Hindawi
Journal of Advanced Transportation
Volume 2019, Article ID 7481489, 14 pages
https://doi.org/10.1155/2019/7481489

Research Article

Distributed Cooperative Backpressure-Based Traffic Light Control Method

Shenxue Hao,[1,2] **Licai Yang**,[1] **Li Ding**,[2] **and Yajuan Guo**[1]

[1] *School of Control Science and Engineering, Shandong University, Jinan 250061, China*
[2] *School of Information and Engineering, Shandong Yingcai University, Jinan 250104, China*

Correspondence should be addressed to Licai Yang; yanglc@sdu.edu.cn

Received 6 November 2018; Accepted 18 February 2019; Published 5 March 2019

Academic Editor: Eneko Osaba

On the foundation of the original backpressure-based traffic light control algorithm, a distributed cooperative backpressure-based traffic light control method is proposed in this paper. The urban traffic network is modeled as a smart agent-controlled queuing network, in which the intersection agents exchange the queue length information and the selected activating light phase information of neighboring intersections through communications and determine the activating light phase at each time slot according to local traffic information. The improved phase pressure computation method considers the phase state of downstream intersections instead of only the queue length of the local intersections. Light phase switching coordination among adjacent intersections is achieved using the consensus-based bundle algorithm, in which the cooperative light phase switching problem is viewed as a task assignment issue among adjacent intersections. Simulation results illustrated that the proposed cooperative backpressure-based traffic light control method obtained better performance than the original backpressure-based and fixed-time traffic control methods.

1. Introduction

With the increase of vehicles travelling on roads in cities, congestion has become a major problem for urban traffic systems. Optimal traffic control strategy can improve traffic efficiency and balance the traffic load, which plays an important role in urban traffic systems. There are three types of conventional traffic control methods: fixed-time traffic control method, actuated traffic control, and adaptive traffic control. In the 1960s, TRANSYT [1] and MAXBAND [2] were proposed based on offline historical traffic data which selected optimal timing plans at different times of the day. However, these fixed-time traffic control methods cannot deal with the fluctuation of traffic demand due to predefined traffic parameters. Next, an actuated traffic control method was proposed to extend the light phase duration according to the detected traffic flow data in real time, using detectors installed at upstream segments. However, the detection of sparse traffic has a considerable influence on delay time [3] and it was applied mainly at isolated intersections [4]. Adaptive traffic control methods (SCOOT [5] and SCATS [6]) were proposed to adjust signal timing plans based on online traffic information for responding to real-time traffic demand. These methods can respond to traffic fluctuation using detector input, historical trends, and predictive models [7]. Although adaptive traffic control systems have been implied in real urban traffic networks, the cooperation of multiple intersections requires a control center, which has obstructed further development. In addition, the centralized signal timing scheme is always calculated according to the current traffic conditions, but is implied in the next cycle when the traffic situation may have already changed [4].

To fix these deficiencies, there is an agreement among many researchers that a distributed traffic control strategy is an ideal alternative. In a distributed traffic control system, signalized intersections are viewed as smart agents, and each agent determines the traffic control parameters according to current local information. For real-time dynamic traffic conditions, this kind of traffic control method can achieve better performance and adaptation [8, 9]. Many agent-based traffic control approaches have been studied in the past decades, and different theories and methods have

been utilized to optimize traffic control among intersections according to different traffic parameters. Researchers have utilized intelligent computation methods to obtain the cycle length and splits by minimizing traffic parameters, such as total travel time, etc. These types of methods avoid huge computations, such as the colony optimization approach [10] and reinforcement learning algorithm [11]. However, it is still difficult for these types of traffic control methods to achieve online traffic timing decision making, since the traffic signal timing problem usually is NP hard, it may take long time to find an optimal solution for simple transportation systems [12, 13]. Therefore, to simplify the traffic control problem, a better way of achieving distributed traffic system control with lower computation is needed. In 2012, Wongpiromsarn *et al.* first introduced a backpressure algorithm to solve traffic control problems and develop a traffic signal control strategy by viewing the traffic network as a queuing network [14]. The backpressure algorithm is usually used in wireless multihop networks as an optimal strategy for resource allocation. It has many features, including throughput optimality, achievable adaptive resource allocation, and simplicity [15].

Since the backpressure-based traffic signal control algorithm can maximize the throughput of the traffic network in a completely distributed manner, researchers have paid more attention to this algorithm and have obtained many achievements. In 2013, Varaiya presented a maximum pressure traffic control strategy based on the backpressure algorithm. At each intersection, the active phase is selected depending on the local queue length, mean turn ratios, and saturation rates [16]. Considering the influence of the routing rate of the queued vehicle on links, Gregoire *et al.* presented a backpressure-based traffic signal control algorithm with unknown routing rates and an estimated aggregated queue length. The vehicle routing information can be detected using detectors on dedicated lanes [17]. In their later research, a routing model of the traffic network was established with partial controllable vehicles to be used for pressure computation [18]. With the development of communication technology, more information could be obtained from a vehicular network and a multicommodity backpressure algorithm for traffic light control was proposed, where it is assumed that all vehicles' routes are known [19]. To further improve the efficiency of the traffic network, Taale *et al.* integrated route guidance technology with traffic signal control based on the backpressure algorithm [20], and Le *et al.* proposed a cyclic phase backpressure control policy with online estimation of turning fraction and measurement of queue size [21]. However, these backpressure-based traffic light control methods only determine the activating light phase according to the phase pressure computed using local information and neglect the possible coordination with adjacent intersections for phase pressure computation.

In this paper, we propose a cooperative backpressure-based traffic control method, in which the phase pressure is computed by considering the phase state of downstream intersections. Actually, the phase pressure is influenced by the queue length on the downstream segment. When the vehicles queued in front of the downstream intersections obtain the right of way, the corresponding phase of the downstream intersection is activated, and the current phase pressure will increase due to the decrease of queued vehicles on downstream segment. These phase pressure changes may affect the choice of the activating light phase. Therefore, we propose a modified phase pressure computation method, which considers the phase state of downstream intersections to achieve cooperative light phase switching among intersections. In addition, for the cooperative light phase switching decision among intersections, the traffic light switching problem is viewed as a task assignment issue. Moreover, the consensus-based bundle algorithm (CBBA), which is usually utilized for decentralized task selection for a multiagent system [22], is introduced to achieve coordination among intersections.

To summarize, there are two main innovations in this paper: (1) A modified phase pressure computation method is proposed considering the phase state of downstream intersections; this method is appropriate for cooperative light phase switching among intersections. (2) CBBA is introduced to solve the conflicts in cooperative light phase switching decision by viewing the cooperative traffic light switching as a cooperative task assignment problem.

The remainder of this paper is organized as follows: In Section 2, the urban traffic network is modeled as an agent-controlled queuing network. The phase pressure computation method is presented in Section 3. In Section 4, the CBBA based traffic light cooperative control algorithm is described. In Section 5, the stability of the proposed algorithm is analyzed. Simulations are carried out, and the results are discussed in Section 6. This paper is concluded in Section 7.

2. Traffic Network Modeling

The urban traffic network is modeled as an agent-controlled queuing network. Road segments are considered link nodes. There are 3 types of link nodes: ingress node, internal node, and exit node. Both ingress nodes and internal nodes have a downstream intersection. Vehicle flows entering a node generate vehicle queues when the corresponding light phase is red. The exit nodes, from which vehicles are leaving the traffic network, have no downstream intersection, and therefore do not generate queued vehicles. It assumes that vehicle flows entering internal nodes are coming completely from an upstream intersection.

