

交通网络最短路径标号算法的实现与效率分析

陈 洁 陆 锋

(中国科学院地理科学与资源研究所资源与环境信息系统国家重点实验室, 北京 100101)

摘 要 标号算法是交通网络最短路径算法族中应用最广泛的算法, 其中以各种 Dijkstra 算法为核心的标号设定算法是各种商用 GIS 平台网络分析算法的首选。然而, 同样隶属于标号算法的标号改正算法在交通网络路径分析中却罕有应用。为了将标号改正算法应用于交通网络路径分析, 首先讨论了标号算法的基本结构; 然后分析了标号设定算法和标号改正算法的实现过程、复杂度、运行特点和适用性, 进而选择了标号设定和标号改正算法中公认的几种优秀算法——基于逼近桶结构和改进四叉堆的 Dijkstra 算法(DIKBA 与 DIKQH)以及 Pallottino 算法(TWO_Q), 并结合交通网络邻接链表结构予以实现; 最后采用城市交通网络数据, 对几种算法的实际运行效率进行了对比试验, 试验结果表明, 标号改正算法和标号设定算法优点各异; 由于交通网络路径算法的应用越来越强调动态性和网络适用性, 而且标号改正算法较之标号设定算法具有更大的适用范围, 因此其在交通网络路径分析中具有极大的应用潜力。

关键词 最短路径算法 标号算法 复杂度 交通网络

中图分类号: TP302.7 P208 文献标识码: A 文章编号: 1006-8961(2005)09-1134-05

Implementation and Evaluation of the Shortest Path Labeling Algorithms in Transportation Networks

CHEN Jie, LU Feng

(LERIS, Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing 100101)

Abstract Labeling algorithms have got broad applications for shortest path finding in transportation networks, among which various fine-tuned Dijkstra's algorithms well known as typical label setting algorithms have been selected by many GIS related software for network analysis. However, label correcting algorithms, the other group in label algorithms family, are rarely used yet in GIS network analysis. After detailed discussion on the structures of labeling algorithms, in this paper, the implementation, complexity and applicability of labeling setting and label correcting algorithms are analyzed. Then three best-known fastest label algorithms, i. e., Dijkstra algorithm implemented with approximate buckets (DIKBA), Dijkstra's algorithm based on quad-heap priority queue (DIKQH) and Pallottino algorithm (TWO_Q), were used to carry out practical evaluation on three real urban road networks. The results showed that for one-to-one shortest path calculation, DIKQH and DIKBA greatly outperformed than TWO-Q algorithm, and DIKQH exhibited the best running efficiency. For one-to-all shortest path calculation, however, TWO-Q algorithm runs a little faster than DIKQH and DIKBA on the selected real road networks. The author argued that more attention should be paid on TWO-Q algorithm for its efficiency and applicability.

Keywords shortest path algorithms, label algorithms, complexity, transportation networks

1 引 言

交通网络是最短路径算法的主要应用领域, 相

关研究层出不穷, 其各种最短路径算法的分析评述
可见文献[1~6]。众所周知, 最短路径算法是资源
分配、路线设计及分析等优化问题的基础, 而交通
网络分析中的其他问题, 如最可靠路径问题、最大容

基金项目: 国家自然科学基金项目(40201043); 中国科学院知识创新工程前沿项目(CXIOG-D04-02)

收稿日期: 2004-09-06; 改回日期: 2005-01-25

第一作者简介: 陈洁(1982~), 女, 2004年获中国地质大学(武汉)学士学位, 现为中国科学院资源与环境信息系统国家重点实验室硕士生。主要从事空间数据库技术和城市交通网络分析算法研究。E-mail: chenj@reis.ac.cn

量路径问题、各种各样的路径导航问题也都可以归并到最短路径问题类型中^[2]。标号算法(labeling algorithms)是最短路径算法族中的重要成员,按照不同的标识结点处理策略,标号算法又可分为标号设定(label setting,简称LS)和标号改正(label correcting,简称LC)两大体系。

对于LS算法,国内外专家学者进行了大量研究。GIS软件产品中的网络分析模块也多是基于LS算法实现,如各种改进的Dijkstra算法^[7-9]。对于LC算法,一般认为该算法的最大优势是可接受网络负权边,但对于交通网络而言,由于无论以距离、时间和费用作为权值,一般都不存在负权边,加之LC算法较之LS算法复杂难懂,因而LC算法长期以来并未得到重视,而且在商用GIS软件平台中也鲜有应用。

