

Received June 20, 2021, accepted July 4, 2021, date of publication July 13, 2021, date of current version July 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3096827

A General Framework for Intersection Traffic Control With Backpressure Routing

SEN BAI^{1,2} AND XIN BAI³

¹Department of Clinical Research, The First Hospital of Jilin University, Changchun 130021, China

²School of Software, Tsinghua University, Beijing 100084, China

³Huawei Technologies Company Ltd., Hangzhou 310007, China

Corresponding author: Xin Bai (baixinbs@163.com)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61702298.

ABSTRACT Traffic congestion has long been a worldwide difficult problem in metropolitan transportation networks, incurring tremendous time waste and exhaust pollution. Extensive efforts have been made to address this problem, among which traffic control at intersections is known to be especially crucial while challenging. Also, it is helpful to the recent advent of autopilot technologies by enhancing the schedule accuracy and reducing the infrastructure cost. In practice, however, existing researches can hardly work well in a pervasive manner since they are essentially limited to two ideal assumptions: 1) each intersection comprises four ways; 2) each way is homogeneously composed of two lanes. Through an in-depth examination of their basic models, we find that the two-fold ideal assumptions are largely compelled by the surprisingly high complexity of converting a practical intersection topology into a theoretical conflict graph. Moreover, existing works seldom consider the modeling of complex intersections in the scene of cooperative control of multiply intersections. Driven by the above understandings, our first effort towards a general framework (for handling multiple-way heterogeneous intersections) is to carefully transform a practical intersection topology into a homomorphic, regular conflict graph which is suited to theoretical modeling and further processing with an affordable complexity. Besides, a maximum weight independent set (MWIS) based approach is proposed to minimize the average waiting time of vehicles at an isolated intersection. In addition, we apply a backpressure-based algorithm to our framework to further optimize the global average waiting time of vehicles in a whole road network. Simulation results have demonstrated that the average waiting time achieved by our approach is merely 80 seconds while traditional traffic light control reaches 370 seconds.

INDEX TERMS Intersection control, backpressure routing, vehicular networks, conflict graph.

I. INTRODUCTION

As a consequence of human beings' rapid urbanization and population growth, urban traffic congestion has become an increasingly severe worldwide problem. It not only impairs the road transportation safety and people's driving experience, but also increases the fuel consumption and exhaust emission (thus aggravating the air pollution) [1]. To this end, in the past few years intelligent transportation systems (ITS) have been taken as a key solution to enhancing the transportation efficiency and mitigating the environmental impact [2], [3]. Specifically, ITS have been employed or expected to support various applications such as traffic surveillance [4], collision avoidance and automatic

transportation pricing [5]. As to all these typical applications, the control of vehicular traffic at road intersections (or simply says intersection traffic control) is known to be crucial for the usefulness and efficiency of ITS [2], [6]–[14].

As a widely used approach to intersection traffic control, intelligent traffic light systems formulate the vehicular traffic signal control processes into certain scheduling problems, particularly to optimize the scheduling of green signals [2], [10], [15], [16]. An efficient schedule algorithm should be able to effectively reduce the waiting time of vehicles at each intersection, and meanwhile increase the traffic throughput. To achieve this goal, recent researches have made use of various computational intelligence approaches [13] such as evolutionary computation algorithm [10], fuzzy logic control [11], [17], [18] and Neural Network [19]. In recent years, *vehicular ad hoc networks* (VANETs) have been exploited

The associate editor coordinating the review of this manuscript and approving it for publication was Arun Prakash^{id}.

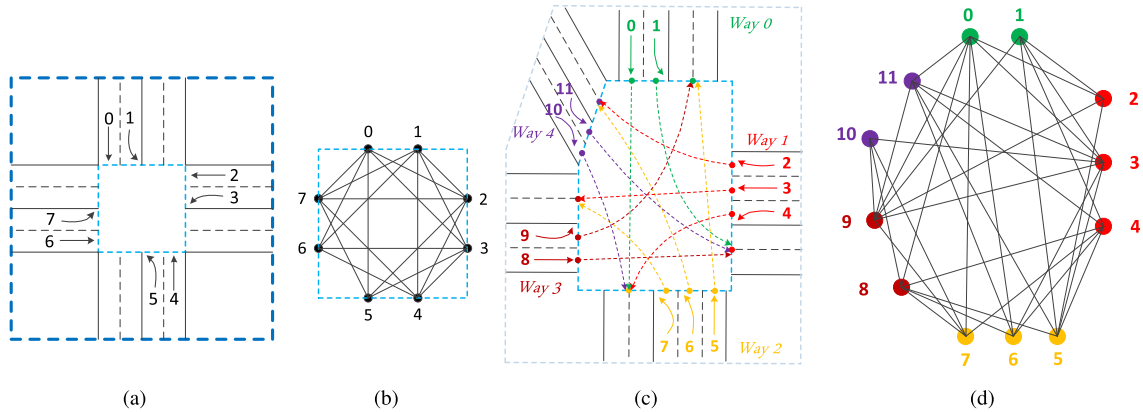


FIGURE 1. (a) A common 4-way intersection. (b) The conflict graph depicting the common 4-way intersection. (c) A practical 5-way intersection. (d) The conflict graph corresponding to the practical five-way intersection.

for intersection traffic control [2], [8], [10], [11]. Compared with roadside equipment [17], [20], VANETs can be leveraged to collect and aggregate fine-grained speed and position information of vehicles. With VANETs, vehicles can use wireless links for vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communications [21]. Therefore, VANET-based intersection traffic control solutions are usually more flexible and efficient.

Besides the common usage scenarios, intersection traffic control is also helpful to the promising automatic driving technologies [22], [23]. Take Google’s self-driving cars as an example, which use the LIDAR system [24] to accurately map out the vehicle’s surroundings with expensive laser radars. Since they rely primarily on pre-programmed route data, they do not obey temporary traffic lights and, in some situations, revert to a slower extra caution mode at complex unmapped intersections. If Google’s self-driving cars were equipped with intelligent traffic control solutions that are applicable to all kinds of intersections, their traffic schedule accuracy would be remarkably enhanced and the aforementioned limitation can be effectively addressed. If this came true, they might even not need the current expensive laser radars.

Despite the fundamental significance of intersection traffic control, existing researches can hardly work well in a pervasive manner since they are essentially limited to two ideal assumptions as demonstrated in Fig. 1(a): 1) each intersection comprises four ways; 2) each way is homogeneously composed of two lanes [2], [8], [10]–[12]. In the real world, however, there are numerous five-way and six-way intersections which involve a crossing of three arterial streets at one junction (as plotted in Fig. 1(c)), especially in urban areas with non-rectangular blocks.

In order to figure out the root causes of the abovementioned two-fold ideal assumptions (note that we believe previous researchers had also noticed the vast existence of multiple-way heterogeneous intersections), we make an in-depth examination of the basic models of existing researches. In particular, we find that an intersection topology was typically converted into a theoretical *conflict graph* using

an intuitive (manual) approach (as demonstrated in Fig. 1(b)), in order to easily and precisely avoid the conflicts of vehicle traffic [8], [10]. The conflict graph of a 4-way, 16-lane intersection is exemplified in Fig. 1(b). Unfortunately, in a practical case, converting the multiple-way heterogeneous intersection topology into a conflict graph turns out to be surprisingly complex. For example, converting the realistic 5-way 22-lane intersection case in Fig. 1(c) into a conflict graph can hardly be accomplished through an intuitive approach. Besides, generating the electronic map for an urban area usually has to deal with tens to thousands of such uncommon intersections. To make things (*i.e.*, constructing a suitable conflict graph) more complex, in some countries/areas special traffic management measures have been taken, such as reversible lanes or intersection traffic signs for no left turn.

