transit priority strategy are satisfied, therefore, the transition green extension is enabled, which means that the residual green time in this traffic direction can

be extended. After green extension fires, a new state is generated by making the occurrence of $M_3[\text{green extension} > M_4]$:

$$M_4(p) = \begin{cases} 1'(G, 15) & \text{if } p = EW_GS_G_{sock}^{out}(phase1) \\ 1'(R, 18) & \text{if } p = EW_TL_R_{sock}^{out}(phase1) \\ 1'(R, 48) & \text{if } p = NS_GS_R_{sock}^{out}(phase1) \\ 1'(R, 78) & \text{if } p = NS_TL_R_{sock}^{out}(phase1) \\ 1'' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

in M_4 , the residual time of the go-straight green light in east-westward traffic direction is extended to 15 s, after that, the transition c2 is enabled again, and then all traffic lights start counting down until the green time in place EW_GS_G is reduced to zero. A new marking is generated by making the occurrence of $M_4[(c2)^{15} > M_5]$:

$$M_5(p) = \begin{cases} 1'(G, 0) & \text{if } p = EW_GS_G_{sock}^{out}(phase1) \\ 1'(R, 3) & \text{if } p = EW_TL_R_{sock}^{out}(phase1) \\ 1'(R, 33) & \text{if } p = NS_GS_R_{sock}^{out}(phase1) \\ 1'(R, 63) & \text{if } p = NS_TL_R_{sock}^{out}(phase1) \\ 1'' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

in M_5 , the transition t21 is enabled, after it fires, we can get a new marking M_6 , which means that the go-straight green light in east-westward traffic direction switches to yellow light.

$$M_6(p) = \begin{cases} 1'(Y, 3) & \text{if } p = EW_GS_Y_{sock}^{out}(phase1) \\ 1'(R, 3) & \text{if } p = EW_TL_R_{sock}^{out}(phase1) \\ 1'(R, 33) & \text{if } p = NS_GS_R_{sock}^{out}(phase1) \\ 1'(R, 63) & \text{if } p = NS_TL_R_{sock}^{out}(phase1) \\ 1'' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

in M_6 , the transition c21 is enabled, when the time stamp in EW_GS_Y is reduced to zero, as shown in M_7 . It indicates the end of phase one of the traffic control system.

$$M_7(p) = \begin{cases} 1'(Y, 0) & \text{if } p = EW_GS_Y_{sock}^{in}(phase2) \\ 1'(R, 0) & \text{if } p = EW_TL_R_{sock}^{in}(phase2) \\ 1'(R, 30) & \text{if } p = NS_GS_R_{sock}^{in/out}(phase2) \\ 1'(R, 60) & \text{if } p = NS_TL_R_{sock}^{in/out}(phase2) \\ 1'' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

in M_7 , transition phase2 is enabled. It means that the phase two can start. After it fires, four new tokens are drop into EW_GS_R , EW_TL_G , NS_GS_R , NS_TL_R , respectively, as shown in M_8 :

$$M_8(p) = \begin{cases} 1'(R, 90) & \text{if } p = EW_GS_R_{sock}^{out}(phase2) \\ 1'(G, 27) & \text{if } p = EW_TL_G_{sock}(phase2) \\ 1'(R, 30) & \text{if } p = NS_GS_R_{sock}^{in/out}(phase2) \\ 1'(R, 60) & \text{if } p = NS_TL_R_{sock}^{in/out}(phase2) \\ 1'' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

This marking indicates that the turn-left green lights turn on and continue for 27 s. After transitions $c3$, $t31$, $c31$ occur in sequence at marking M_8 , a new state is generated by making the occurrence of $M_8[(c3)^{27}t31(c31)^3 > M_9$:

$$M_9(p) = \begin{cases} 1'(R, 60) & \text{if } p = EW_GS_R_{sock}^{in/out}(phase3) \\ 1'(Y, 0) & \text{if } p = EW_TL_Y_{sock}^{in}(phase3) \\ 1'(R, 0) & \text{if } p = NS_GS_R_{sock}^{in}(phase3) \\ 1'(R, 30) & \text{if } p = NS_TL_R_{sock}^{in/out}(phase3) \\ 1'' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

in M_9 , transition $phase3$ is enabled, it can fire. Firing the enabled transition $phase3$ at marking M_9 yields marking M_{10} , it means that the go-straight green lights in north-southward traffic direction turn on and continue for 27 s.

