www.ietdl.org

Published in IET Intelligent Transport Systems
Received on 28th June 2010
Revised on 16th January 2012
doi: 10.1049/iet-its.2010.0207

ISSN 1751-956X

Vehicular *ad hoc* networks enabled traffic controller for removing traffic lights in isolated intersections based on integer linear programming

H. Ghaffarian M. Fathy M. Soryani

Computer Engineering School, Iran University of Science and Technology, Tehran, Narmak, Iran
E-mail: Ghaffarian@iust.ac.ir

Abstract: Traffic management, especially in intersections, is an important part of intelligent transportation systems (ITS). In this study, a new traffic controller is proposed which removes traffic lights in intersections. The heart of the proposed controller is a simple integer linear program (ILP) code. This program manages safe driving patterns crossing a junction while it tries to maximise number of passed vehicles across the intersection. The demanded data are prepared by vehicular *ad hoc* networks (VANET). The proposed safe driving pattern assures the safety of moving vehicles and avoiding any accident. Simulation results show that the proposed controller performs efficiently. The average queue lengths and also tolerated delay by vehicles are far better than traditional approaches. Also, for different probabilities of output selection, the proposed approach remains stable yet.

1 Introduction

Traffic light controllers are inseparable parts of daily life. Looking into the history indicates that the use of traffic control devices certainly began before the dawn of recorded history [1]. Milestones, a form of traffic control devices, were the first devices that were used by the ancient Roman road builders. The Romans used these devices to provide directions for travellers. The first signal lights were used 2600 years ago as a guide returning fishermen to their tribes. In late 19th century and beginning of 20th century, Britain started to use the first generation of current traffic lights which were semaphores with coloured disks [2]. After that, traffic lights became public and were established all around the world.

After World War I, people, especially in the US, used automobiles as the main part of the common transportation system. This was the start point of which roads became full of cars. Therefore the usual traffic lights failed to prepare good support. Intelligent Transportation Systems (ITS) was the solution. 'The term Intelligent Transport Systems (ITS) refers to information and communication technology (applied to transport infrastructure and vehicles) that improve transport outcomes such as transport safety, transport productivity, travel reliability, informed travel choices, social equity, environmental performance and network operation resilience' (http://en.wikipedia.org/wiki/Intelligent_transportation_system). ITS propose to add information and communication technologies to transport systems and vehicles to make them intelligent. It can reduce transportation time, decrease fuel consumption and improve safety. As an example, Abdel-Aty and Pemmmanaboina [3] have combined ITS traffic data with archived weather data of I-4 region in Central Florida. They have developed a crash prediction model for rainy weather in that area.

Traffic flow modelling can improve the quality of ITS applications. One of the best and basic models prepared for three-dimensional traffic flow presentation belongs to Makigami *et al.* [4]. Their model shows the impact of junctions on traffic flow with respect to the space, time and cumulative number of vehicles. An exhaustive traffic flow prediction was made in the PATH program at university of California, Berkeley in 1999 [5]. According to similar models, Ni [6] determined and defined flow, density and space mean speed as traffic-flow characteristics for ITS applications.

To obey compulsory timing of the current light traffics, traffic flows in junctions become slow. Therefore in some places like military zones, traffic lights are not used. ITS can help and solve safety problems in such places. Vehicular *ad hoc* networks (VANET) are one of the best tools for this purpose. Although, the main motivation behind VANET is safety among vehicles along the roads; it also enjoys other strong motivations such as cooperation with infrastructures. The current trends of VANET are reviewed by Abdalla *et al.* [7]. The relationship between VANET and intersections is bidirectional. While using VANET, traffic controllers can be aware of traffic parameters on the streets and act based on these data; the mobility models for VANETs are affected by the intersection management policies [8].

In this paper, a simple intelligent integer linear programming (ILP) traffic controller, named 'simple intelligent ILP-based traffic controller (SI^{2}BTC)', is proposed which could manage vehicles in intersections. It can merge with VANET or any similar system, gathering vehicles information and scheduling them based on safety, priority and time constraints. We have analysed the stability of the proposed approach. Also we have simulate(d) our model and compared it with common traffic light controllers. Simulation results confirm the performance of SI^{2}BTC.

Using linear programming (LP) in ITS and public transportation has a long history. It is used in bus driving, airplane and ship scheduling widely and efficiently. TRACS II is a popular example which has been proposed for scheduling of UK public transportation and especially for bus scheduling [9]. Alternatively, Zhao proposed ZEST, a heuristic approach, for such scheduling problems [10].

Considering LP/ILP history, there is no proposed ILP algorithm for traffic controllers in isolated intersections. The only work in this area is made by Lin and Wang [11]. They propose an ILP formulation for optimising traffic signalling in an arterial with few intersections. For this, they divide the total related area into some cells and focus on parameters related to cell transmission for optimisation. Mirchandani and Zou [12] propose an $M/G/1$ queuing model for adaptive traffic signal controllers. Their model predicts the behaviour of the controller in the green phase distribution, or the conditional busy period. Wunderlich *et al.* [13] propose a quality of service aware signalling scheduling algorithm for isolated intersections. They believe that optimising the performance of isolated junctions contributes to improving the overall performance of networks. Their proposed approach, namely longest queue first–maximal weight matching (LQF–MWM), is based on the MWM and LQF algorithms [14] in network switches. It tries to serve the longest queues first while achieving the maximum weight of transferring vehicles across the intersection. We have simulated LQF–MWM and compared its performance with our controller.

Teodorovic *et al.* [15] present an intelligent isolated intersection control system which it has a two-step process that develops fuzzy rules. In the first step, the best sequence of steps are extracted and selected by genetic algorithm. In the second step, learning the best strategies, specific rules are derived from the set of chosen examples. Peirce and Webb [16] propose the microprocessor optimised vehicle actuation (MOVA) which is a self-optimising system. MOVA is designed to reduce delays and stops in order to maximise vehicle throughput. Dresner and Stone [17–19] propose a multi-agent-based controller for intersections. Their idea is based on a reservation area, called patch. Each movement of vehicles into a new patch in the intersection area must be granted first to avoid collision. For this, the traffic controller calculates the next position of the vehicle according to the current and future position information of vehicles in the intersection area. This concept was followed by Mehani and de La Fortelle [20].