Driven by the above understandings, in this paper we strive towards a general framework for intelligent traffic control at all kinds of intersections. To this end, our first effort is to carefully transform a practical intersection topology into a homomorphic, regular conflict graph (as illustrated in Fig. 1(d)) which is suited to theoretical modeling and further processing (*e.g.*, electronic map design¹) in an affordable complexity. The whole transforming process is automated, and the blueprint (entrance and exit for each lane) of the intersection is the only input.

Based on our constructed models for multiple-way heterogeneous intersections, we design a maximum weight independent set (MWIS) based algorithm to minimize the average waiting time of vehicles at an isolated (or says individual) intersection. Nevertheless, in a road network with multiple intersections, the optimal scheduling at each isolated intersection does not always imply the minimization of all vehicles’ average waiting time in a whole road network. To this end, we further apply a backpressure-based algorithm [15], [16], [25]–[27] to our framework to reduce the global average

¹Electronic map is a core component of intelligent navigation systems. Nowadays, Amap, Google Map and Baidu Map are widely used for vehicle navigation route generation in our daily life.

waiting time of vehicles.² Besides, we develop a completely distributed implementation for the whole algorithm (MWIS + backpressure), so that each vehicle can make separate control decisions at each intersection.

Extensive evaluations on representative workloads confirm the efficacy of our framework. For example, at a 5-way 43-lane intersection (Fig. 11) where the traffic arrival rate is about 8000 vehicles per hour, the average waiting time generated by the practically used traffic light scheduling approach reaches 370 seconds, while the average waiting time achieved by our MWIS-based approach is merely 80 seconds (thus acquiring 78% reduction). Additionally, when we apply the backpressure-based algorithm to the whole urban road network under heavy traffic load, the global average waiting time of vehicles can be further reduced by 7%.

Roadmap: The outline of this paper is as follows. In Section II, we first describe the system model and preliminary definitions; after that, the general framework for conflict graph construction of multiple-way heterogeneous intersection is introduced in detail. In Section III, MWIS-based isolated intersection control is presented. In Section IV, we apply backpressure algorithms in our framework. Evaluation results are included In Section V. Finally, we review the related work in Section VI and conclude the paper in Section VII.

II. CONFLICT GRAPH CONSTRUCTION AND SYSTEM MODEL

In this section, we first define the involved terms and notations, and then propose a novel algorithm of conflict graph construction for generalized intersections. After that, we introduce the vehicular network based system model used in this paper.

A. DEFINITIONS

As shown in Fig. 1(a), most existing intersection traffic control algorithms focus on the typical 4-way intersections. The small blue dashed rectangle represents the core area of the intersection. For simplicity, the lanes entering the core area are numbered from 0 to 7. The large dashed rectangle represents the queue area. A vehicle in the queue area is viewed as in the queue to pass the intersection. A vehicle in the core area is called passing the intersection. The path of a vehicle in the intersection area is determined by the lane it is in.

In order to extend the problem to heterogeneous cases, an example of 5-way 22-lane intersection is given in Fig. 1(c). Five ways are respectively denoted as *Way 0*, *Way 1*, *Way 2*, *Way 3* and *Way 4*. There are totally 12 lanes entering the core area which is represented by the blue dashed pentagon. The queue area can be defined as a proper-sized pentagon larger than the queue area as shown in Fig. 1(c). A vehicle in the queue area is viewed as in the queue to pass the intersection.

²The backpressure routing algorithm is established based on the concept of Lyapunov drift [28], which is crucial for optimal scheduling control in queueing networks.

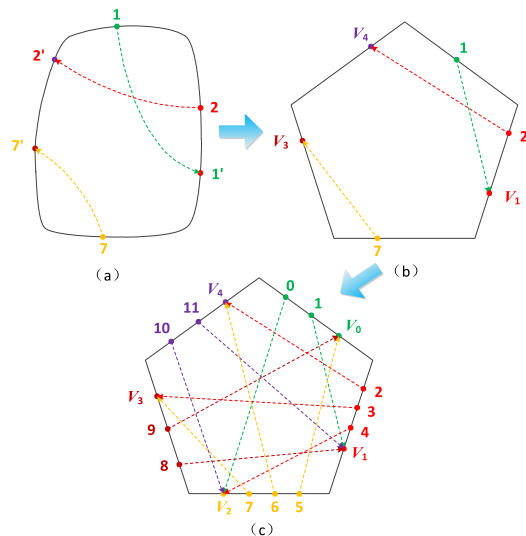


FIGURE 2. The conflict graph construction.

Definition 1 (The Concurrency/Conflict Relationship): If the crossing paths of two vehicles at different lanes intersect inevitably, we say that these two lanes are conflicting with each other. Accordingly, vehicles with non-crossing paths can pass the intersection simultaneously. We call non-conflicting lanes concurrent lanes.

As shown in Fig. 1(c), *lane 1* and *lane 2* are conflicting lanes. *Lane 7* is concurrent with both *lane 1* and *lane 2*. For simplicity, we use a single node to represent the exit of each way as in Fig. 2. For example, three concurrent *lanes 1, 8* and *11* share the same exit, which is denoted as a red point on *Way 1*.

Definition 2 (Conflict Graph): As shown in Fig. 1(d), the conflict relationship among eight lanes can be represented by a conflict graph. The vertices represent the lanes at an intersection and the edges represent the conflict between lanes.

It should be noted that the conflict graph of four-way intersections is fixed as shown in Fig. 1(b). However, it is not trivial to construct the conflict graphs especially when the conflict relationships at a large number of intersections are required.

B. CONSTRUCTION OF CONFLICT GRAPH

From the viewpoint of topology, the core area \mathcal{C} in Fig. 2(a) and the convex pentagon area in Fig. 2(b) can be continuously deformed into each other. In this case \mathcal{C} and the pentagon area are said to be homologous to lie in the same homology class. Thus, we use a regular pentagon area to abstract \mathcal{C} . Each edge of the pentagon represents a way at the intersection. Denote v_1 the exit on *way 1*, v_4 the exit on *way 4*. Moreover, line segments $1v_1$ and $2v_4$ correspond to *lane 1* and *lane 2* respectively. Our approach on conflict graph construction is based on a simple fact as follows.

Fact 1: Denote \mathcal{B} the boundary of a core area \mathcal{C} and a and a' are endpoints of lane 1, and b and b' are endpoints of lane 2. The

Algorithm 1 Conflict Graph Construction

Input: m ways: $\{w_1, \dots, w_m\}$, n lanes: $\{l_1, \dots, l_n\}$; Each way w_i $\{i = 1, \dots, m\}$ corresponds to a set of lanes $L_i = \{l_j, l_{j+1}, \dots, l_{j+k}\}$ ($1 \leq j < j+k \leq n$)
Each lane l_i $\{i = 1, \dots, n\}$ corresponds to a leaving way w_t $\{t = 1, \dots, m\}$, l_i and w_t form a line segment S_i $\{i = 1, \dots, n\}$

Output: a Conflict graph $G_c = (V, E)$

initialized to $V = v_1, \dots, v_n$, $E = \emptyset$.

for $i = 1, i \leq m, i++$ **do**

 Put lanes in L_i as vertices on edges of an m -edge polygon in order

 Traverse all vertices on the edges in clockwise order and number the vertices in sequence

for $i = 1, i \leq n, i++$ **do**

for $j = i, j \leq n, j++$ **do**

if S_i and S_j cross each other (according to Fact 1) **then**

$E \leftarrow E \cup (v_i, v_j)$

boundary \mathcal{B} is traversed in clockwise direction, starting at a , if vertices appear in the order of a - b - a' - b' or a - b' - a' - b , then lane 1 and lane 2 are conflict lanes.