$$M_{10}(p) = \begin{cases} 1'(R, 60) & \text{if } p = EW_GS_R_{sock}^{in/out}(phase3) \\ 1'(R, 60) & \text{if } p = EW_TL_R_{sock}^{out}(phase3) \\ 1'(G, 27) & \text{if } p = NS_GS_G(phase3) \\ 1'(R, 30) & \text{if } p = NS_TL_R_{sock}^{in/out}(phase3) \\ 1''' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

in marking M_{10} , the transition firing sequence $(c4)^{27}t41(c41)^3$ can be executed and results in a new marking M_{11} , which enables the transition $phase4$ in Figure 4e.

$$M_{11}(p) = \begin{cases} 1'(R, 30) & \text{if } p = EW_GS_R_{sock}^{in}(phase4) \\ 1'(R, 30) & \text{if } p = EW_TL_R_{sock}^{in}(phase4) \\ 1'(Y, 0) & \text{if } p = NS_GS_Y_{sock}^{in}(phase4) \\ 1'(R, 0) & \text{if } p = NS_TL_R_{sock}^{in}(phase4) \\ 1''' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

After $phase4$ fires, a new state which means that the turn-left green light in north-southward traffic direction is generated and is represented as the following marking:

$$M_{12}(p) = \begin{cases} 1'(R, 30) & \text{if } p = EW_GS_R_{sock}^{in}(phase4) \\ 1'(R, 30) & \text{if } p = EW_TL_R_{sock}^{in}(phase4) \\ 1'(R, 30) & \text{if } p = NS_GS_R_{sock}^{out}(phase4) \\ 1'(G, 27) & \text{if } p = NS_TL_G_{sock}^{out}(phase4) \\ 1''' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

In Figure 3, besides the green extension priority strategy, the red truncation priority strategy is also illustrated. A specific system state M_{13} which can be generated by the execution of transition firing sequence $(c5)^{17}$ is considered to illustrate this strategy.

$$M_{13}(p) = \begin{cases} 1'(R, 13) & \text{if } p = EW_GS_R_{sock}^{in}(phase4) \\ 1'(R, 13) & \text{if } p = EW_TL_R_{sock}^{in}(phase4) \\ 1'(R, 13) & \text{if } p = NS_GS_R_{sock}^{out}(phase4) \\ 1'(G, 10) & \text{if } p = NS_TL_G_{sock}^{out}(phase4) \\ 1''' PR_RT'' & \text{if } p = \text{Transit Priority Request}(\text{red truncation}) \\ \emptyset & \text{otherwise} \end{cases}$$

M_{13} indicates that transition $c5$ is enabled and fires 17 times continuously until the residual red time in EW_GS_R is reduced to 13 s. Suppose that at this time, an approaching transit vehicle is detected in east-westward traffic direction, then the transition $Init2$ in Figure 4e is enabled. After it fires, a new marking is generated by making the occurrence of $M_{13}[Init2 > M_{14}]$, then transition red truncation in Figure 4e is enabled, which means the residual red time in east-westward traffic direction can be truncated.

$$M_{14}(p) = \begin{cases} 1'(R, 13) & \text{if } p = EW_GS_R_{sock}^{in/out}(\text{red truncation}) \\ 1'(R, 13) & \text{if } p = EW_TL_R_{sock}^{in/out}(\text{red truncation}) \\ 1'(R, 13) & \text{if } p = NS_GS_R_{sock}^{in/out}(\text{red truncation}) \\ 1'(G, 10) & \text{if } p = NS_TL_G_{sock}^{in/out}(\text{red truncation}) \\ 1'10 & \text{if } p = EW_GS_RT \\ \emptyset & \text{otherwise} \end{cases}$$

After transition red truncation fires, a new state is obtained by making the occurrence of $M_{14}[\text{red truncation} > M_{15}]$:

$$M_{15}(p) = \begin{cases} 1'(R, 8) & \text{if } p = EW_GS_R_{sock}^{in}(\text{phase4}) \\ 1'(R, 8) & \text{if } p = EW_TL_R_{sock}^{in}(\text{phase4}) \\ 1'(R, 8) & \text{if } p = NS_GS_R_{sock}^{out}(\text{phase4}) \\ 1'(G, 5) & \text{if } p = NS_TL_G_{sock}^{out}(\text{phase4}) \\ \emptyset & \text{otherwise} \end{cases}$$