The remainder of this paper is organised as follows. Section 2 consists of basic assumptions and preliminaries. In Section 3, the proposed simple intelligent ILP-based scheduling method is introduced. This method is evaluated in Section 4 and finally the paper is concluded in Section 5.

2 Preliminaries

In this section, some basic preliminaries are presented which are the bases for the proposed ILP approach.

2.1 Intersection configuration

The intersection under consideration is shown in Fig. 1. This junction consists of four input approaches and four output approaches. This is a common model in four-way junctions. We have numbered the Input/Output approaches in two ways. The absolute numbers are 0 to 3 counter clockwise. The relative values (just for output approaches) are defined according to the path of the vehicle. A vehicle could turn right (defined by 1), go ahead (defined by 2) or turn left (defined by 3). For example, if a vehicle wants to turn left, regardless of to its current position, its output lane is 3. Turning back in the intersection is prohibited. Using this method, output lanes can be numbered dynamically and independent from the structure of the intersection as presented in [13]. The output lane parameter has arbitrarily been chosen. According to this, variable x_{ikj} ($x_{ikj} \in \{0,1\}$, $i \neq j$) represents kth vehicle from ith input wants to pass into jth output relative while turning back is prohibited.

2.2 Safe driving pattern

Although using traffic lights is the basic method towards collision free movements, it is impossible to use it in some places like construction sites or military zones. In such places, driver must drive in blind crossing situations [21]. In such conditions, infrastructured VANET can be helpful. When vehicles enter into the range of infrastructure, they can join into a group and communicate with the other members of the group via infrastructure. Along this communication, each vehicle could transfer its basic data such as its path, destination, type and speed to a central controller placed in the infrastructure or another vehicle.

Central controller in the infrastructure can do jobs such as scheduling newly arrived vehicles. To guarantee safe driving in scheduling, Li *et al.* [21] define a concept, called 'safe driving pattern', which prepares a platform for simultaneous safe driving of more than one vehicle in the intersection. Suppose vehicles p and q are waiting to enter an intersection. The proposed safe driving pattern has three main rules as follows.

1. Vehicles p and q must not start from the same lane.
2. Vehicles p and q must not enter to the same lane.
3. The starting and ending points of vehicles p and q must be in the left position of each other.

According to the intersection configuration proposed in Section 2.1, here a new safe driving pattern is proposed. This pattern has the following rules:

1. Each pair of the selected vehicles intending to pass the junction must not have the same start point. In other word,

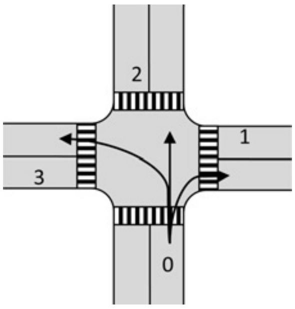

Fig. 1 *Used intersection model*

x_{ikj} and $x_{ik'j'}$ $(k \neq k')$ cannot pass the junction at the same time.
2. Each pair of the selected vehicles intending to pass the junction must not have the same end point. In other word, x_{ikj} and $x_{i'k'j}$ $(i \neq i')$ are not allowed at the same time.
3. Each pair of the selected vehicles to pass the junction must not cut each others' paths along the intersection.

The first two defined rules are similar to the rules defined by Li *et al.* [21]. To remove ambiguity in the third rule of [21], we propose a circular numbering approach instead of finding left position. Fig. 2 shows two samples of safe and unsafe driving. Fig. 2*a* depicts an admissible and safe driving according to the defined safe pattern and Fig. 2*b* depicts a prohibited one.

Selecting numbers 0–3 instead of 1–4 for numbering the approaches can help to reduce the complexity of implementation of the controller. In most of common programming languages, there is a modulus function (MOD). The result of X MOD Y is the remainder left over after dividing X by Y. Each vehicle should put sum of its relative output approach number and its absolute input approach number on the left-hand side of the MOD function. We use a four-way intersection; therefore on the right-hand side of the MOD function, we put number 4. The result, a number between 0 and 3, is the absolute approach number in the intersection.

To find forbidden paths, which cut other paths, a simple approach is defined here. Suppose the area of the intersection is divided into four sub-areas as shown in Fig. 3. If a vehicle in the input approach i wants to turn right, it uses sub-area i. If it wants to go straight ahead, it must use sub-area i and $(i+1)$ mod 4. To turn left, it needs to pass across the sub-areas i, $(i+1)$ mod 4 and $(i+2)$ mod 4. If one sub-area is occupied by a vehicle, for safe driving, other vehicles cannot enter to it. The example in Fig. 2*a* is allowed, because the vehicle in input 0 passes through sub-areas 0, 1 and 2. The remainder sub-area (sub-area number 3) is allocated to the vehicle in input 3. However in Fig. 2*b*, vehicles in inputs 0 and 1 want to capture sub-area 1 simultaneously which is not permitted. The second rule of the safety driving pattern can easily be proved, using these sub-areas. The proposed sub-areas in this paper are different from the patch concept introduced in [17–20]. Although our defined sub-areas are fixed in the intersection area, the patch of each vehicle is a limited area related to the current and next position of it.