To illustrate Fact 1, we present a simple example in Fig. 2(a). 1 - 2 - $1'$ - $2'$ appear on the boundary in clockwise direction. Thus, lane 1 and lane 2 are conflict lanes.

Fact 2: If line segment $1v_1$ and $2v_4$ in Fig. 2(b) intersect with each other, then lane 1 and lane 2 in Fig. 2(a) are conflict lanes. Otherwise, they are concurrent lanes.

Now we are ready to introduce our two-phased modeling method. The details can be referred to Algorithm 1. The time complexity of the algorithm is $O(n^2)$, where n is the number of lanes.

Step 1: Topology of the intersection is continuously deformed into a regular form. For an m -way n -lane intersection, we use regular polygons with m edges to abstract the intersection. Each lane is represented by a vertex which lies on an edge corresponding to their own way.

Step 2: According to the clockwise order in the regular form, number $2n$ endpoints of lanes in sequence. Hence, any conflict relationship of lanes can be computed by Fact 1.

As shown in Fig. 2(c), for 5-way intersections, we use regular polygons with 5 edges to abstract the intersection. Each lane is represented by a vertex which lies on their own edge. According to road rule information, connect the vertices as shown in Fig. 2(c). Therefore, conflict graph in Fig. 1(d) is easily computed by the intersection model in Fig. 2(c).

C. SYSTEM MODEL

We assume that each vehicle has a unique id such as the license plate number. Meanwhile, each vehicle is equipped with an autonomous driving system and an electronic map system of automobile navigation. The electronic map

package is able to include some extra data such as conflict graph for traffic control. In addition, vehicles are assumed to be able to detect the boundary of the queue/core area when it crosses the boundary. With vehicular networks, vehicles can communicate with each other by wireless communication [21]. We also assume that vehicles inside the queue area have transmission ranges larger than the length of the queue area. That is, vehicles can communicate with each other directly. As in literature [8], we use three states to describe the procedure of a vehicle passing the intersection.

- *IDLE:* If a vehicle is out of the queue area, then it is in the idle state.
- *WAITING:* If a vehicle is waiting for the permission to enter the core area, then it is in the waiting state.
- *PASSING:* A vehicle is in the passing state during the time interval between receiving the permission and exiting the core area.

III. MWIS-BASED ISOLATED INTERSECTION CONTROL

A basic problem in intersection control is how to find the sets of concurrent lanes. In [9], all twelve different possible cases of green lights at a 4-way intersection are listed. However, it is a big task to list all the possible cases of concurrent lanes at a complex intersection rapidly, accurately, and then schedule them. In the following, we present the details of our MWIS-based intersection control, including the concurrent lane set construction, details of the proposed intersection control algorithms and our vehicular network based communication mechanism.

A. MAXIMUM WEIGHT INDEPENDENT SET

We model the conflicts between lanes with a conflict graph, so that concurrent lanes in the conflict graph must be independent to each other. From this simple observation, we come to the following theorem.

Theorem 1: A set of concurrent lanes form an independent set in the conflict graph.

Proof: This theorem can be derived from Fact 1.

Definition 3: Maximal independent set (MIS).

In graph theory, an independent set is a set of vertices in a graph, such that no two of which are adjacent. Moreover, *maximal independent set* (MIS) is an independent set that is not a subset of any other independent set.

Definition 4: Right of way (ROW).

At a certain time, there is a set of concurrent lanes in which vehicles have the right to pass through the intersection. We call every lane in this set has the *right of way* at that time.

Fact 3: In order to maximize the utility of the core area, the set of concurrent lanes with ROW forms an MIS in the conflict graph.

A simple example is shown in Fig. 3. As shown in Fig. 3(b), lane 2, lane 10 and lane 11 form an maximal independent set. As shown in Fig. 3(a), every yellow path intersects with at least one of the black paths.

Definition 5: Maximum weight independent set (MWIS).

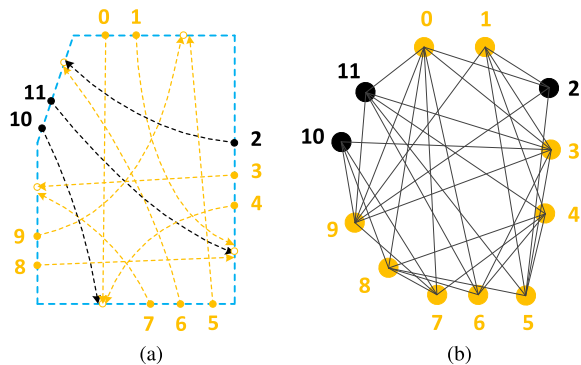


FIGURE 3. Concurrent lanes 2,10,11 form an MIS in conflict graph.

In graph theory, an MWIS is an independent set with maximum total weight. From the perspective of intersection traffic control, weight of a lane may represent different meanings. For instance, the length of the waiting queue or the average waiting time of vehicles. As we mentioned above, a good intersection control algorithm ensures both fairness and efficiency. By giving the weight of lanes different meanings, we can further construct an MWIS in conflict graph as lanes with ROW to achieve fairness and efficiency.

Time Complexity: The problems of finding MWIS and MIS are classical NP-hard optimization problems. Although every graph contains at most $3^{\binom{n}{2}}$ MISs [29], many graphs have far fewer. Several approximation algorithms for MWIS problem in general graph have been proposed in [30]. However, there is no efficient exact algorithm for MWIS problem in general graph. A simple enumeration method is to list all MISs in a graph [31] and compare their weights. Listing all MISs in a graph is often used as a subroutine for solving many NP-complete problems such as maximum independent problem and the maximum clique problem. It can be done in time $3^{\binom{n}{2}}$ [32].

As we described in our system model, road information such as conflict graph of the intersection can be stored in the electronic map package. Similarly, inspired by V2I communication in vehicular networks, the conflict graph also can be stored in a central controller. In order to improve the time efficiency, all MISs can be pre-stored in a central controller, and then we can calculate the optimal solution of MWIS problem.

B. PRIORITY SCHEME

There are two key objectives in intersection traffic control: efficiency and fairness. To achieve the first objective, the main task is to minimize the average waiting time of vehicles. To achieve the second objective, we should minimize the variance of waiting time of vehicles. Therefore, to ensure the effectiveness of the algorithm, two strategies should be considered. The first is to let the number of vehicles pass through the intersection during a time period as many as possible. To do this, an MIS in conflict graph should be selected as lanes with ROW. Another strategy is to reduce the

Algorithm 2 Isolated Intersection Control (Controller)

CoBegin

On receiving $RED(VID,LID)$:

If STATE \leftarrow IDLE

Then

STATE \leftarrow ACTIVE

set a defer timer DTIME

$M_0 \leftarrow \emptyset$

broadcast a $GREEN(LID)$ message

$w_{LID} \leftarrow w_{LID} + 1$

insert TIME to $ATlist_{LID}$

On receiving $PASSING(VID,LID)$:

$w_{LID} \leftarrow w_{LID} - 1$

$p_{LID} \leftarrow p_{LID} + 1$

On receiving $LEAVING(VID,LID)$:

$p_{LID} \leftarrow p_{LID} - 1$

If for every entering lane $LID, w_{LID} \leftarrow 0$ and

$p_{LID} \leftarrow 0$

Then STATE \leftarrow IDLE

else if (for every entering lane LID in M_0 ,

$w_{LID} \leftarrow 0$ and $p_{LID} \leftarrow 0$)

Then terminate the defer timer.