据有关研究表明,对于不同网络形式,各种算法效率差异很大,也没有一种算法针对各种网络形式均有较好的效率^[4]。为了寻求一种较好的算法,以用于交通网络分析,本文针对交通网络,对LS算法族和LC算法族在时间复杂度、空间复杂度、易实现性、运行效率和算法适用性等方面存在什么样的差异,以及如何有针对性地选择合适的路径算法等问题进行了讨论。

2 标号算法分类与算法描述

由于问题特征、网络特性等的纷繁复杂,因此最短路径算法表现出多样性。总的来说,最短路径算法可按问题类型、网络特征和求解技术进行分类^[6]。

最短路径算法按求解技术进行分类时,其中路径搜索技术中的通用技术是最短路径算法研究的重点所在,而标号算法就是路径搜索通用技术分类中组合技术的主要算法类型,它也是绝大多数最短路径算法的核心部分。

2.1 标号设定与标号改正算法

LS算法最早由荷兰数学家Dijkstra于1959年提出,它是理论上最完善、迄今为止应用最广的非负权值网络最短路径算法^[10]。LS算法中的Dijkstra算法的广泛应用,使之成为LS算法的代名词^[11],而其他LS算法多为Dijkstra算法的不同实现方式,其中基于堆结构和桶结构优先级队列的LS算法是研究得最深入、应用也最广泛的LS算法。

针对网络中可能存在负权边的问题,LC算法采用不同的启发式策略,其代表性算法包括Bellman-Ford-Moore算法(即queue算法)、D'Esopo-Pape算法(即deque算法)、Pallottino算法(即2 queue算法或TWO_Q算法)^[1]、门限算法^[12]、拓扑排序算法^[13]和SLF(small label to the front)算法^[11]等等。

实际上,所有的LS或LC算法都可归结为一种更为一般性的算法的特例。它们的不同之处在于对结点扫描的次数以及处理图中所标识结点时采用的优先级队列系统各不相同。

2.2 标号算法描述

单源最短路径标号算法的标号特征如下:

设 $G=(V,E)$ 是一个有向图,其中, V 为结点集合, E 为边集合。对于图中结点集合 V 中每一个结点 j ,赋予以下两个数值(即“标号”):一个是距离标号 $d[j]$,其记录的是从起点到该结点的最短路径长度的上界;另一个是前驱标号 $p[j]$,其记录的是当起点 s 到该结点 j 的一条路径长度取到该上界时,该条路径中结点 j 的直接前驱结点。这样,算法通过不断修改这些标号来进行迭代计算,即可求解最短路径。

迭代计算的过程中,所有结点实际上被分成了以下两类:一类是离起点 s 较近的结点,它们的距离标号表示的是从点 s 到该结点的最短路径长度,其标号不会在以后的迭代中再被改变,故称为永久标号;另一类是离起点 s 较远的结点,它们的距离标号表示的只是从点 s 到该结点的最短路径长度的上界,由于其标号还可能会在以后的迭代中再被改变,因此称为临时标号。

标号算法的目的就是在算法结束时,将所有的临时标号都转变为永久标号。这时,距离标号表示的就是从起点到该结点的最短路径长度。

LS算法通过迭代过程对标号进行逐步修正,每次迭代均选择候选结点集中的标号最小者,然后将其退出候选结点集,并将该结点标号从临时标号转变为永久标号。这是一种基于贪心策略的最短路径算法,其要求在路径选择中的每一步所选择的路径都是到目前为止最好的,以起点至当前结点路径权值之和作为贪心选择策略,在“局部最优而导致总体最优”的假设下,寻求最佳路径。

LC算法在每次迭代时,并不一定将任何结点标号都从临时标号转变为永久标号,只是对临时标号进行一次修正,而其所有结点标号仍然为临时标号,

只有在所有迭代终止时,才将所有结点标号同时转变为永久标号。LC 算法考虑的是“最终最优”,因此最短路径需要等待多次迭代,直到整个算法运行结束才能被确定。

3 标号算法流程及复杂度分析

LS 算法的执行流程如下:

(1) 初始化。令集合 $S = \emptyset$, S 的补集 $S' = V$, $d[s] = 0, p[s] = -1$; 对 V 中的结点 $j (j \notin S)$, 令其初始距离标号 $d[j] = \infty$;

(2) 如果 $S = V$, 则 $d[j]$ 为起点 s 到结点 j 的最短路径长度, 其最短路径可以通过前驱标号 $p[j]$ 所记录的信息反向追踪获得, 结束, 否则继续步骤 3;

(3) 从 S' 中找到距离标号最小的结点 i , 把它从 S' 中删除, 加入 S 。对于所有从 i 出发的弧 $(i, j) \in E$, 其弧长为 $l_{i,j}$, 若 $d[j] > d[i] + l_{i,j}$, 则令 $d[j] = d[i] + l_{i,j}, p[j] = i$, 转步骤 2。

该算法的主要计算量在于步骤 3 循环。它包括以下两个过程: 结点寻找过程(从 S' 中找到距离标号最小的结点 i) 和距离修改过程(修改与结点 i 相邻的结点的距离标号)。

若以 n 表示网络结点数、以 m 表示弧段数(下同), 则在不引入任何优先级队列结构的情况下, 传统的单源 Dijkstra 算法对于结点寻找过程, 第 1 次循环时, 需要 n 次比较操作; 第 2 次循环时, 需要 $n-1$ 次比较操作, ..., 依次类推, 若以 $T(n)$ 表示最坏情况的运行时间, 以 O 记号给出函数 $T(n)$ 的上界^[10](下同), 则该过程的复杂度为 $T(n) = n + (n-1) + \dots + 1 = O(n^2)$ 。对于距离修改过程, 由于其整体效应相当于对每条弧进行一次检查, 因此该过程复杂度为 $O(m)$, 而该算法的时间复杂度为 $O(n^2 + m) = O(n^2)$ 。对于稠密网络, 这是求解单源最短路径问题可能达到的最小的复杂度^[14]。

对于稀疏网络, 若利用各种形式的优先级队列(如堆、桶等)来改进算法的运行结构, 则算法的时间复杂度可不同程度地降低, 例如, 基于 K 叉堆的单源 Dijkstra 算法的复杂度为 $O(m \log n)$, 基于桶结构的单源 Dijkstra 算法的复杂度为 $O(m + nC)$ (其中 C 为常量)等等。

LC 算法的执行流程如下:

(1) 令距离标号 $d[s] = 0$, 前驱标号 $p[s] = -1$; 对其他的结点 j , 令初始距离标号 $d[j] = \infty$;

(2) 如果对所有的弧 (i, j) , 有 $d[j] \leq d[i] + l_{i,j}$, 则结束, $d[j]$ 就是从起点 s 到结点 j 的最短路径长度, 即最短路径可以通过前驱标号获得, 否则进入步骤 3;

(3) 找到一条满足 $d[j] > d[i] + l_{i,j}$ 的弧 (i, j) , 令 $d[j] = d[i] + l_{i,j}, p[j] = i$, 则转步骤 2。算法的总时间复杂度为 $O(n^2 m)$ 。

对比上述 LS 与 LC 算法执行流程可以看出:

由于 LS 算法在候选结点集处理的每一步, 都能得到一条由源点到其余结点的最短路径, 因此在进行 1:1 路径搜索时, 当终点跳出候选结点集时即可结束, 但由于每个结点在整个算法运行过程中只进出候选集一次, 因此 LS 算法不能处理含负权边的网络, 而一般只能用来求解无圈网络和正费用网络的最短路径问题。

LC 算法在处理结点时, 可能需要多次进出优先级队列, 因此起点到终点的 1:1 最短路径需要等到算法全部结束才能得到。值得说明的是, 结点多次出入优先级队列是为了完成结点距离标号的多次迭代计算, 这使得 LC 算法摆脱了 LS 算法对于网络非负权边的约束, 因而其可以处理含负权边的网络, 即对于一般费用网络的最短路径问题均适用。