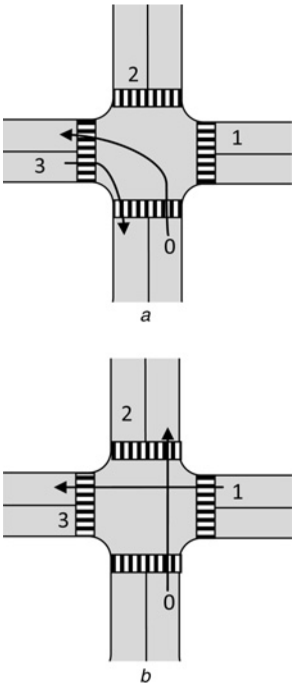

Fig. 2 *Examples of safe and unsafe driving patterns*

a Example of a safe driving pattern: vehicle in approach 0 wants to turn left, whereas vehicle in approach 3 wants to turn right
b Example of an unsafe driving pattern: vehicles in approaches 0 and 1 want to go straight ahead

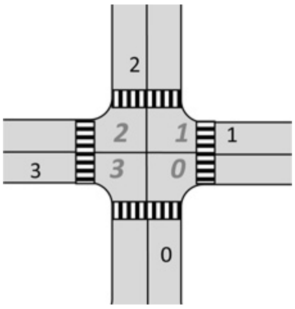

Fig. 3 *Dividing the intersection area into four sub-areas to find the forbidden paths*

2.3 Data constraints

The feasibility of this approach is depending on the following constraints:

1. Each vehicle is well equipped to a global positioning system and knows its position and direction. Also, it should know its path after the junction.
2. Feasibility of VANET approaches is depended onto equipped vehicles. Therefore we suppose that all vehicles are well equipped with completely reliable Vehicle to Infrastructure (V2I) communication systems. With this system, vehicles can communicate with the Vehicle Infrastructure Integration (VII) system and intelligent traffic controller. Any unreliability or failure in the V2I communication system can affect the performance of the controller.
3. The most critical point in intersection management is making decision about the output approach in an intersection. In this paper, we suppose that each driver must decide and announce its exit direction before reaching to the junction. The proposed scheduler will act only based on the gathered information from the vehicles in the head of each approach. Therefore drivers have enough time to make decision about their next trip paths, before reaching to the head of queues. Therefore the submitted data have good degree of reliability. However, after submitting, drivers could not change their decisions in the current round of scheduling. Any changes in their decisions can be applied in the next round of scheduling. To ensure of submitting the exit approach numbers by drivers, road side unites of infrastructured VANET can broadcast special warning messages to ask divers about their decision.

Without losing performance, the above equipments guarantees that there is no need for extra sensors and detectors across the streets. However, they can be used as complementary systems to increase the reliability of the system. The intelligent traffic controller could make a snapshot of the current status of the queue in each direction, using the position information gathered from vehicles.

3 Proposed traffic scheduling approach

In this part the simple ILP scheduler for scheduling vehicles in an intersection is presented. As mentioned previously our studies show that there is not any proposed real-time LP/ILP scheduling approach for safety driving in intersections. The main reason of this is the time complexity of LP/ILP programs. Without good constraints, it is impossible to solve the problem in a bounded time. Especially, when the load of the traffic in the intersection increases, this situation becomes worse. Our proposed ILP algorithm only acts on the four vehicles standing in the head of queues to reduce the time complexity of problem solving. Also, in two-phase intersection controllers, some vehicles will cut paths of other vehicles which is not safe. Therefore we choose a four phase instead of two.

3.1 SI^{2}BTC traffic scheduler

Consider the simple intersection proposed in the previous section. There is only one lane in each approach of this intersection; hence the parameter k is removed (x_{ikj} is simplified to x_{ij}, where $x_{ij} \in \{0,1\}$, $i \neq j$) and the first rule of the safe driving pattern is achieved. The defined ILP algorithm is as follows

$$\text{Max} \quad \sum_{i=0}^{3}\sum_{j=1}^{3} c_{ij}x_{ij} \qquad (1)$$

subject to

$$\sum_{i=0}^{3} x_{ij} \leq 1 \quad \forall j \qquad (2)$$

$$x_{((i+1)\text{mod }4)j} \leq 1 - x_{i2} \quad \forall i, j \qquad (3)$$

$$x_{((i+2)\text{mod }4)3} \leq 1 - x_{i2} \quad \forall i \qquad (4)$$

$$x_{((i+3)\text{mod }4)2} + x_{((i+3)\text{mod }4)3} \leq 1 - x_{i2} \quad \forall i \qquad (5)$$

$$x_{((i+2)\text{mod }4)j} + x_{((i+1)\text{mod }4)j} \leq 2(1 - x_{i3}) \quad \forall i, j \qquad (6)$$

$$x_{((i+3)\text{mod }4)2} + x_{((i+3)\text{mod }4)3} \leq 1 - x_{i3} \quad \forall i \qquad (7)$$

where c_{ij} is the weight of x_{ij} and can be defined as the priority or the waiting delay of a vehicle or the length of queue in the vehicle's approach. Using this variable, it is possible to consider the tolerated delay by vehicles standing in the intersection as a quality of service parameter. The ILP algorithm will try to maximise the number of passed vehicles across the junction, whereas the constraint conditions assure about the safety driving. The first constraint (2) assures the second rule in the safety driving pattern. The remainders are about the third rule in the safety pattern.

The third rule in the safety pattern says that the paths of the vehicles must not cut each other. To ensure this rule some input approaches must be blocked according to the selected output approach. As mentioned before, each vehicle has three choices to pass across the intersection. When it selects to turn right, it does not block any other input approaches. The only constraint in this situation is about the output blocking which is handled by the first constraint (2). If a vehicle selects to go straight ahead in the intersection, it causes some limitations for other input lanes. If this vehicle is located in the input i, it blocks the next input in counterclockwise manner. The blocked input approach is input approach number $(i+1)$ mod 4. This approach is blocked, because the sub-area $(i+1)$ mod 4 is not free and captured by the selected vehicle of input approach i. The second constraint (3) shows this condition. In addition, sub-area i is not free (4); therefore the vehicle in the input approach $(i+2)$ mod 4 cannot turn left. Also, the vehicle in the input approach $(i+3)$ mod 4 is prohibited to go straight ahead or turn left. As both these selections will pass the sub-areas, which are captured by the vehicle in input i, therefore it is forbidden. Constraint (5) presents this situation.

