

# 无线移动感知网络上的数据聚集传输规划

冯 诚 李治军 姜守旭

(哈尔滨工业大学计算机科学与技术系 哈尔滨 150001)

**摘 要** 随着城市交通日益增多,智能交通系统研究受到工业界和学术界的极大关注.智能交通系统上的实时导航、交通监控等应用都需要大量实时车辆速度、位置等信息.由智能车组成的无线移动感知网络的链路变化频繁并且无线干扰严重,现有的车辆网络上的数据聚集算法没有考虑无线链路变化的具体情况,就进行数据聚集和传输的规划.文中分析并求解了移动感知网络上的无冲突数据聚集传输规划问题,将该问题形式化为移动通信拓扑图上的单传聚集传输规划问题,并证明了该问题是 NP 完全的;提出了移动聚集路由树,将无线冲突分为树内冲突和树间冲突,首先构建树间冲突图描述树间冲突,过滤节点的传输候选时刻集以消除树间冲突,然后利用动态规划思想进行聚集路由树上的数据聚集传输时刻规划来消除树内冲突.并且在真实出租车移动轨迹数据集上进行了实验,实验结果表明文中的算法比现有的算法在数据收集率上提高了 1/4 左右,并且收集上来的数据的平均延迟也更小.

**关键词** 无线移动感知网络;车联网;数据聚集;动态规划;数据收集

**中图法分类号** TP301 **DOI 号** 10.3724/SP.J.1016.2015.00685

## Data Aggregation Scheduling on Wireless Mobile Sensor Networks

FENG Cheng LI Zhi-Jun JIANG Shou-Xu

(Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

**Abstract** As urban traffic increases rapidly, the research in Intelligent Transportation Systems gets more and more attentions from the industry and academia. Real time navigations and traffic monitoring applications in ITS require a large number of real-time speed and location information. Such wireless mobile sensor networks composed by vehicles in ITS take such frequent change of wireless links and severe radio interference. Therefore, data collection algorithm in wireless mobile sensor networks should effectively use dynamic wireless links. This paper studies non-conflict data aggregation scheduling problem in wireless mobile sensor networks. The problem is formulated and proved as a NPC problem. Through analyzing inter-subtree conflict and inner-subtree conflict, part interference graph is constructed to filter candidate transmission time sets in order to remove inter-subtree conflicts. Then an algorithm based on dynamic programming is proposed to schedule transmission times on mobile aggregation routing tree. We conduct an extensive experimental study based on real taxi trajectory data sets. Experimental results show that our algorithm increases about 1/4 on the data collection rate compared with the existing VANET data aggregation algorithms and makes average delay less.

**Keywords** wireless mobile sensor network; Internet of Vehicles; data aggregation; dynamic programming; data collection

## 1 引 言

近年来,移动感知网络(Wireless Mobile Sensor Networks)<sup>①</sup>成为无线通信领域继无线传感器网络(Wireless Sensor Networks<sup>[1]</sup>)之后的又一新研究热点.无线移动感知网络由移动感知节点、静态骨架节点(Access Point)<sup>②</sup>以及处理中心构成,用于监测动态环境中的事件.移动感知节点实时产生感知数据,对数据进行初步处理,然后通过其他移动节点的携带和转发,将数据路由到 AP 节点. AP 节点稀疏地部署在监测区域,它们之间以及与处理中心之间通过有线网络进行互联.处理中心通过 AP 节点向移动感知节点派发查询或控制请求,收集移动节点感知的实时数据,并做进一步的数据处理和分析.车辆网络(Vehicular Ad-hoc NETworks)<sup>[2]</sup>就是一种典型的无线移动感知网络,它为智能交通系统(Intelligent Transportation System)<sup>[3]</sup>的应用提供了实现平台.在许多 ITS 应用<sup>[4-5]</sup>中,实时路况信息是许多应用计算的基础,实时路况信息体现在路段上的车辆的平均速度、最大速度等聚集信息.对实时路况信息的收集也是 VANET 研究的关键点.这些聚集信息通常采用网内数据聚集的方式进行收集.

现有的数据聚集算法<sup>[6-18]</sup>主要集中在 WSN 上. WSN 上的传感器节点主要是静态节点,因此 WSN 上的数据聚集算法主要是针对静态无线通信拓扑设计的.现有研究中也有将移动节点引入 WSN,但这些移动节点通常是少量由人工操纵的专用于收集信息的节点,而大量的感知节点仍是静态的<sup>[11-12]</sup>.然而在移动感知网络中大量的感知节点不断地快速移动,少量的固定 AP 节点收集实时感知信息.感知节点不断移动使得节点间的通信拓扑关系不断变化,在这种动态通信拓扑网络环境下 WSN 上的数据聚集算法无法适用.目前在无线移动感知网络上也有一些数据聚集的研究<sup>[19-35]</sup>.这些研究主要是进行协议的设计,通过参数控制调整路由模式,从而优化数据聚集性能.这些方法没有利用整个动态拓扑的细节信息,没有最大程度地优化数据聚集的效率.因此,我们从整个通信拓扑图序列出发,优化聚集传输时刻使得 AP 节点利用最小的传输带宽收集最多的聚集数据.

本文研究的场景是由一个 AP 节点和  $n$  个移动节点构成的移动感知网络.车辆在移动的过程中不断产生速度、位置等实时感知数据.车辆节点携带并

转发这些数据,同时进行网内数据聚集,然后将这些聚集信息路由到 AP 节点. AP 节点周期性地收集这些聚集信息,再将这些信息汇总到处理中心,做进一步应用计算.在该场景下节点之间的通信采用一跳冲突模型,即同一通信半径内的多个节点不能同时传输数据.图 1 显示了由 8 个移动感知节点和 1 个 AP 节点组成的无线移动感知网络从 1 时刻到 4 时刻的网络拓扑变化情况以及数据聚集传输过程.图中  $d_i$  表示节点  $i$  的原始数据,  $ad_{e,f}$  表示节点  $e$  和节点  $f$  的数据聚集后得到的聚集数据,  $ad_{e,f,h,g}$  表示节点  $e$ 、节点  $f$ 、节点  $g$  和节点  $h$  的数据聚集后得到的聚集数据,带箭头的虚线表示通过无线进行数据传输.从图 1 中可以看出, AP 节点固定设置在路边,其他节点不断向右移动,它们通信的拓扑结构不断变化.图中所示的传输规划使得在 4 时刻, AP 节点收集到了 8 个移动节点在 0 时刻产生的感知数据信息.

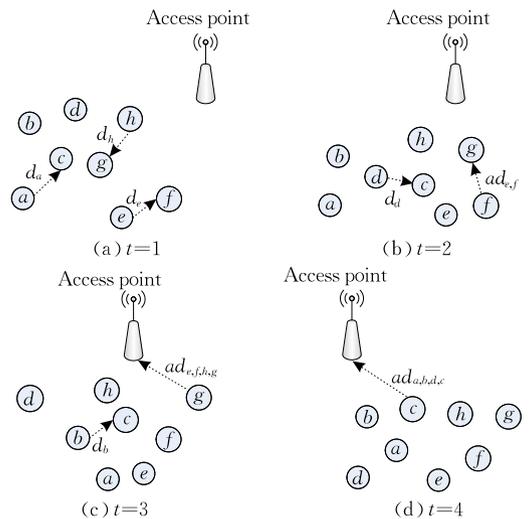


图 1 无线移动感知网络上的数据聚集

本文研究的问题即是规划移动节点数据的路由路径和数据聚集传输时刻,使得 AP 节点收集最多的聚集信息.这里我们需要考虑两个子问题:一个是移动节点选择哪个节点将数据路由到 AP 节点,另一个是在避免节点之间通信冲突的情况下移动节点应该在什么时刻发送数据.对于第一个问题,我们在移动通信拓扑图上建立移动路由聚集树以确定每个节点发送数据的接收节点,利用求解最小生成树的 Prim 算法<sup>[36]</sup>构造移动路由聚集树.本文主要研究的是第二个问题,即在给定的移动路由聚集树上如

① Mobile wireless sensor networks. [http://en.wikipedia.org/wiki/Mobile\\_wireless\\_sensor\\_network](http://en.wikipedia.org/wiki/Mobile_wireless_sensor_network), 2012, 7, 25

② Access Point. [http://en.wikipedia.org/wiki/Wireless\\_access\\_point](http://en.wikipedia.org/wiki/Wireless_access_point), 2012, 7, 25

何规划移动节点的无冲突传输时刻,使得 AP 节点收集最多的聚集信息. 本文的基本思想是根据移动路由聚集树将冲突分成树间冲突和树内冲突,先根据树间边提取出树间冲突,对节点的候选传输时刻集进行过滤,消除树间冲突,然后考虑树内冲突,利用动态规划计算聚集传输的优化时刻.

本文的贡献包括以下几个方面:

(1) 形式化定义了无线移动感知网络上的无冲突数据聚集传输规划问题,并分析了该问题是 NP 完全的.

(2) 提出了求解该问题的算法,主要分成两个步骤,首先消除树间冲突,对节点的候选时刻集进行过滤,然后在过滤后的候选时刻集上进行数据聚集传输规划.

(3) 为消除树间冲突,提出了基于时间段的过滤候选时刻集算法,但是该算法粒度较粗,使得过滤后的候选时刻集较小,导致后续树上的无冲突传输规划的解空间太小,因此提出了基于时间片过滤候选时刻集的改进算法.

(4) 为消除树内冲突,提出了基于动态规划的数据聚集传输规划算法.

(5) 在真实的出租车轨迹数据集上进行了大量实验. 实验结果表明本文提出的算法比现有的算法大幅提高了数据收集率,并且数据的平均延迟也更小.

本文第 2 节介绍 VANET 以及 WSN 上数据聚集的相关工作. 第 3 节描述本文采用的系统模型,第 4 节形式化了  $\Pi$  问题,并对该问题进行简化得到等价的  $\Pi'$  问题,分析该问题是 NP 完全的. 第 5 节给出求解  $\Pi'$  问题的近似算法 PTSDP,针对消除树间冲突,提出构建树间冲突图的 createPIG 算法以及过滤节点候选传输时刻集的 PTS 算法,针对消除树内冲突,提出基于动态规划思想的 DAS-MRT 算法. 第 6 节在真实出租车轨迹数据集上进行仿真实验并对实验结果进行分析. 最后总结全文.