Scheduling().

else if (for every entering lane LID in $M_0 \setminus M_1$,

$p_{LID} \leftarrow 0$)

Then broadcast a $GREEN(M_0 \setminus M_1)$ message.

When the defer timer expires:

Scheduling().

Scheduling()

Compute an MWIS M (all MISs are pre-stored)

Broadcast a $RED(M_0 \setminus M_1)$ message

$M_0 \leftarrow M_1$

$M_1 \leftarrow M$

Set a defer timer DTIME

number of ROW deprivations. If it takes time t for a vehicle to pass through the intersection, then t more time is wasted for a lane to gain the right to pass again. To guarantee the fairness, an effective strategy is to use MWIS instead of MIS in the conflict graph. To ensure that any vehicle will not wait too long at the intersection compared to other vehicles, different parameters can be used as variables of the priority function. In our algorithms, we consider two main parameters for the calculation of priority: L_i : the queue length of vehicles on a lane. T_i : the waiting time of the first vehicle in the queue. One of the advantages of adopting these two parameters is that they are easy to obtain. The priority function of lane i is defined as:

$$f(i) = L_i + \lambda T_i, \quad (1)$$

Algorithm 3 Isolated Intersection Control (Vehicles)

Input: The queue area: \mathcal{Q} The core area: \mathcal{C}

CoBegin //for each vehicle VID from lane LID,

On entering \mathcal{Q} :
 Send a $REQ(VID, LID)$ message to controller.
 STATE \leftarrow WAITING.

On receiving $GREEN(LG)$:
If $(LID \in LG \text{ and } STATE \leftarrow WAITING)$
Then follow the front vehicles to enter \mathcal{C} .

On receiving $RED(LB)$:
If $LID \in LB \text{ and } STATE \leftarrow WAITING$
Then stop entering \mathcal{C} and keep waiting in the queue.

On entering \mathcal{C} :
 Send a $PASSING(VID, LID)$ message to controller.
 STATE \leftarrow PASSING.

On leaving \mathcal{C} :
 Send a $LEAVING(VID, LID)$ message.
 STATE \leftarrow IDLE.

where λ is a tunable parameter. Notice that various priority functions can be chosen. We choose this function cause its value grows fast with T_i . Thus, if a vehicle has waited for a long time, it will have a high priority to pass through. In our experience, the value of λ has very little effect on the algorithm if vehicle arrival rate obeys Poisson distribution.

At a certain time, an MWIS with ROW ensures the efficiency and fairness at the same time. As the vehicles pass through, the weight of MWIS decreases and another MWIS kicks in.

C. ISOLATED INTERSECTION CONTROL USING V2I COMMUNICATION

In the following, we present the details of our control algorithm. Inspired by V2I communication in vehicular networks, a central controller is used to compute the MWIS. Vehicles request to pass the intersection by sending messages to the controller, and based on the response message. The controller stores the information of the intersection such as the conflict graph and maintains the information of vehicles. By integrating vehicles information, an MWIS in conflict graph is computed. Then, a predetermined time period is assigned to each MWIS. We call it a scheduling time cycle. The end of each time cycle will be triggered in two conditions: 1. The predetermined scheduling time cycle is over. 2. All vehicles in lanes corresponding to current MWIS have passed the core area. When either of these two conditions holds, the central controller will compute a new MWIS to replace the old one. Lanes no longer in the MWIS will be deprived of their ROWs. Each new member in MWIS will wait until all vehicles in its conflict lanes have passed the core area, and then it will receive the permission to pass from the central controller.

TABLE 1. Notations and message types.

<i>TIME</i>	<i>The current system time.</i>
<i>STATE</i>	<i>The current state of a vehicle or the control unit.</i>
<i>PASSING</i>	<i>The control unit has two states: IDLE, ACTIVE.</i>
<i>ATlist_{LID}</i>	<i>The arrival times of vehicles in lane with id LID. When a vehicle arrives at lane LID, its arrival time is inserted into ATlist_{LID}. When a vehicle in lane LID enters the core area, its arrival time is removed from ATlist_{LID}. The arrival times of vehicles are added in FIFO (first-in-first-out) order, so the header in ATlist_{LID} is the arrival time of the first vehicle in the queue.</i>
<i>w_{LID}</i>	<i>The number of vehicles waiting in the lane with id LID.</i>
<i>p_{LID}</i>	<i>The number of vehicles in the core area for lane with id LID.</i>
<i>REQ(VID, LID)</i>	<i>A vehicle with id VID sends a request message REQ(VID, LID) to the central controller when it enters the queue area of lane LID.</i>
<i>GREEN(LG)</i>	<i>The central controller broadcasts the GREEN message to allocate the ROW. LG denotes a set of LIDs. Vehicles in lanes with GREEN message are allowed to pass.</i>
<i>RED(LB)</i>	<i>The central controller broadcasts the RED message to take the ROW back. LB denotes a set of LIDs. Vehicles in lanes with RED message are forbidden to enter the core area.</i>
<i>PASSING(VID, LID)</i>	<i>A vehicle sends a message PASSING(VID, LID) when it enters the core area.</i>
<i>LEAVING(VID, LID)</i>	<i>A vehicle sends a message LEAVING(VID, LID) when it leaves the core area.</i>

Notice that ROW deprivations will not happen on common elements in two MWISs.

Our isolated intersection control algorithm consists of two parts. One part is executed on vehicles and another part is executed on the central controller. We present these two parts in Algorithm 2 and Algorithm 3 respectively. Notations and message types in our algorithm are illustrated in Table 1.

IV. BACKPRESSURE INTERSECTION TRAFFIC CONTROL

In queueing theory, backpressure routing algorithm is a method for directing traffic around a queueing network that achieves maximum network throughput, which is established using Lyapunov drift [28]. Before being used in intelligent traffic control in transportation networks, backpressure algorithms are originally developed for routing and scheduling in communication networks. The mathematical optimality properties of backpressure have motivated investigations in city-wide control of traffic flow. Backpressure traffic signal control is implementable in a completely decentralized manner and stability optimal [15], [16], [25]. In contrast to centralized methods, distributed solutions for intersection traffic control are more implementable especially for large urban areas.

A. ROAD NETWORK

Definition 6 (Road Network): The urban road network is modelled as a standard queueing network. A road network of m

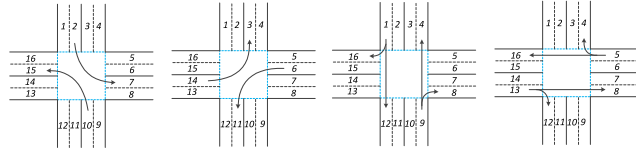


FIGURE 4. Typical phases of a 4-way intersection.

roads and n links is modelled as a directed graph $G = (V, E)$, where $V = \{V_1, V_2, \dots, V_m\}$ and $E = \{E_1, E_2, \dots, E_n\}$. A node in V represents a road with queuing vehicles, and an edge in E enables transfers from node to node. A vehicle exogenously enters the network from a certain road, travels along several roads and leaves the network at a certain road.