The intersections controlled by Smart Traffic Light Control Agents (STLCA) are viewed as junctions that connect link nodes. Although it is unnecessary to explain the light phase of each intersection under backpressure-based traffic control method used in this paper, to describe the topology of vehicle flows in traffic network clearly, it assumes that there are 4 light phases at each intersection: north-south straight phase, north-south left-turn phase, west-east straight phase, and west-east left-turn phase, as shown in Figure 1. There are 3 dedicated lanes on each road segment, i.e., a left-turn lane, a straight lane, and a right-turn lane. It assumes that the vehicles entering a road segment will drive into the dedicated lane immediately and the right-turn vehicle flows are free up to the traffic light. Vehicle flow on the straight lane or left-turn lane is controlled by the corresponding light phases. For example, as shown in Figure 2, there are

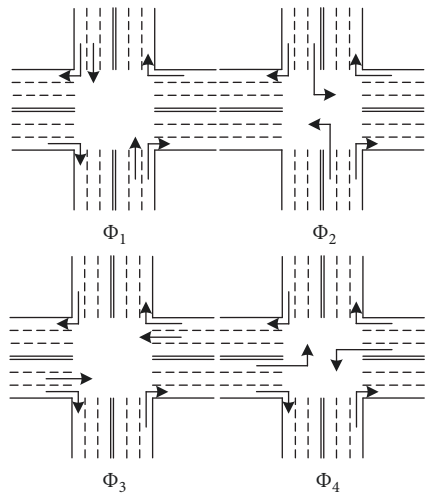

Figure 1: The light phases of one intersection.

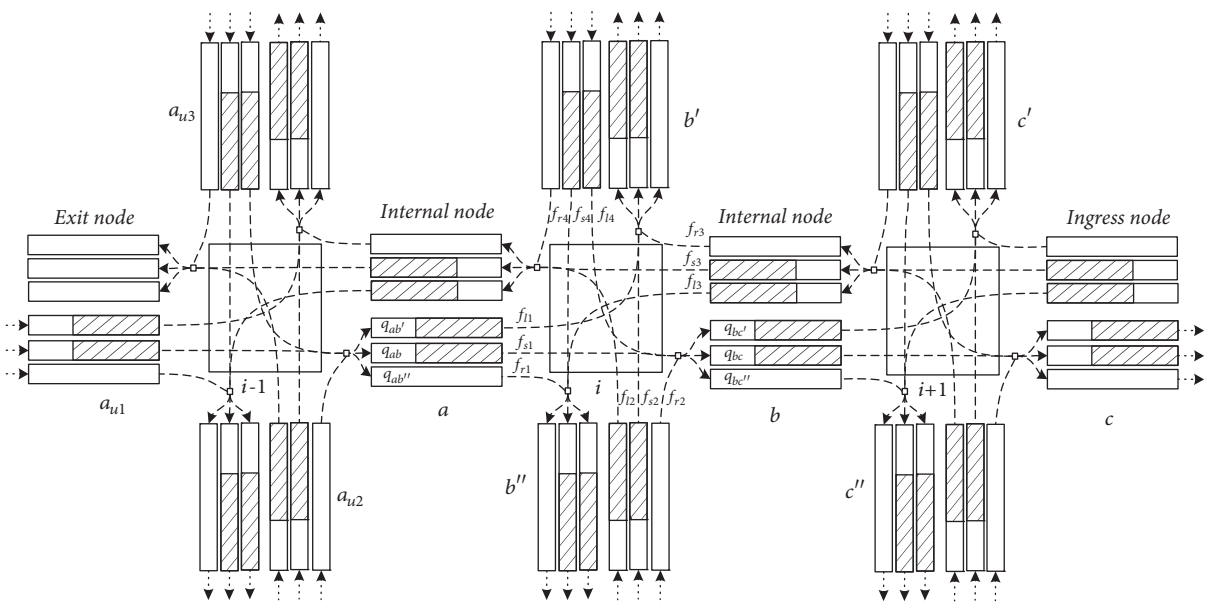

Figure 2: The topology of queuing traffic network.

12 vehicle flows passing through intersection i. The right-turn flows $\{f_{r1}, f_{r2}, f_{r3}, f_{r4}\}$ are not controlled by the traffic light of STLCA i, the straight flows $\{f_{s1}, f_{s3}\}$ and $\{f_{s2}, f_{s4}\}$ are controlled by west-east straight phase and north-south straight phase of intersection i, and the left-turn flows $\{f_{l1}, f_{l3}\}$, $\{f_{l2}, f_{l4}\}$ are controlled by west-east left-turn phase and north-south left-turn phase, respectively.

Time is slotted for the queuing network control. Let $A_a(t)$ denote the exogenous arrivals from upstream intersection, $D_{ab}(t)$ represent the number of vehicles moving from segment a to b, and $T_{ab}(t)$ is the turn ratio of exogenous arrivals, that is, there are $T_{ab}(t)A_a(t)$ vehicles add to queued vehicles $q_{ab}(t)$ during time slot t. Then, queue length q_{ab} on a waiting for moving to b can be computed using (1). Shown as Figure 2, the exogenous arrival $A_a(t)$ of a comes from the three vehicle flows $\{f_{l4}, f_{s1}, f_{r2}\}$ of intersection i-1.

$$q_{ab}(t+1) = q_{ab}(t) + T_{ab}A_a(t) - D_{ab}(t) \tag{1}$$

In the agent-controlled queuing network, STLCA i selects a light phase to activate with a given green time or extends the current activating light phase with a short green time according to the backpressure algorithm at each time slot t. In this paper, we only consider the cooperative light phase switching strategy. The activating phase extension strategy will be researched in the future work.

3. Pressure Computation Method of Light Phase

In the traffic network, vehicle flows on segment a, waiting to pass through the intersection, will generate traffic pressure to the downstream intersection. According to the original

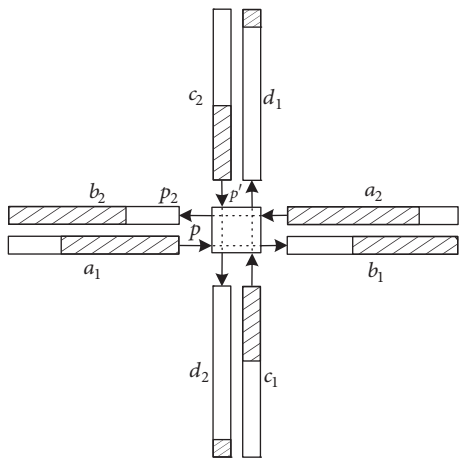

Figure 3: The case with similar queue length for downstream and upstream links.

backpressure algorithm, the pressure of each light phase $pressure_p$ is defined as the sum of the pressure associated with all vehicle flows controlled by light phase p, computed using (2) and (3). The pressure associated with vehicle flow $f_{ab} \in p$ is defined as the current flow rate, weighted by $w_{ab}(t)$ (the difference between q_a and q_b), where $f_{ab} \in p$ indicates that f_{ab} obtains the right of way when p is activated [14, 23]:

$$w_{ab}(t) = q_a(t) - q_b(t) \tag{2}$$

$$pressure_p = \sum_{f_{ab} \in p} \mu_{ab}(p)\, w_{ab}(t) \tag{3}$$

where $q_a(t)$ is the total number of vehicles waiting on segment a at the beginning of slot t and $\mu_{ab}(p)$ is the number of vehicles passing through intersection per unit time when p is activated. If p is not activated, then $\mu_{ab}(p) = 0$.

Only these vehicles queued on segment a, waiting for moving to node b, generate pressure to the light phase p, instead of all the vehicles on segment a. Therefore, in Varaiya's paper [17], the weight $w_{ab}(t)$ was modified as

$$w_{ab}(t) = q_{ab}(t) - \sum_{c} r_{bc} q_{bc}(t) \tag{4}$$

where r_{bc} denotes the proportion of vehicle flow leaving b and entering c and $\sum_c r_{bc} q_{bc}(t)$ indicates the average queue length of b.

According to the backpressure-based traffic control algorithm [15], STLCA i chooses a phase with maximum pressure to activate in the next time slot using (5) to achieve maximum throughput of a single intersection.

$$p* = \arg\max_{p \in P_i} \left(pressure_p(t)\right) \tag{5}$$

where P_i is the light phase set of intersection i and $p*$ is the selected phase to activate in the next time slot.

