

## 基于浮动车数据的快速交通拥堵监控

吴佩莉<sup>1,4</sup> 刘奎恩<sup>2</sup> 郝身刚<sup>3</sup> 张全新<sup>1,4</sup> 谭毓安<sup>1,4</sup>

<sup>1</sup>(北京理工大学计算机学院 北京 100081)

<sup>2</sup>(中国科学院软件研究所 北京 100190)

<sup>3</sup>(南阳师范学院计算机与信息技术学院 河南南阳 473061)

<sup>4</sup>(北京市海量语言信息处理与云计算应用工程技术研究中心 北京 100081)  
(wupl@bit.edu.cn)

## Rapid Traffic Congestion Monitoring Based on Floating Car Data

Wu Peili<sup>1,4</sup>, Liu Kui'en<sup>2</sup>, Hao Shengang<sup>3</sup>, Zhang Quanxin<sup>1,4</sup>, and Tan Yu'an<sup>1,4</sup>

<sup>1</sup>(School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081)

<sup>2</sup>(Institute of Software, Chinese Academy of Sciences, Beijing 100190)

<sup>3</sup>(School of Computer and Information Technology, Nanyang Normal University, Nanyang, Henan 473061)

<sup>4</sup>(Beijing Engineering Research Center of Massive Language Information Processing and Cloud Computing Application, Beijing 100081)

**Abstract** Floating car technology is the essential source to acquire the road traffic information in intelligent transportation systems. It can be used as the data source for large-scale real-time traffic monitoring. It's a challenge of handling stream data effectively in a large number of moving objects because of the huge scale of (floating car data, FCD). In this paper, a congestion companion discovery algorithm is proposed by adopting the idea of similar trajectory clustering and utilizing traffic parameters with congestion characteristics. The candidate congestion FCD can be filtered out from the floating car trajectory stream for approximately predicting the trend of congestion areas. While the load shedding decision-making is determined by the prediction, an algorithm of multi-priority scheduling based on prediction is designed to achieve the whole monitoring process. Our method can effectively reduce the processing cost of FCD, and rapidly monitor traffic congestion. Both efficiency and effectiveness of our method are evaluated by a very large volume of real taxi trajectories in an urban road network.

**Key words** floating car data; traffic congestion; congestion companion discovery; load shedding; trajectory data stream

**摘要** 浮动车技术是近年来智能交通系统中所采用的、获取道路交通信息的先进技术手段之一,可作为大规模实时交通监控的数据源。由于浮动车数据规模庞大,从大量移动对象中有效处理流数据是其中一大难点。采用相似轨迹聚类的思想,结合与拥堵特征相关的交通参数,提出了拥堵同伴发现算法。该算法能从浮动车轨迹流数据中筛选出可能发生拥堵的浮动车数据,从而对拥堵区域变化趋势进行概化预测,由预测结果决定负载处理方式。此外,设计基于预测的多优先级调度算法用以实现整个监控流程。提出的方法可有效降低处理浮动车数据的代价,实现快速交通拥堵监控。通过在城市路网中大规模出租车

收稿日期:2013-05-15;修回日期:2013-08-16

基金项目:国家“八六三”高技术研究发展计划基金项目(2013AA01A212);国家自然科学基金项目(61202064,91124001)

通信作者:郝身刚(nythhsg@sina.com)

轨迹数据上的实测,验证了这种算法的有效性和优势。

**关键词** 浮动车数据;交通拥堵;拥堵同伴发现;降载;轨迹流数据

**中图法分类号** TP311.13

交通拥堵已成为现代生活的特征,并成为一个困扰现代城市发展的严重问题.对于出行者来说,寻求不拥堵的路段,错开严重拥挤路段将大大提高出行效率,可以避免不必要的燃油消费和时间消耗.及时正确地获取道路交通拥堵状况,采取合理预警措施,主动避免交通拥堵,已成为提高道路通行能力的有效手段之一.如何充分利用现代技术有效挖掘监控信息,进一步提升监控系统的智能性,对实现系统功能的增值具有重要的利用意义和实际应用价值<sup>[1]</sup>.

采集交通信息的主要手段包括:线圈、光电、微波、视频和全球定位系统(global position system, GPS)检测等,而浮动车(floating car)技术是伴随着智能交通系统(intelligence transportation system, ITS)新技术应用而在近几年发展起来的新型交通流信息采集技术.浮动车一般是指安装了车载 GPS 定位装置并行驶在城市主干道上的公交汽车和出租车,其基本原理是<sup>[2]</sup>:浮动车通过其装备的车载 GPS 和无线通信接口,周期性地向中心服务器报告自己在其行驶过程中定期记录的车辆位置、行驶方向和速度等信息,服务器每隔一段时间(如 5 min)对所收集的信息应用地图匹配、路径推测等相关的计算模型和算法进行路网匹配和统计计算,使浮动车位置数据和城市道路在时间和空间上关联起来,最终得到浮动车所经过道路的车辆行驶速度以及道路的行车旅行时间等交通网络的交通参数.许多主要城市的出租车已经被嵌入 GPS 传感器,它们的位置以流轨迹的形式定期向交通运输系统报告<sup>[3]</sup>.从移动车辆不断采集的交通更新数据可以得到整个道路网当前的交通状况<sup>[4]</sup>.如果在城市中部署足够数量的浮动车,并将这些浮动车的位置数据通过无线通信系统定期、实时地传输到一个信息处理中心,经过综合处理就可以获得整个城市动态、实时的交通拥堵信息<sup>[5]</sup>.

然而,作为交通信息源的浮动车数据(floating car data, FCD)具有大规模数据量的特点,这使得基于 FCD 的监控应用程序受到很大挑战:不仅需要处理大量的数据和查询,而且其数据的速率也是动态变化的<sup>[6]</sup>.

