

# 基于路由机制的变权网络路径快速生成算法

唐 俊 张栋良

(同济大学电子与信息工程学院 上海 201804)

(同济大学嵌入式系统与服务计算教育部重点实验室 上海 201804)

**摘 要** 在大规模交通流仿真中,车辆个体路径生成环节存在着大量重复计算。为避免重复计算及提高车辆个体路径生成速度,将计算机网络中的路由机制引入到交通流仿真中,提出一种基于路由机制的变权网络路径快速生成算法,即把每个道路路口节点作为路由器,分解并存储原本与车关联的路径作为指路信息。仿真车辆通过访问该指路信息获取下一步行车方向,并且当路网权值发生变化时,能及时响应路网的动态变化,从而给出实时路况下仿真车辆行驶路径的一种方法。

**关键词** 交通流仿真,路径生成,路由机制,动态路网

**中图法分类号** TP391.9 **文献标识码** A

## Routing Mechanism Based Algorithm for Fast Path Generation in Variable -Weight Network

TANG Jun ZHANG Dong-liang

(College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China)

(Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China)

**Abstract** There are a mass of redundant calculations when vast vehicles generate their pathes in the large-scale traffic flow simulation. In order to reduce the redundant calculation and improve the speed of path generation, a routing mechanism based algorithm for fast path generation in variable-weight network was proposed. The algorithm introduced the routing mechanism of network into traffic flow simulation system and took each road intersection as a router. It disassembled and stored the calculated shortest path tree into each relevant router as the guiding information like the next direction from A to B is C. Each vehicle got its next driving direction by visiting the current router when it drove to an intersection. As historical data, the stored guiding information would also improve the speed of future path generation. Furthermore, the algorithm would update the relative region on time when the weight of road network was changed and thus the path of each vehicle is still reasonable and considerable. The experimental results show that this algorithm can reduce the repetitive calculation and improve the speed of path generation.

**Keywords** Traffic flow simulation, Path generation, Routing mechanism, Dynamic road network

## 1 引言

在大规模交通流仿真中,仿真系统为了实现车辆的理性寻径,在生成仿真车辆时需要赋予每辆车一条行驶路径。在计算车辆行驶路径时,以往做法是利用最短路算法<sup>[1,2]</sup>或近似算法为每一辆车赋予一条完整的路径,这条路径伴随着车辆产生和消亡。然而在大规模交通流仿真中,仿真车辆的数量往往以百万计,而且路网庞大,节点以万计,数以百万的仿真车辆行驶路径的计算时间是难以容忍的。此外,因为车辆行驶路径与车辆本身绑定,使得在初期获得规划好的行驶路径后,车辆就会严格按照此路径行驶,这种路径计算方式并不适于实时的响应路网情况的变化。

为了提高仿真系统中车辆行驶路径的计算效率,多种方法被提出,如提出并利用最短路稳定性<sup>[3]</sup>的概念减少每次计

算量,用分层路网<sup>[4]</sup>的思想在较高层次路网上计算粗略路径,然后逐步细化来加速计算等。还有一些最短路改进算法<sup>[5-10]</sup>都在一定程度上提高了计算效率,但因为每次都需要进行重新搜索,重复计算严重,面对仿真中的大量并发路径查询请求也并不能做出及时的响应。

本文针对交通流仿真系统所面临的问题,效仿计算机网络路由思想,提出了一种基于路由机制的变权网络路径快速生成算法。该算法以路由机制为基础,即把每个道路节点作为一个“路由器”,存储从该道路节点往其他任意道路节点的指路信息,通过在路由机制中设计对路由表的更新、维护等操作,将每次计算车辆行驶路径时用最短路算法搜索得到的最短路路径树<sup>[11,12]</sup>(Shortest Path Tree)信息分解并以“下一跳”的标示方法存入到相关节点的路由项中作为指路信息。车辆通过访问节点上的指路信息获得下一步的行驶方向并继续行

到稿日期:2011-06-23 返修日期:2011-09-28 本文受 973 课题(2010CB328101),国家自然科学基金青年基金项目(61003089)资助。