## 2 相关工作

现有的数据聚集研究主要集中在 WSN 上. 关于数据聚集的研究主要考虑时延、聚集数据精度、能量消耗、聚集信息量等因素<sup>[13]</sup>. 由于本文研究的聚集问题与时延和聚集信息量相关,我们主要考虑与这两个方面相关的聚集算法. WSN 上与时延相关的聚集算法的研究包括两个问题,一个是最小化最大聚集延迟问题(MDAT),另一个是在给定的截止

时间内最大化 sink 节点聚集信息量问题(DCDA). MDAT 问题是由 Chen 等人<sup>[6]</sup>最先提出的,并且被证明了该问题是 NP 完全的. 文献[6]给出了 SDA 算法,并证明了该算法的近似比为  $\Delta-1$  (其中  $\Delta$  为节点的最大度数). 现有的关于 MDAT 问题的研究主要思路是先确定路由树,然后在该树上规划无冲突传输. 文献[7]基于最小生成树提出分布式数据聚集规划的算法,并分析了该算法能够达到的上界为  $24R+6\Delta+16$ . 文献[8]提出了基于 Penoy 树的数据传输规划算法,并分析了该算法能够达到的上界为  $15R+\Delta-15$ . 文献[9]分析得出路由树的结构会影响聚集传输规划的性能这一结论,因此提出了同时进行建树和规划的算法. 对于 DCDA 问题最早的研究是由 Hariharan 等人<sup>[15]</sup>提出的. 文献[15]研究的问题是在树型拓扑的 WSN 上规划无冲突的数据传输,使得截止时间内 sink 节点聚集数据量最多. 该文利用二部图最大权值匹配问题的求解,给出了基于动态规划思想的优化算法. 文献[16]研究的场景与文献[15]中的场景基本相同,唯一不同点是文献[15]中每个节点只有一个数据需要进行路由和聚集,而文献[16]中每个节点有多个数据需要路由和聚集,因此这里既要规划节点的传输时刻还要规划每个节点要传输多少数据,即每次传输占用的时间槽数. 该文分析了这一问题为强 NP 难的,并且利用单服务器任务分配近似算法构造基于动态规划思想的规划算法. 文献[18]研究的问题的优化目标与上述不同,这里优化目标是最大化节点子集上的单调压模函数值,它分析了该问题为强 NP 难问题,该文给出了数据转发模式下的常数近似比的近似算法以及数据聚集模式下的双准则近似算法,近似比为  $1/(\min(h, D)+1)$  (其中  $h$  为树高,  $D$  为截止时刻).

目前关于 VANET 上的数据聚集算法的研究主要集中在两个方面,一个是权衡聚集数据的精度和聚集收益的优化<sup>[19-23]</sup>,另一个是权衡延迟和聚集收益的优化<sup>[24-26]</sup>. 聚集收益包括数据聚集率、收集的数据量、节省的传输带宽消耗等方面. Traffic-View<sup>[19]</sup>和 SOTIS<sup>[20]</sup>研究节点上哪些数据进行聚集使得保证聚集精度的前提下最小化聚集数据量. 文献[21-22]引入模糊逻辑,根据建立的模糊规则对数据进行聚集. 文献[23]的研究是针对停车位这类离散信息的应用,它提出的方法只适用于离散性数据的收集,不适用于收集速度、位置等连续性数据. 文献[24-25]提出了基于分簇思想的无损数据压缩的算法. CatchUp<sup>[26-27]</sup>算法利用分布式马尔可夫模型

决策节点数据的转发时刻以最小化传输消耗网络带宽. 文献[28]研究的是限定截止时间内的优化收集聚集信息量问题, 它提出 DB-VDG 协议, 该协议通过调整数据传输过程中的转发和携带策略使得在截止时间内 AP 节点收集的数据量最多.

还有一些其他的关于 VANET 上数据聚集的研究工作. Dietzel 等人<sup>[29]</sup>给出了 VANET 上数据聚集模型, 对不同类型的数据、不同的聚集操作等建立统一的聚集过程模型. 文献[30]研究可扩展的数据聚集模式需要满足的理论性能. Giordano 等人<sup>[31-32]</sup>提出 CORNER 系统用于仿真 VANET 上数据的收集传输. 文献[33]利用 MySQL 数据库构建了用于在 VANET 上收集聚集数据的平台 CANET. 文献[34]提出的 VESPA 平台用于 VANET 上的数据收集和分享.

### 3 系统模型

#### 3.1 动态网络及传输模型

我们考虑基于时间片时钟同步系统. 首先我们对车辆节点的移动进行建模. 设车辆节点  $i$  在时刻  $j$  处于  $(x_i^{(j)}, y_i^{(j)})$  处, 随着车辆节点的移动, 车辆  $i$  在各个时间槽所处位置序对形成车辆  $i$  的移动轨迹  $Trad_i = ((x_i^{(0)}, y_i^{(0)}), (x_i^{(1)}, y_i^{(1)}), \dots, (x_i^{(D)}, y_i^{(D)}))$ . 对于节点之间的无线通信我们采用圆盘通信模型. 车辆  $i$  与车辆  $j$  在  $k$  时刻可以进行通信当且仅当在  $k$  时刻节点  $i$  和节点  $j$  之间的欧氏距离小于无线通信半径  $R$ . 假设已知  $n$  个车辆的移动轨迹, 根据移动轨迹集以及圆盘通信模型, 可以得到  $D$  个时刻的通信拓扑图序列. 然后我们利用通信拓扑图序列构造移动通信拓扑图. 移动通信拓扑图涵盖了  $D$  个时刻的节点之间的连通关系. 下面给出移动通信拓扑图的定义.

**定义 1.** 移动通信拓扑图 MCG. 图  $G=(V, E)$  表示一个移动通信拓扑图, 其中  $V$  为节点集, 包括  $n$  个移动感知节点和一个 AP 节点;  $E$  为链路集,  $E = \{(i, j, CT_{ij}) | i, j \in V\}$ , 其中  $i, j$  为两个移动节点,  $CT_{ij}$  是节点  $i$  与节点  $j$  之间的链路连通时刻集,  $CT_{ij} = \{k | \sqrt{(x_i^{(k)} - x_j^{(k)})^2 + (y_i^{(k)} - y_j^{(k)})^2} \leq R\}$ .  $N_i$  表示节点  $i$  在移动通信拓扑图  $G$  上的邻居节点集. 图  $G$  上的边表示一条链路, 这里假设每个链路的容量为常数, 不失一般性, 设该值为 1. 假设被调度节点需要传输的数据量不超过通信链路的容量.

#### 3.2 树状路由及冲突模型

在给定的移动通信拓扑图上移动节点进行聚集传输数据时, 我们采用以 AP 节点为根的树型结构进行数据路由和聚集. 当移动节点到 AP 节点存在可达路径时, 即可建立树型结构. 节点  $u$  到节点  $v$  存在可达路径指的是存在一条路径  $P_{uv} = (u, e_1, v_1, e_2, \dots, e_k, v)$  且该路径上的链路存在一个递增的连通时刻序列, 即  $\exists t_1, t_2, \dots, t_k, t_1 \in CT_{e_1}, t_2 \in CT_{e_2}, \dots, t_k \in CT_{e_k}$  使得  $t_1 < t_2 < \dots < t_k$ .  $t_k$  为路径  $P_{uv}$  的可达延迟. 下面我们通过移动节点到 AP 节点的可达路径建立移动聚集路由树.

**定义 2.** 移动聚集路由树 MRT. 移动聚集路由树是由所有移动节点到 AP 节点的可达路径构成的树, 表示为  $T=(V, E^T)$ , 其中  $V$  为节点集,  $E^T$  为树  $T$  的边集.  $C_i$  表示节点  $i$  在树  $T$  上的儿子节点集,  $p_i$  表示节点  $i$  在树  $T$  上的父节点.

节点在数据传输过程中, 由于无线干扰会产生冲突. 本文采用一跳冲突模型, 即不能将具有公共节点的两条链路在同一时间激活(包括发送数据和接收数据). 一跳冲突模型包含以下 3 条规则: 一个节点不能在发送数据的同时接收数据; 一个节点不能同时接收来自两个节点的数据; 当一个节点接收数据时, 在它的通信半径内不能有其他节点对正在进行通信. 该模型被广泛应用在蓝牙或 FH-CDMA 系统中.

对于一跳冲突模型, 我们进一步分析可知, 冲突包括两种情况, 一种是节点不能同时接收多个数据, 即同一父节点的多个儿子节点不能同一时刻向父节点发送数据, 同时父节点发送数据的时刻不能和儿子节点发送数据的时刻相同, 这种冲突被称为树内冲突; 另一种是节点在接收数据时不能受到其他发送数据的节点的影响, 即接收节点不能在其他发送节点的通信范围内, 这种冲突被称为树间冲突. 图 2 (a) 所示的情况即是树内冲突, 如图所示节点  $u$  和节点  $v$  有共同的父节点  $x$ , 如果节点  $u$  和节点  $v$  同时

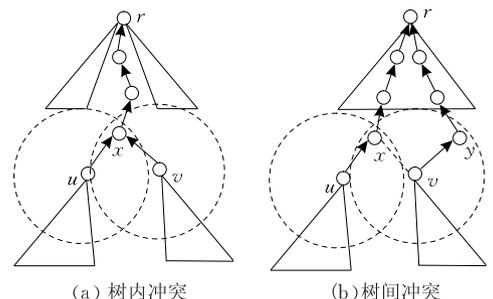


图 2 两种冲突情况

给节点  $x$  发送数据,那么在节点  $x$  处会发生冲突,节点  $x$  无法同时接收节点  $u$  和节点  $v$  的数据. 树间冲突的情况如图 2(b)所示. 节点  $u$  给父节点  $x$  发送数据,节点  $v$  给父节点  $y$  发送数据,但节点  $x$  在节点  $v$  的通信半径内,如果节点  $u$  和节点  $v$  同时发送数据,那么在节点  $x$  处会受到节点  $v$  传输数据的干扰,无法接收到节点  $u$  的数据.

### 3.3 传输规划及数据聚集

在数据聚集传输的过程中,节点发送数据的时刻决定传输是否存在冲突以及 AP 节点最终能收集到的聚集信息量. 因此规划数据聚集传输时刻,即规划节点发送数据的时刻. 节点可以多次发送数据,所以对于每个节点  $i$ ,聚集传输规划得到一个时刻集  $W(i)$ . 因此一个聚集传输规划  $W$  是从移动节点集到时刻子集的映射. 下面我们给出聚集传输规划的定义.

**定义 3.** 聚集传输规划. 一个聚集传输规划是映射  $W: V/\{AP\} \rightarrow 2^{CT}$ , 其中  $CT = \bigcup \{CT_e \mid e \in E^T\}$ ,  $E^T$  为移动聚集路由树  $T$  的边集. 节点  $i$  的  $W(i)$  表示通过聚集传输规划  $W$  得到的发送数据的时刻集,当  $W(i)$  为空集时,表示规划节点  $i$  不发送数据.

在发送数据之前节点要执行数据聚集操作. 我们假设这里的聚集操作为可分函数. 可分函数指的是可以分治计算的函数,例如 MIN、MAX、SUM 等. 这些操作也是数据聚集中常常出现的操作. 同时假设在节点上进行聚集计算的时间是可以忽略的,即不占用单独的时间片.

在初始时刻,每个移动节点产生一个感知数据,即为原始数据. 一个原始数据的信息量为 1. 然后节点在传输数据的过程中通过聚集操作得到聚集数据. 一个聚集数据包含的信息量是参与该聚集数据计算的原始数据的个数,即相关的移动节点的个数. 一个节点上的累积的聚集信息量为该节点原始数据的信息量和该节点接收到的所有聚集数据的信息量之和. 我们下面给出节点上的累积聚集信息量的定义.