在此背景下,本文通过分析浮动车数据特性以

及交通拥堵特征,借鉴相似轨迹聚类的思想,提出拥堵同伴发现算法,并对 FCD 进行 2 次筛选,从而达到降载目的;此外,设计多优先级队列,对负载赋予不同的处理优先级别,以提高系统对重点监控区域变化的及时性;最后,以北京市为例,在大量浮动车数据的基础上,用实验验证了本文方法明显优于以往方法.归纳总结本文的主要贡献如下:

- 1) 提出了拥堵同伴的概念,并给出了定义;
- 2) 提出了拥堵同伴发现算法;
- 3) 设计基于预测的多优先级调度算法实现快速拥堵监控的整个监控流程.

## 1 问题描述

**定义 1.** 区域交通. 给定区域  $R$ 、密度  $k$ 、速度  $v$ 、流量  $q$ , 定义区域交通如式(1)所示:

$$T_R(t) = \langle R, k(t), v(t), q(t) \rangle. \quad (1)$$

一个显而易见的事实是,许多情况下移动对象并不是在二维平面中自由移动的,而是在一个空间嵌入网络中移动(例如公路、铁路等,包括飞机在正常情况下也都遵循规定的路线)<sup>[7]</sup>.早期的研究主要集中在自由空间中的移动对象数据库,他们假设对象的运动是不受约束的,而且是基于欧氏空间的;然而,在现实世界中,物体运动在空间受限的网络中,例如,车辆通常行驶在道路网络中<sup>[8]</sup>.因此,本文主要讨论在道路网空间中的移动对象,即区域  $R$  限定在路网空间中.为了适应不同的研究目标,需要对原有路网模型进行改进,路网模型用于描述区域内的道路网<sup>[9]</sup>,这里主要是针对道路.

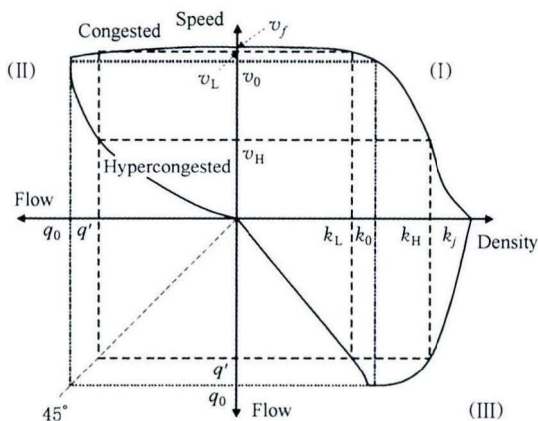
**定义 2.** 道路. 道路  $r$  对应于实际交通网络中的道路、匝道等,这里定义为如下形式:

$$r = (rid, geo, len, lev, Sta, CoC),$$

其中,  $rid$  为道路标识;  $geo$  为该条道路的地理几何形状,可用一条折线(polyline)表示;  $len$  为道路的长度;  $lev$  为道路级别;  $Sta$  为道路当前的交通状态;  $CoC$  为道路当前的拥堵同伴(见定义 3).

通常情况下,当浮动车报告信息时,每个 FCD 记录需要被匹配到道路网络;同时,与之相关道路的查询结果需要被更新.这通常采用交通拥堵模型来

实现,如图 1 所示<sup>[10]</sup>:交通流可由 3 个变量来描述,即密度( $k$ )、速度( $v$ )和流量( $q$ ),分别以每车道平均每千米有多少辆车、每小时多少千米和每车道平均每小时有多少辆车来测量;那么,密度大、流量小以及速度低的区域为拥堵区域。



(I) speed-density curve; (II) speed-flow curve;  
(III) flow-density curve

Fig. 1 The curve diagram of traffic flow.

图 1 交通流曲线图

此外,北京市地方标准《城市道路交通拥堵评价指标体系》<sup>[11]</sup>将道路等级划分为快速路、主干路、次干路和支路 4 个不同类型,将路段的交通状态分别定义为非常畅通、畅通、轻度拥堵、中度拥堵、严重拥堵。参照上述标准,本文将交通状态划分为:高拥堵、拥堵、临界拥堵和未拥堵;相应地,将区域  $R$  的交通状态分别定义为:高拥堵区域( $C_{OH}$ )、拥堵区域( $C_{OL}$ )、临界区域( $C_{CR}$ )和未拥堵区域( $L$ )。

当大量 FCD 记录涌入,并超过系统的处理能力时,处理延迟将增加,而实时监控数据的值也将随时间下降。由于及时性是实时应用服务质量(QoS)的一个重要的衡量,FCD 的一小部分有可能以监控精度降低的代价被丢弃<sup>[6]</sup>。在这种情形下,降载(load shedding)虽然降低了结果的准确性,但可以改善系统的性能。降载的关键是确定降载的方式和实施的降载计划。因此,研究的问题是如何决定多少和哪些 FCD 应该被丢弃,以平衡监控的准确性和及时性。

可以观察到如下事实:如图 2 所示,在时刻  $t_1$  和  $t_2$ ,有些区域的交通状态没有发生多大变化,直到某个时刻  $t_n$  才发生了明显的变化。举例来说:对于高拥堵区域  $C_{OH}$ ,短时间内(监控刷新时间间隔)其状态不会发生太大变化,尤其是当还有大量的车辆涌入该区域时;同样地,对于未拥堵区域  $L$ ,若没有大量的车辆涌入该区域,那么短时间内其状态也不

会发生太大变化。当拥堵发生时,最明显的特征就是交通参数的变化,如车速降低、通行能力减少、车流密度变大以及排队长度增加等。

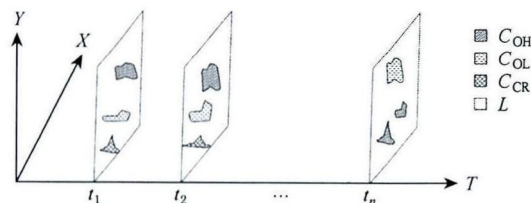


Fig. 2 Regional transportation state transitions.