长期以来, 专家学者们对 LS 和 LC 算法的效率比较进行了深入的研究。文献[12]认为, 对于稠密图, Dantzig 算法优于 D'Esopo-Pape 算法; 而对于稀疏图, 虽然有指数形式的最坏时间界, D'Esopo-Pape 算法仍然是最优的^[12]。文献[3]的研究结果也表明, D'Esopo-Pape 算法可与最好的 LS 算法相媲美^[3]。文献[4]的研究结果表明, 对于不同的网络形式, 不仅各种算法效率差异很大, 而且没有一种算法针对各种网络形式均有较好的效率, 例如对于矩形规则格网型网络, LC 算法中的 Pape 算法和 2que 算法效率最高; 对于稠密网络, 双桶 Dijkstra 算法(即 DIKBD (dijkstra's algorithm implemented with double bucket) 算法)和门限算法效率最高; 对于非平面的规则格网型网络, DIKBD 算法和堆结构的 Dijkstra 算法(即 DIKH (dijkstra's algorithm implemented with heap) 算法)效率最高; 对于无环图, DIKBD 算法和拓扑排列算法效率最高^[4]。

值得注意的是, 理论上最优的算法不一定在实践中最优。总的来说, 基于逼近桶结构的 Dijkstra 算法(即 DIKBA (dijkstra's algorithm implemented with approximate bucket) 算法)^[4]、基于改进四叉堆

结构的 Dijkstra 算法(即 DIKQH(dijkstra's algorithm implemented with quad heap)算法)^[7]和 Pallottino 算法(即 TWO_Q 算法)^[4]分别是 LS 算法和 LC 算法中公认的最为优秀的算法,其时间复杂度分别为 $O(m\beta + n(\beta + C/\beta))$ 、 $O(n \log n)$ 和 $O(n^2m)$ 。本文结合 3 套城市交通网数据分别测试这 3 种算法在 1:1、1:N 最短路径搜索中的运行效率,并将实验结果与理论分析结果进行了对照。

4 标号算法在城市交通网络中的运行效率比较

4.1 运行环境

实验中的算法代码均使用标准 C 语言编写,在 P4 2.8G CPU、512M 内存的 PC 硬件环境、VC++6.0 编译环境中经调试编译通过。

为了比较不同标号算法的运行效率,本文选用 3 套城市交通网数据作为实验交通网络(网络基本信息如表 1 所示)进行了试验对比,计算中,以弧段长度作为边权值。

表 1 实验交通网络

Tab.1 Characteristics of real-world networks in the study

实验网络	结点数	弧段数
A	12 803	18 107
B	11 893	16 689
C	4 893	7 924

4.2 效率比较与分析

各算法的实际运行时间均指执行算法的 CPU 耗时,而不包括数据输入和输出等操作占用的时间。

对于各实验网络,在做 1:1 时间统计时,采取系统抽样方法从网络中选取 100 个结点作为起点、100 个结点作为终点,以保证被选取的结点能覆盖整个网络。求解最短路径时,首先分别统计这 100 个起点中每个结点到每个终点的 1:1 最短路径的运行时间,然后对所有的 1:1 耗时求取平均值;在做 1:N 时间统计时,首先对网络中每个结点分别统计其搜索所有结点所需的时间开销,然后对所有结点的 1:N 耗时求取平均值。

各算法耗时统计结果如图 1 所示。

由图 1 明显可见,以城市交通网 A、B、C 为实验网络,当执行 1:1 路径搜索时,LS 算法族中的 DIKBA 算法和 DIKQH 算法,由于允许到达搜索的终止结点后即可退出算法循环,因此运行效率均明显高于 LC 算法族中的 TWO_Q 算法。由于运行 TWO_Q 算法时,必须等待算法全部结束才能跳出,因此,对于同一网络,1:1 搜索和 1:N 搜索的 CPU 耗时相同。当执行 1:N 最短路径搜索时,3 种算法的实际运行效率的相对差异发生了较大改变,其中,在交通网络 A 和 B 中,TWO_Q 算法运行效率最高,而 DIKQH 算法次之,DIKBA 算法最低;在交通网络 C 中,DIKQH 算法运行效率跃升至最高,TWO_Q 算法次之,DIKBA 算法依然最低。

