

基于车联网的交通管控机制设计与验证

杨 琳 章文利 朱 沁 彭 超

(华东师范大学计算机科学与软件工程学院 上海 200062)

摘 要 近年来,汽车工业飞速发展,汽车保有量不断提高。但与此同时,其引发的道路交通拥堵问题也影响着越来越多的人。为了缓解道路交通拥堵、提高车辆出行效率,提出了一套基于车联网的交通管控机制。首先,结合固定时长轮转法和车流量响应控制法两种现有的主流交通灯控制法思想,提出半实时的交通灯控制方法。该方法中,交通灯控制器根据预测出的车流量信息,得到本路口的最优交通灯相位策略,将本路口所有车辆的等待时间最小化。其次,为了提高个体用户的出行效率,提出启发式的动态路径规划算法,该算法在 Dijkstra 算法的基础上,采用启发式的方法计算路段的权重,以此规划最优路径,使得车辆可以尽可能避开拥堵路段。为了验证提出的交通管控机制的性能,通过交通流量模拟器 SUMO 和网络模拟器 NS3 进行模拟实验。模拟实验结果表明,提出的基于车联网的交通管控机制能够有效地降低车辆的等待时间、缓解交通拥堵情况,从而改善整个道路网络的交通状况。

关键词 车联网,交通灯控制,路径规划,交通管控机制

中图分类号 TP311 文献标识码 A

Traffic Control Mechanism Design and Verification Based on VANET

YANG Lin ZHANG Wen-li ZHU Qin PENG Chao

(School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China)

Abstract In recent years, worldwide rapid development in automobile industries and growing rate of vehicle ownership have been witnessed. Consequently, the thorny issue of traffic congestion has bothered more and more people. To alleviate traffic jams as well as enhance traffic efficiency, this paper proposed a traffic control mechanism based on VANET. At first, we proposed a cooperative traffic light control mechanism for multiple intersections based on semi-real-time processing by combining the concepts of fixed-time control and traffic-response control. In our mechanism, the traffic light controller will use trajectory prediction method to predict the traffic situation of the next period, and then make the optimal decision of traffic light phase setting for the next period according to the prediction result. The aim of our mechanism is to minimize the waiting time of all the vehicles. This paper also proposed a heuristic algorithm for dynamic route planning to enhance individual's travel efficiency in traffic system. On the basis of Dijkstra algorithm, we adopted a heuristic method to calculate the weight of each road. Vehicles can avoid traffic jam as much as possible by adopting this algorithm. To verify the performance of our proposed VANET traffic control mechanism, we ran simulation experiments by combining SUMO with NS3. The simulation results demonstrate that our proposed traffic control mechanism is both effective and practical. It is able to reduce traffic load and average waiting time of vehicles, as well as release traffic jams and divert jammed vehicles, thus can improve road traffic situation of the whole transportation system.

Keywords VANET, Traffic light control, Routing planning, Traffic control mechanism

1 引言

现代科学技术的发展日新月异,汽车工业水平不断提高,全世界的汽车保有量急剧增加。以上海为例,截止到 2014 年年末,全市汽车保有量达到 272.30 万辆,比 2010 年的 170.25 万辆增长了近 60%。虽然汽车保有量一直在以较快的速度持续增加,但是道路交通设施的增量却十分有限,近 4 年间,上海城市道路面积只增长了 8%,中心城市道路里程仅增长了 3%,随之而来的是越来越严峻的交通拥堵问题。据上海

市城乡建设和交通发展研究院统计,2014 年,在上海市 68 个地面道路指数区域中,有 15 个严重拥堵区域在工作日拥堵天数超过 100 天,其中瑞金医院区域的拥堵天数高达 243 个工作日。预计在未来的几年中,上海市道路拥堵的时间将会进一步延长,拥堵里程比例也会继续增加。

为了改善交通环境、缓解交通拥堵,本文提出了一套基于车联网的交通管控机制,该机制主要由两部分组成:半实时的交通灯协同控制算法与启发式的动态路径规划算法。

半实时的交通灯协同控制算法基于固定时长轮转法和车

杨 琳(1991—),女,硕士生,主要研究方向为传感器网络,E-mail:51141500051@ecnu.cn;章文利(1992—),女,硕士生,主要研究方向为传感器网络;朱 沁(1989—),硕士,主要研究方向为传感器网络;彭 超(1978—),男,博士,副教授,CCF 会员,主要研究方向为算法与复杂性、计算机网络,E-mail:cpeng@sei.ecnu.cn(通信作者)。

流量响应控制法,主要通过相邻路口的交通灯控制器之间的协同工作来实现。首先,预先设置固定的交通灯相位时长。在每个相位时段开始时,所有交叉路口的交通灯控制器根据最新接收的附近路边单元收集到的交通数据,计算出当前时刻所在路口的交通情况。由于从邻居路口开来的车辆会影响下一个相位时段本路口的交通情况,因此令相邻的控制器彼此分享当前相位时段内的交通数据。之后,控制器将所在路口与邻居路口的交通信息相结合,预测下一个相位时段开始该路口各方向的车流量。最后,控制器根据预测出的车流量信息,得到本路口的最优交通灯相位策略,使得本路口所有车辆的整体等待时间最小化。

启发式的动态路径规划算法是在经典最短路径 Dijkstra 算法的基础上进行的改进。在现实中,道路网络的交通情况变幻莫测,仅考虑静态的路段长度无法适应复杂的交通情况。而且,比起行驶距离的长短,大多数人其实更关心到达目的地的时间。因此,本文提出一种动态路径规划算法,在计算边的权重时,以车辆通过路段的加权时间估计来代替原本静态的道路长度。每隔一段时间,就更新整个道路网络的交通情况,继而调整每个路段的权重。每辆装有导航的车辆,根据新的路段权重重新规划最优路径,使得车辆能够尽可能地避开拥堵路段,最终在整个道路网络范围之内,达到车辆分流、减少拥堵的目的。

半实时的交通灯协同控制算法是在每个交叉路口的局部范围内对交通状况进行优化;而启发式的动态路径规划算法则是在交通网络的全局范围内进行优化;将二者结合,构成一套基于车联网的交通管控机制。

本文第 2 节对相关工作进行介绍;第 3 节提出一种基于车联网的、半实时的、适用于多路口的交通灯协同控制算法;第 4 节提出适用于车联网的启发式动态路径规划算法;第 5 节将半实时的交通控制方法与启发式的动态路径规划算法相结合,提出一套基于车联网的交通管控机制;第 6 节通过建立两种不同的场景,分别进行模拟实验,验证基于车联网的交通管控机制的可用性及其有效性;最后总结全文。