When the vehicle in the input approach i tries to turn left, it blocks both input approaches $(i+1)$ mod 4 and $(i+2)$ mod 4, because sub-areas i, $(i+1)$ mod 4 and $(i+2)$ mod 4 are captured by the input i. Therefore in (6) a factor of 2 is added on the right-hand side of the equation to show blocking of 2 other approaches if the scheduler selects input approach i. Also, selecting input i, forces input $(i+3)$ mod 4 to restrict its selection domain only to turn right. This happens because the sub-area $(i+3)$ mod 4 is the remaining free sub-area. This constraint is shown in last constraint row (7).

The output of the proposed ILP controller at time t is represented by a 'matching matrix' $M = |M_{ij}(t)|$, $1 \leq i \leq 4$, $1 \leq j \leq 4$; whose binary elements $M_{ij}(t) = 1$, if and only if the vehicle in input approach i is selected by the ILP controller to pass to the output approach j; otherwise $M_{ij}(t) = 0$. Based on the turning back prohibition rule, $M_{ii}(t) = 0$ $\forall i$.

The proposed controller is named SI^{2}BTC. The scheduler becomes active, when at least one vehicle appears in the intersection area and communication range of V2I and VII.

3.2 Stability point of the algorithm

Stability in a queue indicates condition in which the size of the queue be bounded. In this part, a lower bound for stability in the queues of approaches, under running of SI^{2}BTC algorithm is addressed. Let's define 'passing interval' as the time to pass an intersection. To satisfy safety conditions in crossing the junction, we have to use a minimum interval for vehicles' entrance; otherwise there is no guarantee that vehicles do not colloid while crossing. This lower bound is prepared based on passing interval. The SI^{2}BTC determines scheduling of passing the intersection once per each passing interval.

Theorem 1: The average aggregate departed rate of the proposed ILP approach is 2.1 vehicles per passing interval.

Proof: As mentioned before, each vehicle has three possible choices for output. Therefore the total number of possible selections between outputs is $3^4 = 81$. In addition, the number of selected vehicles to pass the intersection in each passing interval is ranged between [1..4]. Results of analysing all 81 cases are shown in Tables 1 and 2.

doi: 10.1049/iet-its.2010.0207

Table 1 Detailed maximum departed rates in different cases

Case	Max	Case	Max	Case	Max	Case	Max	Case	Max	Case	Max	Case	Max	Case	Max	Case	Max
RRRR	4	RRRS	3	RRRL	3	RRSR	3	RRSS	3	RRSL	3	RRLR	3	RRLS	2	RRLL	2
RSRR	3	RSRS	2	RSRL	2	RSSR	3	RSSS	2	RSSL	2	RSLR	3	RSLS	2	RSLL	2
RLRR	3	RLRS	2	RLRL	2	RLSR	2	RLSS	2	RLSL	2	RLLR	2	RLLS	2	RLLL	2
SRRR	3	SRRS	3	SRRL	2	SRSR	2	SRSS	2	SRSL	2	SRLR	2	SRLS	2	SRLL	2
SSRR	3	SSRS	2	SSRL	2	SSSR	2	SSSS	2	SSSL	2	SSLR	2	SSLS	2	SSLL	1
SLRR	3	SLRS	2	SLRL	2	SLSR	2	SLSS	2	SLSL	2	SLLR	2	SLLS	1	SLLL	1
LRRR	3	LRRS	3	LRRL	2	LRSR	2	LRSS	2	LRSL	2	LRLR	2	LRLS	2	LRLL	2
LSRR	2	LSRS	2	LSRL	2	LSSR	2	LSSS	2	LSSL	1	LSLR	2	LSLS	2	LSLL	1
LLRR	2	LLRS	2	LLRL	2	LLSR	2	LLSS	1	LLSL	1	LLLR	1	LLLS	1	LLLL	1

Case: possible case to choose output approach; Max: the maximum number of allowable concurrent passes in one passing interval; R: turn right L: turn left; S: straight ahead

Table 2 Overall maximum departed rates in different cases

Maximum departed rate	1	2	3	4
no. of cases	10	54	16	1
%	0.12	0.67	0.2	0.01

According to Table 2, the average departed rate is calculated as follows

$$\begin{aligned}\text{Average departed rate} &= 1 \times 0.12 + 2 \times 0.67 + 3 \times 0.2 \\ &\quad + 4 \times 0.01 \\ &= 2.1 \text{ veh/passing interval}\end{aligned}$$

It means that on average, most of the times the ILP controller will select two or more vehicles for safe passing through the intersection.

Theorem 2: The queues of the approaches will be stable if aggregate number of arrival vehicles into intersection is less than two vehicles per passing interval.

Proof: The queues are stable, if in a period of time Δt, the difference of the arrival vehicles and departed vehicles becomes zero or less than zero as follows

$$\begin{aligned}\sum_{i=1}^{4} X_i(t+\Delta t) - \sum_{i=1}^{4} X_i(t) &\le \sum_{i=1}^{4}\sum_{k=0}^{\Delta t-1} \lambda_i(t+k) \\ &\quad - \sum_{k=0}^{\Delta t-1} (\text{trace}(\boldsymbol{M}(t+k)^{\text{T}}\boldsymbol{M}(t+k)) \qquad (8)\end{aligned}$$

where $X_i(t)$ is the length of queue i (number of vehicles in queue i) at time t, $\lambda_i(t)$ is the number of newly arrived vehicles to queue i at time t and trace$(M(t)^{\text{T}}\boldsymbol{M}(t))$ is the number of departed vehicles at time t using the data of matching matrix $\boldsymbol{M}$. Queuing systems are stable iff lengths of queues do not grow. The left-hand side of this equation is related to the changes in the length of queues in Δt and the right-hand side of it is related to the difference of arrival rate and departure rate of the intersection. The scheduler must maximise the departure rate to keep the lengths of queues in a stable situation. Selecting 2.1 as the average aggregate number of passing vehicles in each passing interval from Theorem 1, results in

$$\sum_{k=0}^{\Delta t-1} (\text{trace}(\boldsymbol{M}(t+k)^{\text{T}}\boldsymbol{M}(t+k)) \ge 2.1\Delta t > 2\Delta t \qquad (9)$$