唐 俊(1987-),男,硕士生,主要研究方向为移动计算与智能交通,E-mail: tangjun. rojay@gmail. com; 张栋良(1977-),男,博士后,主要研究方向为分布式计算与智能交通。

驶,直到最终到达目的地。每辆车的路径计算充分利用先前车辆路径计算的结果,降低了车辆路径的重复计算。并且当路网权值发生变化时,通过该算法中的动态更新子算法,使得路由表内的指路信息始终适应当前的网络权值变化。模拟实验显示,交通流仿真系统在车辆个体路径生成环节,采用本方法较每次都使用最短路径算法计算每辆车行驶路径的方法在计算时间上有较大优势。

## 2 相关理论基础

**定理 1** 给定一个带权图  $G=(V,E)$ ,其中  $V$  为节点集、 $E$  为边集,定义加权函数  $w: E \rightarrow R$  为从边到权值的映射。设  $p=\langle v_1, v_2, \dots, v_k \rangle$  是从  $v_1$  到  $v_k$  的最短路径。对于任意  $i, j$ , 其中  $1 \leq i \leq j \leq k$ , 设  $p_{ij}=\langle v_i, v_{i+1}, \dots, v_j \rangle$  为  $p$  从顶点  $v_i$  到顶点  $v_j$  的子路径。那么,  $p_{ij}$  是从  $v_i$  到  $v_j$  的最短路径<sup>[13]</sup>,即一条两顶点间的最短路径包含路径上其他的最短路径。

**定理 2** 已知图  $G=(V,E)$ ,在计算源点为  $s$  的最短路径算法时,对每个顶点  $v \in V$ ,设置其前趋顶点  $prev[v]$  为另一顶点或  $NIL$ ,使源于顶点  $v$  的前趋链表沿着从  $s$  到  $v$  的最短路径的相反方向排列。由  $prev$  导出的前趋子图  $G_{prev}=(V_{prev}, E_{prev})$  是以源点  $s$  为根的最短路径树,它包含了从源点  $s$  到  $s$  可达的每个顶点之间的一条最短路径。即设图  $G=(V,E)$  是带权图,以  $s$  为根的最短路径树是<sup>[13]</sup>其子图  $G'=(V',E')$ ,其中  $V' \subseteq V, E' \subseteq E$ ,那么有:

- 1)  $V'$  是  $G$  中从  $s$  可达的顶点集合;
- 2)  $G'$  形成了一棵以  $s$  为根的有根树;
- 3) 对所有的  $v \in V', G'$  中从  $s$  到  $v$  的唯一路径是  $G$  中从  $s$  到  $v$  的最短路径。

## 3 基于路由机制的变权网络路径快速生成算法

根据定理 1 和定理 2,可知利用最短路径算法能获得相应的最短路径树,在该最短路径树上任意祖先节点到其子孙节点的路径为最短路径,且任意子孙节点到其祖先节点的路径也为最短路径。若有仿真车辆  $C_a$ ,其  $OD$  点(Origin-Destination,起讫点)为  $(v_i, v_j)$ ,且在计算其行驶路径时已获得相应最短路径树  $G=(V,E)$ ,其中  $p_{ij}=\langle v_i, v_{i+1}, \dots, v_j \rangle$  是该仿真车辆  $OD$  点的最短路径。将该最短路径树  $G$  保存为历史数据,若有后续仿真车辆  $C_b$  的  $OD$  点  $(v_m, v_n)$  有  $v_m, v_n \in V$  且  $v_m \in \{v_i | v_i = v_n \text{ 或 } v_i = prev[v_i]\}$ ,则  $G$  中已包含了其最短路径  $p_{mn}=\langle v_m, v_{m+1}, \dots, v_n \rangle$ ,而无需再次用最短路径算法进行路径求解,避免了此次最短路径计算。若将每次获得的最短路径树以一定的方式保存,则可以减少后续仿真车辆在生成其行驶路径时的大量重复计算。