2 相关工作

2.1 交通灯控制方法

目前主要有两种交通灯控制方法:固定时长轮转法和车流量响应控制法^[1]。

固定时长轮转法是一种依据历史交通数据来预先设定交通灯相位和周期时长的方法。一天 24 个小时会被划分为多个时段,每个时段内的交通灯相位和周期都相同,不同的时段将执行不同的交通灯策略。固定时长轮转法的优点是计算简单、容易实现。但是,在现实中,城市道路网络的交通流量瞬息万变,即时的交通状况很可能与历史数据有很大的不同,在此情况下,固定时长轮转法既不具备灵活性,又缺乏高效性。

车流量响应控制法是指根据交叉路口的车流量来动态调整交通灯相位的方法,针对交通灯控制的相关研究大多集中于此类方法。大多数车流量响应控制法都需要实时收集交通信息,有些通过环形线圈车辆检测器^[2-3]或者传感器^[4-5]来直接收集信息,有些则是对摄像头拍下的画面进行图像处理,以从中提取交通信息^[6-10]。近年来,越来越多的学者开始研究

如何利用车联网在收集信息方面的优势来进行交通灯的控制^[11-14]。

显然,车流量响应控制法的性能要比固定时长轮转法好。然而,大多数车流量响应控制法的研究都过于强调交通灯控制的实时性,而道路网络的交通情况又极其复杂,对如此庞大的交通数据进行实时计算时所耗费的资源是难以计数的。这就使得此类方法难以真正在大型道路网络中实现,只能局限于相对较小规模的道路网络。

2.2 车辆导航

目前人们广泛使用的导航系统基本上都是基于静态交通信息以及历史交通数据的导航方法。其中,常用的路径规划算法主要有 Dijkstra 算法、A* 算法、在线搜索^[15]、分层搜索等^[16]。但是,当某个道路因为施工或发生意外事故而导致交通不畅时,单纯基于静态信息和历史数据的导航方式无法获得这些信息,即无法及时调整导航路线。

此外,目前的导航系统基本上都是依赖 GPS 获取车辆当前位置的。当车辆位于高架、天桥下或行道树枝叶茂密、城市中心高楼林立时,GPS 的信号很容易丢失。

基于以上两点,越来越多的人开始研究能够及时获取路况信息并根据这些信息及时调整行驶路线的动态导航方法。文献^[17]提出一种路线信息共享(Route Information Sharing)系统,文献^[18]提出一种社交导航网络(Social Navigation Network),两者都是令车辆与车辆通过某个系统或网络进行信息共享,从而能够即时获取路况信息。而在车联网概念被提出之后,考虑到车联网在收集路况信息方面的强大优势,不少学者开始研究基于车联网的动态导航方法^[19-21]。

3 半实时的交通灯协同控制

3.1 问题建模

如图 1 所示,交通道路网络可以建模为一个有向图 $G(V, E)$,其中, V 表示交叉路口的集合, E 表示路段的集合。对于一对相邻的顶点 v_i 和 v_j , $\langle v_i, v_j \rangle$ 表示从路口 v_i 连向路口 v_j 的路段,而 $\langle v_j, v_i \rangle$ 表示从路口 v_j 连向路口 v_i 的路段。 $n(v_i)$ 表示路口 v_i 的邻居路口的集合, $|n(v_i)|$ 表示路口 v_i 的邻居路口的个数。以路口 v_8 为例, $n(v_8) = \{v_4, v_7, v_9, v_{11}\}$, $|n(v_8)| = 4$ 。

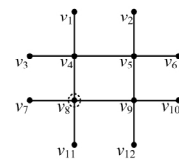


图 1 交通道路网络示例

本文以六车道的交叉路口为例(见图 2),每个交叉路口包括东、南、西、北 4 个方向,每个方向有 3 条进入交叉路口的车道和 3 条离开交叉路口的车道,分别将其简称为进入车道和离开车道。在 3 条进入车道中,为方便研究,本文规定左转弯车辆只能行驶在最左边的车道,中间的车道只能由直行的车辆行驶,而右转弯车辆只能行驶在最右边的车道,并默认所有右拐车道的交通灯始终为绿灯状态,因此本文不考虑右转弯车辆的行驶路径。将剩余的 8 条行驶路径分为 4 组,每一组由两条兼容(即处于这两条行驶路径上的车辆互相不会阻碍到彼

此的前行)的行驶路径组成:

- 从北驶向南(N-S)与从南驶向北(S-N);
- 从东驶向西(E-W)与从西驶向东(W-E);
- 从北驶向东(N-E)与从南驶向西(S-W);
- 从东驶向南(E-S)与从西驶向北(W-N)。

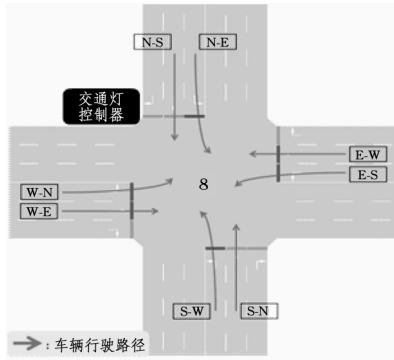


图2 六车道交叉路口示例图

在每个交叉路口,设置有一个交通灯控制器,该控制器与附近的路边单元(Road Side Unit,RSU)相连通。路边单元一方面与其附近装有车载单元(On Board Unit, OBU)的车辆进行无线通信,另一方面又从路边的传感器、摄像头获取信息,最后将收集到的信息汇总发送给该路口的交通灯控制器。同时,每个交叉路口的交通灯控制器还与其邻居路口的控制器通过有线网络连通,以便交换信息,从而进行协同工作。

位于路口 v_i 的交通灯控制器 c_i 负责控制该路口的 8 条进入车道的交通灯状态。令 $l_i(j, k)$ 表示从路口 v_j 过来到达路口 v_i 并要转向路口 v_k 的车道,以图 1 中的路口 v_8 为例,当某辆车行驶在车道 $l_8(7, 11)$ 上,即意味着该车辆正位于路口 v_8 处,它从路口 v_7 开过来,将要在路口 v_8 右转向路口 v_{11} 。

在交通灯相位时段 p 内,将车道 $l_i(j, k)$ 的交通灯状态定义为 $s_i((j, k), p) \in \{green, red\}$,表示交通灯状态。为方便研究,将黄灯视作特殊的红灯状态,因此不将黄灯作为一个单独的交通灯状态进行考虑。