Using formulas (8) and (9), it can be said that the minimum stability point of the algorithm is reached when the following condition is satisfied

$$\sum_{i=1}^{4} X_i(t+\Delta t) - \sum_{i=1}^{4} X_i(t) \le \sum_{i=1}^{4}\sum_{k=0}^{\Delta t-1} \lambda_i(t+k) - 2\Delta t \qquad (10)$$

With respect to the stability definition and (10), it can be concluded that in each passing interval, if two or less vehicles join the queues behind the intersections on average, the algorithm remains stable.

3.3 Extending the scheduler into new spaces

The proposed controller is designed only for the defined case. For other cases, it needs some modifications to cover new created states. Especially in multi-lane approaches, state space of the problem is increased and need some modifications in the current constraints and formulas. These conditions can be more complex when approaches do not have same lanes for inputs and outputs. However, such extensions did not cover in this paper.

4 Evaluation of the proposed method

To evaluate the proposed controller, MATLAB R2008a was used. The simulation parameters are as follows: the average distance between following vehicles is 4 m which consists of the average vehicle length and the average distance from the vehicle in front of it. This value is acceptable because most of current vehicles have lengths between 2.5 and 4 m. Using this value the capacity of intersection can be calculated. For some kinds of vehicles such as buses and trucks, this average length is increased to 8 m. The width of junction is 8 m. Owing to the limited width of the intersection, the effect of acceleration/decelerating is venial. Also, we supposed that vehicles cross the intersection with average crossing speed of 2 m/s (7.2 km/h); therefore each vehicle needs 4 s to pass the intersection (passing interval). It is clear that if we increase this speed, the result is decreasing in the passing interval and increasing the capacity of junction and approaches. Owing to pipeline situation in traditional traffic lights, vehicles can enter into the intersection in turn. Therefore with respect to the defined size of vehicles and the width of the junction, each vehicle only needs 2 s to pass the junction (two vehicles are in the intersection simultaneously).

The fixed-time traffic light controller has three different values as its green light time: 20, 24 and 28 s. Also there are 5 additional seconds as the yellow (or amber) light time (it can be supposed that the yellow light time is 4 s and 1 extra second is used as starting time of vehicles when the light is turned green). The vehicles cannot enter into the intersection when the light is yellow. The traffic lights of approaches are turned green one by one and counterclockwise. According to $M/M/1$ queuing model, for a server with service time S, the maximum arrival rate is

$$\lambda \leq (1/S) \qquad (11)$$

Supposing the intersection as a single server, using the above green and yellow light times, the arrival rate of the 20 s green light time (FxT-20) is 0.4 vehicles per second (1 vehicle per every 2.5 s), the arrival rate of the 24 s green light time (FxT-24) is 0.41 vehicles per second (1 vehicle per every 2.42 s) and the arrival rate of the 28 s green light time (FxT-28) is 0.42 vehicles per second (1 vehicle per every 2.38 s). The intersection has four approaches; so at most, each approach can serve 1 vehicle per every 9.52 s (the maximum arrival rate of vehicles into each approach is 0.105 vehicles per second).

To give variation to the simulations, three types of vehicles are introduced. Vehicle type 1 consists of common personal vehicles. Vehicle type 2 consists of public vehicles such as buses and trucks. This category needs 4 s to pass the junction (double in passing time). The last category is vehicle type 3 which consists of emergency vehicles such as police cars and ambulances. The arrival rate of vehicle type 1 is one vehicle for every 9.6 s; for vehicle type 2, one vehicle for every 2 min and for vehicle type 3, one vehicle for every 30 min. Vehicles are entered into the approaches with Poisson distribution. Therefore the arrival rate of 0.114 vehicle per second (one vehicle per 8.77 s) is chosen as the maximum arrival rate into each input approach (one vehicle per 2.19 s is entered into the intersection in the simulations which is close to the stability point of the proposed algorithm in Theorem 2).

In the simulations, the SI^{2}BTC algorithm is used in three different forms. In the first form, there is not any differentiation between vehicles (namely SI^{2}BTC without priority or NP-SI^{2}BTC). In the second form, the weight of vehicle type 1 is set to 1, the weight of vehicle type 2 is set to 2 and the weight of vehicle type 3 is set to 3 (namely SI^{2}BTC with priority or P-SI^{2}BTC). In the last form, the length of queue which vehicle belongs to it is selected as input weight of controller (namely queue length aware SI^{2}BTC or QL-SI^{2}BTC). In addition to these, LQF–MWM is also implemented.

Simulation runtime is 4500 s, in which the first 500 s and the last 400 s are used as setup and final times. The results of these 900 s are removed in the final result analysis. Consequently, the main simulation runtime is 3600 s or 1 h. The simulations are repeated ten times, each loaded from 5% till 100% of the possible arrival rates and the average results of all repetitions are shown here. Fig. 4 depicts the average number of vehicles served in different simulations. Based on the given parameters, 92% of the arrived vehicles are from type 1, 7.5% from type 2 and 0.5% from type 3. Therefore the weighted average of passing interval will be 4.3 s.