图 2 区域交通状态转换示意图

更重要地是区域的交通状态是以增量方式变化的。因此,根据以前的状态以及交通参数的最新变化,可对区域的状态转换进行预测,这分为状态转换时间预测和区域状态预测,分别如式(2)和式(3)所示:

$$T_R(t_0) \xrightarrow{?} \begin{cases} C_{OH}(t_0 + \Delta t_1), \\ C_{OL}(t_0 + \Delta t_2), \\ C_{CR}(t_0 + \Delta t_3), \\ L(t_0 + \Delta t_4), \end{cases} \quad (2)$$

$$R(t_0) \rightarrow R'(t_0 + \Delta t), \quad (3)$$

式(3)中, $R$  代表原区域状态, $R'$  代表预测的区域状态。如果能够准确预测到各区域的状态转换时间和区域状态,那么无疑能以最有效的方式优化负载,达到最优的降载效果。然而,上述想法只是一种理想的状态,这主要是由于现有的预测技术通常是采用预测模型来实现的<sup>[12]</sup>,且大多数都存在模型比较复杂、开销大和成本高、而精确度低等问题。考虑到系统开销等因素,本文采用筛选出可能发生拥堵的那些 FCD,并以此概化估计拥堵区域的变化趋势,从而达到降载的目的。因此,本文所研究的问题是:1)降载方式,如何发现可能发生拥堵的那些 FCD;2)降载计划,如何设计与实现降载决策执行及快速监控。为了进一步说明,先作如下定义。

**定义 3. 拥堵同伴。**在浮动车轨迹流数据中,沿着同一道路且同一行驶方向上平均速度低于拥堵阈值的行进在一起的对象组,称为拥堵同伴(congestion companion)。

**定义 4. 拥堵同伴发现。**从浮动车轨迹流数据中发现拥堵同伴,称为拥堵同伴发现(congestion companion discovery, CoCD)。

本文采用相似轨迹聚类的思想,并考虑到与拥堵特征相关的交通参数,提出了拥堵同伴发现算法来筛选出可能发生拥堵的那些 FCD,并设计了基于预测的多优先级调度算法来实现降载决策执行及整个监控流程。

## 2 相关工作

由于在很多应用领域的实际重要性,流数据降载成为近来研究的一个热点问题.然而,现有的方法不能有效地处理道路网络的交通监控问题.例如文献[13]中,对于给定的一组形成查询网络的查询,具有最高成本的那些查询被逐渐丢弃,直到降载目标得到实现,此方法是基于查询的处理成本和降载目标来策略性地进行降载的,其目标是最大限度地提高处理的查询数量.本文优化的目标除了降载外,还需要实现尽可能高的精确度,而丢弃高成本负载的方法不能保证获得高的精确度;此外,在本文优化的负载中有着非常相似的处理成本.

文献[14]提出一种基于反馈循环来控制降载的方法.其降载目标取决于处理能力和输入的流数据:根据以前的数据,如果某个查询的准确性较低,那么这个查询的采样率对于即将到来的流数据会提高;否则应该降低.此外,该方法假定数据以相似的速率到达,即输入数据的速率是稳定的.然而,在本文的研究中速度变化起着很重要的角色.

文献[6]使用道路的速度变化作为降载决策的一个重要参数.仅仅当某条道路的速度发生了变化,才能触发对这条道路交通状况的更新操作.此方法将降载策略的关注点集中在数据属性,其优化的重点在于达到尽可能高的查询精确度.这一方法虽然与本文都旨在实现相同的降载目标,但它不能有效地解决本文的问题.这主要是由于速度变化仅为交通拥堵所呈现出来的表象之一,仅以速度这一数据属性作为降载决策是不太全面的.

交通拥堵呈现出的是一种群体特征,这使得在制定降载决策时,不仅需要考虑速度变化,还需要分析移动对象在路网中的运动模式.基于上述想法,本文采用相似轨迹聚类思想,并考虑到与拥堵特征相关的交通参数,从 FCD 轨迹流数据中发现可能发生拥堵的那些对象组,并以此实现对拥堵区域的概化预测,从而决定多少和哪些 FCD 数据应该被丢弃.相似轨迹聚类研究有很多,如 Convoys<sup>[15]</sup> 和 Swarms<sup>[16]</sup> 等.然而,它们中的大部分是设计工作在静态数据集中,有些方法还需要多次扫描数据,或者不能以增量方式输出结果.

文献[17]提出了基于检测到的异常值的时空属性构造异常值因果关系树的算法,这些因果关系树的频繁子结构不仅揭示时空异常值之间的循环相互

作用,而且还能发现当前交通网络设计的潜在缺陷,它针对时空交通数据中的异常值检测,主要是研究解决被检测到的异常值之间的因果关系发现.文献[18]提出了一个新颖的分区和组框架来找到基于子轨迹的集群.本文需要发现的是一种拥堵关系,即发现行驶(或拥堵)在一起的对象组.

文献[19]研究了从流轨迹中发现行进在一起的对象组(即旅行同伴)的问题.为了提高发现算法的效率,设计了一种称为旅行伙伴(traveling buddy)的数据结构,提出了基于伙伴的同伴发现方法.这种技术在科学研究、运输管理和军事侦察领域有广泛的应用,但它是基于欧氏空间和自由形状的对象组为研究背景的,且没有考虑交通参数,若要应用到路网空间的移动车辆,尚有不足之处.