However, when the vehicle queue on the downstream segments of an intersection is longer than the vehicle queue on an upstream segment, or when the two queue lengths are close (as shown in Figure 3), according to the backpressure algorithm, phase p' will be activated instead of phase p in the next time slot. In this situation, the two pairs of queue lengths (q_{a1}, q_{b1}) and (q_{a2}, q_{b2}) both have a small difference, and the two pairs of (q_{c1}, q_{d1}) and (q_{c2}, q_{d2}) have a larger difference. This results in the pressure of phase p being smaller than phase p', and phase p' with a smaller volume is to be activated in the next time slot. It is obvious that in this circumstance the backpressure-based phase switching strategy described above cannot achieve an ideal effect.

Considering this situation, the cooperative traffic control strategy may be a good choice for some subareas of the traffic network. Based on this view, in this paper the phase pressure is calculated considering the phase state of downstream intersections to achieve a cooperative backpressure-based phase switching algorithm. Shown in Figure 4(a), for the straight phase $p^i{}_j$ of intersection i, there is $f_{ab} \in p^i{}_j$. For the downstream intersection i+1, there are $f_{bc'} \in p_k^{i+1}$ and $f_{bc''} \in p_{k+1}^{i+1}$. $p^i{}_j$ denotes the jth light phase of intersection i. $d_{bc'}$ and $d_{bc''}$ denote the number of vehicles departing from b to c' or c'', respectively. If $p^j{}_k$ or p_{k+1}^j is activated, then the queued vehicles on b will decrease, and the pressure of $p^i{}_j$ will increase. Then, we can conclude that the pressure of $p^i{}_j$ of intersection i is affected by the phase state of the downstream intersection i+1. A similar situation of the left-turn light phase of intersection i is shown in Figure 4(b). According to the above-stated analysis, there are two cases for the phase switching coordination of adjacent intersections. Case 1 occurs when intersection i is determining the activating phase of the next time slot, the corresponding downstream phase p_k^{i+1} or p_{k+1}^{i+1} is still activating. Case 2 occurs when the two intersections i and i+1 are determining the activating phase at the same time. For these two cases, the phase pressure computation method of $p^i{}_j$ should be modified as (6)-(9).

$$w_{ab}(t) = q_{ab}(t) - \sum_{c \in \{c', c''\}} r_{bc}\left(q_{bc}(t) - d_{bc}(t)\right) \tag{6}$$

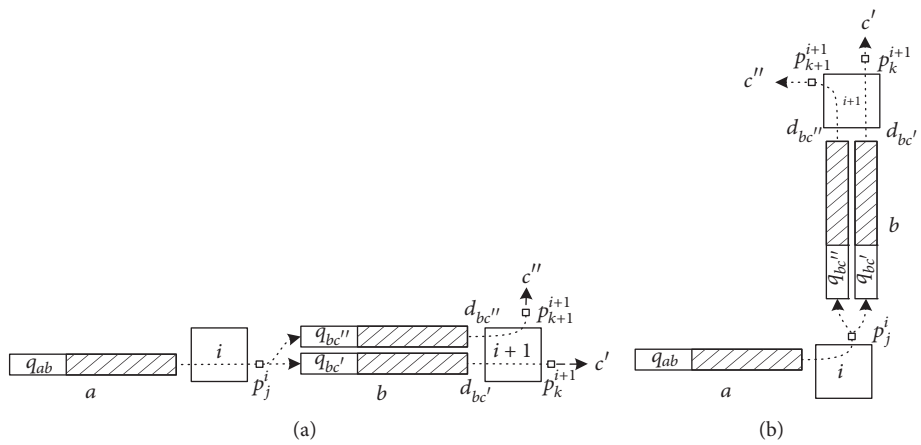

Figure 4: Pressure of straight phase and left-turn phase of intersection i affected by downstream intersection i+1. (a) Straight phase of intersection i affected by downstream light phase; (b) left-turn phase of intersection i affected by downstream light phase.

$$pressure_p = \sum_{f_{ab}\in p} \mu_{ab}(p)\left(q_{ab}(t) - \sum_{c\in\{c',c''\}} r_{bc}\left(q_{bc}(t) - d_{bc}(t)\right)\right) \quad (7)$$

where $d_{bc}(t)$ is the number of vehicles departing from b to c defined as (8), (9).

$$d_{bc}(t) = \begin{cases} s(p_k(t))\, q_{bc}(t) & \text{for } f_{bc} \in p_k(t) \text{ if } q_{bc}(t) < f_{\max} \\ s(p_k(t))\, f_{\max} & \text{for } f_{bc} \in p_k(t) \text{ if } q_{bc}(t) >= f_{\max} \end{cases} \quad (8)$$

where

$$s(p_k(t)) = \begin{cases} 1 & \text{if } p_k(t) \text{ is activated} \\ 0 & \text{if } p_k(t) \text{ is not activated} \end{cases} \quad (9)$$

where $f_{\max}$ is the maximum number of vehicles passing through the intersection when the downstream light phase p_k is activated.

Using the proposed method of phase pressure computation by considering the phase state of downstream intersections, the light phase is to be switched in cooperation mode. The light phase switching of adjacent intersections should be able to make a cooperative decision without conflicts. However, it is difficult to obtain the global optimal traffic control strategy among multiple intersections with a complex traffic state. CBBA, which is usually used for task assignment problems [22], is introduced to solve this problem in the next section.

4. Cooperative Light Phase Switching Decision Using CBBA

According to the idea of the backpressure traffic light control algorithm, the phase with maximum pressure will be selected to activate in the next time slot. However, the phase pressure computed using the proposed method is affected by the phase state of the downstream intersection. The selected phase $p^i{}_j$ of intersection i may be in conflict with the phase p_k^{i+1} of downstream intersection i+1. This is because the pressure of $p^i{}_j$ may be computed by assuming that phase p_{k+1}^{i+1} is activated. To solve the conflicts of phase switching among intersections, the CBBA is introduced in this section.

CBBA is a market-based distributed task assignment algorithm and has been shown to produce a conflict free solution [24, 25]. There are two stages in CBBA [22], i.e., the auction stage and the consensus stage. In the auction stage, each agent bids on a task and calculates a score based on the local information to decide whether it assigns a new task to the task bundle or not. The task bundle of agent i is defined as the possible task set with task score. If agent i does not win any tasks in the auction stage, the task bundle of agent i is empty. In the consensus stage, agents exchange the task information through communication. If an agent is outbid for a task, then it is released from the task bundle. For a task, the agent with the highest score is the winner. Based on the idea of CBBA, in the next section, the cooperative phase switching problem in urban traffic network is modeled as a task assignment problem.

4.1. Cooperative Light Phase Switching Modeled as Task Assignment. There are four light phases for the light controller of each intersection, which are managed by a smart agent, STLCA. The task of STLCA is to determine which phase is to be activated in the next time slot. For each intersection, there are two vehicle flows with opposite directions, controlled by each light phase. For each vehicle flow, there is one downstream intersection at most. If there is one downstream intersection for one vehicle flow, then there are three cooperative phase switching possibilities. Shown in Figure 5, vehicle flows f_1 and f_2 are controlled by phase $p^i{}_j$. When $p^i{}_j$ is activated, the vehicles waiting on segments a_1 and a_2 obtain the right of way to pass through intersection i. The pressure of $p^i{}_j$ is consists of two parts: the pressure

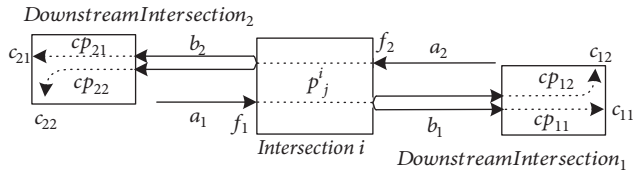

Figure 5: Coordination relationships of light phases among adjacent intersections.