At this state, the countdown transition $c5$ can be fired continuously until the residual green time in NS_TL_G is reduced to zero, then the transition $t51$ is enabled and after its firing, the turn-left green lights in north-southward direction turn yellow, that is $M_{15}[(c5)^5 t51 > M_{16}]$:

$$M_{16}(p) = \begin{cases} 1'(R, 3) & \text{if } p = EW_GS_R_{sock}^{in}(\text{phase4}) \\ 1'(R, 3) & \text{if } p = EW_TL_R_{sock}^{in}(\text{phase4}) \\ 1'(R, 3) & \text{if } p = NS_GS_R_{sock}^{out}(\text{phase4}) \\ 1'(Y, 3) & \text{if } p = NS_TL_Y(\text{phase4}) \\ \emptyset & \text{otherwise} \end{cases}$$

in M_{16} , the countdown transition $c51$ can be fired three times continuously until the yellow time is reduced to zero, then transition $t6$ is enabled. After $t6$ fires, a token is deposited in $Init$ place which implies that the marking of the signal control system is transferred to a new marking M_{17} which is similar to the initial marking M_0 except two places Transit Priority Request not having tokens.

Appendix B

Appendix B.1. Arcs Used in Figure 4a

Arc1:1→2 phase1'enter_all_in_red:{c=R}
 Arc2:1→3 green_extension'Init1:{s="PR_GE", n=10}
 Arc3:1→4 red_truncation'Init2:{s="PR_RT", n=10}
 Arc4:2→5 phase1'c1:{EGtick=2, ELtick=2, NGtick=2, NLtick=2}
 Arc5:2→6 green_extension'Init1:{s="PR_GE", n=10}
 Arc6:2→7 red_truncation'Init2:{s="PR_RT", n=10}
 Arc11:5→9 phase1'c1:{EGtick=1, ELtick=1, NGtick=1, NLtick=1}
 Arc12:5→10 green_extension'Init1:{s="PR_GE", n=10}
 Arc13:5→11 red_truncation'Init2:{s="PR_RT", n=10}
 Arc19:9→13 phase1'phase1:{t=0}

Arc20:9→14 green_extension'Init1:{s="PR_GE", n=10}
 Arc21:9→15 red_truncation'Init2:{s="PR_RT", n=10}

Appendix B.2. Arcs Used in Figure 4b

Arc181:95→101 phase1'c2:{EGtick=15, ELtick=18, NGtick=48, NLtick=78}
 Arc185:98→104 phase1'c1:{EGtick=12, ELtick=15, NGtick=45, NLtick=75}
 Arc186:98→105 green_extension'Init1:{s="PR_GE", n=10}
 Arc187:98→106 red_truncation'Init2:{s="PR_RT", n=10}
 Arc192:101→107 phase1'c2:{EGtick=14, ELtick=17, NGtick=47, NLtick=77}
 Arc196:104→110 phase1'c2:{EGtick=11, ELtick=14, NGtick=44, NLtick=74}
 Arc197:104→111 green_extension'Init1:{s="PR_GE", n=10}
 Arc198:104→112 red_truncation'Init2:{s="PR_RT", n=10}
 Arc203:107→113 phase1'c2:{EGtick=13, ELtick=16, NGtick=46, NLtick=76}
 Arc207:110→116 phase1'c2:{EGtick=10, ELtick=13, NGtick=43, NLtick=73}
 Arc208:110→117 green_extension'Init1:{s="PR_GE", n=10}
 Arc209: 110→118 red_truncation'Init2:{s="PR_RT", n=10}
 Arc210:111→119 green_extension'green_extension:{EGtick=11, NGtick=44, ELtick=14, NLtick=74, t=10}
 Arc214:113→119 phase1'c2:{EGtick=12, ELtick=15, NGtick=45, NLtick=75}
 Arc218:116→122 phase1'c2:{EGtick=9, ELtick=12, NGtick=42, NLtick=72}
 Arc219:116→123 green_extension'Init1:{s="PR_GE", n=10}
 Arc220:116→124 red_truncation'Init2:{s="PR_RT", n=10}
 Arc221:117→95 green_extension'green_extension:{EGtick=10, ELtick=13, NGtick=43, NLtick=73, t=10}
 Arc225:119→126 phase1'c2:{EGtick=11, ELtick=14, NGtick=44, NLtick=74}
 Arc229:122→128 phase1'c2:{EGtick=8, ELtick=11, NGtick=41, NLtick=71}
 Arc230:122→129 green_extension'Init1:{s="PR_GE", n=10}
 Arc231: 122→130 red_truncation'Init2:{s="PR_RT", n=10}
 Arc232:123→101green_extension'green_extension:{EGtick=9, ELtick=12, NGtick=42, NLtick=72, t=10}
 Arc237:126→132 phase1'c2:{EGtick=10, ELtick=13, NGtick=43, NLtick=73}
 Arc240:128→134 phase1'c2:{EGtick=7, ELtick=10, NGtick=40, NLtick=70}
 Arc241:128→135 green_extension'Init1:{s="PR_GE", n=10}
 Arc242: 128→136 red_truncation'Init2:{s="PR_RT", n=10}
 Arc243:129→107 green_extension'green_extension:{EGtick=8, ELtick=11, NGtick=41, NLtick=71, t=10}
 Arc251:134→140 phase1'c2:{EGtick=6, ELtick=9, NGtick=39, NLtick=69}
 Arc252:134→141 green_extension'Init1:{s="PR_GE", n=10}
 Arc253:134→142 red_truncation'Init2:{s="PR_RT", n=10}
 Arc254:135→113 green_extension'green_extension:{EGtick=7, ELtick=10, NGtick=40, NLtick=70, t=10}
 Arc262: 140→146 phase1'c2:{EGtick=5, ELtick=8, NGtick=38, NLtick=68}
 Arc263:140→147 green_extension'Init1:{s="PR_RT", n=10}
 Arc264:140→148 red_truncation'Init2:{s="PR_RT", n=10}
 Arc265:141→119 green_extension'green_extension:{EGtick=6, ELtick=9, NGtick=39, NLtick=69, t=10}