定义 1 在交通灯相位时段 p 内,路口 v_i 的交通灯相位为本路口所有进入车道的交通灯状态集合,以 $S(v_i, p)$ 表示:

$$S(v_i, p) = \cup s_i((j, k), p) (v_j \in n(v_i), v_k \in \{n(v_i) - v_j\}) \quad (1)$$

以图 2 中的六车道交叉路口为例, $|n(v_i)| = 4$,因为右转弯车道默认为绿灯,在本文中不做分析, $|n(v_i) - v_j - 1| = 2$,所以 $S(v_i, p)$ 一共包括 $4 \times 2 = 8$ 条进入车道的交通灯状态。

在每个交通灯相位时段内,只有车流量最大的那一组行驶路径上的车辆被允许放行,即为绿灯状态,其他 3 组行驶路径对应的车道只能为红灯状态,以此来确保在每个交通灯相位时段内,4 个方向上的所有车辆总的等待时间最短,进而保证整个道路网络中所有车辆总的等待时间最短。

另外,为了避免有些车辆因为处在车流量较小的行驶路径上而一直等待红灯无法通行的情况发生,对于每辆等在车道 $l_i(j, k)$ 上的车辆,交通灯控制器 c_i 会记录其已等待时间。一旦车辆离开路口 v_i ,其已等待时间将被重置为 0。用 $w_i((j, k), p)$ 表示在时段 p 的一开始,车道 $l_i(j, k)$ 上等待最久的车辆的已等待时间,相位时长表示为 t_p 。在时段 p 开始时,首先判断是否存在 $w_i((j, k), p) + t_p$ 超过等待阈值,若存

在,则将该车道 $l_i(j, k)$ 在 p 时段的交通灯状态置为绿灯。若存在多个车道的最长已等待时间超过等待阈值,则选择等待时间最长的车辆所在车道;若等待时间相同,则选择整体等待时间最长的车道。

3.2 相关参数

本文提出的半实时的交通灯协同控制算法是以交通灯相位时长固定为前提的,且需要提前设定好该定值。在每个交通灯相位时段 p 内,路口 v_i 的交通灯控制器 c_i 收集该路口的交通信息,并与相邻路口的交通灯控制器进行信息交换,从而预测在下一个交通灯相位时段 $p+1$ 时该路口的 4 组行驶路径的车流量,最后根据预测结果计算出下一个交通灯相位的最优决策状态。

以车道 $l_i(j, k)$ 为例,在交通灯相位时段 p 内,预测过程涉及到的一些参数定义如下:

本文用 $Tr_i((j, k), p)$ 表示在时段 p 的一开始,已经到达路口 v_i 且位于车道 $l_i(j, k)$ 上的车辆总数。车辆的数量可以通过分析路边的传感器和摄像头收集到的信息来得到。

本文用 $Tr_i((j, k), p)$ 表示在当前时段 p 内正行驶在路段 $\langle v_j, v_i \rangle$ 上,并在下个时段 $p+1$ 开始之前,将会到达路口 v_i 且位于车道 $l_i(j, k)$ 上的车辆的总数。在当前时段 p 内正行驶在路段 $\langle v_j, v_i \rangle$ 上的车辆可分为两部分:1) 为与车联网相连并设置导航系统的车辆,这一类车辆可以根据导航路线而确定的下一个路口的信息发送给路段 $\langle v_j, v_i \rangle$ 上的路边单元;2) 未进行导航的车辆,此类车辆的信息由摄像头和传感器收集后发送给路边单元。路边单元将收集到的所有信息发送给交通灯控制器,控制器可以根据车辆的历史轨迹来预测其下一个路口。本文提出的管控机制采用基于概率后缀树(Probabilistic Suffix Tree, PST)的轨迹预测方法^[26]。然后,交通灯控制器选出那些下一个路口为 v_k 的车辆,再根据这些车辆距离路口 v_i 的路程以及行驶速度,判断在下个时段 $p+1$ 开始之前,这些车辆是否会到达路口 v_i 且位于车道 $l_i(j, k)$ 上,进而得到 $Tr_i((j, k), p)$ 。需要注意的是,由于预测并不能保证 100% 的准确,因此在得到第二类车辆的数量后,需要将其数量乘以相应的预测准确率,从而得到这类车辆的期望数量,进而减小因为预测不准确而引起的误差。

本文用 $Td_i((j, k), p)$ 来表示在时段 p 内,从车道 $l_i(j, k)$ 离开路口 v_i 的车辆总数。 $Td_i((j, k), p)$ 可以通过分析相关的历史数据并结合离开车辆的当前速度来计算得到。需要注意的是,如果车道 $l_i(j, k)$ 在时段 p 内对应的交通灯状态为红灯,即 $s_i((j, k), p) = red$,那么所有在车道 $l_i(j, k)$ 上的车辆都不能在时段 p 内离开路口 v_i ,因此在这种情况下, $Td_i((j, k), p) = 0$ 。

在时段 p 内,会有车辆从路口 v_i 的邻居路口 v_j 离开,驶向路口 v_i 。其中,有一部分车辆会在下个时段 $p+1$ 开始之前到达路口 v_i 处的车道 $l_i(j, k)$ 上,用 $Tc_i((j, k), p)$ 来表示此类的车辆数量。 $Tc_i((j, k), p)$ 的计算过程分为两步:先由路口 v_i 的邻居路口 v_j 处的交通灯控制器 c_j 计算出所有离开路口 v_j 并开上路段 $\langle v_j, v_i \rangle$ 的车辆,并将此信息发送给路口 v_i 的控制器 c_i ;然后与计算 $Tr_i((j, k), p)$ 时一样,由控制器 c_i 进行分析计算,最后得到 $Tc_i((j, k), p)$ 。

根据以上参数即可预测在下一个时段 $p+1$ 开始之前将

会到达路口 v_i 并处于车道 $l_i(j, k)$ 上的车辆总数,用 $T_i((j, k), p+1)$ 来表示,并用式(2)进行定义:

$$T_i((j, k), p+1) = T_i((j, k), p) + Tr_i((j, k), p) + Tc_i((j, k), p) - Td_i((j, k), p) \quad (2)$$