4.1 Equal route selection probabilities

In the first series of simulations, the output approach of each vehicle is selected randomly with equal probabilities, from each of the three possible output options. Fig. 5 depicts the average tolerated delay of vehicles for different loads. As shown in this figure, below 85% of relative load,

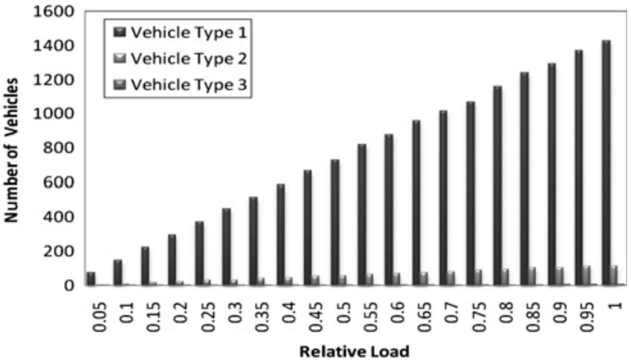

Fig. 4 *Number of vehicles in different simulations*

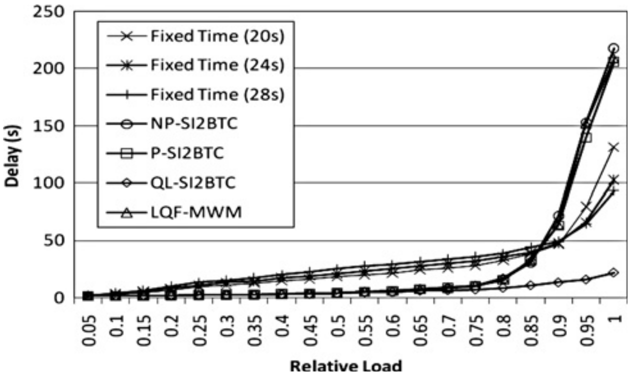

Fig. 5 *Average tolerated delay by vehicles under different relative loads and equal route selection probabilities*

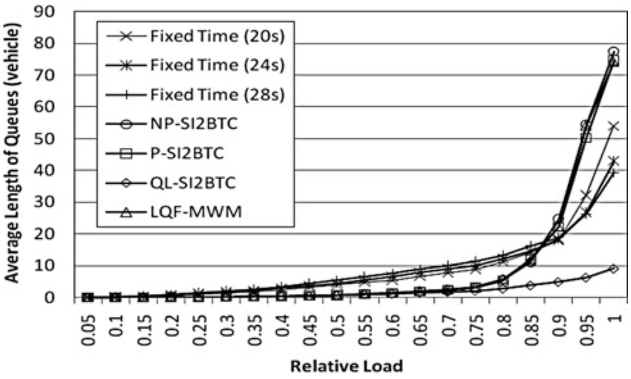

Fig. 6 *Average length of queues behind the intersection approaches and equal route selection probabilities*

doi: 10.1049/iet-its.2010.0207

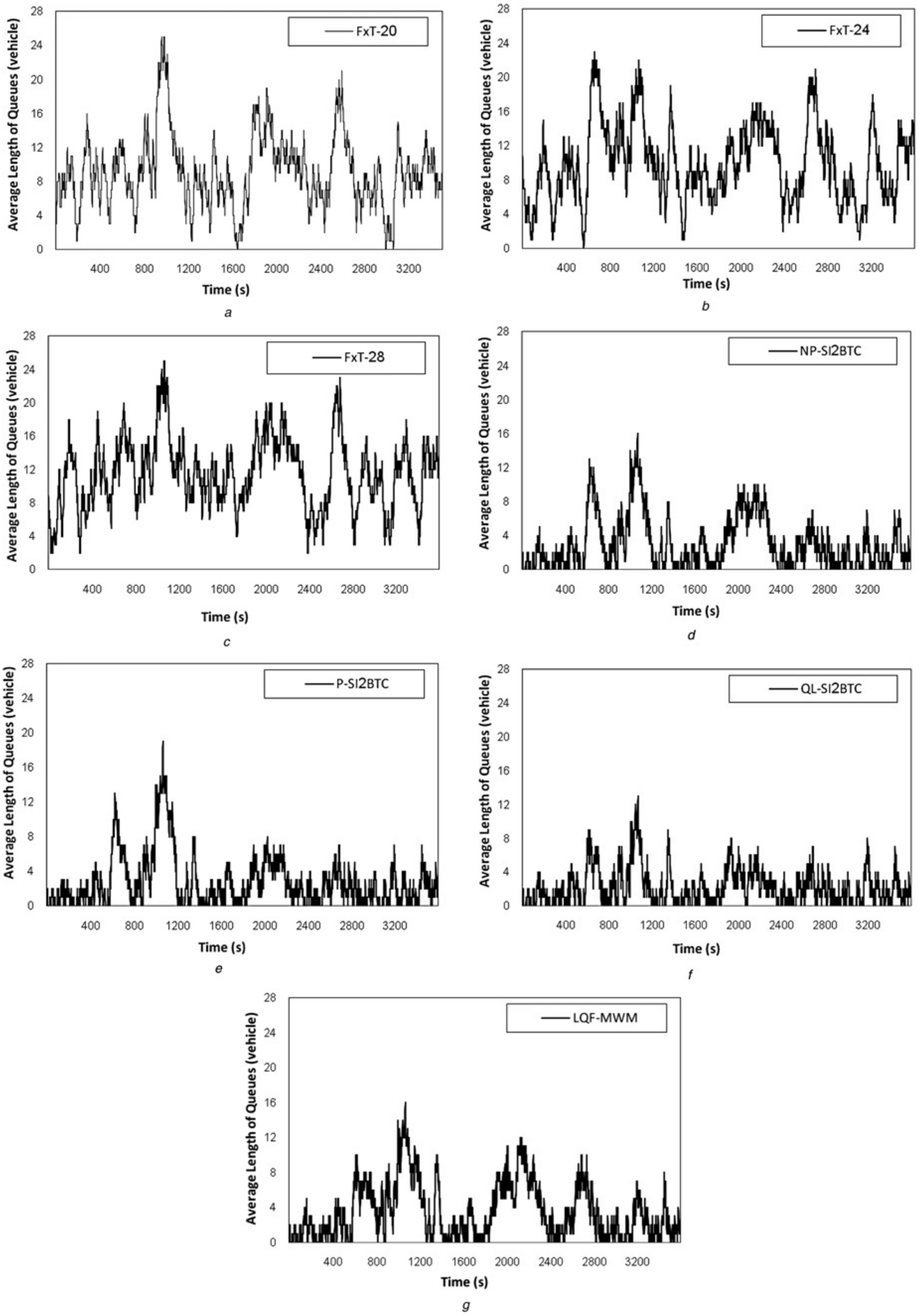

Fig. 7 *Average queue length of different traffic management methods for 75% load*

a Average queues length variations in the 20 s fixed time controller
b Average queues length variations in the 24 s fixed time controller
c Average queues length variations in the 28 s fixed time controller
d Average queues length variations in NP-SI^{2}BTC controller
e Average queues length variations in P-SI^{2}BTC controller
f Average queues length variations in QL-SI^{2}BTC controller
g Average queues length variations in LQF–MWM controller