A traffic movement corresponding to vehicles exiting V_a and entering V_b is represented by an edge (V_a, V_b) . For a road network with \mathcal{J} intersections, each intersection $(J_i)_{i \in \mathcal{J}}$ consists of a set of edges. For an intersection J , $\mathcal{I}(J)$ and $\mathcal{O}(J)$ denote input nodes and output nodes of J . Such that if (V_a, V_b) is an edge of J , then $V_a \in \mathcal{I}(J)$ and $V_b \in \mathcal{O}(J)$. Denote $(V_a, V_b) \in J$. As standard in queuing network control, time is slotted. At any time slot t , $Q_{ab}(t)$ denotes the number of vehicles queued in a road V_a waiting to move to an adjacent road V_b . Denote $Q_a(t) = \sum_b Q_{ab}(t)$ the total queue length at road V_a .

B. FIXED-PHASED BACKPRESSURE TRAFFIC SIGNAL CONTROL

Existing works on backpressure traffic signal control assume that an intersection V_i is controlled by activating a predefined finite set of phases P_i . Typical phases at a four-way intersection are shown in Fig. 4. For example, a typical 4-way intersection has four possible phases $P_i = \{p_i^1, p_i^2, p_i^3, p_i^4\}$:

$$\begin{aligned} p_i^1 &= (V_2, V_7), (V_{10}, V_{15}) \\ p_i^2 &= (V_{14}, V_3), (V_6, V_{11}) \\ p_i^3 &= (V_1, V_{12}), (V_1, V_{16}), (V_9, V_4), (V_9, V_8) \\ p_i^4 &= (V_5, V_4), (V_5, V_{16}), (V_{13}, V_{12}), (V_{13}, V_8) \end{aligned}$$

Denote $p_i(t)$ the activated phase at intersection V_i during time slot t , thus $p_i(t) \in P_i$. During a time slot, a feasible phase is activated to control traffic flows around the intersection. Let $P(t)$ denote global phase control of the road network at time slot t , that is $P(t) = (p_i(t))_{i \in \mathcal{M}}$. Furthermore, denote $\mu_{ab}(p_i(t))$ the maximum number of vehicles transferred from V_a to V_b if phase $p_i(t)$ is activated. That is, if (V_a, V_b) is not in $p_i(t)$, then $\mu_{ab}(p_i(t)) = 0$. Thus, if global phase $P(t)$ is activated during time slot t , the number of vehicles transferring from V_a to V_b is:

$$f_{ab}(t) = \min(Q_{ab}(t), \mu_{ab}(P(t))) \quad (2)$$

Let $A_a(t)$ be the number of vehicles exogenously arrive at node N_a during time slot t . We assume that the ratio of vehicles added to $Q_{ab}(t)$ is rate-convergent with rate $r_{ab} \in [0, 1]$, thus r_{ab} represents the long-term routing ratios of vehicles

Algorithm 4 Fixed-Phased Backpressure Intersection Control

Input: Intersection J_i

Edge set of J_i

$\mathcal{I}(J_i)$ and $\mathcal{O}(J_i)$

Output: $p_i(t) \in P_i$

for $V_a \in \mathcal{I}(J_i)$ and $V_b \in \mathcal{O}(J_i)$ **do**

$W_{ab}(t) \leftarrow$

$d_{ab}(t) \max(\text{Pressure}(Q_a)(t) - \text{Pressure}(Q_b)(t), 0)$

$p_i(t) \leftarrow \text{argmax}_{p_i(t) \in P_i} \sum_{a,b} W_{ab}(t) \mu_{ab}(p_i)$

Algorithm 5 MWIS-Based Backpressure Intersection Control

Input: Intersection J_i

Edge set of $J_i: \mathcal{E}$

n_c : For a road V_c , n_c is the total number of lanes of V_c .

Output: $p_i(t)$

for $(L_a, L_b) \in \mathcal{E}$, L_b is a lane of road V_c **do**

$W_{ab}(t) \leftarrow \max(\text{Pressure}(Q_a)(t) - \frac{\text{Pressure}(Q_c)(t)}{n_c}, 0)$

for $\{W_{ab}(t) | (L_a, L_b) \in \mathcal{E}\}$, compute an MWIS $p_i(t)$

entering N_a and the exit rate at node N_a is $1 - \sum_b r_{ab} \geq 0$. The queue dynamics is:

$$Q_{ab}(t+1) = Q_{ab}(t) - f_{ab}(t) + r_{ab}(t) \left(\sum_c f_{ca}(k) + A_a(t) \right), \quad (3)$$

where $r_{ab}(t)$ is rate-convergent with rate r_{ab} .

In backpressure traffic signal control, pressures at intersections in the road network are computed based on the vehicle queue lengths. A simple linear pressure function is to use the queue length as the pressure exerted by a node, that is:

$$\text{Pressure}(Q_a) = Q_a \quad (4)$$

Let $d_{ab}(t) \in \{0, 1\}$ denote the presence/absence of vehicles on V_a waiting to depart from V_a to V_b . The outline of backpressure traffic signal control is in Algorithm 4.

C. MWIS-BASED BACKPRESSURE INTERSECTION CONTROL

In fixed-phased backpressure traffic control, an intersection controller chooses traffic movements during a time slot to maximize the pressure release. This is a max weight problem as follows:

$$\text{Maximize : } \sum_{(V_a, V_b) \in J_i} W_{ab}(t) \mu_{ab}(p_i) \quad (5)$$

$$\text{Subject to : } p_i \in P_i \quad (6)$$

where,

$$W_{ab}(t) \leftarrow d_{ab}(t) \max(\text{Pressure}(Q_a)(t) - \text{Pressure}(Q_b)(t), 0) \quad (7)$$

Considering the modelling method for multiple-way heterogeneous intersections and the proposed MWIS-based intersection control in this paper, let $L = \{L_1, L_2, \dots, L_p\}$ denote lane set of the road network. Each edge in E connects a pair of lanes in L . Such that if (L_a, L_b) is an edge of J , denote $(L_a, L_b) \in J$. Moreover, we propose a new max weight problem for the backpressure intersection control:

$$\text{Maximize : } \sum_{(L_a, L_b) \in p_i} W_{ab}(t) \tag{8}$$

$$\text{Subject to : } (L_a, L_b) \in J_i \tag{9}$$

and edges in p_i form an MIS in the conflict graph, where

$$W_{ab}(t) \leftarrow \max(\text{Pressure}(Q_a)(t) - \frac{\text{Pressure}(Q_c)(t)}{n_c}, 0) \tag{10}$$

L_b is a lane of road V_c and V_c has n_c lanes. The queue lengths of L_a, L_b and V_c are Q_a, Q_b and Q_c . That is, the max weight problem for backpressure traffic control reduces to an MWIS problem in Algorithm 5. Interestingly, for the first time, our study illustrates that how to increase the throughput of an isolated intersection and the entire road network at the same time.

V. PERFORMANCE EVALUATION

We evaluate the effectiveness of our proposed approach by experiments under various situations. Network simulator ns-3 (version 3.25) and traffic simulator SUMO (version 0.30) are used to conduct experiments. First, we simulate an adaptive traffic signal control algorithm as proposed in [10]. Compared with the existing method, our MWIS-based isolated intersection control method proves to be efficient in the case of 4-way intersections. Second, to validate the proposed isolated intersection control at heterogeneous multiple-way intersections, a real signalized 5-way intersection of Renmin Street, Nanhu Road and Gongnong Road, in Changchun, China, is used as a case study. Third, we test the effect of isolated and backpressure intersection control on urban road network throughput under various population. A real block in Changchun City is used as the road network. It is shown whether the vehicles arrival rate is high or low, our MWIS-based approach generally outperforms fixed-time traffic signal control and other adaptive intersection control approaches.