3.3 半实时的交通灯协同控制算法

算法 1 描述了本文提出的半实时的交通灯协同控制算法的主要思想。

算法 1 半实时的交通灯协同控制

```

1. initialize S(i, p+1);
   // set traffic lights for  $v_i$ ;
2. for all  $v_j$  in  $n(v_i)$  &  $v_k$  in  $\{n(v_i) - v_j\}$  do
3.   get  $w_i((j, k), p+1)$ ;
4. end for;
5. if  $\max(w_i((j, k), p+1)) + t_p > \max WT$  then
6.    $s_i((j, k), p+1) = \text{green}$ ;
   //将车道  $l_i(j, k)$  的兼容车道也置为绿灯状态
   //用  $s_i'((j, k), p+1)$  来表示兼容车道的交通灯状态
7.    $s_i'((j, k), p+1) = \text{green}$ ;
8.   return;
9. else then
10.  for all  $v_j$  in  $n(v_i)$  &  $v_k$  in  $\{n(v_i) - v_j\}$  do
11.    get  $T_i((j, k), p)$ ,  $Tr_i((j, k), p)$ ,  $Td_i((j, k), p)$ ,  $Tc_i((j, k), p)$ ;
12.    share  $Td_i((j, k), p)$  to  $c_k$ ;
13.     $T_i((j, k), p+1) = T_i((j, k), p) + Tr_i((j, k), p) + Tc_i((j, k), p) - Td_i((j, k), p)$ ;
14.  end for
   //用  $T_i'((j, k), p+1)$  表示车道  $l_i(j, k)$  的兼容车道的车辆总数
15.  get  $\max(T_i((j, k), p+1) + T_i'((j, k), p+1))$ ;
16.   $s_i((j, k), p+1) = \text{green}$ ;
   //将车道  $l_i(j, k)$  的兼容车道也置为绿灯状态
17.   $s_i'((j, k), p+1) = \text{green}$ ;
18. end if

```

在该算法中,需要预先设定固定的交通灯相位时长。设当前交通灯相位时段为 $p+1$ 。在 $p+1$ 时段开始时,首先判断是否存在 $p+2$ 时段开始时超出等待时间阈值的车道,若有,则直接在 $p+1$ 时段将该车道及其兼容车道设置为绿灯状态;若存在多个车道超出阈值,则选择等待时间最长的车道;若多个超出时间阈值的车道的最长等待时间相同,则选择整体等待时间最长的车道(见算法 1 中第 2—8 行)。否则,位于路口 v_i 的交通灯控制器 c_i 将本路口的交通灯相位设置为在上个时段 p 时计算出的 $p+1$ 时段的最优交通灯决策的状态,具体表述为:控制器 c_i 获取本路口每条进入车道的已到达车辆总数 $T_i((j, k), p)$ 、将到达车辆总数 $Tr_i((j, k), p)$ 以及将离开的车辆总数 $Td_i((j, k), p)$ 。接着,把将离开本路口 v_i 的车辆信息发送给它们将要开往的邻居路口 v_k 的控制器。同时控制器 c_i 也从每个邻居路口的控制器接收将要离开该邻居路口离开并开向本路口 v_i 的车辆信息,根据这些信息计算得到每条进入车道的 $Tc_i((j, k), p)$ 。利用式(2)即可计算出在时段 p 开始之前路口 v_i 对应的各条进入车道中等待的车辆总数 $T_i((j, k), p+1)$ (见算法 1 中第 10—14 行)。最后,控制器 c_i 计算出在下个时段 $p+1$ 开始之前车流量最大的一组兼容行驶路径,并将该组行驶路径对应的车道的交通灯状态置为绿灯,其余 3 组行驶路径对应的车道的交通灯状

态置为红灯(见算法 1 中第 15—17 行)。如果车流量最大的兼容车道不止一组,则控制器 c_i 比较这几组车道的总的车辆的已等待时间,将已等待时间最长的那组车道置为绿灯。

4 启发式的动态路径规划算法

目前广泛应用的导航系统通常使用的都是 Dijkstra 算法,将有向图 $G(V, E)$ 的边 $\langle v_i, v_j \rangle$ 的权重设置为连接路口 v_i 与路口 v_j 的路段长度。然而,在现实情况下,道路的交通状况是持续变化的,若某些路段已经严重拥堵,仅考虑静态路段长度的导航因为无法得到最新的路况消息会继续将更多的车辆导航到这些拥堵路段上,导致拥堵越来越严重。

在现实中,大部分司机设置导航都是为了能够在最短的时间内到达目的地,为了避免以上情况的发生,在 Dijkstra 算法中不再以路段的长度作为边的权重,而是通过启发式的方法,以车辆通过某条路段的加权时间估计作为该条路段对应的权重。又因为在不同时段,车辆通过路段的加权时间估计会因为交通情况的不同而发生变化,所以每隔一段时间就需要对整个道路网络中所有路段的加权时间估计进行更新。如此一来,原本要经过拥堵路段的车辆可以绕开这些路段,避免越来越多的车辆堵在同一个路段上。

对于道路网络 $G(V, E)$,为了计算车辆通过某条路段 $r_i \in E$ 的加权时间估计 \tilde{t} ,需要考虑以下 3 个方面。

(1) 车辆通过路段 r_i 所需的平均时间。该平均时间可以用路段 r_i 的长度 l_i 除以目前路段 r_i 上所有车辆的平均车速 v_{mean} 来得到。如果此时路段 r_i 上没有车辆,则将 v_{mean} 设置为该路段的最高限速。

(2) 车辆通过路段 r_i 需要等待红灯的时间。该时间可以用车辆等待红灯的期望时间来表示,即用车辆遇到红灯的概率 P_{red} 乘以交通灯相位时长 p 。车辆在路段 r_i 的路口有 3 种可能的转向,即右转、直行和左转。在第 3 节中,默认所有右转车辆无需等待红灯,而对于非右转车辆,其遇上红灯的概率为 $3/4$,因为如 3.1 节中所述,每个路口共有 4 组行驶路径,在每个交通灯相位时段内,只有 1 组行驶路径为绿灯,其余 3 组都为红灯。因此,车辆在路段 r_i 的路口遇到红灯的概率为 $P_{red} = 2/3 \times 3/4 = 1/2$ 。

(3) 路段 r_i 的拥堵系数。本文使用目前路段 r_i 上的车辆密度 ρ_i 乘以密度修正权重 α 再加上 1 来表示该路段当前时间的拥堵系数。密度 ρ_i 可以用目前路段 r_i 上的车辆总数 N_{veh} 除以其最多可容纳的车辆数来计算。而路段 r_i 最多可容纳的车辆数可以用路段 r_i 的长度 l_i 除以平均车辆长度 l_{veh} 与最短车距 l_{gap} 的和,再乘以车道数 n_i 来计算。注意,以一条六车道路段为例,在计算其最多可容纳的车辆数时,不能将其车道数视作 6,因为实际上,该路段有 2 个方向,每个方向各 3 个车道,故应将其车道数视作 3。而密度修正权重 α 表示路段密度对车辆通过路段的加权时间估计的影响度。 α 的取值非常重要,若取值太小,则拥堵路段的权重不够大,达不到令车辆绕开拥堵路段的效果;但若 α 取值过大,会使得某些没有拥堵的路段也被错误判断为拥堵路段,导致车辆反而绕了远路。至于加上 1 的原因是为了保证当路段 r_i 上没有车辆时拥堵系数不会被算作 0。