**定义 4.** 节点上的累积聚集信息量. 该值是节点  $i$  原始数据的信息量和节点  $i$  在聚集传输规划  $W$  下接收的聚集数据的信息量之和,记为  $M_i(W)$ .  $M_i(W)$  定义如下:

$$M_i(W) = \begin{cases} 1, & i \in V^L \\ 1 + \sum_{j \in C_i} |W(j)| M_j(W), & i \in V/V^L \end{cases} \quad (1)$$

其中  $V^L$  为移动聚集路由树  $T$  中的叶节点集合,  $C_i$  为节点  $i$  的儿子节点集.

## 4 问题定义及分析

### 4.1 问题定义

无线移动感知网络上的数据聚集传输规划问题是已知  $n$  个移动节点和 1 个 AP 节点构成的移动通信拓扑图  $G=(V, E)$ , 以 AP 节点为根的移动聚集路由树  $T=(V, E^T)$ , 在  $T_0$  时刻移动节点  $i$  产生感知数据  $d_i$ , 求聚集传输规划  $W$ , 使得满足数据传输无冲突的情况下最大化 AP 节点收集的聚集信息量  $M_{AP}(W)$ . 我们将无线移动感知网络上的数据聚集传输规划问题进行形式化定义, 记为  $\Pi$  问题.

#### $\Pi$ 问题

$$\text{Maximize } M_{AP}$$

$$\text{Subject to } W(i) \in 2^{CT_{i,p_i}}, \forall i \in V/\{AP\}$$

$$W(i) \cap W(k) = \emptyset, \forall i \in V/\{AP\}, \forall k \in C_i$$

$$W(i) \cap W(j) = \emptyset, \forall i, j \in C_k, \forall k \in V/V^L$$

$$W(i) \cap W(u) = \emptyset, \forall i \in V, \forall j \in N_i/\{p_i\},$$

$$\forall u \in C_j$$

其中  $W(i) \cap W(j) = \emptyset$  即表示了节点  $i$  和节点  $j$  不能同时发送数据. 限制条件中的第 2 项表示的是父节点发送数据的时刻不能与儿子节点发送数据的时刻相同, 第 3 项表示的是同一父节点的多个儿子节点都传输数据时传输数据的时刻不能相同, 避免父节点在接收数据时发生冲突, 第 4 项表示的是一个节点在向父节点发送数据时, 邻居节点不能接收数据, 即邻居节点的儿子节点不能发送数据.

### 4.2 问题的简化

从上述形式化定义中, 我们可以看出  $\Pi$  问题中节点可以多次发送数据, 因此规划的节点发送数据的时刻是一个集合, 如果限定每个节点最多只发送数据一次, 即  $W(i)$  中元素的个数小于等于 1, 我们得到单传数据聚集传输规划问题, 记为  $\Pi'$  问题. 定义为如下形式:

#### $\Pi'$ 问题

$$\text{Maximize } M_{AP}$$

$$\text{Subject to } W(i) \in 2^{CT_{i,p_i}}, \forall i \in V/\{AP\}$$

$$|W(i)| \leq 1, \forall i \in V/\{AP\}$$

$$W(i) \cap W(k) = \emptyset, \forall i \in V/\{AP\}, \forall k \in C_i$$

$$W(i) \cap W(j) = \emptyset, \forall i, j \in C_k, \forall k \in V/V^L$$

$$W(i) \cap W(u) = \emptyset, \forall i \in V, \forall j \in N_i/\{p_i\},$$

$$\forall u \in C_j$$

**定理 1.**  $\Pi'$  问题的优化解也是  $\Pi$  问题的优化解.

证明. 假设  $\Pi$  问题的优化解中节点  $i$  有  $k$  次发送的数据经过转发最终到了 AP 节点, 分别在时刻  $\omega_1, \omega_2, \dots, \omega_k$  发送数据包  $p_1, p_2, \dots, p_k$ , 假设  $\omega_1 < \omega_2 < \dots < \omega_k$ . 由于数据包  $p_1, p_2, \dots, p_k$  可以被聚集为一个包, 因此, 将这些数据包聚集成一个数据包在时刻  $\omega_k$  发送出去, 同分别在时刻  $\omega_1, \omega_2, \dots, \omega_k$  发送数据包  $p_1, p_2, \dots, p_k$ , 对于 AP 节点来说, 收集到的聚集信息量是相同的. 因此, 通过  $\Pi$  问题的优化解能够构造  $\Pi'$  问题的解.

设  $M_1^*$  为  $\Pi$  问题的优化解, 则构造满足假设条件的解为  $M_2$ , 则  $M_1^* = M_2$ . 设  $M_2^*$  为  $\Pi'$  问题的优化解, 则  $M_2 \leq M_2^*$ . 我们知道  $\Pi'$  问题的优化解  $M_2^*$  是  $\Pi$  问题的一个可行解, 因此  $M_2^* \leq M_1^*$ . 综上可知,  $\Pi'$  问题的优化解也是  $\Pi$  问题的优化解. 证毕.

通过定理 1 可知, 通过解  $\Pi'$  问题即可得到  $\Pi$  问题的优化解, 也就是在形式化问题时加入的每个节点最多只发送一次数据的条件并没有影响数据聚集问题的实质. 通过这一条件可以简化问题. 由于  $\Pi'$  问题中每个节点最多只发送一次数据, 因此我们将节点  $i$  的被规划的发送数据时刻记为  $\omega(i)$ , 当节点  $i$  被规划不发送数据时,  $\omega(i)$  为 0.

### 4.3 单传数据聚集传输规划问题是 NP 完全问题

由于  $\Pi'$  问题是受几何约束的传输规划问题, 因此我们利用受限平面 3SAT 问题进行规约, 从而证明该问题是 NP 完全的. 受限平面 3SAT 判定问题是指是否存在 1 个赋值使得所有的可平面的布尔公式均满足, 其中每个子句最多有 3 个变量, 每个变量最多出现在 3 个子句中, 每个变量的正形式和负形式均出现.

**定理 2.**  $\Pi'$  问题是 NP 完全的.

证明. 我们将受限平面 3SAT 问题规约到  $\Pi'$  问题. 给定受限平面 3SAT 问题的实例  $\pi(n, m, G_\pi)$ ,  $n$  为变量个数,  $m$  为合取子式个数,  $G_\pi$  为布尔表达式平面图. 通过实例  $\pi(n, m, G_\pi)$  构造一个平面图和其上的路由树, 从而构造截止时间内将节点数据聚集到根节点的  $\Pi'$  问题实例.

首先我们构造平面图  $G_k$  ( $k$  为正整数). 如图 3 所示. 对于每个变量  $x_i$  ( $1 \leq i \leq n$ ), 我们构造环  $X_i$  和两条路径  $P_i, P'_i$ .  $X_i$  包括  $10k$  个节点, 依次等距为  $p_i, q_i, r_i, s_i, t_i, s'_i, r'_i, q'_i, p'_i, t'_i$  节点,  $P_i$  是一条节点  $o_i$  和节点  $p_i$  之间长度为  $(6i-5)k-1$  的路径,  $P'_i$  是一条  $o'_i$  和  $p'_i$  之间长度为  $(6i-5)k-1$  的路径.  $X_i \cap P_i = \{p_i\}$ ,  $X_i \cap P'_i = \{p'_i\}$ . 对于每个子式  $c_i$  ( $1 \leq i \leq m$ ), 我们构造边  $C_i$ .  $C_i$  是一条节点  $b_i$  和节点  $c_i$  之间

的边. 对于  $G_\pi$  上的每条边  $x_i c_j$  (或  $x'_i c_j$ ) ( $1 \leq i \leq n, 1 \leq j \leq m$ ), 构造边  $I_{ij}$ . 边  $I_{ij}$  的一个端点是  $b_j$ , 另一个端点在  $\{s'_i, r'_i\}$  (或  $\{r_i, s_i\}$ ) 中. 下面根据边  $I_{ij}$  构造边  $C_i$  的连通时刻集, 设子式  $c_i = x_{i1} \cdot \neg x_{i2} \cdot x_{i3}$ , 当边  $I_{i1,i}$  中端点为  $s'_i$  或  $s_i$  时, 构造边  $C_i$  的一个连通时刻为  $(6i_1-2)k-1$ , 当边  $I_{i1,i}$  中端点为  $r'_i$  或  $r_i$  时, 构造边  $C_i$  的一个连通时刻为  $(6i_1-3)k-1$ . 同理构造其他两个连通时刻, 因此边  $C_i$  的连通时刻集包括 3 个连通时刻.  $Q_i$  是一条节点  $t_i$  到节点  $t_{i+1}$  长度为  $k$  的路径.  $\pi_i$  是一条节点  $b_i$  到节点  $b_{i+1}$  长度为  $k$  的路径. 还有两条路径分别是节点  $t_n$  到节点  $b_1$  长度为  $k$  的路径和  $b_m$  到节点 root 长度为  $k$  的路径, 上述构造得到图  $G_k$ , 如图 3 所示.

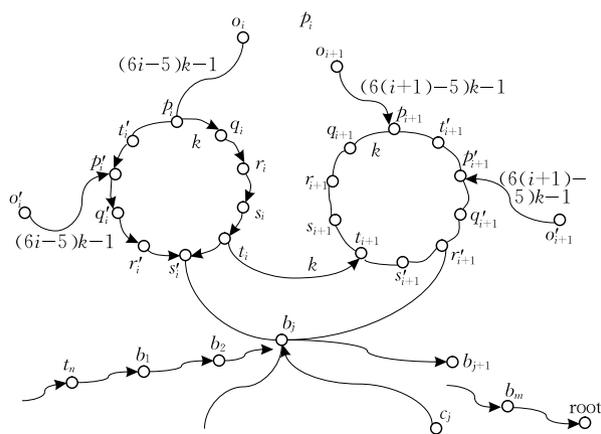


图 3 构造平面图  $G_k$  示例

设该图  $G_1$  的节点个数为  $g$ ,  $l = 120g^2$ , 将  $G_1$  中的每条边用长度为  $l$  的路径替代得到图  $G'_1$ . 通过文献[6]中的定理可知, 进行平面化嵌入的时间复杂度为  $O(f(g))$ , 其中  $f(g)$  为  $g$  的多项式函数. 由文献[6]中的引理 2 可知, 图  $G'_1$  为网格图的子图, 且构造时间为  $n$  和  $m$  的多项式.

然后构造路由树  $T_k$ . 构造树的关键是给出节点的父节点或祖父节点. 这里我们不区分父节点和祖父节点, 均称为父节点.  $o_i$  (或  $o'_i$ ) 的父节点为  $p_i$  (或  $p'_i$ ),  $t'_i$  的父节点为  $p'_i$ ,  $p_i$  (或  $p'_i$ ) 的父节点为  $q_i$  (或  $q'_i$ ),  $q_i$  (或  $q'_i$ ) 的父节点为  $r_i$  (或  $r'_i$ ),  $r_i$  (或  $r'_i$ ) 的父节点为  $s_i$  (或  $s'_i$ ),  $s_i$  (或  $s'_i$ ) 的父节点为  $t_i$ ,  $t_i$  的父节点为  $t_{i+1}$ ,  $t_n$  的父节点为  $b_1$ ,  $b_i$  的父节点为  $b_{i+1}$ ,  $b_m$  的父节点为 root. 其中箭头线段构成路由树.