A. 4-WAY HOMOGENEOUS INTERSECTIONS

In our study, some parameters in communication device and vehicles are selected for calibration as follows. We set queue area of intersections to be 100m*100m. The transmission range of the communication device is set to be 200m. IEEE 802.11 is adopted as the communication protocol. The saturation flow for each lane in the link is 1800 vehicles/h. We model the vehicular traffic arrival process as a Poisson arrival process. The injection of vehicles follows a Poisson distribution with rate λ . Thus, the vehicle inter-arrival times are exponentially distributed with inverse rate parameter r ,

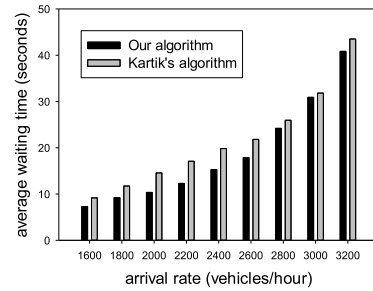


FIGURE 5. 4-way intersections, uniform volume, average waiting time.

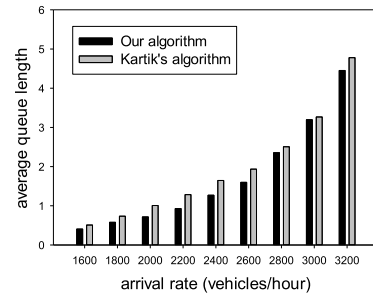


FIGURE 6. 4-way intersections, uniform volume, average queue length.

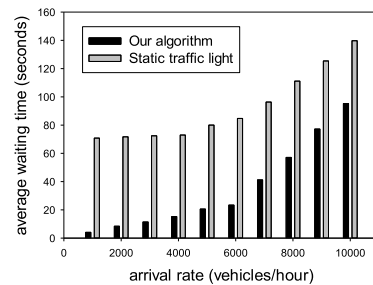


FIGURE 7. 5-way intersections, average waiting time, uniform volume.

where $r = \lambda^{-1}$. For each set of experiment, we conduct 1000 runs with different traffic arrival rate and report the average. As shown in Fig. 5, 6, 7, 8, 9 and 10, for each arrival rate, we run each of the algorithms to be tested for 20 minutes. To measure the performance of our proposed algorithm, some following metrics are used.

Average waiting time: the average time duration that a vehicle stays in waiting state.

Average queue length: the average length of the queue of waiting vehicles in each lane.

Experiments results in 4-way intersections are presented in Fig. 5, 6 and 8, 9 according to different performance metrics. The arrival rate in Fig. 5, 6 and Fig. 8, 9 is sum of arrival rates in all eight lanes. The results from our MWIS-based algorithm and Kartik's algorithm are illustrated respectively. With the increase of arrival rate of vehicles, the average queue length and the average waiting time increase accordingly. Generally, the value of waiting time is dominated by the queue length. As shown in the figures, the performances of these two algorithms are close. Traditional job scheduling based adaptive traffic signal control performs well in the

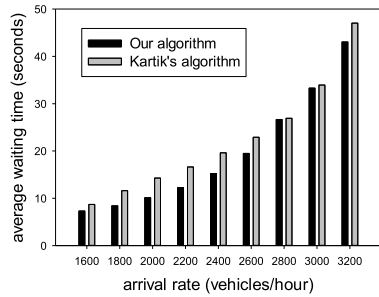


FIGURE 8. 4-way intersections, non-uniform volume, average waiting time.

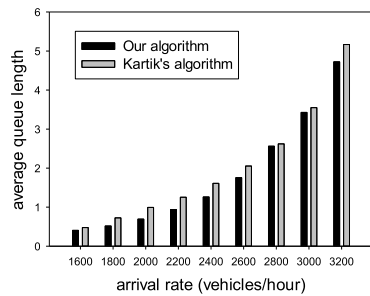


FIGURE 9. 4-way intersections, non-uniform volume, average queue length.

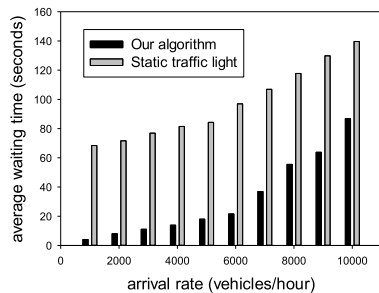


FIGURE 10. 5-way intersections, average waiting time, non-uniform volume.

case of 4-way intersections. However, we can see that the MWIS-based approach still performs a little bit better at any arrival rate.

In order to test the impact of traffic volume on our algorithm, we set two different patterns: the uniform volume and the non-uniform volume. In the pattern of uniform volume, each lane has the same vehicle arrival rate. In the pattern of non-uniform volume, the volume in north and south lanes are two times of that in west and east lanes. The results in Fig. 5, 6 and Fig. 8, 9 show that with uniform traffic volume, both average queue length and average waiting time are shorter than non-uniform cases. Fig. 16 shows the effect of predetermined time period on the average waiting time. The scheduling time cycle is set between 10~20s. We can see that the scheduling time cycle almost has no influence on the average waiting time when the arrival rate is between 1600 and 2800 vehicles/hour. When the arrival rate is between 3200 and 3600 vehicles/hour, the average waiting time decreases as the scheduling time cycle increases. This indicates that at a light

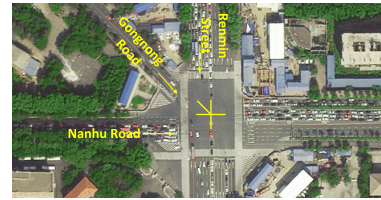


FIGURE 11. A 5-way intersection in Changchun, China.

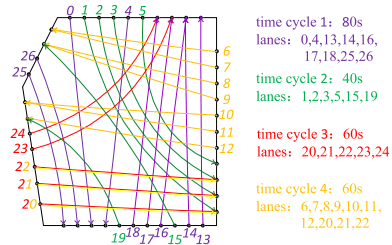


FIGURE 12. Currently used fixed-time traffic signal control strategy at the 5-way intersection in Fig. 13.

traffic arrival rate, the predetermined time period will not be triggered. At a heavy traffic arrival rate, long scheduling time cycle means small number of ROW deprivations but the sacrifice of fairness at the same time.

B. 5-WAY HETEROGENEOUS INTERSECTIONS

To validate the effectiveness of our algorithm at complex intersections, we use a real 5-way intersection as a case study. The main purpose of our experiment is to illustrate the necessity of using adaptive intersection control instead of non-adaptive fixed-time signal control at multiple-way heterogeneous intersections. As shown in Fig. 12, the currently used control strategy consists of four fixed-time time cycles. The following conclusions can be made from the results:

- 1) Our MWIS-based algorithm provides good results at complex intersections. Especially with a low traffic volume, MWIS-based algorithm gets much shorter average waiting time.
- 2) The MWIS-based algorithm can delay the occurrence of traffic congestion. Moreover, better adaptive traffic control at heterogeneous multiple-way intersection is crucial for easing traffic pressure during rush hours.

We first test the performance of our MWIS-based algorithm and fixed-time signal control method at different arrival rates, the results are shown in Fig. 7, 10. Fig. 7 shows the average waiting time under uniform traffic volume. Fig. 10 shows the average waiting time under non-uniform traffic volume. We can see that under light traffic volume, our algorithm achieves extremely short average waiting time. This is easy to understand that adaptive intersection control allows the arriving vehicles to pass as soon as they reach the intersection. Another weakness of static traffic light approach is that it schedules green signal patterns without considering the dynamics of concurrency of lanes. In our approach, vehicles in busy lanes can pass concurrently with large chance. As shown in the figures, when the traffic arrival rate raise up to

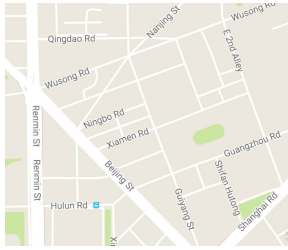


FIGURE 13. The used road network. It is an area around Victory Park in Changchun City, China.