综上所述,车辆通过路段 r_i 的加权时间估计 \tilde{t} 的计算公式为:

$$\tilde{l} = (\alpha \times \rho_i + 1) \times \left(\frac{l_i}{v_{mean}} + \frac{1}{2} \times p \right) \quad (3)$$

其中, 路段 r_i 上的车辆密度 ρ_i 的计算公式为:

$$\rho_i = \frac{N_{veh}}{\frac{l_i}{l_{veh} + l_{gap}} \times n_i} \quad (4)$$

算法 2 描述了启发式的动态路径规划算法。为了能够动态更新道路网络交通信息, 该算法设定每隔一段时间 d , 车联网云端服务器就根据最新收到的道路交通信息, 按照式(3)更新道路网络 $G(V, E)$ 中每条边的权重, 得到更新后的道路网络 $G'(V, E)$, 将其发送给所有路边单元, 再由路边单元发送给附近的车辆。在获得最新的 $G'(V, E)$ 后, 车载单元先确定车辆的当前位置 s' , 然后将当前位置 s' 作为新的目的点输入, 根据最新的 $G'(V, E)$ 重新计算从当前位置到达目的地的最优路径。

算法 2 启发式的动态路径规划算法

输入: 目的点 target

1. get newest $G'(V, E)$ at regular intervals;
2. get current position s' ;
3. Dijkstra($G'(V, E), s', d$);
4. set new path;

5 基于车联网的交通管控机制设计

将第 3 节提出的半实时的交通灯协同控制算法与第 4 节提出的启发式的动态路径规划算法相结合, 提出一套基于车联网的交通管控机制, 如图 3 所示。

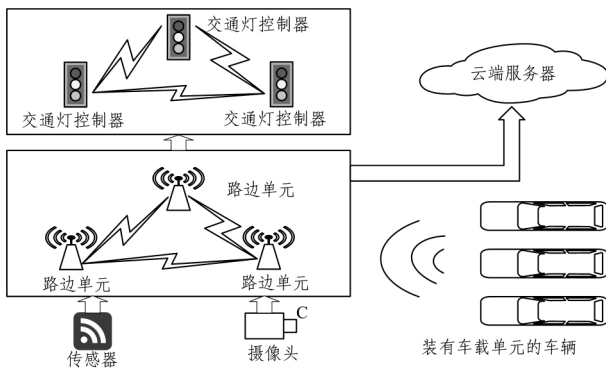


图 3 基于车联网的交通管控机制

在此管控机制中, 路边单元扮演了非常重要的桥梁角色。路边单元一方面与附近装有车载单元的车辆进行无线通信, 获取这些车辆的行驶速度、历史轨迹、行驶目的地等信息; 另一方面, 从附近道路上的传感器、摄像头那里收集未装有车载单元的车辆信息。然后, 路边单元每隔一段时间就将这些信息汇总起来, 通过有线网络, 分别发送给附近路口的交通灯控制器以及车联网的云端服务器。

每个交通灯控制器定时更新本路口附近的交通信息, 并将此信息分享给相邻路口的控制器。通过与相邻路口控制器之间的协作, 每个交通灯控制器计算出本路口的最优交通灯相位策略, 最终达到使整个交通网络的车辆等待时间最小的目的。

与交通灯控制器只掌握局域交通信息不同, 云端服务器拥有的是整个道路网络的全局交通信息。云端服务器每隔一段时间就更新全局网络信息, 并以此为依据, 更新每个路段的权重, 接着将整个道路网络的权重数据下发给所有路边单元,

路边单元再将此数据通过无线传输发送给所有附近装有车载单元的车辆。搭载在车载单元上的导航应用根据最新收到的道路网络权重数据及时调整本车的最佳行驶路径。

在整个交通管控机制中, 无论是路边单元、车载单元, 还是交通灯控制器、云端服务器, 各自的工作量都不大, 这使得它们对硬件性能的要求并不高, 因此从经济的角度来说, 该机制有着较高的可实现性。对于其在改善交通状况、缓解交通拥堵方面的有效性, 将在第 6 节通过模拟实验进行验证。

6 实验与验证

6.1 实验工具介绍

SUMO(Simulation of Urban Mobility)是一个开源的、微观的、多元化的交通流量模拟器。SUMO 生成的每一辆车都是独立定义的, 至少需要 3 个属性: 唯一标识、出发时间以及在道路网络中的行驶路线^[23-24]。

NS3 是一个开源的离散事件网络模拟器。NS3 通过 C++ 语言将现实网络中的软硬件对象抽象为类, 其中最关键的几个概念分别为: 节点(node), 对应于现实网络中具有联网功能的对象, 可以是一台计算机, 也可以是一个基站; 网络设备(net device), 相当于网卡及其驱动的总和; 信道(channel), 即可以传输网络数据的媒介^[25]。

利用 SUMO 提供的 TraCI(Traffic Control Interface)接口, 在 NS3 与 SUMO 之间建立起反馈循环。本文用 SUMO 构建交通网络场景并生成车流量, 用 NS3 在 SUMO 生成的车辆上部署节点和网络设备, 并沿着 SUMO 生成的道路部署路边单元。通过 TraCI 接口, SUMO 可以将交通场景中的各条道路、各个车辆的相关数据通过反馈指令传递给 NS3。与此同时, NS3 可以对接收到的数据进行分析处理, 并向 SUMO 发送控制指令控制 SUMO 中交通灯的状态以及车辆的速度、行驶路线等。

本文的模拟实验所用的 SUMO 版本为 0.20.0, NS3 的版本为 3.19。

6.2 模拟实验场景的构建与参数选择

为了尽可能模拟真实的道路环境, 本文利用 SUMO 构建一幅简略版上海中心市区地图。首先在 OpenstreetMap 开源地图上截取上海市在经度 121.342 到 121.505、纬度 31.154 到 31.282 范围内的地图(见图 4)。然后利用 NETCONVERT 工具将该地图导入 SUMO 中。再通过筛选, 只保留主干道, 最后得到一幅共有 69 个交叉路口、318 条路段的道路网络图(见图 5)。