## 3 拥堵同伴发现

为了发现同伴,监视系统应该聚类(cluster)每个快照的对象,并交叉(intersect)聚类结果,最终检索运动在一起的对象,即旅行同伴发现(traveling companions discovery, TrCD)有以下 3 个关键问题<sup>[19]</sup>:

- 1) 聚类(cluster). 同伴应该是运动在一起的对象.
- 2) 一致性(consistency). 同伴应该一致,并足以持续几个快照的时间.
- 3) 数量(size). 很多情况下,需要的是足够大的对象组;所以,对同伴的大小,即对象组中移动对象的数目可能会有要求.

与旅行同伴(traveling companions)相比,除了上述 3 个关键问题之外,拥堵同伴(congestion companion)还具有以下 2 个明显的特征:

- 1) 行驶方向相同(direction). 同一拥堵同伴中,每个成员具有相同的行驶方向.
- 2) 平均速度低于拥堵阈值(speed). 同一拥堵同伴中,所有成员的行驶速度的平均值低于拥堵阈值.

因此,在基于伙伴的 TrCD 算法的基础上<sup>[19]</sup>,结合交通拥堵特征,本文提出了拥堵同伴发现算法(见算法 1). 算法所涉及符号的含义如表 1 所示.

### 算法 1. 拥堵同伴发现算法.

输入: 数量阈值  $\delta$ 、持续时间阈值  $\delta_t$ 、拥堵阈值  $\delta_s$ 、候选集合  $H$ 、索引  $BI$ 、轨迹数据流  $S$ ;

输出: 合格的拥堵同伴  $q$ .

① for each coming snapshot  $s$  of  $S$  do  
② initialize new candidate set  $H'$ ;  
③ buddy based clustering; /\* 文献[19]算法 4 \*/  
④ update  $BI$  and corresponding candidates;  
⑤ for each candidate  $h_i$  in  $H$  do  
⑥ if  $\text{size}(h_i) < \delta_s$  then break; endif;  
⑦ for each cluster  $c_j$  in  $s$  do  
⑧  $h'_i \leftarrow \text{buddy-based-intersection}(h_i, c_j)$ ;  
⑨  $\text{duration}(h'_i) = \text{duration}(h_i) + \text{duration}(s)$ ;  
⑩ remove intersected objects and buddies from  $h'_i$ ;  
⑪ if  $\text{size}(h'_i) \geq \delta_s$  then add  $h'_i$  to  $H'$   
⑫ if  $\text{duration}(h'_i) \geq \delta_t$  then  
⑬ output  $h'_i$  as a qualified companion  
 $p_k$   
⑭ endif; endif;  
⑮ endfor;  
⑯ for each cluster  $c_j$  do  
⑰ if  $c_j$  is closed then add to  $R'$ ; endif;  
⑱ endfor; endfor;  
⑲  $H \leftarrow H'$ ;  
⑳ endfor;  
㉑ for each qualified companion  $p_k$  do 算法 2  
㉒ endfor;  
㉓ 算法 3

算法 1 首先得到符合要求的同伴(行⑪~⑱);其次,执行行驶方向聚类算法(行㉑),它对符合要求的每一个同伴,根据行驶方向再做一次聚类(cluster)操作,如果在一个同伴中,其成员的行驶方向不同,则这一同伴将会被拆分为多个,以使每个新同伴中的成员具有相同的行驶方向,并将符合大小阈值的新同伴作为候选拥堵同伴输出,见算法 2;最后,算法 1 将对候选拥堵同伴进行平均速度的判定(行㉓),它对每一个候选拥堵同伴计算其平均速度,并将平均速度不高于拥堵阈值  $\delta_c$  的候选拥堵同伴作为合格的拥堵同伴输出,见算法 3.

算法 2. 行驶方向聚类算法.

输入:候选拥堵同伴  $p_k$ 、数据阈值  $\delta_s$ ;  
输出:合格的拥堵同伴  $q'_n$ .

① for 每一个候选拥堵同伴  $p_k$  do  
② for  $i = \text{first}(p_k)$  to  $\text{size}(p_k)$  do

③ for  $j = i + 1$  to  $\text{size}(p_k)$  do  
④ if  $o_i.dc = o_j.dc$  then  
⑤ add  $o_j$  to  $q'_n$ ;  
⑥ remove object  $o_j$  from  $p_k$ ;  
⑦ endif; endfor;  
⑧ add  $o_i$  to  $q'_n$ ;  
⑨ if  $\text{size}(q'_n) \geq \delta_s$  then  
⑩ 输出合格的拥堵同伴  $q'_n$ ;  
⑪  $n++$ ;  
⑫ endif; endfor; endfor;

算法 3. 平均速度判定算法.

输入:候选拥堵同伴  $q'_n$ 、拥堵阈值  $\delta_c$ ;

输出:合格的拥堵同伴  $q$ .

① for 每一个候选拥堵同伴  $q'_n$  do  
② calculate average speed of  $q'_n$  to  $v_m$ ;  
③ if  $v_m \leq \delta_c$  then  
④ output  $q'_n$  as a qualified congestion companion  $q$ ;  
⑤ endif; endfor;

Table 1 List of Notations  
表 1 符号列表

Notation	Explanation	Notation	Explanation
$S$	The trajectory stream	$s$	The snapshots in stream
$C$	The cluster set	$c_j$	The clusters
$Q$	The congestion companion set	$h_i$	The companion candidates
$H$	The candidate set	$o_i, o_j$	The objects
$O$	The object set	$o_i.dc$	The direction of objects
$\delta_s$	The size threshold	$q$	The congestion companion
$\delta_t$	The duration threshold	$p_k$	The companions
$\delta_c$	The congestion threshold	$q'_n$	The congestion companion candidates

需要说明的是,对速度而言,道路级别不同,其拥堵临界值也不同,例如快速路上拥堵的临界值为 50 km/h,主干路上拥堵临界值为 30 km/h. 设定算法中的速度阈值为最高的拥堵临界值(如 50 km/h),则可将可能发生拥堵的所有级别的道路筛选出来. 这样,对于拥堵临界值低的那些道路,可能会将未发生拥堵的 FCD 数据也筛选出来;所以,对于筛选出来的数据还需要再做进一步的处理,下面将进行详细阐述.