受限平面 3SAT 实例  $\pi$  是可满足的, 当且仅当  $\Pi'$  问题实例  $(G_1, T_1, V(G_1), |V(G_1)|)$  可实现在  $O(ik)$  轮内将  $V(G_1)$  中的所有节点聚集到  $t_n$  节点. 由于受限平面 3SAT 问题是 NP 完全的, 所以  $\Pi'$  问题是 NP 完全的. 证毕.

## 5 数据聚集传输规划问题的求解算法

### 5.1 算法框架

问题求解的是节点什么时刻传输数据. 在规划节点的传输时刻时, 需要满足两个约束, 一个是考虑单个节点自身, 该节点的传输时刻需要在该节点与父节点的链路的连通时刻集中取值, 另一个是考虑节点之间一跳冲突, 一个发送数据的节点不能干扰到其他发送节点的接收节点的数据接收. 由于一跳冲突分为树内冲突和树间冲突, 因此, 我们考虑成两步处理, 首先处理树间冲突, 使得不同子树的节点在发送数据时不会发生冲突, 我们通过保证在通信范围内的处于不同子树的节点的可选传输时刻集不相交达到这种冲突的避免, 然后我们再处理树内冲突, 即在规划一个父节点的儿子节点们的传输时刻时, 将儿子节点和传输时刻进行匹配, 从而避免多个儿子节点同时发送数据.

我们提出了求解该问题的算法框架: PTS DP (Partition Time Set Dynamic Programming) 算法, 如图 4 所示. 该近似算法的基本思想就是分别处理两种冲突, 首先过滤节点的可选传输时刻集, 保证不同子树上的节点的传输规划不会产生干扰, 然后再从远到近按层规划节点的儿子节点与可选传输时刻的优化匹配, 使得同一父节点的儿子节点们传输时不产生冲突.

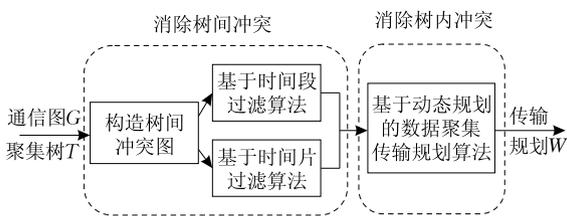


图 4 PTS DP 算法框图

PTS DP 算法的伪码如下.

#### 算法 1. PTS DP 算法.

输入: 移动通信拓扑图  $G=(V, E)$ , 移动聚集路由树  $T=(V, E^T)$

输出: 所有节点的聚集传输规划时刻

算法:

1. 构造树间冲突图算法
2. 输出树间冲突图  $G'$
3. 过滤候选传输时刻集算法
4. 输出新的移动聚集路由树  $T'$
5. 在树  $T'$  上规划无冲突数据传输时刻
6. 输出节点的聚集传输时刻

其中步 1~4 进行消除树间冲突, 详细内容在 5.2 节展开. 步 5、6 消除树内冲突并进行聚集传输规划, 输出无冲突的优化聚集传输时刻, 详细内容在 5.3 节展开.

### 5.2 消除树间冲突

为了描述树间冲突我们引入一个重要定义, 树间冲突图 (Partial Interference Graph).

**定义 5** 树间冲突图 PIG. 已知移动通信拓扑图  $G=(V, E)$ , 记  $G'=(V', E')$  为树间冲突图, 其中节点集  $V'$  中的元素是  $V$  的子集, 即  $V'=\{v \in 2^V\}$ , 边集  $E'$  中的元素为  $(u, v)$ , 则  $u, v \in V'$ , 且  $\exists i \in u, \exists j \in v, \exists k \in CT_{i, p_i} \cup CT_{j, p_j}, \omega(i) = \omega(j) = k$ .

如图 5(a) 所示, 节点  $d$  为移动聚集路由树的根节点, 其他节点存在到节点  $d$  的唯一路径, 节点  $a$  和节点  $j$ , 节点  $i$  和节点  $c$  以及节点  $g$  和节点  $j$  之间存在无线连通链路. 在图 5(b) 中, 八面体表示树间冲突图的节点, 冲突图的节点是  $G$  的节点集  $V$  的子集, 冲突图上的节点有两种类型, 一种是图  $G$  中的单个节点, 例如图 5(b) 中由节点  $a$  构成的节点 1, 由节点  $j$  构成的节点 5, 另一种是树  $T$  上的兄弟节点集, 例如图 5(b) 中由节点  $c$  和节点  $l$  构成的节点 2, 由节点  $f$ 、节点  $h$  和节点  $i$  构成的节点 4. 冲突图上的边是由在图  $G$  中而不在树  $T$  上的边构造而成的. 例如图  $G$  中的边  $e_{aj}$  的存在导致了节点  $a$  的传输时刻不能与节点  $c$  和节点  $l$  的传输时刻相同, 节点  $j$  的传输时刻不能与节点  $f$ 、节点  $h$  和节点  $i$  的传输时刻相同, 也就是在  $G'$  中的两条边, 节点 1 和节点 2 之间的边, 节点 4 和节点 5 之间的边.

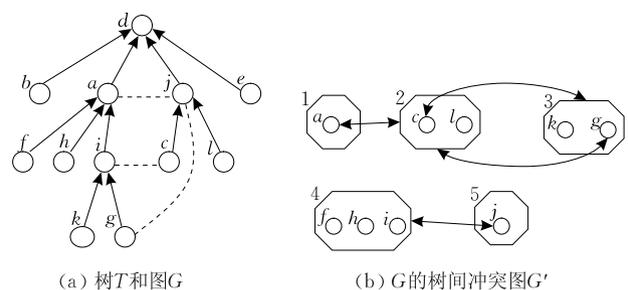


图 5 树间冲突图示例

我们利用树间冲突图刻画传输规划中的树间冲突, 即当不同子树上的节点之间存在无线链路时, 一个节点向它的父节点发送数据时, 另一个节点不能接收儿子节点的数据, 即此时另一个节点的儿子节点不能发送数据. 树间冲突图中节点之间存在边表示两个节点的候选传输时刻集之间不能重叠. 因此根据树间冲突图的结构将边的两个端点中节点的候

选传输时刻集中相交的部分进行划分,对相交的部分在一个端点的节点的候选传输时刻集中保留,在另一个端点的节点的候选传输时刻集中去除,最终使得交集为空,即得到了避免树间冲突的目的.因此,消除树间冲突部分包括两个子算法,构造树间冲突图算法和过滤候选时刻集算法.

### 5.2.1 构造树间冲突图算法

该算法输入移动通信拓扑图  $G=(V, E)$  和移动聚集路由树  $T=(V, E^T)$ , 输出树间冲突图  $G'=(V', E')$ . 该算法主要有两个部分, 首先根据在移动通信拓扑图  $G$  上而不在移动聚集路由树  $T$  上的边  $e_{uv}$ , 构造树间冲突图  $G'$  上的两条边,  $e_{uv}$  的存在会导致两个冲突, 节点  $u$  和  $C_v$  中的节点不能同时传输, 节点  $v$  和  $C_u$  中的节点不能同时传输.

如图 6 所示, 节点  $u$  有 3 个儿子节点  $a, b, c$ , 节点  $v$  有两个儿子节点  $d$  和  $e$ , 在  $G'$  中节点  $\{u\}$  和节点  $\{d, e\}$  之间有一条边, 节点  $\{v\}$  和节点  $\{a, b, c\}$  之间有一条边, 这就是  $G'$  中的两条边的构造过程. 然后我们要对节点进行合并, 这是因为对于  $u \in V$ , 如果  $u \in v_1 \cap v_2, v_1, v_2 \in V', u$  在  $v_1$  和  $v_2$  分别分配的候选时刻不能有冲突, 因此需要将具有相同元素的节点进行合并. 通过节点合并以后就得到了树间冲突图  $G'$ . 下面给出 createPIG 算法的伪码.

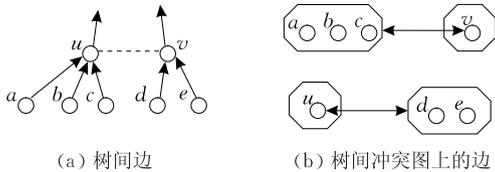


图 6 树间冲突图上的边的构造

#### 算法 2. createPIG 算法.

输入: 移动通信拓扑图  $G=(V, E)$ , 移动聚集路由树  $T=(V, E^T)$

输出: 树间冲突图  $G'=(V', E')$

算法:

1. 初始化  $V' = NULL, E' = NULL$
2. FOR  $((u, v) \in E \setminus E^T)$
3. { 将  $\{u\}$  作为一个节点添加到  $V'$ , 将  $C_v$  作为一个节点添加到  $V'$
4. 构造  $\{u\}$  和  $C_v$  之间的边, 加入  $E'$
5. 将  $\{v\}$  作为一个节点添加到  $V'$ , 将  $C_u$  作为一个节点添加到  $V'$
6. 构造  $\{v\}$  和  $C_u$  之间的边, 加入  $E'$
7. FOR  $(u, v \in V')$
8. IF  $u$  是  $v$  的子集, 则将  $u$  和  $v$  合并成  $v$  节点

该算法从步 2 到步 6 循环了  $|E \setminus E^T|$  次, 步 7 的循环次数是  $|V'|^2$ . 设  $m = |E \setminus E^T|, |V'| \leq 2m$ , 因此

createPIG 算法的时间复杂度是  $O(4m^2)$ .

### 5.2.2 过滤候选时刻集算法

createPIG 算法构造得到树间冲突图, 通过树间冲突图我们可以获得不同子树上的节点的冲突信息, 下面的关键就是将冲突的候选传输时刻集划分使得可能冲突的节点的候选时刻集之间没有交集. 不同的划分规则就决定了不同节点上的候选传输时刻集, 进而影响传输时刻的规划. 我们将划分树间冲突并过滤候选传输时刻集的算法称为 PTS 算法.

树间冲突图表示的是整个时间段内节点之间可能存在的冲突情况. 因此, 从树间冲突图考虑的是整个时刻集上的冲突. 由于这样处理的粒度粗, 导致了部分冲突图复杂, 过滤后的候选时刻集较小, 使得后续树上的无冲突传输规划的解空间较小, 结果较差. 如果考虑单时刻的部分冲突图, 求单个时刻的部分冲突图的最大权独立集, 将该时刻分配给独立集中的节点使用, 并将该时刻下其他节点的链路的连通时刻集中过滤掉, 这样会使得被过滤掉的候选时刻更少, 即后续树上的无冲突传输规划的解空间大一些.