Appendix B.3. Arcs Used in Figure 4c

Arc1318:725→735 phase'c5:{EGtick=16, ELtick=16, NGtick=16, NLtick=13}
 Arc1335:735→746 phase'c5:{EGtick=15, ELtick=15, NGtick=15, NLtick=12}
 Arc1336:735→747 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1353:746→757 phase'c5:{EGtick=14, ELtick=14, NGtick=14, NLtick=11}
 Arc1354:746→758 red_truncation'Init2:{s="PR_RT", n=10}

Arc1355:747→759 red_truncation' red_truncation:{EGtick=15, ELtick=15, NGtick=15, NLtick=12, t=10}
 Arc1371:757→769 phase'c5:{EGtick=13, ELtick=13, NGtick=13, NLtick=10}
 Arc1372:757→770 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1373:758→771 red_truncation' red_truncation:{EGtick=14, ELtick=14, NGtick=14, NLtick=11, t=10}
 Arc1374:759→771 phase'c5:{EGtick=10, ELtick=10, NGtick=10, NLtick=7}
 Arc1390:769→781 phase'c5:{EGtick=12, ELtick=12, NGtick=12, NLtick=9}
 Arc1391:769→782 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1392:770→783 red_truncation' red_truncation:{EGtick=13, ELtick=13, NGtick=13, NLtick=10, t=10}
 Arc1393:771→783 phase'c5:{EGtick=10, ELtick=10, NGtick=10, NLtick=7}
 Arc1409:781→792 phase'c5:{EGtick=11, ELtick=11, NGtick=11, NLtick=8}
 Arc1410:781→793 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1411:782→794 red_truncation' red_truncation:{EGtick=12, ELtick=12, NGtick=12, NLtick=9, t=10}
 Arc1412:783→794 phase'c5:{EGtick=8, ELtick=8, NGtick=8, NLtick=5}
 Arc1426:792→802 phase'c5:{EGtick=10, ELtick=10, NGtick=10, NLtick=7}
 Arc1427:792→803 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1428:793→804 red_truncation' red_truncation:{EGtick=11, ELtick=11, NGtick=11, NLtick=8, t=10}
 Arc1429:794→804 phase'c5:{EGtick=7, ELtick=7, NGtick=7, NLtick=4}
 Arc1442:802→812 phase'c5:{EGtick=9, ELtick=9, NGtick=9, NLtick=6}
 Arc1443:802→813 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1444:803→759 red_truncation' red_truncation:{EGtick=10, ELtick=10, NGtick=10, NLtick=7, t=10}
 Arc1445:804→814 phase'c5:{EGtick=6, ELtick=6, NGtick=6, NLtick=3}
 Arc1458:812→821 phase'c5:{EGtick=8, ELtick=8, NGtick=8, NLtick=5}
 Arc1459:812→822 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1460:813→771 red_truncation' red_truncation:{EGtick=9, ELtick=9, NGtick=9, NLtick=6, t=10}
 Arc1461:814→823 phase'c5:{EGtick=5, ELtick=5, NGtick=5, NLtick=2}
 Arc1473:821→830 phase'c5:{ELtick=7, EGtick=7, NGtick=7, NLtick=4}
 Arc1474:821→831 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1475:822→783 red_truncation' red_truncation:{EGtick=8, ELtick=8, NGtick=8, NLtick=5, t=10}
 Arc1488:830→839 phase'c5:{EGtick=6, ELtick=6, NGtick=6, NLtick=3}
 Arc1489:830→840 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1490:831→794 red_truncation' red_truncation:{EGtick=7, ELtick=7, NGtick=7, NLtick=4, t=10}
 Arc1505:840→804 red_truncation' red_truncation:{EGtick=6, ELtick=6, NGtick=6, NLtick=3, t=10}
 Arc1503:839→848 phase'c5:{EGtick=5, ELtick=5, NGtick=5, NLtick=2}
 Arc1504:839→849 red_truncation'Init2:{s="PR_RT", n=10}
 Arc1505:840→804 red_truncation' red_truncation:{EGtick=6, ELtick=6, NGtick=6, NLtick=3, t=10}
 Arc1521:849→814 red_truncation' red_truncation:{EGtick=5, ELtick=5, NGtick=5, NLtick=2, t=10}