#### 4 基于预测的多优先级调度算法

通过 CoCD 可从大量的轨迹流数据中筛选出可能发生拥堵的 FCD, 再将这些数据进行路网匹配后就可筛选出可能发生拥堵的道路。

对匹配到的道路, 将其原存储的  $Sta$  和  $CoC$  信息(分别记为  $OLD(Sta)$ ,  $OLD(CoC)$ )与其新发现的  $CoC$  信息(记为  $NEW(CoC)$ )进行比较, 即可对该区域的拥堵变化趋势作概化预测如下:

1) 当  $OLD(CoC) \neq \text{"null"}$  时: 如果  $OLD(CoC) \subseteq NEW(CoC)$ , 则说明该条道路的交通状态有恶化的趋势; 否则, 说明该条道路的交通状态有好转的趋势。

2) 当  $OLD(CoC) = \text{"null"}$  时: 说明该条道路有拥堵的趋势。

没有匹配到的道路, 说明其上未发现拥堵情况。因此, 这些道路的  $Sta$  新值(记为  $NEW(Sta)$ )均为“未拥堵”。对于这部分道路, 找出  $OLD(Sta) \neq \text{"未拥堵"}$  的刷新对应地监控信息, 并将其  $Sta$  更新为“未拥堵”即可。

此外, 区域的拥堵程度越高所关注度就越高, 监控信息的更新也要最快。因此, 处理各区域的先后顺序是有区别的。在本文中, 根据交通状态的级别赋予负载不同的优先级别: 对于级别高的其优先级也高, 将优先进行处理。为此, 设计了 4 个负载处理队列(如图 3 所示):

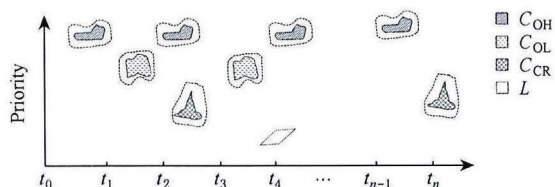


Fig. 3 Multi-priority scheduling based on prediction.

图 3 基于预测的多优先级调度

$COH$  队列(记为  $N_1$ ): 当前匹配到的道路中,  $OLD(Sta)$  为“高拥堵”的进入此队列;

$COL$  队列(记为  $N_2$ ): 当前匹配到的道路中,  $OLD(Sta)$  为“拥堵”的进入此队列;

$CCR$  队列(记为  $N_3$ ): 当前匹配到的道路中,  $OLD(Sta)$  为“临界拥堵”的进入此队列;

$L$  队列(记为  $N_4$ ): 当前匹配到的道路中,  $OLD(Sta)$  为“未拥堵”的进入此队列;

以上 4 个队列的优先级从高到低为:  $N_1, N_2, N_3, N_4$ 。此外, 设计了 1 个更新队列(记为  $N_5$ ), 用于

将交通状态有变化的(即需要刷新监控信息的)的区域(道路)插入此队列。由上所述, 可设计基于预测的多优先级调度算法(multi-priority scheduling based on prediction)(见算法 4)。

**算法 4.** 基于预测的多优先级调度算法。

输入: 拥堵同伴集合  $Q$ 、道路集合  $R$ ;

输出: 刷新监控信息。

```

① congestion companion discovery; /* 算法 1 */
② initialize new matched road set  $R'$ ;
③ initialize new queue  $N_1, N_2, N_3, N_4, N_5$ ;
④ for each FCD in  $Q$  do road network matching;
   endfor;
⑤ add all matched road  $r_i$  to  $R'$ ;
⑥ for each road  $r'_i$  of  $R'$  do
⑦   if  $OLD(r'_i, Sta) = \text{"OH"}$  then add  $r'_i$  to  $N_1$ ; endif;
⑧   if  $OLD(r'_i, Sta) = \text{"OL"}$  then add  $r'_i$  to  $N_2$ ; endif;
⑨   if  $OLD(r'_i, Sta) = \text{"CR"}$  then add  $r'_i$  to  $N_3$ ; endif;
⑩  if  $OLD(r'_i, Sta) = \text{"L"}$  then add  $r'_i$  to  $N_4$ ;
   endif;
⑪ endfor;
⑫ for each road  $r'_i$  of  $N_1$  do
⑬   if  $NEW(r'_i, CoC) \subseteq OLD(r'_i, CoC)$  then
     Call  $SCP(r'_i)$ ; endif;
⑭ endfor;
⑮ for each road  $r'_i$  of  $N_2$  do
⑯   if Not( $NEW(r'_i, CoC) = OLD(r'_i, CoC)$ )
     then Call  $SCP(r'_i)$ ; endif;
⑰ endfor;
⑱ for each road  $r'_i$  of  $N_3$  do
⑲   if Not( $NEW(r'_i, CoC) = OLD(r'_i, CoC)$ )
     then Call  $SCP(r'_i)$ ; endif;
⑳ endfor;
㉑ for each road  $r'_i$  of  $N_4$  do
㉒   if  $OLD(r'_i, CoC) \subseteq NEW(r'_i, CoC)$  then
     Call  $SCP(r'_i)$ ; endif;
㉓ endfor;
㉔ for each road  $r_i$  of  $R$  do
㉕   if ( $r_i \notin R'$ ) and ( $OLD(r_i, Sta) \neq \text{"L"}$ )
     then
㉖      $r_i, Sta = \text{"L"}; r_i, CoC = \text{"null"};$ 
㉗     add  $r_i$  to  $N_5$ ;
㉘   endif; endfor;