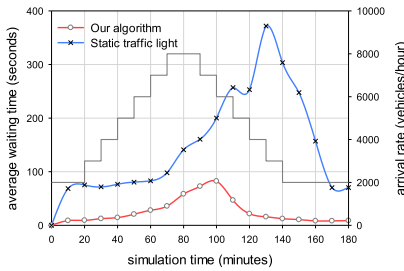


FIGURE 14. 5-way intersections, average waiting time, dynamic traffic volume.

10000 vehicle/hour, the advantage of our proposed algorithm is not that great but still achieve less than 100s average waiting time when the static traffic light method needs 140s.

In order to make our experiment more close to the real circumstances, we vary the traffic arrival rates in a 3-hour time period. The results are shown in Fig. 14, 15. We start out the experiment at a light traffic arrival rate, and gradually increase the vehicle arrival rate to 8000 vehicles/hour. As shown in the figures, the rush hour appears at around 80 minute. After that, the traffic volume decreases accordingly. The results show that our MWIS-based algorithm can adapt to the change of traffic flow. Our approach can delay the occurrence of traffic congestion due to its good performance at low traffic rate and recovers from congestion much faster. Notice that in Fig. 7, 10, for each arrival rate we test our algorithm for 20 minutes. In Fig. 14, since the heavy traffic arrival rate lasts for a long time, congestion happens and the average waiting time increases to about 370s. It is interesting to see that fixed-time traffic control approach cannot digest the congested vehicles in time and the vehicles will keep accumulating at the intersection. In reality most traffic congestions in the rush hour can be avoided.

C. ISOLATED AND BACKPRESSURE INTERSECTION CONTROL

To test the effect of our algorithms on the global throughput of road networks, a road network from a central region in Changchun City, China has been used as a case study. The network comprised of 25 signalized intersections is depicted in Fig. 13. Traffic flows are generated by ActivityGen in SUMO supporting tools. ActivityGen considers the road network as a city. We test the scenarios with population ranging from 1000 to 8000.

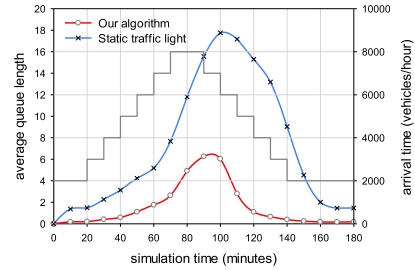


FIGURE 15. 5-way intersections, average queue length, dynamic traffic volume.

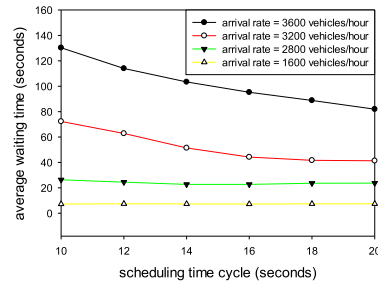


FIGURE 16. The effect of scheduling time cycle on the average waiting time.

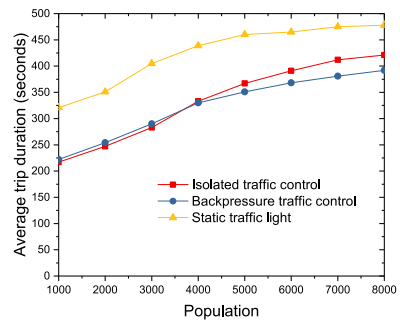


FIGURE 17. The average trip duration of vehicles travelling in the road network.

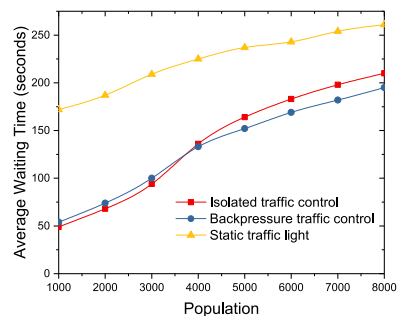


FIGURE 18. The average waiting time of vehicles travelling in the road network.

Three algorithms are tested: static traffic light, MWIS-based isolated intersection control and MWIS-based backpressure traffic control. For the backpressure method, we consider that each time slot is 20 seconds. All simulations are performed for 3600 seconds. Average trip durations and waiting time of

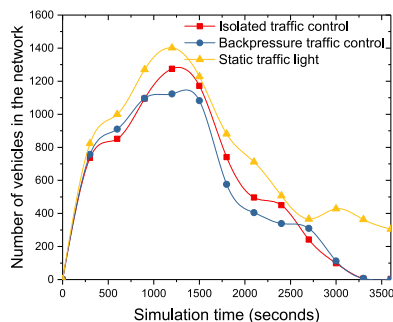


FIGURE 19. The number of running vehicles in the road network when population is 6000.

vehicles have been shown in Fig. 17, 18, both MWIS-based isolated traffic control and backpressure control outperform static traffic light approach. Compared with backpressure method, MWIS-based isolated intersection control has tiny advantage when the population is less than 4000. However, for a population of 4000 to 8000, backpressure traffic control shows its advantage. Therefore, the performance gain of backpressure control mainly occurs at heavy traffic load. Moreover, with light network traffic, MWIS-based isolated intersection control should be used to help the vehicles to arrive at their destinations as fast as they can. Fig. 19 shows the number of running vehicles in the road network when population is 6000. As the figure shows, the peak of traffic flows appears at around 1300s and backpressure traffic control is the best approach to ease the traffic congestion.

VI. RELATED WORK

A recent work called CityDrive [33] implements a speed-advisory driving system to suggest proper speed for drivers so that they arrive at intersections in green phase. As in [33], most existing intersection control approaches focus on traffic light control systems. However, the rise of autonomous driving technologies and the emergence of vehicular networks have promote new perspectives on urban traffic control in recent years. For example, in [34], [35] optimal control frameworks on coordinating autonomous vehicles at an isolated or two adjacent intersections in urban area have been proposed. With the development of autopilot technology, navigation of self-driving vehicles in complex urban area will greatly benefit from automotive automatic control technology. Vehicular network is another promising technology for urban traffic control. Communication among vehicles is a fundamental requirement for coordination among vehicles. Several works have proposed their vehicular network based solutions on intersection traffic control [2], [8], [10], [11]. The mutual exclusion-based method in [8] is the only distributed intersection traffic control algorithm which focuses on communication mechanisms. Backpressure traffic signal control is another type of adaptive traffic signal control approach [15], [16], [25]. The implementation of these controllers requires real-time traffic information and aims to increase the global throughput of road network.

Backpressure methods address network level traffic control locally and provide good performance compared to traditional fixed time schedule controllers. A feasible improvement of backpressure traffic control is the dynamic rerouting scheme for vehicles.

Our work focuses on a fundamental problem in urban traffic control: with smart intersection management technology, how well does distributed intersection control solve the traffic congestion in urban environment.

VII. CONCLUSION

In this paper, we propose an MWIS-based intersection traffic control approach. We focus on multiple-way heterogeneous intersections and propose a novel model on conflict graph construction. Rather than traditional traffic signal control, the proposed algorithm takes advantages of fine-grained information from vehicular networks. Our evaluation results show that the MWIS-based intersection control approach can handle various traffic cases at a complex urban intersection. Moreover, we develop a completely distributed implementation to increase the global throughput of road networks. Compared with traditional isolated intersection control methods which do not care about the global optimal of road networks, our distributed manner not only makes the intersection control framework in this paper more general, but also provides a new prospective on isolated intersection control. Although backpressure routing has shown its advantages, in practice cloud systems are required to achieve real-time and cooperative control of multiply intersections. The design and implementation of backpressure based traffic control systems in real-world settings will be a key issue in our future work.