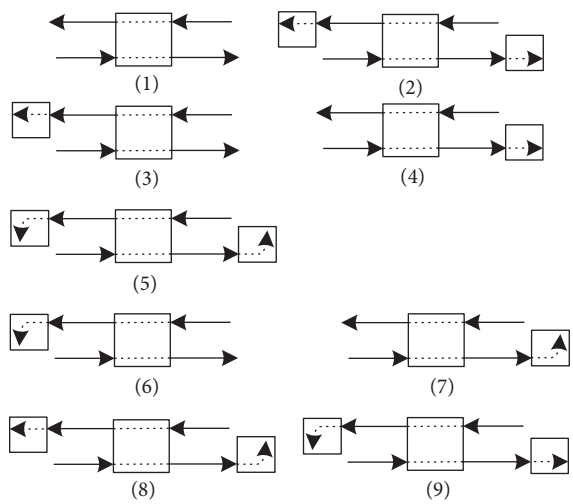

Figure 6: Nine cooperative phase switching possibilities of one light phase.

from f_1 and the pressure from f_2. The pressure generated from vehicle flow f_1 is associated with the cooperative phases cp_{11} and cp_{12}. If cp_{11} or cp_{12} are activated, vehicles on segment b_1 can drive into segments c_{11} or c_{12}. In this case, the number of vehicles departing from segment b_1 should be considered. If cp_{11} and cp_{12} are not to be activated or the $DownstreamIntersection_1$ does not exist, the number of vehicles departing from segment b_1 is zero. Similarly, the computation of pressure generated from vehicle flow f_2 should consider the downstream phase states cp_{21} or cp_{22}. Therefore, when considering the downstream phase state, there are 9 possible cooperative phase switching strategies for phase $p^i{}_j$, as shown in Figure 6. For each intersection, there are at most 36 possible cooperative phase switching.

In order to solve the cooperative phase switching problem using CBBA, these possible cooperative phase switching strategies are viewed as the task bundle of STLCA i. Each possible phase switching strategy is viewed as a possible task of intersection i. The task score is represented by the phase pressure calculated using (6)-(9) according to the downstream phase state.

4.2. Cooperative Backpressure-Based Light Phase Switching Algorithm. In this section, the CBBA is utilized to solve the cooperative phase switching problem among intersections. According to the CCBA, the cooperative backpressure-based light phase switching algorithm consists of three steps as follows.

(*a*) Task Bundle Construction. STLCA i constructs the task bundle according to its topology in the traffic network. The possible switching strategies of intersection i are considered as the possible tasks of STLCA i. The amount of possible switching strategies for a particular intersection is fixed, since the topology of an intersection in urban traffic network is fixed. The task bundle of each intersection is constructed before phase switching decision making can improve the algorithm performance. Shown in Table 1, there are 9 possible phase switching strategies for one phase of intersection i. Therefore, there are 36 possible switching strategies in one intersection at most.

(*b*) Phase Pressure Computation. STLCA i collects the queue length information of each segment and broadcasts the information to the adjacent intersections. The pressure of each strategy is calculated using (6)-(9) according to the local queue length information and queue length information from adjacent intersections. The possible strategies of intersection i are ordered by the phase pressure, and all the possible strategies are set to be available at beginning. After the pressure computation, for each intersection, the strategy with maximum pressure is selected to be the candidate strategy, and the responding phase to be the candidate light phase for activating in the next time slot.

(*c*) Activating Phase Decision and Conflicts Resolution. In this step, STLCA i determines the candidate phase as the activating phase of intersection i at time t+1 directly if the candidate strategy need not coordinate with the neighboring downstream intersections. These intersections whose light phase has been determined are called activating-phase-determined intersections. If STLCA i has determined the activating phase, STLCA i broadcasts this information

```
If SelectedStrategy.DownstreamIntersection_1 = 0 And SelectedStrategy.DownstreamIntersection_2 = 0 Then
   Intersection_i.DeterminedFlag = True
   Intersection_i.SelectedPhase = SelectedStrategy.phase
   Let the selected phase be the activating phase in the next time slot.
   Broadcast the decision of Intersection_i to the adjacent intersections.
End If
If Intersection_i.DeterminedFlag = False Then
   If STLCA_i receives the decisions from adjacent intersections Then
      For each AdjacentIntersection of Intersection_i
         If AdjacentIntersectionj.DetermineFlag = True Then
            Release the strategies of Intersection_i conflicting with AdjacentIntersection_j.SelectedPhase
         End If
      Next
      Reselect the Intersection_i.SelectedStrategy with maximum pressure from the available strategies
      Broadcast Intersection_i.SelectedStrategy to the adjacent intersections.
   End If
   Elect the maximum pressure strategy among the available strategies of all intersections using distributed election algorithm.
   If Intersection_i.SelectedStrategy is the elected strategy with maximum pressure Then
      Intersection_i. DeterminedFlag = True
      Intersection_i. SelectedPhase = SelectedStrategy.phase
      Let the selected phase be the activating phase in the next time slot.
      Broadcast the decision of Intersection_i to the adjacent intersections.
   End If
End If
```

Algorithm 1: The algorithm of intersection i for activating phase decision and conflicts resolution.

Table 1: The possible cooperative phase switching strategies of $p^i{}_j$.

Strategy Number	*Light phase*	$DownstreamIntersection_1$	$CooperativePhase_1$	$DownstreamIntersection_2$	$CooperativePhase_2$
1	$p^i{}_j$	0	×	0	×
2	$p^i{}_j$	1	cp_{11}	0	×
3	$p^i{}_j$	1	cp_{12}	0	×
4	$p^i{}_j$	0	×	1	cp_{21}
5	$p^i{}_j$	0	×	1	cp_{22}
6	$p^i{}_j$	1	cp_{11}	1	cp_{21}
7	$p^i{}_j$	1	cp_{11}	1	cp_{22}
8	$p^i{}_j$	1	cp_{12}	1	cp_{21}
9	$p^i{}_j$	1	cp_{12}	1	cp_{22}

Note: × indicates that the cooperative phase is not to be activated. 0 indicates that the downstream intersection does not exist or does not coordinate with $p^i{}_j$. cp_{11}, cp_{12}, cp_{21}, and cp_{22} are the cooperative phases of *downstream intersection*$_1$ and *downstream intersection*$_2$.

to its adjacent intersections. Then, the adjacent STLCAs of intersection i release the strategies conflict with intersection i. For example, if intersection i has determined the activating light phase to be west-east straight phase, the strategies of neighboring intersections (if the neighboring intersections are not activating-phase-determined intersections) which are conflict with the activating light phase of intersection i, will be released, that is, the invalid flag of these strategies are set to be true. For these activating-phase-undermined intersections, STLCAs elect a strategy with maximum pressure in distributed mode by communication. The elected intersection determines the activating phase and broadcasts it to adjacent intersections. Step (a) and (b) are repeated until all STLCAs have determined the activating phase. A detailed description of the algorithm is given in Algorithm 1.

Based on the three steps, the cooperative backpressure-based traffic control method (simplified as CBP method) can obtain a conflict free phase switching strategy with maximum phase pressure. The backpressure-based traffic light control method (simplified as BP method) only considers the queue length of the current intersection and neglects the decrease of queued vehicles on the downstream segment when the corresponding downstream phase is activated. The CBP method fixes this deficiency and considers the phase state of downstream intersections to achieve coordination among intersections. Furthermore, all the switching

possibilities based on BP method (that is, the strategies with $DownstreamIntersection_1 = 0$ and $DownstreamIntersection_1 = 0$) are contained in the task bundle of each intersection in CBP method. In other words, the CBP method considering downstream phase state can obtain equal or greater traffic performance compared to the BP method. To illustrate the feasibility of the proposed CBP method, the stability is discussed in the next section.

5. Stability Analysis

In this section, the stability of the proposed CBP method is analyzed. As described in references [26] and [14], for a network with queue vector $\mathbf{U} = \{U_1, U_2, \ldots, U_N\}$, a sufficient condition for stability can be provided using Lyapunov drift, which is given as below.