#### 5.2.2.1 基于时间段的过滤候选时刻集算法

我们提出从整个时间段的树间冲突图考虑节点间的冲突关系进而过滤节点的候选时刻集的算法, 记为 PTS-Period 算法. 首先根据树间冲突图划分冲突子集使得子集的个数最少. 然后将整个时间段根据子集个数进行划分, 形成不重叠的时刻子集. 最后将冲突子集与时刻子集进行匹配. 因此, PTS-Period 算法包括 3 个部分: 冲突子集的划分、时刻子集的划分和冲突子集与时刻子集的匹配.

PTS-Period 算法的伪码如下.

#### 算法 3. PTS-Period 算法.

输入: 树间冲突图  $G'=(V', E')$ , 移动聚集路由树  $T=(V, E^T)$

输出: 新移动聚集路由树  $T'=(V, E'^T)$

算法:

1. 划分冲突子集: 对树间冲突图进行顶点着色
2. 划分时刻子集: EI 划分或 A 划分
3. 构造颜色号和时刻子集的映射, 计算权值  $w_{ij}$
4. 求最大权二部图匹配

下面我们展开每个部分的详细阐述.

对于冲突子集的划分, 我们利用贪心算法对树间冲突图进行顶点着色<sup>[36]</sup>. 同一颜色的节点组成一个冲突子集. 首先根据节点的度进行从大到小排序, 第 1 个颜色对应第 1 个节点着色, 并按照节点排列的次序对与前面着色点不邻接的每个点着相同颜

色,重复上述步骤直至所有顶点着色完成,设共用了  $CN$  个颜色。

在时刻子集的划分中,时刻集  $TS = \{1, 2, \dots, D\}$ , 划分形成的  $CN$  个子集  $TS_1, TS_2, \dots, TS_{CN}$  需要满足如下两个条件:

- (1)  $\bigcup_{i=1}^{CN} TS_i = TS$ ;
- (2)  $TS_i \cap TS_j = \emptyset$ .

划分的方法有很多,最直接的划分方法是将该时刻集  $TS$  等分为  $CN$  个子集,得到  $\{1, 2, \dots, \lfloor TS/CN \rfloor\}, \{\lfloor TS/CN \rfloor + 1, \lfloor TS/CN \rfloor + 2, 2\lfloor TS/CN \rfloor\}, \{CN\lfloor TS/CN \rfloor, CN\lfloor TS/CN \rfloor + 1, \dots\}$ , 记为 EI(Equal Interval)划分. 这种划分方法基本思想是每个节点被分配一段可用时间,如果节点的候选传输时刻集和这段时间较匹配,则该节点可在这段时间内规划传输时刻,但当超过这段时间时,该节点就没有传输机会了. 这种划分方法使得节点的候选传输时刻集中在较小的时刻或者较大的时刻. 另一种划分方法是将时刻集  $TS$  抽取分为  $CN$  个子集,得到  $\{1, CN + 1, \dots, \lfloor TS/CN \rfloor \times CN + 1\}, \{2, CN + 2, \dots, \lfloor TS/CN \rfloor \times CN + 2\}, \dots, \{CN, 2CN, \dots\}$ , 记为 A(Alternate)划分. 这种划分的基本思想是让节点在各个时间段内都有传输的机会,但在每个时间段内可用时刻更少,即把传输机会分散开。

冲突子集与时刻子集的匹配是 PTS 算法的核心. PIG 中的每个节点属于一个冲突子集,该节点本身包含多个移动节点,每个移动节点有候选传输时刻集,当建立时刻子集与冲突子集的匹配,意味着该时刻子集中的时刻可以属于该冲突子集的节点的候选传输时刻集,因此匹配结果直接影响节点的候选传输时刻集. 节点的候选传输时刻集决定了传输时刻规划的解空间. 下面我们分析影响节点优化传输时刻的 3 个因素,分别是节点在路由树上的位置、节点之间的关系以及节点和父节点的链路的连通时刻集。

#### (1) 考虑节点在路由树上的位置

节点按照路由树的结构传输数据,因此传输时刻存在偏序关系,也就意味着离根节点越远的节点要越早发送数据,因为除了干扰影响传输进程,节点和根节点的路径长也影响节点的数据能否在  $D$  时刻之前到达根节点,离根节点越近的节点应该越晚发送数据,节点应该尽可能等待子节点传输的数据,能尽可能多地收集数据再向上传输. 通过上述分析,我们将节点与根节点的距离作为建立映射的权值. 设节点  $i$  离根节点的距离为  $h_i$ , PIG 中涂  $j$  颜色的

节点集为  $V'_j = \{v'_{j1}, v'_{j2}, \dots, v'_{jbn}\}, V'_j \subset V', v'_{jk} = \{v_{(1)}, v_{(2)}, \dots, v_{(m)}\}, \forall v'_{jk} \in V'_j$ . 定义  $j$  颜色的节点平均高度为  $h^j$  如式(2).

$$h^j = \sum_{v' \in V'_j, v_0 \in v'} h_{v_0} / |V'_j| \quad (2)$$

#### (2) 考虑节点之间的关系

父节点在接收儿子节点的数据时,儿子节点之间是竞争的关系,因此当一个节点的兄弟节点较多时,应该候选传输时刻更晚. 设节点  $i$  的兄弟节点个数为  $b_i$ , PIG 中涂  $j$  颜色的节点集为  $V'_j = \{v'_{j1}, v'_{j2}, \dots, v'_{jbn}\}, V'_j \subset V', v'_{jk} = \{v_{(1)}, v_{(2)}, \dots, v_{(m)}\}, \forall v'_{jk} \in V'_j$ . 定义  $j$  颜色的平均兄弟数为  $b^j$  如式(3).

$$b^j = \sum_{v' \in V'_j, v_0 \in v'} b_{v_0} / |V'_j| \quad (3)$$

#### (3) 考虑节点和父节点的链路的连通时刻集

节点的候选时刻集是规划传输时刻的解空间,因此候选时刻集越大,规划的机会越多,优化的空间也越大. 节点候选时刻集大小由两个因素决定,一个是节点与父节点的链路的连通情况,这是本问题的输入信息,另一个是分配的时刻子集与连通时刻集的匹配情况,匹配程度越高,连通时刻集中被保留作为候选时刻集的元素越多,后续规划传输时刻的优化空间也越大. 设 PIG 中涂  $j$  颜色的节点集为  $V'_j = \{v'_{j1}, v'_{j2}, \dots, v'_{jbn}\}, V'_j \subset V', v'_{jk} = \{v_{(1)}, v_{(2)}, \dots, v_{(m)}\}, \forall v'_{jk} \in V'_j$ . 节点  $v$  与父节点  $p_v$  的链路连通时刻集为  $CT_{v, p_v}$ . 计算  $j$  颜色的节点的连通时刻集与时刻子集  $TS_i$  的匹配度  $m_{ij}$  如式(4).

$$m_{ij} = \sum_{v' \in V'_j} \sum_{v \in v'} |CT_{v, p_v} \cap TS_i| / \sum_{v' \in V'_j} \sum_{v \in v'} |CT_{v, p_v}| \quad (4)$$

下面我们定义冲突子集  $j$  对应时刻子集  $TS_i$  的权值为  $w_{ij}$ , 如式(5)所示,这里  $\alpha, \beta, \gamma$  为比例因子。

$$w_{ij} = \alpha h^j + \beta b^j + \gamma m_{ij} \quad (5)$$

根据权值进行冲突子集与时刻子集匹配的算法的基本思想是对于时刻子集  $TS_i$  找出所有冲突子集的  $w_{ij}$  值最大的,将该时刻子集匹配该冲突子集,重复此过程,直至所有时刻子集均匹配。

下面我们分析 PTS-Period 算法的时间复杂度. 设图  $G'$  的顶点数为  $n$ , 总时刻个数为  $D$ . 步 1 顶点着色算法时间复杂度为  $O(n^2)$ , 步 2 划分时刻子集为  $O(1)$ , 步 3 计算权值的时间复杂度为  $O(nD)$ , 步 4 匹配算法时间复杂度为  $O(n \log(n))$ , 综上 PTS-Period 算法的时间复杂度为  $O(n^2 + nD)$ .

#### 5.2.2.2 基于时间片的过滤候选时刻集算法

PTS-Period 算法中将整个时刻集分成时刻子集,然后对时刻子集进行颜色号分配,这里时刻子集

的划分采用等分或等间隔的方式,可能会将更优化的传输时刻规划漏掉,因此,我们考虑可以对每个时刻分别考量而不是对一个时刻子集进行考量.图7给出了  $\Pi'$  问题的实例,如图所示由  $a$  到  $l$  组成的10个节点构成移动通信拓扑图,右侧为各链路的连通时刻集,左侧为以节点  $a$  为根的一棵移动聚集路由树.图8给出了采用 PTS-Peroid 算法的 PTSDP 算法的解,根节点  $a$  截止时刻7时收集的最大聚集信息量为9,树边上标注的是节点规划的发送数据的时刻.对于树间冲突图中的节点  $\{f\}$  分配的可用时刻子集为  $\{6,7\}$ ,节点  $\{d,e\}$  分配的可用时刻子集为  $\{4,5\}$ ,节点  $\{j,k,l\}$  分配的可用时刻子集为  $\{1,2,3\}$ ,图8的右侧即为 PTS-Peroid 算法求得的解,可以看出该算法的解求得的优化目标的值为9,节点  $e$  和节点  $f$  的数据没有被规划传输的机会.

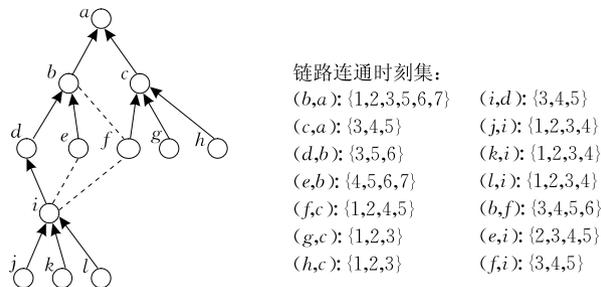


图7  $\Pi'$  问题实例

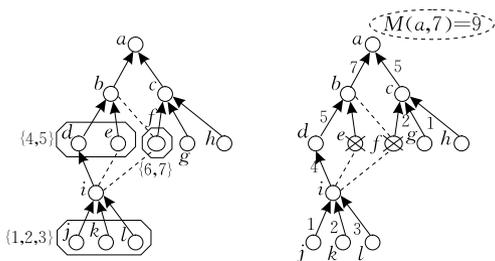


图8 采用 PTS-Peroid 算法的 PTSDP 算法得到的解

由于 PTS 算法中对节点的候选传输时刻集进行了过滤,经过 PTS 算法后规划传输时刻的解空间缩小了.因此,PTS 算法应该尽量减少缩小的解空间范围.从图8中可以看出,经过 PTS-Peroid 算法,节点  $e$  的候选传输时刻集由  $\{2,3,4,5,6\}$  过滤为  $\{4,5\}$ ,节点  $f$  的候选传输时刻集由  $\{3,4,5\}$  过滤为空集,如果不采用时间段去过滤,而是针对单个时刻进行考虑时,就可以使得从候选时刻集中被过滤出去的时刻更少,也就是得到的解空间更大,因此,我们提出了 PTS-Slot 算法,采用 PTS-Slot 算法的 PTSDP 算法得到结果如图9所示,根节点  $a$  截止时刻7时收集的最大聚集信息量为11.