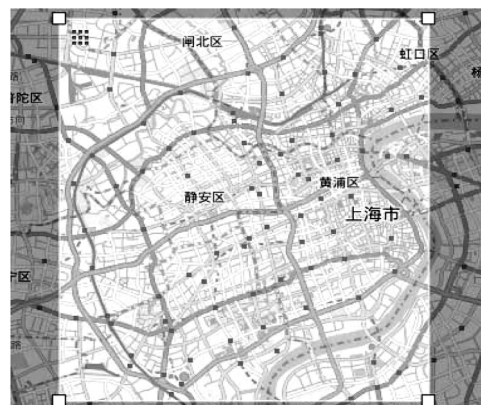


图 4 OpenstreetMap 导入地图

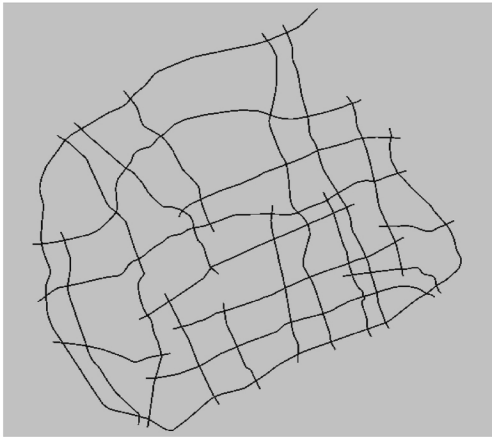


图 5 SUMO 生成的道路网络图

为了验证基于车联网的交通管控机制在不同情况下的性能,将道路网络分为 9 个区域,周围一圈 6 个区域以及中心 3 个区域。然后利用 SUMO 构建两种不同的场景:道路车辆密度均匀的场景和道路车辆密度不均匀的场景。

在道路车辆密度均匀的场景中,通过 O/D 矩阵工具生成 10000 辆车的虚拟车流。这些车辆较为均匀地分布在 9 个区域,并将在一个小时之内出发,随机开往 9 个区域中的某一个。而在道路车辆密度不均匀的场景中,共安排 8000 辆车在一个小时内出发,并且为了验证本文提出的交通管控机制在缓解交通拥堵方面的效率,在此场景中设置 3 处交通拥堵区域。

模拟实验中生成的车辆参数如表 1 所列。

参数	值
车辆加速度	3m/s
车辆制动加速度	-6m/s
车身长	5m
最小车间距	1m

在模拟实验中,对于交通灯相位时长的设置,有两方面的考虑因素:1)交通灯相位时长越短,整个交通网络中的车辆等待总时间就会越短;2)不能忽视行人过马路的基本需求。通常,一个六车道城市主干道宽为 20~23m,加上非机动车道和道路当中的绿化带,行人过马路时需要走 30m 左右^[27]。而文献^[28]建议在计算交通灯相位间隔时间时,将行人的行走速度设置为 1.05m/s(3.5ft/s)。所以为了尽量缩短交通灯的相位间隔时间,同时又使行人能够在此间隔时间内过完马路,本文将交通灯相位时长定为 30s。将更新全网道路权重的时间间隔设置为 10s。

为了在计算道路权重时能够确定恰当的密度修正权重 α 的值,在模拟实验中对 α 分别取区间 $[1,10]$ 内的 10 个值,共进行 10 次模拟实验,得到当密度修正权重 $\alpha=5$ 时,实验效果最好。因此,在接下来的实验中将密度修正权重 α 设置为 5。

此外,为了尽可能模拟真实的道路情况,将实验所用的交通网络中的道路分为 3 种:1)模拟城市主干道,道路限速为 50km/h,此类路段共有 76 条;2)模拟普通道路,限速为 40km/h,此类路段共有 218 条;3)模拟路况较差的道路(例如施工路段),此类路段限速为 30km/h,共有 24 条。

6.3 模拟实验结果

为了验证本文提出的基于车联网的交通管控机制的性能,在 6.2 节提到的两个场景中分别实行以下几种不同的交通管控机制:机制 1,只对实验场景中的所有交通灯实行固定轮转控制,并假设所有车辆都没有连入车联网而只能进行基于经典 Dijkstra 算法的静态路径规划;机制 2,对实验场景中的所有交通灯实行半实时的协同控制,并假设所有车辆都没有连入车联网而只能进行基于经典 Dijkstra 算法的静态路径规划;机制 3,在对所有交通灯实行半实时协同控制的同时,随机选取 50%的车辆,将它们连入车联网中,令这些车辆能够利用车联网进行启发式的动态路径规划,而剩余 50%的车辆则没有连入车联网,进行静态路径规划;机制 4,对所有交通灯实行半实时的协同控制,并假设理想情况下,所有车辆都连入车联网,能够进行启发式的动态路径规划。

图 6 显示了在道路车辆密度均匀的场景和道路车辆密度不均匀的场景中,4 种不同的交通管控机制分别所对应的车辆平均等待时间。在道路车辆密度均匀的场景中,仅机制 2 就可以达到很好的效果,其车辆平均等待时间为 163.85s,比机制 1 的 472.22s 降低了 65.30%。相比之下,此场景中加入启发式的车辆动态路径规划的机制 3 和机制 4 并没有取得明显的优化效果,车辆平均等待时间分别为 157.25s 和 144.66s,比机制 1 降低了 66.70%和 69.37%,但仅仅比机制 2 降低了 4.02%和 11.71%。这是因为在车辆进行动态路径规划时,每个路段的权重随着其密度变大而增加,而在道路车辆密度均匀的场景中,所有路段的密度都差不多,导致所有路段的权重之比与静态路径规划时没有明显的差别,所以两者得到的车辆行驶路线也没有太大不同,因而启发式的动态路径规划比静态路径规划的车辆平均等待时间降低得并不多。在道路车辆密度不均匀的场景中,虽然安排的车辆数比道路车辆密度均匀的场景要少 20%,但如果使用机制 1,车辆的平均等待时间却要比道路车辆密度均匀的场景多 35.19%。可见交通拥堵会大大增加车辆的等待时间。在此情况下,机制 3 和机制 4 的平均等待时间比机制 2 要降低 16.28%和 28.10%。可见,在道路网络发生拥堵时,基于启发式的动态路径规划的优化效果比静态路径规划有明显的提高。这是因为本文提出的动态路径规划算法能够使车辆绕开拥堵路段,从而减少等待时间。可见,本文提出的基于车联网的交通管控机制能够有效地减少交通网络中的车辆等待时间并缓解交通拥堵。