```



1)算法 4 进行 CoCD 算法筛选 FCD(行①);2)拥堵同伴中的 FCD 进行路网匹配(行④);3)对匹配到的道路,根据其  $OLD(Sta)$  将它们分别插入到 4 个负载处理队列中(行⑥~⑪);4)按多优先级调度算法,对 4 个负载处理队列中的每条道路分别进行处理(行⑪~⑲),  $N_1$  中对  $NEW(CoC) \subset OLD(CoC)$  的道路进行 SCP 处理(见算法 5),  $N_2$  中对除  $NEW(CoC)=OLD(CoC)$  之外的道路进行 SCP 处理,  $N_3$  中对除  $NEW(CoC)=OLD(CoC)$  之外的道路进行 SCP 处理,  $N_4$  中对  $OLD(CoC) \subset NEW(CoC)$  的道路进行 SCP 处理;5)对未匹配到的道路,将  $OLD(Sta) \neq$  “未拥堵”的道路插入到更新队列,并更新它们的  $Sta$  = “未拥堵”,  $CoC$  = “null”(行⑳~㉔)。

**算法 5.** 统计计算处理 (statistical calculation processing, SCP)算法。

- 输入:匹配道路  $r'_i$  ;  
输出:刷新监控信息。
- ① congestion model calculation for  $r'_i$  ;
  - ② if  $OLD(r'_i, Sta) \neq NEW(r'_i, Sta)$  then
  - ③  $r'_i, Sta = NEW(r'_i, Sta)$  ;
  - ④ add  $r'_i$  to  $N_5$  ;
  - ⑤ if  $NEW(r'_i, Sta) = "L."$  then
  - ⑥  $r'_i, CoC = "null"$  ;
  - ⑦ else
  - ⑧  $r'_i, CoC = NEW(r'_i, CoC)$  ;
  - ⑨ endif;endif;

算法 5 首先根据新采集的 FCD,按交通拥堵模型计算确定道路的交通状态(行①);其次,将状态发生变化的道路插入更新队列,并更新其  $Sta = NEW(Sta)$  (行②~④);最后,对  $NEW(Sta) =$  “未拥堵”的道路,更新它的  $CoC =$  “null”,否则  $CoC = NEW(CoC)$  (行⑤~⑧)。

此外,对更新队列进行监控,一旦有新值插入,则触发执行刷新对应的监控信息的操作。

通过这种方法可过滤掉大量的不需要处理的 FCD 数据,从而提高监控系统的性能,快速实现交通拥堵监控。

5 实验结果与分析

5.1 实验设计

为了验证本文算法,实验使用北京市的真实 GIS 地图数据以及北京市出租车过万的真实时空数

据。为了实验方便,仅选择了 3 级以上的道路。此外,通过对浮动车数据时间粒度的分析,以 5 min 为时间粒度的时间间隔数保留更为完整,仅有较少的数据缺失,最低缺失也只有 98.3%,而以 1 min 为时间粒度的时间间隔数最低比例则达到了 89.2%<sup>[1]</sup>。因此,实验采用的监控刷新时间间隔为 5 min。表 2 列出了实验的主要参数。实验主要对以下 3 种方法进行了测试对比:

**方法 1.** 每当 FCD 报告信息时,每个 FCD 记录被匹配到道路网络(称为路网匹配操作,记为 P1);同时,与之相关道路按交通流模型重新计算其密度、流量和平均速度(称为统计计算操作,记为 P2);最后,将计算结果分别与设定的阈值相比较,得出最新的交通状态,并更新交通监控信息(称为更新操作,记为 P3)。将这种方法记为 TM。

**方法 2.** P1;同时,与之相关道路计算其平均速度,并对速度发生变化的, P2; P3。将这种方法记为 VM。

**方法 3.** 对新报告的 FCD 数据,应用 CoCD 算法进行筛选,并对筛选出的 FCD 数据, P1;通过概化预测再作进一步筛选,并对筛选出的 FCD 数据, P2; P3。将这种方法记为 PM。

Table 2 Main Parameters of the Experiments  
表 2 模拟实验的主要参数

Parameter	Value	Meaning
$N_o$	13 000	Number of moving objects/floating cars
$N_r$	26 941	Number of routes in the traffic network
$\delta_s$	20~100, default 40	Congestion companion size threshold
$\delta_t/\text{min}$	5~15, default 10	Congestion companion duration threshold
$\delta_s/(\text{km} \cdot \text{h}^{-1})$	20~50, default 50	Congestion threshold/Speed thresholds
Duration/s	57 600~14 4000	Duration of the simulation runs

5.2 实验结果与分析

本文实验首先评估了 3 种方法的时间成本,图 4 所示为实验结果。从图 4 可看出,PM 方法达到了最佳性能。这主要是由于 PM 只对筛选出来的负载进行处理,故所需的计算开销大大降低,特别是路网匹配的计算量;而 TM 需要为每一个 FCD 付出计算成本;VM 虽然也通过速度变化进行了筛选处理,但它耗费了与 TM 一样的路网匹配计算量。在交通拥堵

监控中,很大的一个瓶颈就在于路网匹配的计算量,而 PM 方法通过计算开销较低的 CoCD 筛选后再进行路网匹配,使这一计算量大大降低,故在 3 种方法中消耗的时间成本最低。

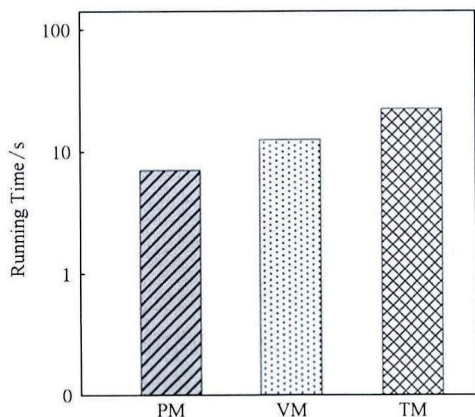


Fig. 4 Efficiency analysis.