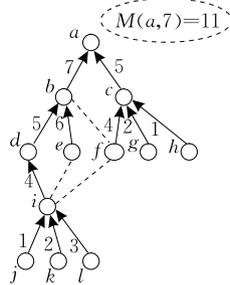


图9 采用 PTS-Slot 算法的 PTSDP 算法得到的解

PTS-Slot 算法的核心是每个时刻应该划分给哪些节点作为可用传输时刻. PTS-Slot 算法首先用单时刻的树间冲突图  $t$ -PIG 描述在一个时刻时节点之间的冲突关系,这个冲突关系可以通过树间冲突图提取出来,然后将该时刻划分给部分节点可用.下面给出 PTS-Slot 算法的伪码.

**算法4.** PTS-Slot 算法.

输入: 树间冲突图  $G'=(V',E')$ , 移动聚集路由树  $T=(V,E^T)$

输出: 新移动聚集路由树  $T'=(V,E'^T)$

算法:

1. FOR 时刻  $t$  从 1 到  $D$
2. 根据树间冲突图  $G'$  得到  $t$ -PIG 图.
3. 计算  $t$ -PIG 图上的所有节点的权值  $w(t,i)$ .
4. 在  $t$ -PIG 上求最大权独立集  $IS$ : 计算每个节点  $u$

的  $w(u)/\sum_{v \in N(u)} w(v)$  值,选择最大值的节点加入  $IS$ ,将该节点及邻居节点删除,重复直至处理完所有节点.

从树间冲突图中提取单时刻树间冲突图的基本思想就是将包含某时刻  $t$  的可规划传输时刻集对应的节点保留,当 PIG 上的一个节点内包含的所有节点的可规划传输时刻集中都不包括该时刻时,这个节点不包含在  $t$ -PIG 的节点集中,相连的边也被删除.

对每个  $t$ -PIG,该图上节点之间的边  $e_{ij}$  表示在  $t$  时刻节点  $i$  中的节点的传输时刻和节点  $j$  中的节点的传输时刻不能同时规划时刻  $t$ ,因此时刻  $t$  保留在哪个节点的候选传输时刻中是  $t$ -PIG 图上的独立集问题.我们求解  $t$ -PIG 图的最大权独立集,将时刻  $t$  从  $t$ -PIG 图上的其他节点的候选时刻集中去掉.这里的核心就在于权值如何设计.在设计权值参数时要考虑两个因素,分别是节点在路由树上的高度和兄弟节点的个数.考虑节点在路由树上的高度和规划时刻时,节点在路由树上离根节点越远应该越早规划,离根节点越近应该越晚规划.考虑节点的兄弟节点个数和规划时刻时,如果节点的兄弟节点个数越多,竞争越激烈,可规划的时刻应该更多一些.利用式(2)计算  $t$ -PIG 图上单节点的平均高度,利用

式(3)计算  $t$ -PIG 图上单节点的平均兄弟个数, 我们给出权值的定义为函数  $\omega: TS \times V \rightarrow R$ , 其中  $TS$  为  $\{1, 2, \dots, D\}$ ,  $V$  是  $t$ -PIG 图的顶点集,  $R$  为实数集.

$$\omega(t, i) = \frac{[(t - bt_r/2) \cos \alpha - (sh_i/b_i - h_T/2) \sin \alpha]^2}{2p} - \frac{[(t - bt_r/2) \sin \alpha + (sh_i/b_i - h_T/2) \cos \alpha]^2}{2q} + \omega_0 \quad (6)$$

其中  $\alpha = 2\pi - \arctan D/h_T$ ,  $p, q, s$  为 3 个静态参数.

求解最大权独立集采用一个贪心的算法<sup>[37]</sup>, 该算法由 WMIN 算法<sup>[37]</sup> 改进得到, 基本思想是每次从节点集选择  $\omega(u) / \sum_{v \in N(u)} \omega(v)$  值最大的节点加入独立集, 在原图上删除该节点及其邻居节点, 重复上述过程直至图上没有节点.

下面分析 PTS-Slot 算法的时间复杂度. 设图  $G'$  的顶点数为  $n$ , 边数为  $m$ . 步 2 需要遍历  $G'$  时间复杂度为  $O(m)$ , 步 3 为  $O(n)$ , 步 4 为贪心法求最大权独立集的时间复杂度为  $O(n)$ , 步 2、3、4 要循环  $D$  次, 因此 PTS-Slot 算法的时间复杂度为  $O((n+m)D)$ .

### 5.3 消除树内冲突

通过 PTS 算法, 我们得到新移动聚集路由树, 在该树上需要给出无冲突传输规划使得根节点收集的聚集数据最多. 我们提出了基于动态规划思想的 DAS-MRT 算法. 用  $X[i, W]$  表示节点  $i$  在传输时刻为  $W$  时发送的聚集数据包含的最多聚集信息量, DAS-MRT 算法对节点所有的候选传输时刻根据节点在树上的深度从底向上逐层计算  $X[i, W]$ , 在计算  $X[i, W]$  时, 需要选择节点  $i$  的儿子节点的传输时刻, 使得儿子节点之间的传输时刻不冲突, 我们将该子问题转化成二部图的最大权匹配问题, 通过 KM 算法<sup>[38]</sup> 进行求解. 该二部图的构造方法为节点集分为  $A$  和  $B$  两个不相交子集, 集合  $A$  为节点  $i$  的儿子节点集, 集合  $B$  为儿子节点的候选传输时刻集的并集, 集合  $A$  中的节点  $j$  与集合  $B$  中的时刻  $t$  之间有边当且仅当时刻  $t$  为节点  $j$  的传输候选时刻集中的元素, 该边的权值为  $X[j, t]$ . 求解该二部图的最大权匹配即可求得父节点的  $X[i, W]$  值. DAS-MRT 算法的伪码如下.

#### 算法 5. DAS-MRT 算法.

输入: 新移动聚集路由树  $T' = (V, E^{T'})$

输出: 所有节点的聚集传输时刻

算法:

1. 对于所有的叶子节点, 初始化  $X[i, W] = 1, \forall i \in V^L, \forall W \in CT_{i, p_i}$

2. 对于所有的节点, 初始化  $X[i, 0] = 0$ , 如果  $1 \in CT_{i, p_i}, X[i, 1] = 1$
3. 考虑除了根节点  $r$  和叶节点的任一节点  $i$ , 计算  $X[i, W], \forall i \in V \setminus (V^L \cup \{r\}), \forall W \in CT_{i, p_i}$
4. 设节点  $i$  有  $k$  个儿子节点  $c_1, c_2, \dots, c_k$ . 任一儿子节点  $c_j$  的  $\omega(c_j)$  从  $CT_{c_j, i} \cap \{1, \dots, W-1\} \cap \{0\}$  中选取.
5. 设临时变量  $NC = C_i$
6. (1) 从  $W-1$  开始, 将没有出现冲突的时刻分发给相应的节点, 设  $m = 1$
7. WHILE  $((\omega - m) > 1)$  {
8. 如果  $|\{c_j | (\omega - m) \in CT_{c_j, i}\}| = 0, m + 1$
9. 如果  $|\{c_j | (\omega - m) \in CT_{c_j, i}\}| = 1$ , 将  $c_j$  的  $W$  设为  $W - m$ , 且更新  $NC = NC / \{c_j\}$
10. 如果  $|\{c_j | (\omega - m) \in CT_{c_j, i}\}| > 1$ , 跳出循环}
11. (2) 对剩下未处理的节点进行处理且从  $\omega - m$  开始, 对于  $NC$  中的节点从  $CT_{c_j, i} \cap \{1, \dots, W - m\}$  中选择最大的前  $|NC|$  个, 如果不足则全选
12. 用  $NC$  的节点和可选时刻集构造最大权值匹配问题, 调用 EM 算法
13. 计算  $X[r, D]$

步 1、2 为初始化步骤. 从步 3~13 为逐层计算除根节点以外的所有节点的  $X[i, 1]$  值, 其中步 6~10 是将不冲突的时刻直接分配, 步 11~12 是将剩余的未规划节点利用 KM 算法进行规划. 步 13 是计算根节点的  $X[r, D]$ .

下面分析 DAS-MRT 算法的时间复杂度. 设移动聚集路由树  $T$  的高度为  $h$ , 树上节点的最大度为  $\Delta$ , 节点总数为  $n$ , MRT 中边的连通时刻集的基数的最大值为  $m$ . 二分图最优匹配问题的 KM 算法<sup>[36]</sup> 的复杂度为  $O(q^3)$ , 其中  $q$  为二分图的边数. 在每个节点  $i, \forall W \in CT_{i, p_i}$ , 我们需要计算  $X[i, W]$ .

对于叶子节点来说,  $\forall i \in V^L, \forall W \in CT_{i, p_i}, X[i, W]$  均设为 1, 因此计算需要时间为  $O(m)$ .

对于第  $k$  层的节点来说, 对于  $\forall i \in V^{(k)}, \forall W \in CT_{i, p_i}$ , 计算  $X[i, W]$  需要的时间取决于 EM 算法的时间复杂度, 这里二部图的边数为  $O(\Delta^2)$ , 由于 KM 算法的时间复杂度为  $O(q^3)$ ,  $q$  为边数, 因此计算  $X[i, W]$  的时间复杂度为  $O(\Delta^6)$ , 从而计算第  $k$  层一个节点的所有可传输时刻的  $X$  值的时间复杂度为  $O(m\Delta^6)$ . 由于同一层的节点的  $X$  值相互不影响, 可以并行计算, 因此第  $k$  层的节点的  $X$  值计算的时间复杂度为  $O(m\Delta^6)$ .

第  $k$  层节点的  $X$  值计算取决于第  $k+1$  层节点的  $X$  值. 第  $h$  层为叶子节点, 从  $h-1$  层开始逐层计算, 计算到第 0 层的根节点的时间复杂度为  $O(m +$

$(h-1)m\Delta^6)$ .

因此, DAS-MRT 算法的时间复杂度为  $O(m + (h-1)m\Delta^6)$ .

## 6 仿真实验及其结果分析

本文通过仿真实验分析采用不同的 PTS 算法的 PTSDP 算法的性能, 同时与 DB-VDG<sup>[28]</sup> 进行了比较. DB-VDG 是由 Claudio 等人提出的在车辆感知网络上进行限定延时的数据收集协议. 这个协议包括两种策略, 分别是 SBSS 和 DBSS, 在文献[28]中 SBSS 的性能更好, 因此, 我们采用 SBSS 策略的 DB-VDG 协议. PTSDP 近似算法框架中核心是 PTS 算法的选择, PTS 算法分为两类, 分别是 PTS-Period 算法和 PTS-Slot 算法, 这里 PTS-Period 算法可以采用两种划分时间子集方法, EI 划分和 A 划分, 分别记为 PTSDP-PEI 算法, PTSDP-PA 算法, 将采用 PTS-Slot 的算法记为 PTSDP-S 算法. 仿真实验中将本文提出的 PTSDP-PEI 算法, PTSDP-PA 算法和 PTSDP-S 算法与 DB-VDG 进行比较.