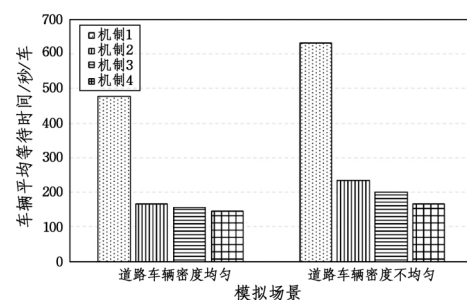


图 6 车辆平均等待时间

由于机制 3 和机制 4 的区别仅在于可利用车联网进行动态路径规划的车辆比例的大小,且图 6 已经说明比例越大,车

辆平均等待时间越小,故机制4更能突出动态路径规划的作用。因此,在接下来的讨论中,为简化实验,本文直接使用机制1、机制2、机制4进行对比。

图7显示了在道路车辆密度均匀和道路车辆密度不均匀的两个场景中,车辆等待时间与行驶时间的比值,其中,车辆行驶时间为车辆总的出行时间减去其等待时间的部分。为了更清晰直观地显示不同交通管控机制之间的性能差异,本文将每种管控机制对应的等待时间与行驶时间的比值都按照升序排列。需要注意的是,两张图的纵坐标是不同的。从图7中可以看出,在道路车辆密度不均匀的场景中,车辆等待时间与行驶时间的比值明显高于道路车辆密度均匀的场景。可见,道路拥堵会严重降低车辆的行驶效率。在实验中,当执行机制1时,在道路车辆密度均匀的场景中(见图7(a)),只有个别车辆的等待时间大于其行驶时间(比值大于1);而在道路车辆密度不均匀的场景中(见图7(b)),26.71%的车辆等待时间大于其行驶时间,车辆等待时间与行驶时间的比值最高达到5.99。当执行机制2和机制4时,在道路车辆密度不均匀的场景中,等待时间大于行驶时间的车辆比例分别下降至5.58%及0.99%,同时,车辆等待时间与行驶时间的最高比值分别减小到4.51和3.22。这个实验数据表明,本文提出的基于车联网的交通管控机制能够降低车辆等待时间与行驶时间的比值并提高车辆的出行效率,在道路拥堵的情况下,该管控机制的优化效果尤为明显。

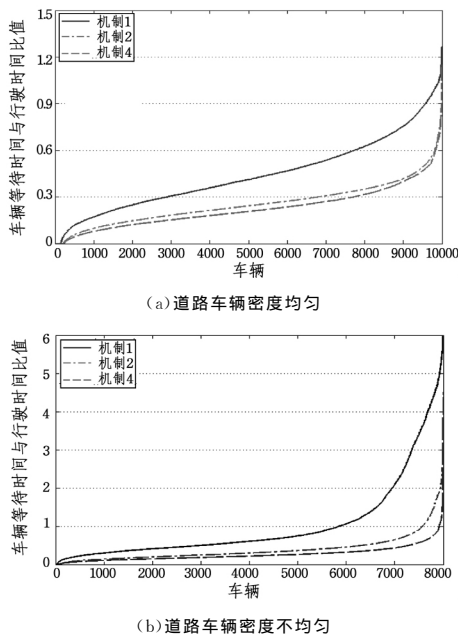
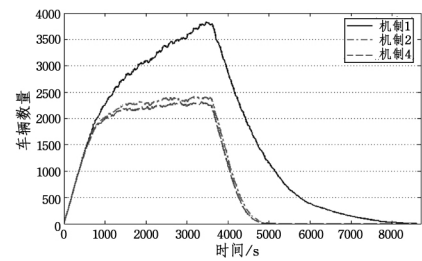
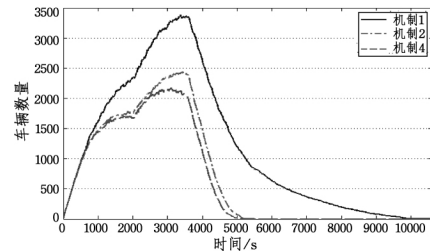


图7 车辆等待时间与行驶时间的比值

图8显示了在道路车辆密度均匀和道路车辆密度不均匀这两个场景中,每一秒内整个道路网络中的车辆总数。从图8中可以看出,机制1,2,4,在4000s之后,都能明显地减少模拟实验场景内的车辆总数。如图8(a)所示,在道路车辆密度均匀的场景中,机制4减少的车辆总数并没有明显少于机制2。而由图8(b)可以看出,在道路车辆密度不均匀的场景中,2500s之后,机制4每秒车辆总数明显少于机制2。这些数据表明,本文提出的车联网交通管控机制能够有效地减少整个交通网络的车辆负载,而当发生交通拥堵时,其车辆负载的降低效率尤为明显。



(a)道路车辆密度均匀



(b)道路车辆密度不均匀

图8 实验场景中车辆总数随时间的变化

结束语 本文首先提出了半实时的交通灯协同控制算法,该算法融合了固定时长轮转法和车流量响应控制法的思想,通过相邻路口的交通灯控制器之间的协同合作,预测下个时段该路口的交通状况并以此为依据制定最优的交通灯策略。然后,本文在Dijkstra算法的基础上,采用启发式的权重计算方法,以车辆通过路段的加权时间估计代替原本静态的道路长度来计算边的权重,并每隔一段时间,就利用车联网根据每条路段的当前交通数据调整其权重。车辆获得所有路段的新权重后就重新规划最优行驶路径。最后,将半实时的交通灯协同控制算法与启发式的动态路径规划算法相结合,提出一套基于车联网的交通管控机制。为了验证本文提出的基于车联网的交通管控机制的性能,将交通流量模拟器SUMO与网络模拟器NS3相结合来进行模拟实验。实验的结果表明,本文提出的基于车联网的交通管控机制具备较高的可用性和效率,实验数据显示该交通管控机制能够显著降低车辆的等待时间,缓解交通拥堵状况,从而达到提高车辆出行效率并改善整个道路网络交通状况的目的。

为方便研究,本文规定左转车辆只能行驶在最左边的车道,中间的车道只能由直行的车辆行驶,而右转车辆只能行驶在最右边的车道,并默认所有右拐车道的交通灯始终为绿灯状态,但可能不太符合现实路况情况;不仅如此,实验所用的数据多为模拟器产生的模拟数据,所有的车辆都是同一种类型的,其行驶模式都是从起点出发,到达目的地后就自动被移出实验场景,与真实的交通情况存在一定差距。在未来的工作中,我们将会考虑左右车道均可直行或右车道可直行等更加符合现实场景的状况;并收集真实的车辆轨迹数据,构建更贴近真实交通状况的实验场景,以验证提出的车联网交通管控机制在更真实、更复杂的交通情况下的性能。