References

1. Garyfalia, M.; Papageorgiou, M.; Diakaki, C.; Papamichail, I.; Dinopoulou, V. State-of-the-art and -practice review of public transport priority strategies. *IET Intell. Transp. Syst.* **2015**, *9*, 391–406.
2. Jeng, A.A.K.; Jan, R.H.; Chen, C.; Chang, T.L. Adaptive urban traffic signal control system with bus priority. In Proceedings of the IEEE Vehicular Technology Conference, Dresden, Germany, 2–5 June 2013.
3. Hassan, M.; Hawas, Y.E. A methodology for rearranging transit stops for enhancing transit users generalized travel time. *J. Traffic Transp. Eng. (English Edition)* **2017**, *4*, 14–30. [[CrossRef](#)]
4. Hu, J.; Park, B.; Parkany, A. Transit Signal Priority with Connected Vehicle Technology. *Transp. Res. Rec. J. Transp. Res. Board* **2014**, *2418*, 20–29. [[CrossRef](#)]
5. Tan, C.W.; Park, S.; Liu, H.; Xu, Q.; Lau, P. Prediction of transit vehicle arrival time for signal priority control: Algorithm and performance. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 688–696.

6. Lee, J.; Shalaby, A. Rule-based transit signal priority control method using a real-time transit travel time prediction model. *Can. J. Civ. Eng.* **2013**, *40*, 68–75. [[CrossRef](#)]
7. Gardner, K.; Souza, C.D.; Hounsell, N.; Shrestha, B.; Bretherton, D. *Review of Bus Priority at Traffic Signals Around the World, Final Report of Working Program Bus Committee*; UITP Working Group: Bruxelles, Belgium, 2009.
8. He, Q.; Head, K.L.; Ding, J. Multi-modal traffic signal control with priority, signal actuation and coordination. *Transp. Res. Part C Emerg. Technol.* **2014**, *46*, 65–82. [[CrossRef](#)]
9. Ekeila, W.; Sayed, T.; Esawey, M. Development of dynamic transit signal priority strategy. *Transp. Res. Rec. J. Transp. Res. Board* **2009**, *2111*, 1–9. [[CrossRef](#)]
10. Niittymäki, J.; Maenpää, M. The role of fuzzy logic public transport priority in traffic signal control. *Traffic Eng. Control* **2001**, *42*, 22–26.
11. Wahlstedt, J. Impacts of bus priority in coordinated traffic signals. *Procedia-Soc. Behav. Sci.* **2011**, *16*, 578–587. [[CrossRef](#)]
12. Wolput, B.; Tegenbos, R.; Skabardonis, A.; Tampere, C.M.J. Control strategies for transit priority: Comparing adaptive priority with full transit signal priority. In Proceedings of the Transportation Research Board Annual Meeting, Washington, DC, USA, 11–15 January 2015.
13. Ahmed, F.; Hawas, Y.E. An integrated real-time traffic signal system for transit signal priority, incident detection and congestion management. *Transp. Res. Part C Emerg. Technol.* **2015**, *60*, 52–76. [[CrossRef](#)]
14. Oliveira-Neto, F.; Loureiro, C.; Han, L. Active and passive bus priority strategies in mixed traffic arterials controlled by SCOOT adaptive signal system. *Transp. Res. Rec. J. Transp. Res. Board* **2009**, *2128*, 58–65. [[CrossRef](#)]
15. Murata, T. Petri nets: Properties, analysis and applications. *Proc. IEEE* **1989**, *77*, 541–580. [[CrossRef](#)]
16. Girault, C.; Valk, R. *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications*; Springer Science & Business Media: Berlin, Germany, 2013.
17. Wu, N.Q.; Zhou, M.C. 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.* **2012**, *9*, 446–454.
18. Wu, N.Q.; Zhou, M.C. Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation. *IEEE Trans. Autom. Sci. Eng.* **2012**, *9*, 203–209.