Lemma 1. *Suppose* $\mathbf{E}\{U_i(t)\} < \infty$ *for all* $i \in \{1, 2, \ldots, N\}$, *and there exist constants* $B > 0$ *and* $\varepsilon > 0$ *which satisfies*

$$\mathbf{E}\{L(\mathbf{U}(t+1) - L(\mathbf{U}(t)) \mid \mathbf{U}(t)\} \leq B - \varepsilon \sum_{i=1}^{N} U_i(t) \tag{10}$$

then the network is stable, where the Lyapunov function is defined as

$$L(U) = \sum_{i=1}^{N} U_i^2 \tag{11}$$

To describe simplicity, define the function $V_{ab}^{in}(p(t))$ and $V_{ab}^{out}(p(t))$ and denote the vehicles entering $q_{ab}(t)$ and vehicles departing from $q_{ab}(t)$ under the current light phase switching strategy $P(t)$ during slot t.

$$V_{ab}^{out}(P(t)) = \sum_{f_{ab} \in p_i(t)} \mu_{ab}(p_i(t)) \tag{12}$$

where $p_i(t)$ is the activated light phase of intersection i, and vehicle flow f_{ab} is controlled by $p_i(t)$; b is the downstream node of a.

$$V_{ab}^{in}(P(t)) = \sum_{f_{ca} \in p_{i-1}(t)} \mu_{ca}(p_{i-1}(t)) \tag{13}$$

where $p_{i\text{-}1}(t)$ is the activated light phase of intersection i-1 and vehicle flow f_{ca} is controlled by $p_{i\text{-}1}(t)$; c is the upstream node of a.

Then, (1) is rewritten as

$$q_{ab}(t+1) = q_{ab}(t) + V_{ab}^{in}(P(t)) - V_{ab}^{out}(P(t)) \tag{14}$$

According to the Lyapunov function and (14), $V_{ab}^{in}(P(t))$ and $V_{ab}^{out}(P(t))$ are simplified as V_{ab}^{in} and V_{ab}^{out}, we obtain

$$\begin{aligned} &L(U(t+1)) - L(U(t)) \\ &\quad = \sum_{a,b} \left(\left(V_{ab}^{in}\right)^2 + \left(V_{ab}^{out}\right)^2 \right) \\ &\qquad - 2\sum_{a,b} \left(q_{ab}(t) \left[V_{ab}^{out} - V_{ab}^{in} \right] \right) - 2\sum_{a,b} V_{ab}^{out} V_{ab}^{in} \\ &\quad = \sum_{a,b} \left(\left(V_{ab}^{in}\right)^2 + \left(V_{ab}^{out}\right)^2 \right) \\ &\qquad - 2\left(\sum_{a,b} \left(q_{ab}(t) \left[V_{ab}^{out} - V_{ab}^{in} \right] \right) + \sum_{a,b} V_{ab}^{out} V_{ab}^{in} \right) \end{aligned} \tag{15}$$

For queue network in this paper, it assumes that the arrival vehicles of ingress node are all come from the upstream nodes. For example, shown in Figure 7, there are two vehicle queues $\{q_{bc1}, q_{bc2}\}$ on node b; the arrival vehicle V_{bc1}^{in} is computed using $V_{bc1}^{in} = r_{bc1} V_{ab}^{out}$, where r_{bc1} is the proportion of vehicles entering q_{bc1} from q_{ab} when the corresponding light phase is activated. Similarly, there is $V_{bc2}^{in} = r_{bc2} V_{ab}^{out}$. Furthermore, for the node b, under a given light phase strategy $P(t)$, one of V_{bc1}^{out} and V_{bc2}^{out} is zero at least.

Based on these properties of the traffic network, the right term of (15) is expanded as

$$\begin{aligned} &\sum_{a,b} \left(q_{ab}(t) \left[V_{ab}^{out} - V_{ab}^{in} \right] \right) + \sum_{a,b} V_{ab}^{out} V_{ab}^{in} \\ &\quad = \left(q_{ab}(t) V_{ab}^{out} - q_{ab}(t) V_{ab}^{in} \right) \\ &\qquad + \left(q_{bc1}(t) V_{bc1}^{out} - q_{bc1}(t) V_{bc1}^{in} \right) \\ &\qquad + \left(q_{bc2}(t) V_{bc2}^{out} - q_{bc21}(t) V_{bc2}^{in} \right) + \ldots + V_{ab}^{in} V_{ab}^{out} \\ &\qquad + V_{bc1}^{in} V_{bc1}^{out} + V_{bc2}^{in} V_{bc2}^{out} + \ldots \\ &\quad = q_{ab}(t) V_{ab}^{out} - q_{bc1}(t) r_{bc1} V_{ab}^{out} - q_{bc2}(t) r_{bc2} V_{ab}^{out} \\ &\qquad - q_{ab}(t) V_{ab}^{in} + \ldots \\ &\qquad + \left(V_{ab}^{in} V_{ab}^{out} + r_{bc1} V_{ab}^{out} V_{bc1}^{out} + r_{bc2} V_{ab}^{out} V_{bc2}^{out} + \ldots \right) \\ &\quad = q_{ab}(t) V_{ab}^{out} - q_{bc1}(t) r_{bc1} V_{ab}^{out} - q_{bc2}(t) r_{bc2} V_{ab}^{out} \\ &\qquad + r_{bc1} V_{ab}^{out} V_{bc1}^{out} + r_{bc2} V_{ab}^{out} V_{bc2}^{out} \\ &\qquad + \left(\underline{-q_{ab}(t) V_{ab}^{in} + V_{ab}^{in} V_{ab}^{out}} \right) + \ldots \end{aligned} \tag{16}$$

For the term underline-marked in the above equation, $-q_{ab}(t)V_{ab}^{in}+V_{ab}^{in}V_{ab}^{out}$, a is an ingress node. It assumes that node x is a virtual node with q_x=0, V_x^{out} is the vehicle input on node a, and r_{ab} is the proportion of V_x^{out} for vehicle entering q_{ab} during time slot t. Then, this term can be rewritten as follows:

$$\begin{aligned} &-q_{ab}(t) V_{ab}^{in} + V_{ab}^{in} V_{ab}^{out} \\ &\quad = q_x(t) V_x^{out} - q_{ab}(t) r_{ab} V_x^{out} + r_{ab} V_x^{out} V_{ab}^{out} \end{aligned} \tag{17}$$

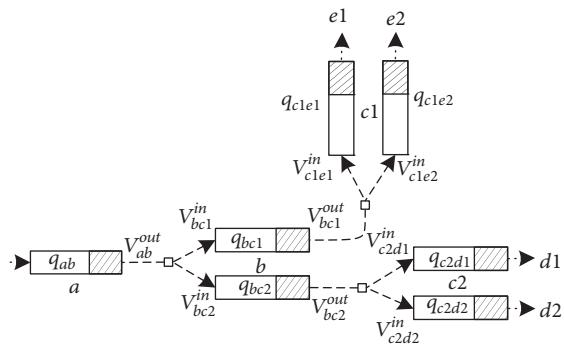

Figure 7: The relationship of arrival and departing vehicle flow in traffic network.