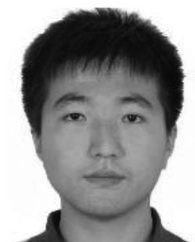
REFERENCES

- [1] X. Chang, B. Y. Chen, Q. Li, X. Cui, L. Tang, and C. Liu, "Estimating real-time traffic carbon dioxide emissions based on intelligent transportation system technologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 469–479, Mar. 2013.
- [2] M. B. Younes and A. Boukerche, "Intelligent traffic light controlling algorithms using vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 5887–5899, Aug. 2016.
- [3] M. Ferreira and P. M. d'Orey, "On the impact of virtual traffic lights on carbon emissions mitigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 284–295, Mar. 2012.
- [4] H.-Y. Cheng and S.-H. Hsu, "Intelligent highway traffic surveillance with self-diagnosis abilities," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1462–1472, Dec. 2011.
- [5] A. Y. S. Lam, "Combinatorial auction-based pricing for multi-tenant autonomous vehicle public transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 859–869, Mar. 2016.
- [6] M. Ahmane, A. Abbas-Turki, F. Perronnet, J. Wu, A. E. Moudni, J. Buisson, and R. Zeo, "Modeling and controlling an isolated urban intersection based on cooperative vehicles," *Transp. Res. C, Emerg. Technol.*, vol. 28, pp. 44–62, Mar. 2013.
- [7] L.-W. Chen and C.-C. Chang, "Cooperative traffic control with green wave coordination for multiple intersections based on the internet of vehicles," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1321–1335, Jul. 2017.
- [8] W. Wu, J. Zhang, A. Luo, and J. Cao, "Distributed mutual exclusion algorithms for intersection traffic control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 65–74, Jan. 2015.
- [9] B. Zhou, J. Cao, X. Zeng, and H. Wu, "Adaptive traffic light control in wireless sensor network-based intelligent transportation system," in *Proc. IEEE 72nd Veh. Technol. Conf.-Fall*, Sep. 2010, pp. 1–5.

- [10] K. Pandit, D. Ghosal, H. M. Zhang, and C. N. Chuah, "Adaptive traffic signal control with vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1459–1471, May 2013.
- [11] J. Cheng, W. Wu, J. Cao, and K. Li, "Fuzzy group-based intersection control via vehicular networks for smart transportations," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 751–758, Apr. 2017.
- [12] R. Wunderlich, C. Liu, I. Elhanany, and T. Urbanik, "A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 536–547, Sep. 2008.
- [13] D. Zhao, Y. Dai, and Z. Zhang, "Computational intelligence in urban traffic signal control: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 485–494, Jul. 2012.
- [14] P. Mirchandani and L. Head, "A real-time traffic signal control system: Architecture, algorithms, and analysis," *Transp. Res. C, Emerg. Technol.*, vol. 9, no. 6, pp. 415–432, Dec. 2001.
- [15] A. A. Zaidi, B. Kulcsár, and H. Wymeersch, "Back-pressure traffic signal control with partial routing for urban vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2134–2143, Aug. 2016.
- [16] J. Gregoire, S. Samaranyake, and E. Frazzoli, "Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 6753–6758.
- [17] B. P. Gokulan and D. Srinivasan, "Distributed geometric fuzzy multiagent urban traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 714–727, Sep. 2010.
- [18] J. Qiao, N. Yang, and J. Gao, "Two-stage fuzzy logic controller for signalized intersection," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 1, pp. 178–184, Jan. 2011.
- [19] D. Srinivasan, M. C. Choy, and R. L. Cheu, "Neural networks for real-time traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 261–272, Sep. 2006.
- [20] K. Y. Chan, S. Khadem, T. S. Dillon, V. Palade, J. Singh, and E. Chang, "Selection of significant on-road sensor data for short-term traffic flow forecasting using the taguchi method," *IEEE Trans. Ind. Informat.*, vol. 8, no. 2, pp. 255–266, May 2012.
- [21] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," in *Proc. IEEE Veh. Technol. Conf. (VTC Spring)*, May 2008, pp. 2036–2040.
- [22] J. Levinson, J. Askeland, J. Dolson, and S. Thrun, "Traffic light mapping, localization, and state detection for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 5784–5791.
- [23] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Res. A, Policy Pract.*, vol. 77, pp. 167–181, Jul. 2015.
- [24] R. W. Wolcott and R. M. Eustice, "Visual localization within LIDAR maps for automated urban driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 176–183.
- [25] J. Gregoire, X. Qian, E. Frazzoli, A. D. L. Fortelle, and T. Wongpiromsarn, "Capacity-aware backpressure traffic signal control," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 164–173, Jun. 2015.
- [26] D. Ma, J. Xiao, X. Song, X. Ma, and S. Jin, "A back-pressure-based model with fixed phase sequences for traffic signal optimization under oversaturated networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 29, 2020, doi: 10.1109/TITS.2020.2987917.
- [27] L. Li and S. E. Jabari, "Position weighted backpressure intersection control for urban networks," *Transp. Res. B, Methodol.*, vol. 128, pp. 435–461, Oct. 2019.
- [28] L. Tassioulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.
- [29] J. W. Moon and L. Moser, "On cliques in graphs," *Isr. J. Math.*, vol. 3, no. 1, pp. 23–28, 1965.
- [30] S. Sakai, M. Togasaki, and K. Yamazaki, "A note on greedy algorithms for the maximum weighted independent set problem," *Discrete Appl. Math.*, vol. 126, nos. 2–3, pp. 313–322, 2003.
- [31] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou, "On generating all maximal independent sets," *Inf. Process. Lett.*, vol. 27, no. 3, pp. 119–123, Mar. 1988.
- [32] D. Eppstein, "Small maximal independent sets and faster exact graph coloring," in *Graph Algorithms and Applications 4*. Singapore: World Scientific, 2006, pp. 131–140.
- [33] Y. Zhao, Y. Zhang, T. Yu, T. Liu, X. Wang, X. Tian, and X. Liu, "CityDrive: A map-generating and speed-optimizing driving system," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2014, pp. 1986–1994.
- [34] Y. J. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Optimal control and coordination of connected and automated vehicles at urban traffic intersections," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 6227–6232.
- [35] J. Gregoire and E. Frazzoli, "Hybrid centralized/distributed autonomous intersection control: Using a job scheduler as a planner and inheriting its efficiency guarantees," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 2549–2554.



wireless *ad-hoc* networks and intelligent transportation systems.



SEN BAI received the B.S. degree from the School of Software Engineering, Huazhong University of Science and Technology, China, in 2008, the M.S. degree from the Department of Computer Science, Jilin University, China, in 2013, and the Ph.D. degree from Jilin University, in 2016. He held a postdoctoral position with the School of Software, Tsinghua University, under the supervision of Prof. Y. Liu, until 2019. He is currently working with Jilin University. His research interests include

XIN BAI received the B.S. degree from the Department of Mathematics, Shanghai Jiao Tong University, China, in 2008, the M.S. degree from the Department of Computer Science, Jilin University, China, in 2013, and the Ph.D. degree from Jilin University, in 2018. He is currently working with Huawei Technologies Company Ltd. His research interest includes intelligent transportation systems.

...