图 4 时间效率分析

3 种方法的有效性分析如图 5 所示。从图 5 可以看出, TM 方法的精确度和召回率最高,这是因为它计算了每一个 FCD 数据,没有遗漏故而准确度最高。与之相比, PM 方法虽略有不足,这主要是由于筛选数据引入的误差,但其精确度和召回率都能满

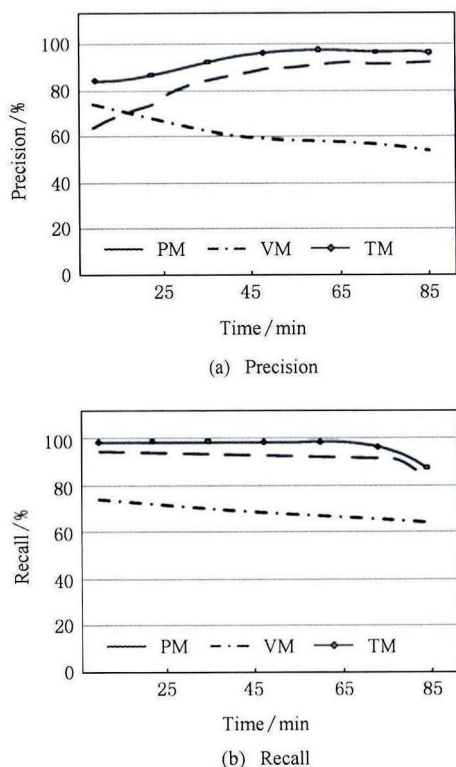


Fig. 5 Effectiveness analysis.

图 5 有效性分析

足实际应用的要求。与之相比, VM 方法略显不足。

因此,通过实验证明我们提出的方法明显优于以往的方法,并且是以损失百分率很小的精确度和召回率换来了系统性能的大大提升。此外,通过 PM 明显优于 VM,说明仅以速度这一数据属性作为降载决策还是不太全面,这主要是由于速度变化仅为交通拥堵所呈现出来的表象之一。

### 5.3 CoCD 参数影响分析

$\delta_s$ ,  $\delta_t$  和  $\delta_c$  是 3 个很重要的参数,它们直接影响到 CoCD 算法的性能。图 6 显示了  $\delta_s$  对算法的影响,我们取它的不同值评估了算法。一般来说,当  $\delta_s$  的取值越来越大时,在每个快照将会删除更多的候选拥堵同伴,从而使过滤机制就更有效;此外,空间成本会显著降低,运行时间也会由于交叉操作的减少而降低。

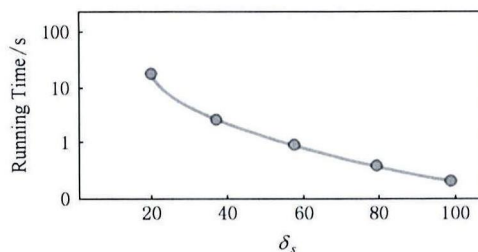


Fig. 6 Impact analysis of the parameter  $\delta_s$ .

图 6  $\delta_s$  参数影响分析

我们也取不同的  $\delta_t$  值评估了算法,图 7 显示了它对算法的影响。当  $\delta_t$  的取值变大时,由于很多候选拥堵同伴不能保持足够长时间的一致性,从而使 CoCD 算法的运行时间会减少。

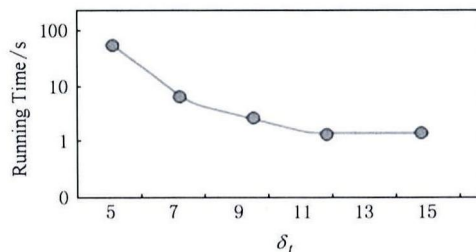


Fig. 7 Impact analysis of the parameter  $\delta_t$ .

图 7  $\delta_t$  参数影响分析

同样地,我们取不同的  $\delta_c$  值评估了算法,图 8 显示了它对算法的影响。当  $\delta_c$  的取值变大时,由于拥堵阈值的提高会产生更多的候选拥堵同伴,从而使 CoCD 算法的运行时间会增加。

由于 TM 和 VM 方法不受这 3 个参数的影响,故这里只讨论了它们对 CoCD 的影响。



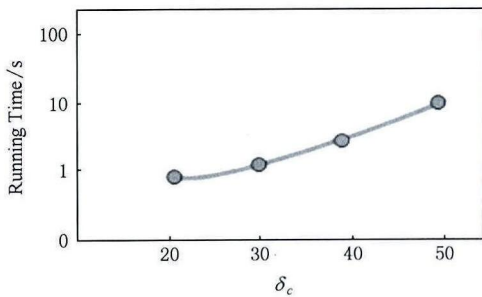


Fig. 8 Impact analysis of the parameter  $\delta_c$ .

图 8  $\delta_c$  参数影响分析

### 5.4 调度策略影响分析

在基于预测的多优先级调度算法中,我们采用了多优先级队列,优先级高的队列优先处理,同一队列按轮询的策略来进行调度.今后,我们将尝试一些不同的策略,如启发式调度策略等,以进一步提高系统性能.

此外,我们下一步将在拥堵同伴的发现上进行深入研究和实验,以期进一步提高算法的有效性.

## 6 结束语

通常情况下(非高峰时间)发生拥堵的区域是很有限的.因此,本文首先筛选出可能发生拥堵的FCD,再作进一步处理,这样大大降低了系统处理负载的开销.本文将同伴发现技术引入交通拥堵监控中,为解决此类应用中存在的问题作了一定的尝试.今后,还将作进一步的深入研究,挖掘出更多的数据属性,用以解决大规模轨迹流数据所带来的诸多问题.