Similarly, the queues in the exit nodes also can be expressed as the similar forms. Therefore, the right term of (15) can be rewritten as

$$\begin{aligned}
&\sum_{a,b}\left(q_{ab}(t)\left[V^{out}_{ab}-V^{in}_{ab}\right]\right)+\sum_{a,b}V^{out}_{ab}V^{in}_{ab}=\sum_{a,b}q_{ab}(t)V^{out}_{ab}\\
&\quad-q_{bc1}(t)r_{bc1}V^{out}_{ab}-q_{bc2}(t)r_{bc2}V^{out}_{ab}+r_{bc1}V^{out}_{ab}V^{out}_{bc1}\\
&\quad+r_{bc2}V^{out}_{ab}V^{out}_{bc2}=\sum_{a,b}\left(q_{ab}(t)-q_{bc1}(t)r_{bc1}-q_{bc2}(t)\right.\\
&\quad\left.\cdot r_{bc2}+r_{bc1}V^{out}_{bc1}+r_{bc2}V^{out}_{bc2}\right)V^{out}_{ab}=\sum_{a,b}\left(q_{ab}(t)\right.\\
&\quad-\left(q_{bc1}(t)r_{bc1}+q_{bc2}(t)r_{bc2}-r_{bc1}V^{out}_{bc1}\right.\\
&\quad\left.\left.-r_{bc2}V^{out}_{bc2}\right)\right)V^{out}_{ab}=\sum_{a,b}\left(q_{ab}(t)\right.\\
&\quad\left.-\sum_{c}\left(q_{bc}(t)r_{bc}-r_{bc}V^{out}_{bc}\right)\right)V^{out}_{ab}=\sum_{a,b}\left(q_{ab}(t)\right.\\
&\quad\left.-\sum_{c}r_{bc}\left(q_{bc}(t)-V^{out}_{bc}\right)\right)V^{out}_{ab}
\end{aligned}\tag{18}$$

Then, $L(U(t+1))$-$L(U(t))$ can be expressed as

$$\begin{aligned}
&L(U(t+1))-L(U(t))\\
&\quad=\sum_{a,b}\left(\left(V^{in}_{ab}\right)^2+\left(V^{out}_{ab}\right)^2\right)\\
&\qquad-2\sum_{a,b}\left(q_{ab}(t)-\sum_{c}r_{bc}\left(q_{bc}(t)-V^{out}_{bc}\right)\right)V^{out}_{ab}
\end{aligned}\tag{19}$$

Let $B=\sum_{a,b}((\sup(V^{in}_{ab}))^2+(\sup(V^{out}_{ab}))^2)$, then there is

$$\begin{aligned}
&L(U(t+1))-L(U(t))\\
&\quad\le B-2\sum_{a,b}\left(q_{ab}(t)-\sum_{c}r_{bc}\left(q_{bc}(t)-V^{out}_{bc}\right)\right)V^{out}_{ab}
\end{aligned}\tag{20}$$

In the above equation, V^{out}_{bc} is the departing vehicles from q_{bc}, and if the corresponding downstream light phase of node a is not activated, V^{out}_{bc} will be zero. Here V^{out}_{bc} is same to d_{bc} in (6). Inject (6), (12) into (17); we obtain

$$\begin{aligned}
&L(U(t+1))-L(U(t))\\
&\quad\le B-2\sum_{a,b}\sum_{f_{ab}\in p(t)}\mu_{ab}(p(t))\,w_{ab}(t)
\end{aligned}\tag{21}$$

Since the queuing network modeled in this paper is similar to the model in reference [14, 16], there are similar properties:

$$\begin{aligned}
&\sum_{a,b}q_{ab}(t)\left[V^{out}_{ab}(P(t))-V^{in}_{ab}(P(t))\right]\\
&\quad=\sum_{a,b}\sum_{f_{ab}\in p(t)}\mu_{ab}(p(t))\left[q_{ab}(t)-\sum_{c}r_{bc}q_{bc}(t)\right]
\end{aligned}\tag{22}$$

From (22), the following equation can be deduced:

$$\begin{aligned}
&\sum_{a,b}q_{ab}(t)\left[V^{out}_{ab}(P(t))-V^{in}_{ab}(P(t))\right]\\
&\quad=\sum_{a,b}\sum_{f_{ab}\in p(t)}\mu_{ab}(p(t))\left[q_{ab}(t)-\sum_{c}r_{bc}q_{bc}(t)\right]\\
&\quad\le\sum_{a,b}\sum_{f_{ab}\in p(t)}\mu_{ab}(p(t))\\
&\quad\cdot\left[q_{ab}(t)-\sum_{c}r_{bc}\left(q_{bc}(t)-d_{bc}(t)\right)\right]\\
&\quad\le\sum_{a,b}\sum_{f_{ab}\in p(t)}\mu_{ab}(p(t))\,w_{ab}(t)
\end{aligned}\tag{23}$$

It means that, under the same given light phase switching strategy $P(t)$, if the downstream phase is coordinated with the phase of current intersection, it achieves equal or better throughput. If $d_{bc}(t)$ is zero, the CBP method is degenerated to the BP method. When the downstream light phase is coordinated with the current light phase, it achieves better throughput.

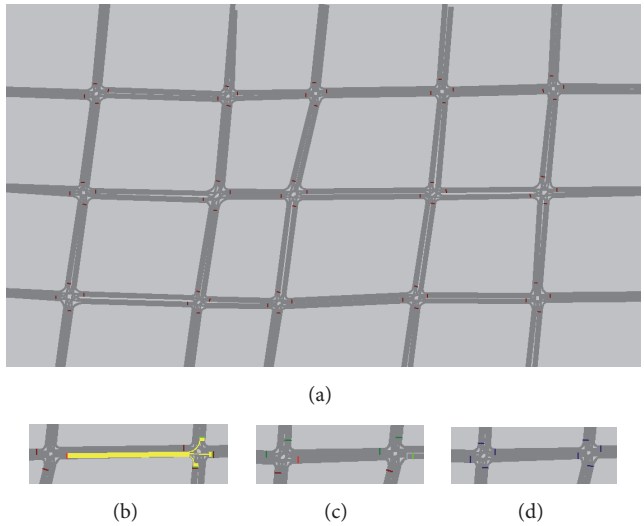

Figure 8: Topology and settings of simulation traffic network. (a) The topology of simulation traffic network; (b) route decisions setting; (c) travel time sections setting; (d) queue Counters setting.

According to (21) and (23), we obtain

$$\begin{aligned} & L(U(t+1)) - L(U(t)) \\ & \leq B - 2\sum_{a,b}\sum_{f_{ab}\in p(t)} \mu_{ab}(p(t))\, w_{ab}(t) \\ & \leq B - 2\sum_{a,b} q_{ab}(t)\left[V_{ab}^{out}(p) - V_{ab}^{in}(p)\right] \end{aligned} \tag{24}$$

In additional, in [14], it has been proved that, for the admissible arrival rate $\boldsymbol{\lambda}$ in capacity region Λ, and $\varepsilon > 0$, there is

$$\mathbf{E}\left\{V_{ab}^{out}(P(t)) - V_{ab}^{in}(P(t)) \mid U(t)\right\} = \lambda_{ab} + \varepsilon \tag{25}$$

Then, we obtain

$$\mathbf{E}\{L(\mathbf{U}(t+1)) - L(\mathbf{U}(t)) \mid \mathbf{U}(t)\} \leq B - 2\varepsilon\sum_{ab} q_{ab}(t) \tag{26}$$

According to *Lemma* 1, the network controlled by the proposed CBP method is stable under the admissible arrival rate $\boldsymbol{\lambda}$ in capacity region Λ. To illustrate the effectiveness of the proposed CBP method, simulations are conducted, and a description is given in the following section.

6. Simulation and Discussion

Three traffic light control methods are implemented, including fixed-time control, BP control, and CBP control. The BP method implemented in simulations is proposed by Varaiya [16]. Although there are several improved versions of backpressure-based traffic control methods, these methods do not consider the coordination of light phase switching among neighboring intersections. The CBP control method considering the downstream light phase state is proposed on the foundation of basic backpressure-based traffic control method proposed by Varaiya. Therefore, in this section we compare the three traffic control methods. In the further research work, based on these improved versions the coordination, should be considered.