本文将计算机网络中的路由机制引入到车辆寻径环节,路由机制中的路由表以道路节点(路口)作为路由节点进行设计。

**定义 1** 定义路由表的路由格式为三元组:  $\langle curId, destId, nextId \rangle$ ,其中  $curId$  为当前所在的道路节点  $Id$ ,  $destId$  为目的地所在的道路节点  $Id$ ,  $nextId$  为仿真车辆从  $curId$  开往  $destId$  时,其下一步应往  $nextId$  方向行驶。

通过设计该路由机制中的更新算法将最短路径树分解并保存到路由表中作为历史数据;通过设计该路由机制中的动态更新算法使得算法能适应动态路网的道路权值变化情况。

在该路由机制的基础上提出了一种适合交通仿真环境的仿真车辆的路径快速生成算法,其基本流程见图 1 左图。仿真车辆在生成其  $OD$  点后,通过访问路由表查询其下一步行驶方向,若路由表中已存在相关指路信息,则直接根据该指路信息驶向下一个道路节点后再次查询路由表。否则计算其  $OD$  点之间的最短路径,并将计算出的最短路径树用路由表的更新算法将其更新到路由表中,仿真车辆通过访问路由表获得下一步行驶方向。在道路权值发生变化时通过动态更新子算法(见图 1 右图),更新一定区域内的所有节点对之间的最短路径信息,使得该区域内的车辆行驶路径依然是一条能及时响应道路情况的行驶路径。

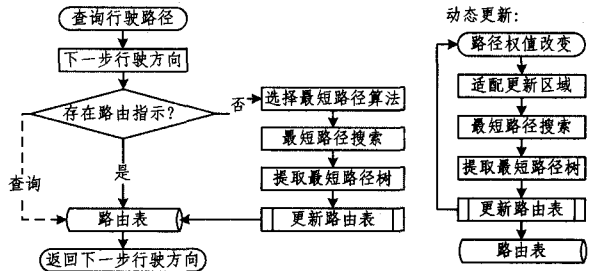


图 1 路径快速生成算法流程

路由表的更新算法将最短路径树分解并保存到路由表中。系统进行一次车辆  $OD$  点的最短路径搜索(Dijkstra 算法)后,更新算法将得到的最短路径树上的节点间的前驱/后继信息以定义 1 的形式更新到路由表中,避免了接下来在已存储点对之间的重复计算。其基本的更新思想是遍历这棵最短路径树  $G'=(V',E')$ ,将任意满足祖先/子孙关系的两节点的最短路径信息分解并保存到路由表中。此更新算法的时间复杂度为:

$$C = \sum_{i=1}^{\lfloor \log_2 n + 1 \rfloor} 2i \times k^{i-1} = O(n \lg n)$$

其算法伪代码为:

```
FUNCTION: updateStrategy
A = {x | x ∈ V' & x ≠ leaf}
∀ a ∈ A, 令 B = {x | prev[x] = a}
  ∀ b ∈ B, 令 C = {x | x = b | ∃ y ∈ C 有 prev[x] = y}
    ∀ c ∈ C, 令 d = prev[c]
      updateTable(a, c, b)
      updateTable(c, a, d)
```

路由表的动态更新算法在路网上的权值(以车辆通过该路段所花费的平均时间为该路段权值)发生变化时,例如某条道路因交通堵塞而权值剧增,动态更新路网中部分区域内的最短路径指路信息,使得原本将要经过该条已堵塞的道路的仿真车辆能获得一条较原先不改变行驶路径更合理的路径。当道路权值的变化超过一定的阈值时,程序会更新路由表中一定范围内的相关路由信息,以动态适应路网的变化。其思想是以道路权值发生变化的那一段路两边的道路节点为中心,往外延伸至多 5 层道路节点为一个更新区域。重新计算该区域内任意两道路节点之间的最短路径并用路由表的更新算法更新到路由表中,使路由表内的指路信息能反映出路网的动态变化。

其算法伪代码为:

```
FUNCTION: dynamicUpdateStrategy(id1, id2)
id1.level = id2.level ← 1
```

```

Queue={id1,id2},Close=φ
for first node(id)∈Queue
if (id.level>5) break the loop;
Queue=Queue-id,Close←-id
A={x|x∈id's adjacentNode&x∈Queue&x∉Close}
∀a∈A,a.level←-id.level+1,Queue←-a
∀b∈Close
Dijkstra(b) in Close
updateStrategy()