SI^{2}BTC family and LQF–MWM have the best delays. However for 90% relative loads and especially 95 and 100% of the loads, it can be seen that only QL-SI^{2}BTC acts properly. It shows that the stability point of QL-SI^{2}BTC is 100% and the stability point of NP-SI^{2}BTC, P-SI^{2}BTC and LQF–MWM is less than 90% of the maximum load which is in agreement with Theorem 2. On average, the delays in SI^{2}BTC with priority are somewhat better than LQF–MWM and SI^{2}BTC without priority.

For the fixed time traffic light controllers, it can be seen that the average tolerated delay by vehicles in the shorter fixed times is more than the tolerated delays by longer fixed times. This happened because in the shorter fixed time controllers, the time of yellow light with respect to the time of green light is longer. In addition, after 90% of the load, the tolerated delays for the fixed time traffic lights are grew up exponentially (stability point of the simulated fixed time controllers).

Another important factor that shows the performance degree of a traffic controller is the average length of queues in each traffic control method. Fig. 6 depicts this for different traffic loads. As it can be seen until 85% of the relative load, the results of SI^{2}BTC family and LQF–MWM are the best and after that, again, only QL-SI^{2}BTC acts very well. It is interesting that always NP-SI^{2}BTC, P-SI^{2}BTC and LQF–MWM algorithms follow each other and their results are similar. However, none of them can follow QL-SI^{2}BTC after 80% of load.

As an example and to clearly show the variation of the length of queues along the 1 h simulation, the 75% load is selected. Fig. 7 depicts the average length of queues for different methods along the 75% of load simulation. As this figure shows, the tolerance of the length of queues in SI^{2}BTC family is far better than the tolerance of the length of queues in the fixed time traffic controllers. Even their results are better than the achieved results of LQF–MWM controller. The queue length aware SI^{2}BTC scheme can handle average length of queues very well. The tolerance of the average queue length in this algorithm placed in lowest condition. Also among the fixed time controllers, it can be seen that the tolerance of the shorter fixed times is better than the longer ones.

4.2 Non-equal route selection probabilities

Another interesting traffic case study in intersections is related to the time when selected output approaches do not have equal probabilities. Alike [13], we suppose an intersection where in each input approach, 70% of drivers are interested to go straight ahead, 20% turn right and 10% turn left. Other simulation parameters remain as prior. Figs. 8 and 9 show the average tolerated delay and average queues' length, respectively. As can be seen in these figures, under non-equal route selection probabilities, only LQ-SI^{2}BTC has better performance than the common fixed time schedulers, owing to its greedy reaction to service the long queues first. When the arrival rate into an output increases, it becomes a bottleneck and its related queues become long. Therefore LQ-SI^{2}BTC leads to solve this problem. Also, the average results of reaction of fixed time schedulers do not change which is very good. Theorem 2 in [13] says the scheduling and length of queues will have stable conditions if and only if the output rate into each output approach is less than $C/3$, where C is the total capacity of the intersection. In heavy load conditions, the probability of violating such condition is high. As shown in [22], in the priority networks, some instabilities in measures are plausible (see simulation results of [13]). Here, a similar problem occurs in the output results of simulations with relative loads greater than 0.8 which is clear in Figs. 8 and 9 in results of NP-SI^{2}BTC, P-SI^{2}BTC and LQF–MWM methods. Although, in different relative loads ranged between 0.8 and 1, occasionally the best rank of performance between these three algorithms is changed; using the average of ten different simulation runs' results as final results, hide most of such instabilities in the figures.

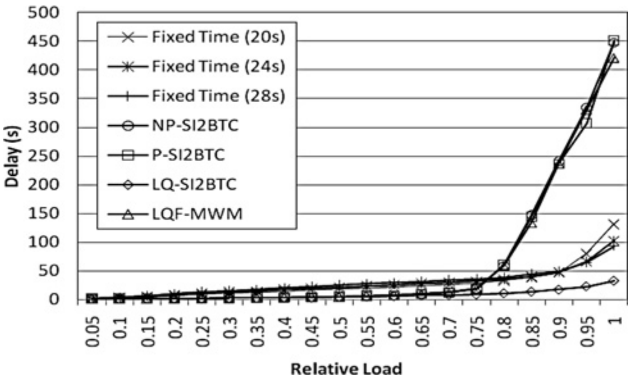

Fig. 8 *Average tolerated delay by vehicles under different relative loads for non-equal route selection probabilities*

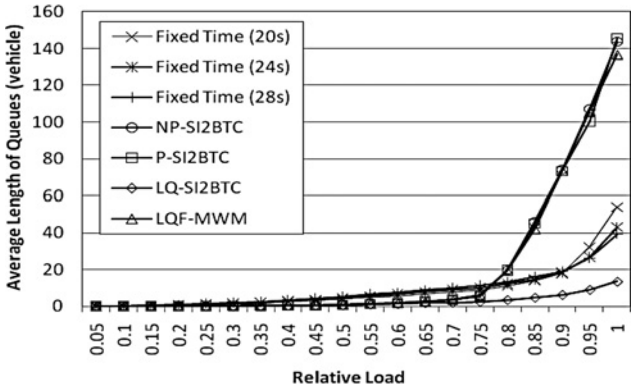

Fig. 9 *Average length of queues behind the intersection approaches for non-equal route selection probabilities*