The simulation traffic network consists of 15 intersections and 76 links constructed in Vissim. The network includes 16 ingress links and 16 exit links, as shown in Figure 8(a). The vehicle input of ingress links is 800veh/h. There are 15 signal controllers and 4 light phases for each controller, i.e., north-south straight, north-south left-turn, west-east straight, and west-east left-turn. There are 3 lanes on each link. The left-turn vehicle flow drives on Lane 3 and the straight vehicle flow drives on Lane 2. The two vehicle flows are controlled by a traffic light. The right-turn vehicle flow drives on Lane 1 and is free of the traffic light. To obtain the traffic parameters during simulation, 60 queue counters, 60 travel time sections, and 60 routing decisions are laid on the simulation traffic network, as shown in Figures 8(b), 8(c), and 8(d). During the simulation, the green light time of each light phase is assumed to be 21s, which is the pedestrian clearance time (the road width is about 21m, and the pedestrians speed is about 1m/s) [4]. The yellow light time is assumed to be 3s.

The three traffic light control methods are implemented in Visual Studio 2010. The simulation programs communicate with Vissim through the Vissim COM programming interface to obtain the traffic parameters and decide the traffic light signal at each time slot. The simulation runs for 7,200s, and the traffic network performance is evaluated from 1,000s to 7,200s using Vissim. This is because in the first 1,000s of simulation time there are not enough vehicles entering the simulation traffic network.

The simulation results of traffic network performance are given in Table 2. The results show that under similar traffic volume condition, the fixed-time control algorithm results in a higher delay time and travel time due to the lower

Table 2: Traffic network performance comparison.

Parameters	Fixed-Time Method	BP Method	CBP Method
Total travel time (h)	1494.11	1440.434	1332.647
Total delay time (h)	824.652	761.403	648.577
Number of Stops	120850	70359	68554
Total stopped delay (h)	593.744	587.938	482.059
Average delay time (s)	133.379	123.199	105.175
Average number of stops	5.43	3.162	3.088
Average stopped delay (s)	96.032	95.131	78.172
Average speed (km/h)	23.549	24.761	26.965
Number of vehicles in network	938	880	806
Number of vehicles left network	21320	21369	21394
Total Number of vehicles	22258	22249	22200

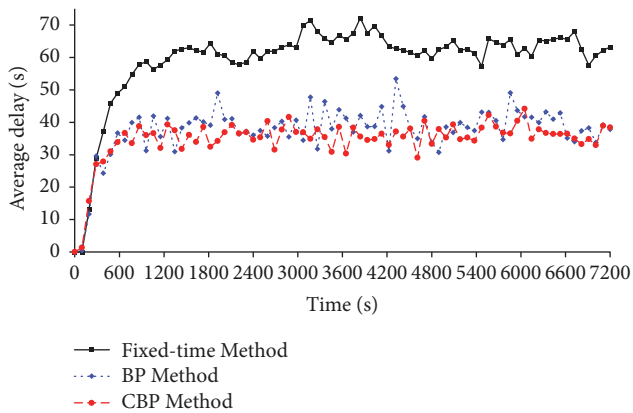

Figure 9: Average vehicle delay during simulation.

average speed. Vehicles stopping in front of intersections have to wait for the next cycle to get the right of way to pass through the intersection, which results in an increase in the vehicle delay time. Although the BP method can switch light signal according to the pressure of each light phase, it cannot deal well with the situation described in Figure 3. This kind of situation results in an increase of vehicle delay, because the vehicles on a_1 and a_2 cannot be released in time. The CBP method solves this problem well. From Table 2, we can conclude that the CBP control can obtain better traffic network performance than the other two methods.

To further illustrate the effectiveness of the CBP method, the average delay, average stop delay, average queue length, and the maximum queue length during the simulation are extracted from the Vissim evaluation files and listed in Figures 9, 10, 11, and 12.

Figure 9 displays the average delay of the three methods during simulation time. The delay time of fixed-time method is larger than the other two methods since each light phase is activated in same time interval in the fixed-time control, and the segments with longer queue length cannot get the right of way in priority. The BP and CBP methods have a close average vehicle delay, because these two methods have similar light phase switching strategy by selecting the light phase with the maximum pressure for activation. The CBP method considers the phase state of downstream intersections and achieved coordination among intersections. It can effectively speed up the release of queued vehicles. Therefore, the CBP method obtains a smaller average delay during simulation.

For the average stop delay, shown in Figure 10, the CBP method shows a smaller average stop delay during simulation than the other two methods. By considering the phase state of downstream intersections, the CBP method achieves coordination among intersections in a distributed pattern, which releases more vehicles due to the cooperative light phase switching among adjacent intersections.

During the simulation, there is an average of three queue lengths increase in the first 1,000s with the increase of simulation time, as shown in Figure 11. Under the same vehicle input, the average queue length of the CBP method generates a smaller average queue length in the traffic network. Figure 11 illustrates that the cooperative backpressure-based traffic light control method can speed up the queued vehicle releasing.

From Figure 12, according to the curve of the fixed-time traffic light control method, queue lengths of some segments in the traffic network reaches the maximum value at about 2,600s. The queue length of the BP method reaches the maximum value at about 6,600s. The queue length of the CBP method does not reach the maximum value in the

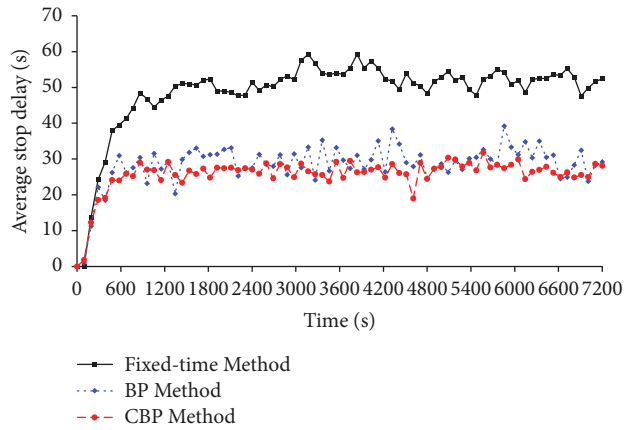

Figure 10: Average stop delay during simulation.

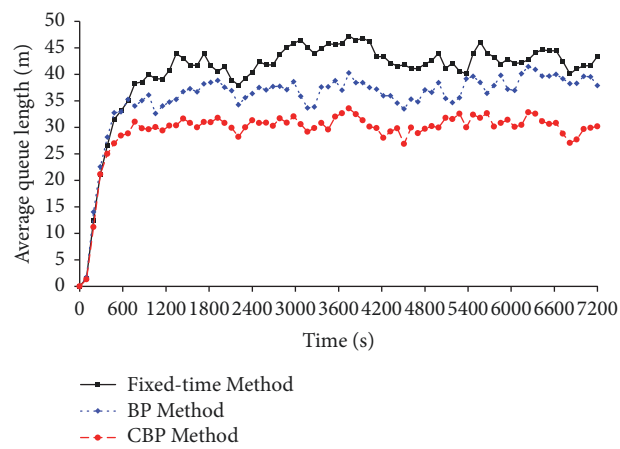

Figure 11: Average queue length during simulation.

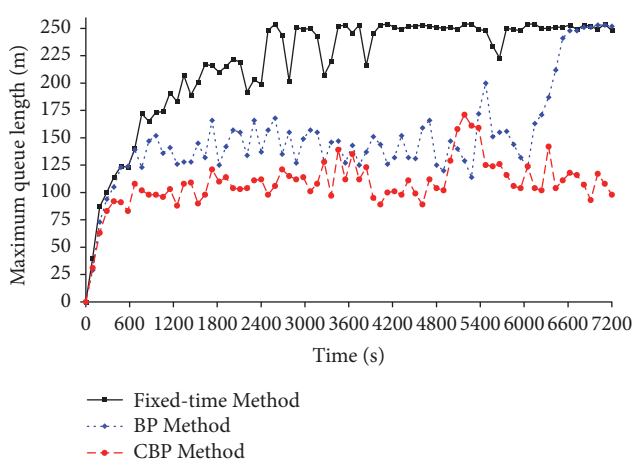

Figure 12: Maximum queue length during simulation.

entire simulation time. This simulation result indicates that the fixed-time control method is more likely to lead to queue spillover in the segments whose queue length reaches the maximum value. The queue spillover of some segments may result in traffic congestion propagation or a traffic collapse. This is a seriously negative effect to urban traffic. Although the time of the queue length for reaching the maximum value of the BP method is later than the fixed-time control method, it still reaches the maximum queue length after all. As shown in Figure 12, the queue lengths of all the segments

do not reach the maximum value during the CBP method simulation. This result indicates that the CBP method can obtain better performance in the traffic network.