```

#### 4 性能测试

为了测试本文算法的性能,我们进行了模拟实验。实验采用 9328 个道路节点的上海市导航地图路网。在该路网的基础上分别进行 2 项测试:

1) 大规模的仿真车辆并发查询时,该算法实际的最短路计算次数。

2) 采用最短路算法(Dijkstra 算法及 A\* 算法)与本文所采用算法的时间开销对比。

第一项性能测试中,定义 10000 辆仿真车辆在同一时刻进行并发查询为 1 次测试,共进行连续的 20 次测试,统计这 20 次测试中每次实际进行的最短路计算次数。实验结果(见图 2)显示,基于路由机制的动态路网路径快速生成算法因为保存了先前计算的历史数据,其每次测试时实际进行的最短路计算次数少于车辆的路径查询次数,并在测试的中后期,每次只需进行极少量的最短路计算。这是因为路由表中已经保存了足够多的历史数据,这些历史数据的存在避免了大量重复计算,具有良好的性能表现。

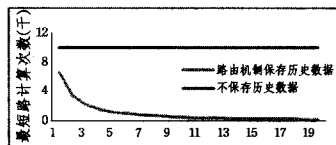


图 2 实际进行的最短路计算次数对比

第二项性能测试中,以每次有 10000 辆新上路的车并发查询为 1 次查询,对比优先队列的 A\* 算法、优先队列的 Dijkstra 算法及路径快速生成算法对此的时间开销为一组数据,连续测试 20 次。实验结果显示(见图 3),采用优先队列的 A\* 启发式算法每次的时间开销约为 6s(平均每次 A\* 搜索时间约在 0.6ms),采用优先队列的 Dijkstra 算法的每次时间开销约为 14s。而本文提出的路径快速生成算法最初的时间开销约为 20s,但随着测试次数的增加,路由表内的指路信息快速丰富,该算法的计算时间下降明显以致趋近于 0,性能优势明显。需要说明的是,之所以路径快速生成算法中头几组数据的时间开销要比 A\* 算法甚至 Dijkstra 算法的时间开销还要高,是因为其采用的是以 Dijkstra 算法为核心的路径搜索算法。首先 Dijkstra 算法的时间开销约是 A\* 算法的时间开销的 2~3 倍(见图 3),其次路径快速生成算法还需要占用部分时间开销来更新路由表内的指路信息。而更新路由表所花费的时间暂时还要超过刚开始计算时减少的相对较少的重复计算省下的时间。

定义每次测试时的该路径快速生成算法的加速比=优先队列的 A\* 算法总计算时间/路径快速生成算法总计算时间。其加速比(见图 4)也从首次运行时的约 0.3 逐渐上升到了第

20 次测试时的 8,加速效果较理想,实验结果较理想。

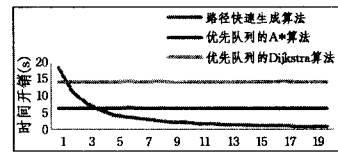


图 3 不同路径生成算法的时间开销对比

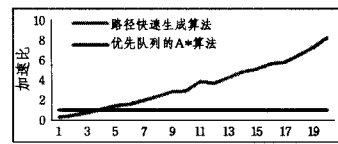


图 4 路径快速生成算法的加速比

**结束语** 本文针对目前交通流仿真系统在生成大量仿真车辆时对其行驶路径的重复计算,借鉴计算机网络中的路由思想,提出了一种基于路由机制的变权网络路径快速生成算法。该算法通过路由机制分解并保存先前仿真车辆最短路计算后的最短路树作为指路信息,车辆通过访问该指路信息获取下一步行车方向直到到达目的地。当路网权值发生变化时,通过动态更新子算法,使得仿真车辆的行驶路径依然是一条能及时响应道路变化的路径。实验表明,仿真系统采用路径快速生成算法能有效减少重复计算,节约时间开销,且能适应路网权值的动态变化。