### 6.1 实验数据及其参数设置

本文采用真实出租车移动轨迹数据集 CSGPS 进行仿真实验. CSGPS 数据集是由 Michal Piorkowski, Natasa Sarafijanovic-Djukic, Matthias Grossglauser<sup>[39]</sup> 在 2008 年于美国旧金山采集的出租车 GPS 移动轨迹数据. 该数据集的数据是由 cabspotting<sup>①</sup> 项目组提供的. 该数据集包含在旧金山湾区的大约 500 辆出租车的 30 天的 GPS 数据. 数据采集的开始时间为 2008-05-17, 结束时间为 2008-06-10. 数据集中每辆车的轨迹数据存储为以该车辆 ID 命名的 txt 文件, 该文件中每一条记录为一个时刻该车辆的 GPS 数据. 该数据集具体数据以及更详细的信息可在 <http://crawdad.cs.dartmouth.edu/epfl/mobility/CSGPS> 上获取.

我们对数据进行预处理, 首先对整个数据采集时间段进行分片, 将每个时间片内出现的 GPS 数据整理为一个文件, 形成单时间片 GPS 数据文件, 然后根据给定的通信半径以及圆盘模型, 形成单时间片通信拓扑图文件. 实验中关于数据集和算法的默认参数见表 1.

我们采用 NS2 模拟器模拟无线网络传输, 这里 MAC 层采用 802.11p 协议. 我们将处理后的 GPS 轨迹导入 NS2, 在 NS2 上运行规划算法, 按照得到的结果进行实际数据传输.

表 1 默认参数

参数	默认值
Node number	[40, 50]
Rounds	20
Communication Range	500 m
MAC layer	802.11p
Cardinal Number of TS	10
Delay Constraint	10
Time Frame	100 s
$\alpha$	1
$\beta$	1
$\gamma$	1
$p$	1
$q$	1
$s$	1

### 6.2 性能指标

为了分析数据收集算法的性能, 我们定义 3 个指标来评测数据收集的有效性、效率和时效性:

(1) 数据收集率是度量数据收集算法的有效性的指标, 也就是数据收集算法将相关的节点数据路由到 AP 的能力. 该度量是 AP 收集到的节点数据占节点总数的比例, 定义如下

$$effectiveness = \frac{N_{delivered}}{N_{nodes}} \quad (7)$$

这里  $N_{delivered}$  是 AP 收集到的节点的数据总量,  $N_{nodes}$  是节点总数.

(2) 平均传输数据率是评测传输数据时消耗的带宽量的指标, 即度量算法的效率. 该度量是节点发送的数据总量与 AP 收集到的节点的数据总量的比值, 定义如下

$$efficiency = \frac{\sum_{i \in delivered} |S_i|}{N_{delivered}} \quad (8)$$

这里  $S_i$  为节点  $i$  发送的数据量,  $delivered$  是 AP 节点收集到的数据对应的节点集,  $N_{delivered}$  是 AP 收集到的节点的数据总量.

(3) 平均延迟是度量数据收集算法的时间性能的指标. 该度量是收集上来的数据的延迟的平均值, 定义如下

$$averageDelay = \sum_{i \in received} t_i / N_{received} \quad (9)$$

这里  $t_i$  为节点  $i$  的数据到达 AP 节点的时刻,  $received$  是 AP 节点收集到的数据对应的节点集,  $N_{received}$  为 AP 收集到的节点的数据总量.

### 6.3 性能对比

为了分析 PTSDP 系列近似算法的性能, 我们将该算法与 DB-VDG 算法从数据收集的有效性、效

① Cabspotting Project. <http://cabspotting.org/about.html>

率和时效性 3 个方面进行对比. 从表 2 中可以看出本文提出的 PTSDP-PEI 算法、PTSDP-PA 算法和 PTSDP-S 算法从数据收集的有效性、时效性和效率上均优于现有的最好的 DB-VDG 算法. 3 种算法中数据收集的有效性最高的是 PTSDP-S 算法, 这是由于该算法考虑部分冲突的粒度是单个时刻, 而相比较 PTSDP-PEI 算法和 PTSDP-PA 算法考虑部分冲突的粒度是时间段, PTSDP-S 算法处理的粒度更细, 因此效果更好. 对于数据收集的时延, 本文提出的 3 种算法略小于 DB-VDG 算法, 这是由于 DB-VDG 算法在数据聚集过程中传输的数据量较大, 因此最大延时会更高. 本文提出的算法在数据收集效率的评测指标上的结果均为 1, 这是由于本文提出的算法进行了完全的聚集, 即根据移动路由由聚集树结构, 当父节点等到所有规划传输的儿子节点发送完数据以后, 父节点才将所有已收到的数据和自己的数据进行聚集发送出去, 之后父节点不再进行数据传输, 因此, 每个节点只发送一次数据, 所以收集到的聚集数据量即为发送数据的节点个数, 因此, 数据收集效率为 1, 而 DB-VDG 算法在传输数据过程中是通过阈值控制数据转发的方式, 存在大量节点需要多次发送数据的情况, 即没有实现最大化的数据聚集, 因此, 在数据收集的效率上本文提出的 3 个算法要远远好于 DB-VDG 算法.

表 2 默认参数下仿真结果的对比

	<i>effectiveness</i>	<i>averageDelay</i> (/100 s)	<i>efficiency</i>
DB-VDG	0.647	7.13	3.96
PTSDP-PEI	0.786	6.12	1.00
PTSDP-PA	0.693	6.43	1.00
PTSDP-S	0.817	6.64	1.00

### 6.3.1 节点数量对性能的影响

本组实验主要研究节点数量对算法性能的影响. 除节点数量以外的其他参数采用默认参数, 见表 1. 从图 10 中可以看出随着节点个数的增加数据收集率逐渐下降. 本文提出的 PTSDP-S 算法比其他算法的下降趋势缓慢得多, 因此, 在数据收集的有效性上该算法要远远优于 DB-VDG 算法. PTSDP-PEI 算法和 PTSDP-PA 算法的下降非常迅速, 这是由于当节点个数增加时, 树间冲突图中的节点和边的个数都会增加, 使得时间槽个数远远小于分配颜色数, 在过滤时会造成很多的候选时刻被去除, 使得规划传输的机会大大降低, 但 PTSDP-S 算法是针对单个时刻构造树间冲突图, 因此树间冲突图的大小变化不大, 也就使得数据收集的有效性很高. 从

图 11 中可以看出数据收集的平均延迟随着节点个数的增加变化不大, 4 种算法相比较来看, DB-VDG 算法的平均延迟更高, 这是由于 DB-VDG 采用阈值控制数据的转发, 因此只有在达到阈值时采用转发数据, 否则采用携带策略, 这也就导致了收集上来的数据的延迟较高. PTSDP 系列算法是采用完全聚集的策略, 因此数据会尽可能早地路由给父节点, 这也就使得收集到的数据的平均延迟较小.

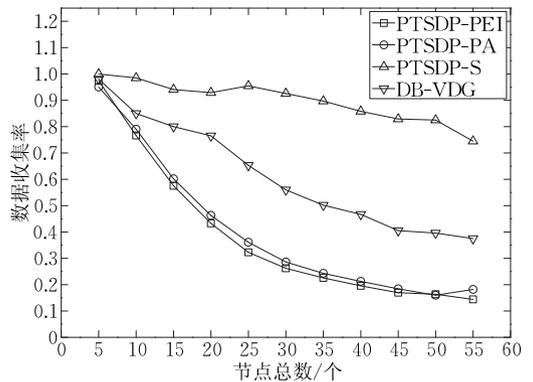


图 10 节点总数对数据收集率的影响

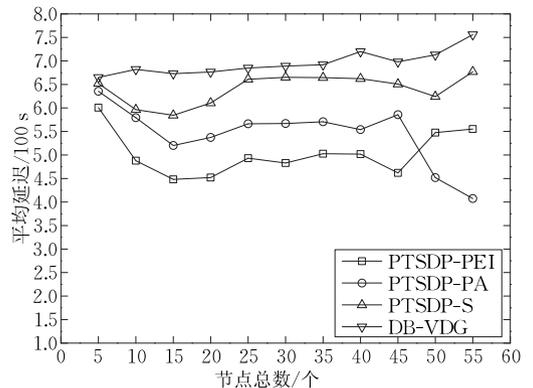


图 11 节点总数对平均延迟的影响

### 6.3.2 延迟界对性能的影响

本组实验主要研究延迟界对性能的影响. 除延迟界以外的其他参数采用默认参数, 见表 1. 从图 12 和图 13 中可以看出随着延迟界的增加, 数据收集率和平均延迟均是逐渐增加的. 图 12 中 4 个算法中 PTSDP-S 算法的数据收集率增长趋势更快, DB-VDG 算法是缓慢增加的, 这说明了 PTSDP-S 算法随着延迟界的增加, 可规划的空间变大, 优化的机会更多, 因此性能提高更显著. PTSDP-PEI 算法和 PTSDP-PA 算法随着延迟界的增加, 数据收集率的增长并不迅速, 这是由于虽然延迟界的增加使得规划时刻个数增加, 但还是小于树间冲突图的颜色数, 因此仍然存在大量候选时刻被过滤掉的问题.