Based on the above discussion, the proposed traffic control method performs better than the other two methods in the simulations. However, the CBP method requires communication among intersections due to the distributed phase switching decision. This results in an increase of the communication cost. With the development of an intelligent transport system, the communication problem in the proposed method can be solved completely. Therefore, this proposed cooperative traffic light control method may be an useful attempt for improving the efficiency of the urban traffic system.

7. Conclusion

In this paper, a cooperative backpressure-based traffic light control method is proposed to improve the efficiency of urban traffic network. The traffic network is modeled as a queuing network, in which the light phases of each intersection are controlled by STLCA. The light phase pressure is computed using the proposed light phase pressure computation method considering the phase state of downstream intersections, which is the basis for cooperative light phase switching decision among intersections. In the cooperative light phase switching decision, the CBBA is introduced to solve the conflicts in the switching strategies of adjacent intersections in a distributed mode. Simulations show that the proposed method obtains better performance than the original backpressure-based traffic light control method and fixed-time traffic light control method.

However, there are further works to be done in the future. Firstly, the pressure computation should be improved to consider other factors such as the capacity of the downstream segment, the priority of queued vehicles, and the number of passengers in queued vehicles. These factors may influence the light phase switching decision in different scenarios. Secondly, for simplicity, the cooperative backpressure-based traffic light control method only considers situations of synchronous traffic light switching. In reality, traffic lights do not switch at the same time, and there are also different green times at different intersections. Actually, some group-based traffic control methods have considered this problem, in these methods the current light phase can be decided to extend the green time or finish it according to the max-pressure signal control policy [27, 28]. In the future work, on the foundation of these methods the cooperative light phase switching method can be further improved for better traffic network performance. Thirdly, the physical traffic model should be designed in the future research work to achieve the actual application in the urban traffic network.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by Shandong Provincial Natural Science Foundation, China [Grant no. ZR2018FM027], Project of Shandong Province Higher Educational Science and Technology Program [Grant no. J17KB181], Key Research and Development Project in Shandong Province, China [Grants nos. 2015GGX101048, 2016GSF120009], and National Natural Science Foundation of Youth Fund Project [Grant no. 61501284].

References

[1] D. I. Robertson, "TRANSYT: a traffic network study tool," Tech. Rep. LR-253, Road Res. Lab., Berkshire, UK, 1969.

[2] J. D. C. Little, M. D. Kelson, and N. H. Gartner, "MAXBAND: a program for setting signal on arteries and triangular network," *Transportation Research Record*, no. 795, pp. 40–46, 1981.

[3] S. Araghi, A. Khosravi, and D. Creighton, "A review on computational intelligence methods for controlling traffic signal timing," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1538–1550, 2015.

[4] B. Cesme and P. G. Furth, "Self-organizing traffic signals using secondary extension and dynamic coordination," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 1–15, 2014.

[5] P. B. Hunt, D. I. Robertson, R. D. Bretherton, R. Winton, and L. Scoot, "A traffic responsive method of coordinating signals," Tech. Rep. LR1014, TRRL, Crowthorne, UK, 1981.

[6] A. G. Sims, "The Sydney coordinated adaptive traffic system," in *Proceedings of the Engineering Foundation Conference on Research Directions in Computer Control of Urban Traffic Systems*, pp. 12–27, Calif, USA, 1979.

[7] H. M. Abdelghaffar, H. Yang, and H. A. Rakha, "Isolated traffic signal control using a game theoretic framework," in *Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems, ITSC 2016*, pp. 1496–1501, Rio de Janeiro, Brazil, November 2016.

[8] Q. Wu, B. Li, and K. Chen, "A multi-agent traffic control model based on distributed system," *Sensors & Transducers*, vol. 173, pp. 60–67, 2014.

[9] D. McKenney and T. White, "Distributed and adaptive traffic signal control within a realistic traffic simulation," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 574–583, 2013.

[10] A. Jovanović, M. Nikolić, and D. Teodorović, "Area-wide urban traffic control: a bee colony optimization approach," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 329–350, 2017.

[11] J. Jin and X. Ma, "Hierarchical multi-agent control of traffic lights based on collective learning," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 236–248, 2018.

[12] P. Shao, L. Wang, W. Qian, Q.-G. Wang, and X.-H. Yang, "A distributed traffic control strategy based on cell-transmission model," *IEEE Access*, vol. 6, pp. 10771–10778, 2018.

[13] L. Wang, C. Li, M. Z. Q. Chen, Q.-G. Wang, and F. Tao, "Connectivity-based accessibility for public bicycle sharing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 99, no. 4, pp. 1–12, 2018.

[14] T. Wongpiromsarn, T. Uthaicharoenpong, Y. Wang, E. Frazzoli, and D. Wang, "Distributed traffic signal control for maximum network throughput," in *Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems, ITSC 2012*, pp. 588–595, Anchorage, Alaska, USA, September 2012.

[15] Z. Jiao, B. Zhang, C. Li, and H. T. Mouftah, "Backpressure-based routing and scheduling protocols for wireless multihop networks: a survey," *IEEE Wireless Communications Magazine*, vol. 23, no. 1, pp. 102–110, 2016.

[16] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[17] J. Gregoire, E. Frazzoli, A. De La Fortelle, and T. Wongpiromsarn, "Back-pressure traffic signal control with unknown routing rates," in *Proceedings of the 19th IFAC World Congress on International Federation of Automatic Control, IFAC 2014*, vol. 47, pp. 11332–11337, South Africa, August 2014.

[18] J. Gregoire, S. Samaranayake, and E. Frazzoli, "Back-pressure traffic signal control with partial routing control," in *Proceedings of the 55th IEEE Conference on Decision and Control, CDC 2016*, pp. 6753–6758, Las Vegas, Nev, USA, December 2016.

[19] A. A. Zaidi, B. Kulcsar, and H. Wymeersch, "Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2134–2143, 2016.

[20] H. Taale, J. Van Kampen, and S. Hoogendoorn, "Integrated signal control and route guidance based on back-pressure principles," *Transportation Research Procedia*, vol. 10, pp. 226–235, 2015.

[21] T. Le, P. Kovács, N. Walton, H. L. Vu, L. L. H. Andrew, and S. S. P. Hoogendoorn, "Decentralized signal control for urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 431–450, 2015.

[22] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[23] J. Gregoire, X. Qian, E. Frazzoli, A. De La Fortelle, and T. Wongpiromsarn, "Capacity-aware backpressure traffic signal control," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 164–173, 2015.

[24] X. Hu, J. Cheng, and H. Luo, "Task assignment for multi-uav under severe uncertainty by using stochastic multicriteria acceptability analysis," *Mathematical Problems in Engineering*, vol. 2015, Article ID 249825, 10 pages, 2015.

[25] W. Zhao, Q. Meng, and P. W. H. Chung, "A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario," *IEEE Transactions on Cybernetics*, vol. 46, no. 4, pp. 902–915, 2016.

[26] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[27] S. Lee, S. C. Wong, and P. Varaiya, "Group-based hierarchical adaptive traffic-signal control part I: formulation," *Transportation Research Part B: Methodological*, vol. 104, pp. 1–18, 2017.

[28] S. Lee, S. C. Wong, and P. Varaiya, "Group-based hierarchical adaptive traffic-signal control Part II: implementation," *Transportation Research Part B: Methodological*, vol. 105, pp. 376–397, 2017.