### 参 考 文 献

[1] He Weinan. Spatial-temporal distribution characteristics of urban recurrent traffic congestion based on floating car data [D]. Beijing: Institute of Transportation, Beijing Jiaotong University, 2012 (in Chinese)  
(何巍南. 基于浮动车数据的城市常发性交通拥堵时空分布特征研究[D]. 北京: 北京交通大学交通运输学院, 2012)

[2] Xin Feifei, Chen Xiaohong, Lin Hangfei. Study on space-time distribution characteristics of floating car data based on large samples [C] //Proc of 2010 Int Conf on Optoelectronics and Image Processing. Piscataway, NJ: IEEE, 2010: 449-452

[3] Tang Luan, Zheng Yu, Xie Xing, et al. Retrieving  $k$ -nearest neighboring trajectories by a set of point locations [C] //Proc of the 12th Int Symp on Spatial and Temporal Databases. Berlin: Springer, 2011: 223-241

[4] Lü Weifeng, Wu Dongdong, Zhu Tongyu. A heuristic path-estimating algorithm by using vector-based recognition [J]. Chinese Journal of Computers, 2009, 32(7): 1443-1450 (in Chinese)  
(吕卫锋, 吴东东, 诸彤宇. 基于向量识别的启发式路径推测算法[J]. 计算机学报, 2009, 32(7): 1443-1450)

[5] Yuan Jing, Zheng Yu, Zhang Chengyang, et al. T-drive: Driving directions based on taxi trajectories [C] //Proc of the 18th SIGSPATIAL Int Conf on Advances in Geographic Information Systems. New York: ACM, 2010: 99-108

[6] Liu Kuiren, Deng Ke, Ding Zhiming, et al. MOIR/MT: Monitoring large scale road network traffic in real-time [J]. Proc of the VLDB Endowment, 2009, 2(2): 1538-1541

[7] Güting R H, Schneider M. Moving Objects Databases [M]. Translated by Jin Peiquan, Yue Lihua. Beijing: Higher Education Press, 2009 (in Chinese)  
(Güting R H, Schneider M. 移动对象数据库[M]. 金培权, 岳丽华, 译. 北京: 高等教育出版社, 2009)

[8] Meng Xiaofeng, Chen Jidong. Moving Objects Management: Models, Techniques and Applications [M]. Beijing: Tsinghua University Press, 2010

[9] Ding Zhiming, Han Jingyu, Li Man, et al. Network-constrained moving objects database based traffic flow statistical analysis model [J]. Journal of Computer Research and Development, 2008, 45(4): 646-655 (in Chinese)  
(丁治明, 韩京宇, 李曼, 等. 基于网络受限移动对象数据库的交通流统计分析模型[J]. 计算机研究与发展, 2008, 45(4): 646-655)

[10] Lindsey C R, Verhoef E T. Congestion modelling [OL]. Amsterdam: Tinbergen Institute, 1999. [2013-03-22]. <http://papers.tinbergen.nl/99091.pdf>

[11] Beijing Transportation Research Center. DB11/T 785—2011 Provincial standard of Beijing: Indices system of urban road traffic congestion evaluation [S]. Beijing: Beijing Municipal Administration of Quality and Technology Supervision, 2011 (in Chinese)  
(北京交通发展研究中心. DB11/T 785—2011 北京市地方标准: 城市道路交通拥堵评价指标体系[S]. 北京: 北京市质量技术监督局, 2011)

[12] Tawara K, Mukai N. Traffic signal control by using traffic congestion prediction based on pheromone model [C] //Proc of 2010 Int Conf on Tools with Artificial Intelligence. Piscataway, NJ: IEEE, 2010: 27-30

[13] Tatbul N, Çetintemel U, Zdonik S, et al. Load shedding in a data stream manager [C] //Proc of the 29th Int Conf on Very Large Data Bases. San Francisco: Morgan Kaufmann, 2003: 309-320

[14] Babcock B, Datar M, Motwani R. Load shedding for aggregation queries over data streams [C] //Proc of the 20th Int Conf on Data Engineering (ICDE'04). Piscataway, NJ: IEEE, 2004: 350-361

- [15] Jeung H, Yiu M L, Zhou Xiaofang, et al. Discovery of convoys in trajectory databases [J]. Proc of the VLDB Endowment, 2008, 1(1): 1068-1080
- [16] Li Zhenhui, Ding Bolin, Han Jiawei, et al. Swarm: Mining relaxed temporal moving object clusters accurate discovery of valid convoys from moving object trajectories [J]. Proc of the VLDB Endowment, 2010, 3(1): 723-734
- [17] Liu Wei, Zheng Yu, Chawla S, et al. Discovering spatio-temporal causal interactions in traffic data streams [C] //Proc of the 17th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2011: 1010-1018
- [18] Lee J G, Han J, Whang K Y. Trajectory clustering: A partition-and-group framework [C] //Proc of the 2007 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2007: 593-604
- [19] Tang Luan, Zheng Yu, Yuan Jing, et al. On discovery of traveling companions from streaming trajectories [C] //Proc of the 28th Int Conf on Data Engineering (ICDE'12). Piscataway, NJ: IEEE, 2012: 186-197



**Wu Peili**, born in 1975. PhD candidate in Beijing Institute of Technology. Her main research interests include database systems, spatio-temporal data management, and information security.



**Liu Kui'en**, born in 1983. PhD and assistant professor in the Institute of Software, Chinese Academy of Sciences. Member of China Computer Federation. His main research interests include spatio-temporal data management, moving objects database and data mining.



**Hao Shengang**, born in 1977. Master. His research interests include information security and computer network.



**Zhang Quanxin**, born in 1974. PhD, member of China Computer Federation. His main research interests focus on storage algorithm and mobile computing.



**Tan Yu'an**, born in 1972. PhD, professor and PhD supervisor in Beijing Institute of Technology. Senior member of China Computer Federation. His main research interests include network storage, computer information security, embedded system.