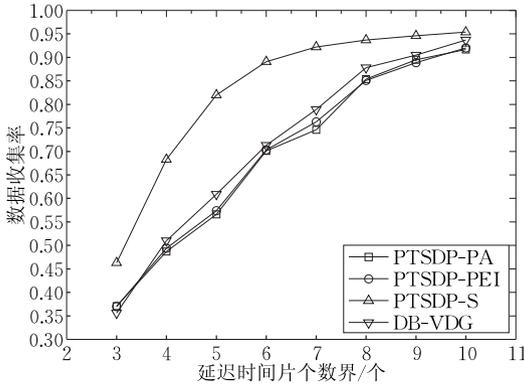


图 12 延迟界对数据收集率的影响

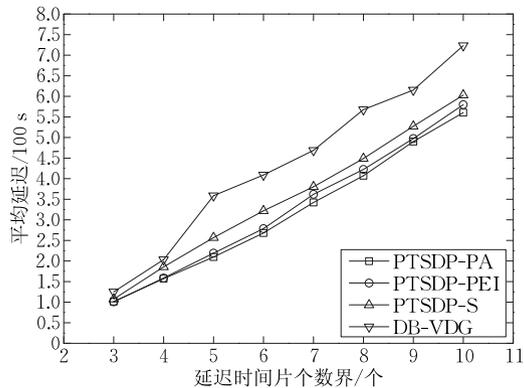


图 13 延迟界对平均延迟的影响

从图 13 中,我们可以看出平均延迟几乎随着延迟界同步增长的,但本文算法的增长趋势比 DB-VDG 算法逐渐变缓,这是由于本文采用的路由树在构造时考虑了延迟的优化,因此在保证数据收集率的情况下平均延迟也是优化的。

## 7 结 论

实时感知数据的采集和聚集是智能交通系统中各种应用的基础.而现有的车辆上的数据聚集研究大部分采用通过阈值进行调整的策略,没有利用无线通信网络本身的动态通信拓扑信息,因此,本文分析现有研究的不足,提出基于移动通信拓扑图在移动路由聚集树规划无冲突的数据传输规划问题.首先我们形式化了无线移动感知网络上的单传无冲突数据聚集传输规划问题,并证明了该问题是 NP 完全的.然后提出了求解该问题的 PTSDP 算法框架,基本思想是将传输冲突分两种情况处理,将发送节点发送数据时干扰其他正在接受数据的节点的情况用树间冲突图进行描述,将发生冲突的时刻划分给一方可用于规划,其他节点不能使用该时刻传输数据,然后再用移动路由聚集树上的动态规划算法处

理一个接收数据节点的多个发送数据节点之间传输不冲突问题.最后我们通过利用真实出租车轨迹数据集上的实验验证了本文提出的近似算法比现有的最好的 DB-VDG 算法在数据收集的有效性、效率以及评价延迟上都高出 1/4 左右.本文研究移动感知网络上的数据聚集问题采用先建立路由聚集树然后进行传输规划的思路,树的结构影响规划的效果,因此,更优化的思路是建树和传输规划一起做,边建树边规划合理的传输时刻.我们下一步即研究在移动通信拓扑图上不依托给定路由树结构的无冲突数据传输规划问题。

## 参 考 文 献

- [1] Ian F. Wireless Sensor Networks. UK: Wiley Press, 2010
- [2] Hassnaa M, Yan Zh. Vehicular Networks: Techniques, Standards and Applications. USA: Auerbach, 2009
- [3] Kashif Naseer Q, Abdul Hanna A. A survey on intelligent transportation systems. Middle-East Journal of Scientific Research, 2013, 15(5): 629-642
- [4] Nasim R, Kassler A. Distributed architectures for intelligent transport systems; A survey//Proceedings of the 2nd Symposium on Network Cloud Computing and Applications. London, UK, 2012: 130-136
- [5] Zhu Jing, Roy S. MAC for dedicated short range communications in intelligent transport system. Journal on IEEE Communications Magazine, 2006, 41(12): 60-67
- [6] Chen X J, Hu X D, Zhu J M. Minimum data aggregation time problem in wireless sensor networks. Journal on Lecture Notes in Computer Science, 2005, 5(4): 133-142
- [7] Huang S C, Peng J W, Li Y S. Nearly constant approximation for data aggregation scheduling in wireless sensor networks//Proceedings of the 26th IEEE International Conference on Computer Communications. Anchorage, AK, 2007: 366-372
- [8] Wang P, He Y, Huang L S. Near optimal scheduling of data aggregation in wireless sensor networks. Journal on Ad Hoc Networks, 2013, 11(4): 1287-1296
- [9] Bagaa M, Derhba A, Lasla N, Ouadjaout A. Semi-structured and unstructured data aggregation scheduling in wireless sensor networks//Proceedings of the 31th IEEE International Conference on Computer Communications. Orlando, FL, 2012: 2671-2675
- [10] Kuo T, Tsai M. On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: NP-Completeness and approximation algorithms//Proceedings of the 31th IEEE International Conference on Computer Communications. Orlando, FL, 2012: 2873-2877
- [11] Chen Z Y. QoS-based data aggregation technology in wireless sensor networks. Journal on Applied Mechanics and Materials, 2011, 135(2012): 702-708
- [12] Mario O. Scheduling and Data Aggregation for Periodic Data Gathering in Wireless Sensor Networks [Ph. D. dissertation]. University of London and Diploma of Imperial College, UK,

- 2011
- [13] Wang Ch, Jiang C J, Liu Y B, Li M. Aggregation capacity of wireless sensor networks; Extended network case// Proceedings of the 30th IEEE International Conference on Computer Communications. Shanghai, China, 2011: 1701-1709
- [14] Akkaya K, Younis M, Youssef M. Efficient aggregation of delay-constrained data in wireless sensor networks// Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications. Washington, USA, 2005: 904-909
- [15] Srikanth H, Ness B. Deadline constrained scheduling for data aggregation in unreliable sensor network// Proceedings of the 9th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks. Princeton, USA, 2011: 140-147
- [16] Srikanth H, Ness B. Maximizing aggregated revenue in sensor networks under deadline constraints// Proceedings of the 48th IEEE Conference on Decision and Control. Beijing, China, 2009: 4846-4851
- [17] Srikanth H, Ness B. Maximizing information in unreliable sensor networks under deadline and energy constraints. Journal on IEEE Transactions on Automatic Control, 2013, 58(6): 1416-1429
- [18] Zheng Zizhan, Ness B. Maximizing a submodular utility for deadline constrained data collection in sensor networks// Proceedings of the 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks. Paderborn, Germany, 2012: 116-123
- [19] Tamer N, Sasan D, Chunyuan L, Liviu I. TrafficView: Traffic data dissemination using car-to-car communication. Journal on ACM Mobile Computing and Communications Review, Special Issue on MDM, 2004, 8(3): 6-19
- [20] Wischoff L, Ebner A, Rohling H, Lott M. SOTIS—A self-organizing traffic information system// Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference. Seoul, Korea, 2003: 2442-2446
- [21] Schoch E, Dietzel S, Bako B, Kargl F. A structure-free aggregation framework for vehicular ad hoc networks// Proceedings of the 6th International Workshop on Intelligent Transportation. Hamburg, Germany, 2009: 2359-2368
- [22] Dietzel S, Bako B, Schoch E. A fuzzy logic based approach for structure-free aggregation in vehicular ad-hoc networks// Proceedings of the 6th ACM International Workshop on Vehicular Ad Hoc Networks. Beijing, China, 2009: 79-88
- [23] Bjorn S, Christian L, Jędrzej R, Martin M. A probabilistic method for cooperative hierarchical aggregation of data in VANETs. Journal on Ad Hoc Networks, 2010, 8(5): 518-530
- [24] Khaled I, Michele C. CASCADE: Cluster-based accurate syntactic compression of aggregated data in VANETs// Proceedings of the 2008 Global Communications Conference Workshop. New Orleans, LO, 2008: 1-10
- [25] Khaled I, Michele C. Accurate data aggregation for VANETs// Proceedings of the 4th ACM International Workshop on Vehicular Ad-Hoc Networks. New York, USA, 2007: 71-72
- [26] Yu Bo, Gong Jiayu, Xu Chengzhong. Catch-Up: A data aggregation scheme for VANETs// Proceedings of the 5th ACM International Workshop on Vehicular Inter-Net Working. New York, USA, 2008: 49-57
- [27] Yu Bo, Xu Chengzhong. Adaptive forwarding delay control for VANET data aggregation. Journal on IEEE Transactions on Parallel and Distributed Systems, 2012, 23(1): 11-18
- [28] Claudio E, Fabio P, Pedro M. Delay-bounded data gathering in urban vehicular sensor networks. Journal on Pervasive and Mobile Computing, 2012, 8(2): 180-193
- [29] Dietzel S, Kargl F, Heijenk G, Schaub F. On the potential of generic modeling for VANET data aggregation protocols// Proceedings of the Vehicular Networking Conference. Jersey City, NJ, 2010: 78-85
- [30] Bjorn S, Christian L, Jędrzej R, Martin M. A fundamental scalability criterion for data aggregation in VANETs// Proceedings of the Annual International Conference on Mobile Computing and Networking. Beijing, China, 2009: 285-296
- [31] Giordano E, Frank R, Pau G. CORNER: A realistic urban propagation model for VANET// Proceedings of the 7th International Conference on Wireless On-demand Network Systems and Services. Kranjska Gora, Slovenia, 2010: 57-60
- [32] Giordano E, Frank R, Pau G. CORNER: A step towards realistic simulations for VANETs// Proceedings of the 6th ACM International Workshop on Vehicular Ad-hoc Networks. New York, USA, 2010: 41-50
- [33] Mohammad N, Ana A. Data gathering for sensing applications in vehicular networks// Proceedings of the 2011 Vehicular Networking Conference. Amsterdam, Holland, 2011: 222-229
- [34] Bruno D, Thierry D, Sergio I, Jose-Luis Z, Nicolas C. Data aggregation in VANETs; The VESPA approach// Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. Brussels, Belgium, 2008: 13-25
- [35] Hu Yong-Li, Sun Yan-Feng, Yin Bao-Cai. Information sensing and interaction technology in Internet of Things. Chinese Journal of Computers, 2012, 35(6): 1147-1163(in Chinese) (胡永利, 孙艳丰, 尹宝才. 物联网信息感知与交互技术. 计算机学报, 2012, 35(6): 1147-1163)
- [36] Wang Gui-Ping. Graph Theory and Algorithm: Theory, Implementation and Application. Beijing: Peking University Press, 2011(in Chinese) (王桂平. 图论算法理论、实现及应用. 北京: 北京大学出版社, 2011)
- [37] ShuiChi S, Mitsunori T, Koichi Y. A note on greedy algorithms for the maximum weighted independent set problem. Journal on Discrete Applied Mathematics, 2003, 126(2/3): 313-322
- [38] Thomas H, Char E, Ronald L, Clifford Stein. Introduction to Algorithm. 2nd Edition. USA: The MIT Press, 2008
- [39] Michal P, Natasa S, Matthias G. A parsimonious model of mobile partitioned networks with clustering// Proceedings of the Communication Systems and Networks. Bangalore, India, 2009: 156-163



**FENG Cheng**, born in 1986, Ph. D. candidate. Her research interests include data collection and data aggregation on mobile wireless sensor networks.

**LI Zhi-Jun**, born in 1977, Ph. D. , associate professor. His research interests include P2P, pervasive computing, WSN and operation system .

**JIANG Shou-Xu**, born in 1968, Ph.D. , professor, Ph.D. supervisor. His research interests include P2P, data-base, WSN and pervasive computing.

## Background

Real-time data gathering is the basic research work of Intelligent Traffic System. For ITS applications, the goal of data gathering is to get accurate real-time sensor data as much as possible with dynamic wireless resource. Data aggregation is a useful technology in the process of data gathering in order to effectively use communication bandwidth. Much research results have been gained in this area recently, but they did not consider data timeliness. We study deadline constrained data aggregation on wireless mobile sensor networks. It is analyzed as NPC problem. Approximation

algorithm frame is proposed. Its main idea is to separate different kinds of conflicts and eliminate them in the two phases. This paper is supported by the National Natural Science Foundation of China under Grant Nos. 61370214 and 61300210. These projects focus on the data processing in VANETs. Our group has been working on the in-network data processing methods and data routing and gathering in VANETs. Some research results have been accepted by VTC'12, WAIM'12 or other conferences and international journals.