该算法暂时还存在一些不足,例如其虽然能通过路由表的动态更新算法,使得仿真车辆的行驶路径能适应路网权值的动态变化,但却无法保证其行驶路径为其 OD 点之间的最短路径,而只能保证仿真车辆的行驶路径要优于未响应路网权值变化的原先的行驶路径。而且当路网权值大范围且高频密集变化(超过阈值)时(路网中一般不会发生该情况),其动态更新算法的性能也有待改进。但由于交通流仿真系统是仿真现实生活中车辆行驶行为,而生活中的车辆并不会严格按照 OD 点之间最短路径行驶,因此这些不足目前还可令人接受。

该算法目前已应用于 863 普适计算国家重点示范项目“路网中的个人信息服务”子课题中的路径引导和同济大学交通网格信息平台出行服务中的路径规划。接下来将尝试改进该算法中的动态更新算法,并利用其路由机制能保存历史数据的优势,以一种新的尝试方法将其应用于 LBS(Location Based Service)路网 k 近邻算法<sup>[14]</sup>的研究中。

#### 参考文献

- [1] Dijkstra E W. A note on two problems in connection with graphs [J]. *Numerische Mathematik*, 1959, 1(4): 269-271
- [2] Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths in graphs [J]. *IEEE Trans. Syst. Sci. and Cybernetics*, 1968, SSC-4(2): 100-107
- [3] 林澜, 闫春钢, 辛肖刚, 等. 基于稳定分支的变权网络最优路径算法 [J]. *电子学报*, 2006, 34(7): 1222-1225
- [4] 李兵, 郑四发, 曹剑东, 等. 基于路网分层策略的多源点最短路径距离 [J]. *计算机工程*, 2008, 34(12): 37-39
- [5] 张学敏. GIS 环境下的动态交通最优路径算法研究 [D]. 长沙: 中南大学, 2009
- [6] 姜贵山, 江志斌, 刘树军. 改进的引导式领域搜索算法求解周期性车辆路径问题 [J]. *上海交通大学学报*, 2010, 44(9): 1171-1175

(下转第 124 页)

管也是直线,但有部分点出现了偏离。

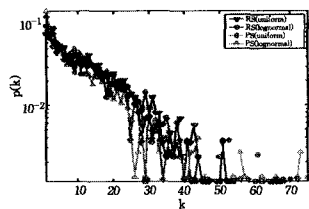


图 16 两种方式形成的网络度分布比较

图 17 考查了种子数对两种方式下形成的网络的平均路径长度(AVL)、聚类系数  $C$  和最大子网节点数在网络中所占比例数(SubNet)的影响。图例中第一个字母为“R”代表随机选择,“P”代表优先选择。从图中可以看出,在网络规模不变的情况下,种子节点越多,形成的网络包含的节点数就越多,即网络规模越大。随着种子数的增加,随机选择方式的网络平均路径明显变短,而优先选择方式的平均路径长度变化不大。总的来看,优先选择方式的聚类系数比随机选择方式大。大量实验结果表明,两种方式形成的网络的平均路径长度增加的速度远远小于网络规模按对数方式增长的速度。比如,在种子数取 25 的情况下,按随机方式和优先方式形成的规模为 1000 的网络的平均路径长度分别为 2.54 和 2.65,形成的规模为 2000 的网络平均路径长度分别为 2.78 和 2.68。因此,基于两种方式形成的网络都具有小世界效应<sup>[1]</sup>。

根据仿真形成的逻辑网络的平均路径长度、聚类系数和度分布,并结合复杂网络相关理论<sup>[1]</sup>,可以确定,P2P 逻辑网络是度分布为指数分布的小世界网络。

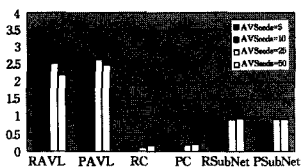


图 17 不同种子数对网络形成的影响

**结束语** P2P 逻辑网络的拓扑结构对网络的性能、健壮性和病毒的传播都有重要影响,因而一直是 P2P 网络研究的重点和热点。尽管如此,迄今为止还是没有明确 P2P 逻辑网络的拓扑结构。混淆 P2P 逻辑网络与覆盖网络以及研究中采用的测量方法难以在极短的时间内获取网络中所有节点的相关信息,是无法确定 P2P 逻辑网络拓扑结构的主要原因。鉴于此,本文首先廓清了 P2P 逻辑网络和 P2P 覆盖网络的关