参考文献

- [1] National Electrical Manufacturers Association. Traffic Controller Assemblies with NTCIP Requirements; TS 2-2003 v02. 06 [S]. NEMA Standards Publication, 2003.
- [2] ZHOU B B, CAO J N, WU H J. Adaptive traffic light control of

- multiple intersections in WSN-Based ITS[C]// Proceedings of IEEE Vehicular Technology Conference. Yokohama, Japan, 2011;1-5.
- [3] ZHU Y M, LIU X M, LI M L, et al. POVA: Traffic light sensing with probe vehicles, on parallel and distributed systems[J]. IEEE Transaction on Parallel and Distributed Systems, 2013, 24(7):1390-1400.
- [4] FAYE S, CHAUDET C, DEMEURE I. A distributed algorithm for multiple intersections adaptive traffic lights control using a wireless sensor networks[C]// Proceedings of the first workshop on Urban networking (UrbaNe). New York, NY, USA, 2012;13-18.
- [5] SWAPNALI S, CHILVERI P G. Hand gesture recognition using accelerometer sensor for traffic light control system[C]// Proceedings of Electronics and Communication Systems (ICECS). Coimbatore, India, 2014;1-5.
- [6] GOMEZ A, ALENCAR F, PRADO P, et al. Traffic lights detection and state estimation using Hidden Markov Models[C]// Proceedings of IEEE Intelligent Vehicles Symposium. Dearborn, MI, USA, 2014;750-755.
- [7] NIENHUSER D, DRESCHER M, ZOLLNE J M. Visual state estimation of traffic lights using Hidden Markov Models[C]// Proceedings of International IEEE Conference on Intelligent Transportation Systems (ITSC). Funchal, Portugal, 2010;1705-1710.
- [8] CHOUDEKAR P, BANERJEE S, MUJU M K. Implementation of image processing in real time traffic light control[C]// Proceedings of International IEEE Conference on Electronics Computer Technology (ICECT). Kanyakumari, India, 2011;94-98.
- [9] KANUNGO A, SHARMA A, SINGLA C. Smart traffic lights switching and traffic density calculation using video processing [C]// Proceedings of Engineering and Computational Sciences (RAECS). Chandigarh, India, 2014;1-6.
- [10] ASARE S K, SOWAH R A. Design and development of a micro-controller based traffic light control system using image processing techniques: A case study prototype for Legon-Okponglo Junction, University of Ghana[C]// Proceedings of International IEEE Conference on Adaptive Science & Technology (ICAST). Kumasi, Ghana, 2012;59-64.
- [11] NEUDECKER T, AN N, TONGUZ O K, et al. Feasibility of virtual traffic lights in non-line-of-sight environments[C]// Proceedings of ACM International Workshop on Vehicular Inter-Networking, Systems, and Applications (VANET). New York, NY, USA, 2012;103-106.
- [12] SONG C, WU J, YANG W S, et al. Catching up with traffic lights for data delivery in vehicular ad hoc networks[C]// Proceedings of ACM Annual International Workshop on Mission-oriented Wireless Sensor Networking (MiSeNet). New York, NY, USA, 2013;21-26.
- [13] ABBAS M K, KARSITI M N, NAPIAH M, et al. Traffic light control via VANET system architecture[C]// Proceedings of Wireless Technology and Applications (ISWTA). Langkawi, Malaysia, 2011;174-179.
- [14] KWATIRAYO S, ALMHANA J, LIU Z. Adaptive traffic light control using VANET; A case study[C]// Proceedings of Wireless Communications and Mobile Computing Conference (IWC-MC). Sardinia, Italy, 2013;752-757.
- [15] SUMIYA K, SEGAWA T, SUGIHARA K, et al. Web-in-Car: A web search method not requiring keyword input for car navigation systems[C]// Proceedings of ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS). Irvine, CA, USA, 2008.
- [16] WEN F, GAO X H, GEN M. A novel approach to route selection in car navigation systems by a multiobjective genetic algorithm [C]// Proceedings of Annual Conference on Genetic and Evolutionary Computation (GECCO). Atlanta, Georgia, USA, 2008;767-768.
- [17] YAMASHITA T, IZUMI K, KURUMATANI K, et al. Smooth traffic flow with a cooperative car navigation system[C]// Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems. New York, NY, USA, 2005;478-485.
- [18] KARIMI H, BENNER J, ANWAR M. A model for navigation experience sharing through social navigation networks (So-NavNets)[C]// Proceedings of IEEE International Conference on Information Reuse and Integration (IRI). Las Vegas, Nevada, USA, 2011;557-560.
- [19] CHIM T, YIU S, HUI L, et al. VSPN: VANET-based secure and privacy-preserving navigation [J]. IEEE Transactions on Computers, 2014, 63(2):510-524.
- [20] CHANG I C, TAI H T, HSIEH D L, et al. Design and implementation of the travelling time-and energy-efficient Android GPS navigation app with the VANET-based A* route planning algorithm[C]// Proceedings of International Symposium on Biometrics and Security Technologies (ISBAST). Chengdu, China, 2013;85-92.
- [21] WU Y, SHAO Z H, LI W F, et al. A novel design and realization of the vehicular driving navigation system based on VANET[C]// Proceedings of Intelligent Control and Automation (WCICA). Jinan, China, 2010;4419-4423.
- [22] CORMEN T H, LEISERSON C E, RIVEST R L, et al. Introduction to Algorithms(3rd Edition)[M]. The MIT Press, 2009.
- [23] BEHRISCH M, BIEKER L, ERDMANN J, et al. SUMO-Simulation of Urban MObility, An Overview[C]// Proceedings of International Conference on Advances in System Simulation (SIMUL). 2011;63-68.
- [24] KRAJZEWICZ D. Traffic Simulation with SUMO-Simulation of Urban Mobility[J]. Fundamentals of Traffic Simulation International Series in Operations Research & Management Science, 2010, 145:269-293.
- [25] 马春光, 姚建盛. ns-3 网络模拟器基础及应用[M]. 北京:人民邮电出版社, 2014.
- [26] RON D, SINGER Y, TISHBY N. The power of amnesia: Learning probabilistic automata with variable memory length[J]. Machine Learning, 1996, 25(2/3):117-149.
- [27] POTTS I B, HARWOOD D W, RICHARD K R. Relationship of lane width to safety for urban and suburban arterials[J]. Journal of the Transportation Research Board, 2007, 2023(2023):63-82.
- [28] AUSTRALIA S. Manual of uniform traffic control devices: part 2; traffic control devices for general use[M]. 2009.