5 Conclusion

Traditional traffic controllers have become inefficient in recent decades. ITS can efficiently improve the operation of traditional controllers. In this paper a new method, called SI^{2}BTC, is proposed which tries to remove fixed traffic light controllers from isolated intersections. The proposed controller uses a safe driving pattern to clarify safe driving and non-accident conditions when vehicles pass the intersection. Although SI^{2}BTC is aware about the vehicle safety, it tries to maximise the number of passed vehicles across the intersection. Using this, it achieves low tolerated delay per vehicle and a shorter queue length behind the intersection. The simulation results show that the proposed SI^{2}BTC algorithm acts very well below the 90% of the possible load in the intersection. However with selecting queue lengths as input weights, its performance is increased very well over this value. In addition, for unbalanced output selections, the proposed approach shows high performance. Extending the proposed method for multi-lane approaches is our program as a future work.

6 Acknowledgment

This work is supported by Iranian Research Institute for ICT.

7 References

1 Mueller, E.A.: 'Aspects of the history of traffic signals', *IEEE Trans. Veh. Technol.*, 1970, **VT-19**, (1), pp. 6–17

2 Breckenridge, F.C.: 'Fifty years of signal lighting' (Illuminating Engineering Society, 1958), pp. 311–317

3 Abdel-Aty, M.A., Pemmmanaboina, R.: 'Calibrating a real-time traffic crash-prediction model using archived weather and ITS traffic data', *IEEE Trans. Intell. Transp. Syst.*, 2006, **7**, (2), pp. 167–174

4 Makigami, Y., Newell, G.F., Rothery, R.: 'Three dimensional representation of traffic flow', *Transp. Sci.*, 1971, **5**, (3), pp. 302–313

5 Smilowitz, K.R., Daganzo, C.F.: 'Predictable of time-dependent traffic backups and other reproducible traits in experimental highway data', Working Paper No. UCB-ITS-PWP-99–5, California PATH Program, University of California, Berkeley, March 1999

6 Ni, D.: 'Determining traffic-flow characteristics by definition for application in ITS', *IEEE Trans. Intell. Transp. Syst.*, 2007, **8**, (2), pp. 181–187

7 Abdalla, G.M.T., Ali Abu-Rgheff, M., Mohammed Senouci, S.: 'Current trends in vehicular ad hoc networks', *Ubiquit. Comput. Commun. J.*, 2008, **4**, (3), pp. 29–38

8 Harri, J., Filali, F., Bonnet, C.: 'Mobility models for vehicular ad hoc networks: a survey and taxonomy', *IEEE Commun. Surv. Tutor.*, 2009, **11**, (4), pp. 19–41, Fourth Quarter

9 Fores, S., Proll, L., Wren, A.: 'TRACS II: a hybrid IP/heuristic driver scheduling system for public transport', *J. Oper. Res. Soc.*, 2002, **53**, (10), pp. 1093–1100

10 Zhao, L.: 'A heuristic method for analyzing driver scheduling problem', *IEEE Trans. Syst. Man Cybern. A, Syst. Hum.*, 2006, **36**, (3), pp. 521–531

11 Lin, W.-H., Wang, C.: 'An enhanced 0–1 mixed-integer lp formulation for traffic signal control', *IEEE Trans. Intell. Transp. Syst.*, 2004, **5**, (4), pp. 238–245

12 Mirchandani, P.B., Zou, N.: 'Queuing models for analysis of traffic adaptive signal control', *IEEE Trans. Intell. Transp. Syst.*, 2007, **8**, (1), pp. 50–59

13 Wunderlich, R., Liu, C., Elhanany, I., Urbanik, T.: 'A novel signal-scheduling algorithm with quality of service provisioning for an isolated intersection', *IEEE Trans. Intell. Transp. Syst.*, 2008, **9**, (3), pp. 536–547

14 McKeown, N., Mekkittikul, A., Anantharam, V., Walrand, J.: 'Achieving 100% throughput in an input-queued switch', *IEEE Trans. Commun.*, 1999, **47**, (8), pp. 1260–1269

15 Teodorovic, D., Lucic, P., Popovic, J., Kikuchi, S., Stanic, B.: 'Intelligent isolated intersection'. 10th IEEE Conf. on Fuzzy Systems, Melbourne, Australia, December 2001, vol. 1, pp. 276–279

16 Peirce, J.R., Webb, P.J.: 'MOVA control of isolated traffic signals-recent experience'. Third'. IEEE Conf. on Road Traffic Control, London, UK, May 1990, pp. 110–113

17 Dresner, K., Stone, P.: 'Multiagent traffic management: a reservation-based intersection control mechanism'. Third Int. Joint Conf. on Autonomous Agents and Multiagent Systems, New York, USA, July 2004, pp. 530–537

18 Dresner, K., Stone, P.: 'Multiagent traffic management: an improved intersection control mechanism'. Fourth Int. Joint Conf. on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, July 2005, pp. 471–477

19 Dresner, K., Stone, P.: 'A multiagent approach to autonomous intersection management', *J. Artif. Intell. Res.*, 2008, **31**, pp. 591–656

20 Mehani, O., de La Fortelle, A.: 'Trajectory planning in a crossroads for a fleet of driverless vehicles'. 11th Int. Conf. on Computer Aided Systems Theory, Las Palmas de Gran Canaria, Spain, February 2007

21 Li, L., Wang, F.-Y., Kim, H.: 'Cooperative driving and lane changing at blind crossings'. IEEE Intelligent Symp., Las Vegas, NV, USA, June 2005, pp. 435–440

22 Bramson, M.: 'Stability of queuing networks', in 'Lecture Notes in Mathematics' (Springer, 2008, vol. 5), pp. 169–345