系;在此基础上,考虑到仿真的方法能及时快速地获取网络中所有节点的信息,采用仿真的方法研究了 P2P 逻辑网络的拓扑结构。在深入分析 P2P 网络协议的基础上,设计并实现了 P2P 网络仿真系统。为了正确地确定 P2P 逻辑网络的拓扑结构,基于开发出的仿真系统进行了大规模仿真实验。仿真实验主要对网络的 3 大特征系数进行考查。仿真出的 P2P 逻辑网络都有较小的平均路径长度和较大的聚类系数,而度分布都为指数分布。根据这 3 个网络特征参数的特点并利用复杂网络理论可以确定,P2P 逻辑网络是一个度分布为指数分布的小世界网络。

## 参考文献

- [1] 汪小帆,李翔,陈关荣. 复杂网络理论及其应用[M]. 北京:清华大学出版社,2006
- [2] Ripeanu M, Foster I, Iamnitchi A. Mapping the Gnutella Network: Properties of Large-scale Peer-to-Peer Systems and Implications for System Design[J]. IEEE Internet Computing Journal (special issue on peer-to-peer networking), 2002, 6(1): 50-57
- [3] Liang J, Kumar R, Ross K W. The KaZaA overlay: A measurement study[C] // Proc. of the 19th IEEE Annual Computer Communications Workshop. Florida, 2004
- [4] Stutzbach D, Rejaie R. Characterizing the Two-tier Gnutella Topology [C] // ACM SIGMETRICS. Extended Abstract, June 2005
- [5] The Gnutella Protocol Specification v0.4 Document Revision 1.2 [EB/OL]. <http://www.clip2.com>
- [6] 王跃武,荆继武,向继,等. 拓扑相关蠕虫仿真分析[J]. 软件学报, 2008, 19(6): 1508-1518
- [7] eMule website[EB/OL]. <http://www.emule-project.net/>
- [8] 蒋君. eMule 系统中的覆盖网络研究[D]. 上海:上海交通大学, 2008
- [9] Stutzbach D. Measuring and Characterizing Properties of Peer-to-Peer System[D]. America: University of Oregon, 2006
- [10] Ilie D. On Unicast QoS Routing in Overlay networks[D]. Sweden: Blekinge Institute of Technology(BTH), 2008
- [11] 刘斯伟. 基于测量的 eMule/Kad 网络 k 桶特性研究[D]. 北京: 北京交通大学, 2008
- [12] Thommes R, Coates M. Epidemiological Modeling of Peer-to-Peer Viruses and Pollution[C] // Proceedings of IEEE INFOCOM. April 2006

(上接第 112 页)

- [7] 谭德荣,严新平. 变权值加快收敛的路径寻优实时算法[J]. 交通运输工程学报, 2004, 4(1): 118-120
- [8] 胡腾波,叶建栋. GIS 时变权值网络最短路径算法研究[J]. 计算机与现代化, 2008, 11: 22-24
- [9] Huang Yi-bing. An improved dijkstra algorithm for the graphs of Shortest-Path tree[J]. Journal of Beijing Institute of Machinery, 2002, 17(4): 50-55
- [10] 雷东升,诸彤宇. 一种基于实时路况信息的动态路径规划算法[J]. 计算机科学, 2008, 35(4): 28-30

- [11] 王涛,李伟生. 低代价最短路径树的快速算法[J]. 软件学报, 2004, 15(5): 660-665
- [12] 孙知信,高艳娟,王文鼎. 更新最短路径树的完全动态算法[J]. 吉林大学学报:工学版, 2007, 37(4): 860-864
- [13] Cormen T H, Leiserson C E, Rivest R L, et al. 算法导论[M]. 北京:机械工业出版社, 2007: 357-359
- [14] Shin S-H, Lee S-C, Kim S-W, et al. Efficient Shortest Path Finding of k-Nearest Neighbor Objects in Road Network Databases [C] // SAC'10. 2010: 1661-1665