19. Wu, N.Q.; Chu, F.; Chu, C.B.; Zhou, M.C. Petri net modeling and cycle time analysis of dual-arm cluster tools with wafer revisiting. *IEEE Trans. Syst. Man Cybern. Syst.* **2013**, *43*, 196–207. [[CrossRef](#)]
20. Wu, N.Q.; Zhou, M.C.; Bai, L.P.; Li, Z.W. Short-term scheduling of crude oil operations in refinery with high fusion point oil and two transportation pipelines. *Enterp. Inf. Syst.* **2016**, *10*, 581–610. [[CrossRef](#)]
21. Chen, Y.F.; Li, Z.W.; Al-Ahmari, A.; Wu, N.Q.; Qu, T. Deadlock recovery for flexible manufacturing systems modeled with petri nets. *Inf. Sci.* **2017**, *381*, 290–303. [[CrossRef](#)]
22. Yang, F.J.; Wu, N.Q.; Qao, Y.; Zhou, M.C.; Li, Z.W. Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *47*, 502–516. [[CrossRef](#)]
23. Zhang, S.W.; Wu, N.Q.; Li, Z.W.; Qu, T.; Li, C.D. Petri net-based approach to short-term scheduling of crude oil operations with less tank requirement. *Inf. Sci.* **2017**, *417*, 247–261. [[CrossRef](#)]
24. Bai, L.P.; Wu, N.Q.; Li, Z.W.; Zhou, M.C. Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 1456–1467. [[CrossRef](#)]
25. Dicesare, F.; Kulp, P.; Gile, K.; List, G. The application of Petri nets to the modeling, analysis and control of intelligent urban traffic networks. In *Application and Theory of Petri Nets 1994, Volume 815, Ser. Lecture Notes in Computer Science*; Springer: New York, NY, USA, 1994; pp. 2–15.
26. Ng, K.M.; Reaz, M.B.I.; Ali, M.A.M. A review on the applications of Petri nets in modeling, analysis, and control of urban traffic. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 858–870. [[CrossRef](#)]
27. Tolba, C.; Lefebvre, D.; Thomas, P.; Moudni, A.E. Continuous and timed Petri nets for the macroscopic and microscopic traffic flow modelling. *Simul. Model. Pract. Theory* **2005**, *13*, 407–436. [[CrossRef](#)]
28. Dotoli, M.; Fanti, M.P.; Iacobellis, G. An urban traffic network model by first order hybrid Petri nets. In Proceedings of the IEEE International Conference on System Man and Cybernetic, Singapore, 12–15 October 2008.

29. Huang, Y.; Weng, Y.; Zhou, M. Modular design of urban traffic-light control systems based on synchronized timed Petri nets. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 530–539. [[CrossRef](#)]
30. Huang, Y.; Weng, Y.; Zhou, M. Design of Regulatory Traffic Light Control Systems with Synchronized Timed Petri Nets. *Asian J. Control* **2017**, *20*, 174–185. [[CrossRef](#)]
31. List, G.F.; Cetin, M. Modeling traffic signal control using Petri nets. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 177–187. [[CrossRef](#)]
32. Li, L.; Lin, W.H.; Liu, H. Traffic signal priority/preemption control with colored Petri nets. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems, Vienna, Austria, 16 September 2005.
33. Huang, Y.; Weng, Y.; Zhou, M. Design of traffic safety control systems for emergency vehicle preemption using timed Petri nets. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2113–2120. [[CrossRef](#)]
34. Jensen, K.; Kristensen, L.M.; Wells, L. Coloured Petri nets and CPN Tools for modelling and validation of concurrent systems. *Int. J. Softw. Tools Technol. Transf.* **2007**, *9*, 213–254. [[CrossRef](#)]
35. Jensen, K. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*; Springer Science & Business Media: Berlin, Germany, 2013.
36. PTV. *VISSIM 5.30 User Manual*; PTV: Melbourne, Australia, 2011.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).