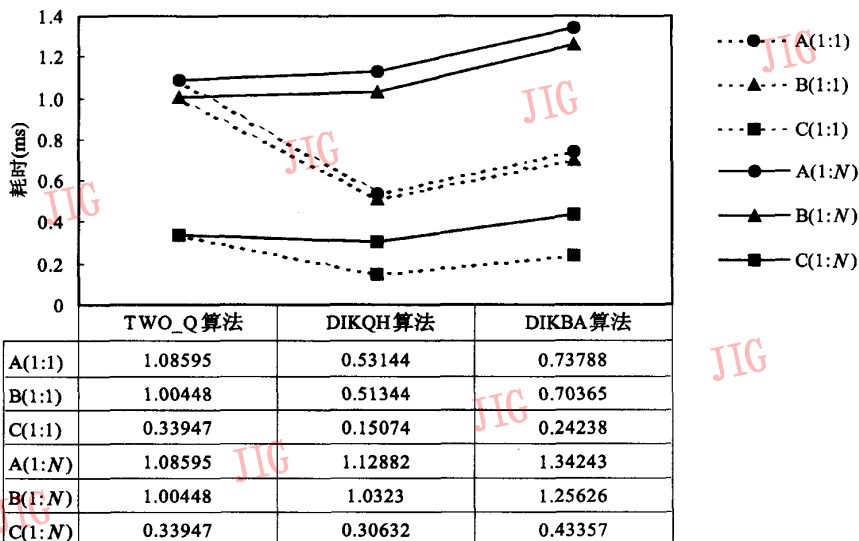


图 1 TWO_Q 算法、DIKQH 算法、DIKBA 算法路径搜索 CPU 耗时(ms)

Fig.1 Consumed CPU time(ms) of TWO_Q, DIKQH, DIKBA algorithm in shortest path searching

根据上一节对 LS 算法和 LC 算法的时间复杂度的理论分析, DIKBA 算法的最坏时间复杂度为 $O(m\beta + n(\beta + C/\beta))$, 而 TWO_Q 算法的最坏时间复杂度为 $O(n^2m)$, 可见 TWO_Q 算法的时间复杂度远大于 DIKBA 算法。然而, 在实际运行过程中, 算法的运行效率与所选网络的具体特征关系密切, 其运行效率的高低并不一定与与理论上的时间复杂度保持一致性关系, 这是需要引起高度注意的地方。

5 结 论

据本文研究与实验过程可以得出以下结论:

(1) 在交通网络最短路径搜索中, 当处理 1:1 最短路径搜索时, LS 算法允许搜索到终止结点即可退出运算过程, 而 LC 算法则需要完成全部结点的最短路径搜索才能得到 1:1 的最短路径。LS 算法族中最优秀的 DIKBA 算法和 DIKQH 算法较之 LC 算法族中最优秀的 TWO_Q 算法效率高。由于 DIKBA 算法要求整数权值, 因此对于以长度、时间或者费用等浮点类型的权值需要做整形化处理, 而且由于采用以权值为地址的桶运行结构, 因而使得 DIKBA 算法的空间复杂度较之 DIKQH 算法略大, 算法结果也存在一定的误差。在 1:N 最短路径搜索中, TWO_Q 算法与 DIKQH 算法效率接近, TWO_Q 算法略快。而 DIKBA 算法较之 TWO_Q 算法与 DIKQH 算法略差。

(2) LS 算法族与 LC 算法族优点各异, 其中, LC 算法族中的 TWO_Q 算法既允许网络中存在负权边, 又不会对边造成权值精度损失, 而且其 1:N 路径搜索效率高于 LS 算法族中的最优算法具有可比性。此外, 由于交通网络路径算法的应用越来越强调动态性和网络适用性, 而且 LC 算法族中的 TWO_Q 算法较之 LS 算法族中最优的 DIKBA 算法和 DIKQH 算法具有更大的适用范围, 因此其在交通网络路径分析具有极大的应用潜力。

参考文献 (References)

1 Pallotino S. Shortest path methods: complexity, interrelation and

new propositions[J]. Networks, 1984, 14: 257 ~ 267.

2 Deo N, Pang C Y. Shortest-path algorithms: taxonomy and annotation [J]. Networks, 1984, 14: 275 ~ 323.

3 Gallo G, Pallotino S. Shortest paths algorithms [J]. Annals of Operations Research, 1988, 13: 3 ~ 79.

4 Cherkassky B V, Goldberg A V, Radzik T. Shortest paths algorithms: theory and experimental evaluation [J]. Mathematical Programming, 1996, 73: 129 ~ 174.

5 Zhan F B, Noon C E. Shortest path algorithms: an evaluation using real road networks [J]. Transportation Science, 1998, 32(1): 65 ~ 73.

6 Lu Feng. Shortest path algorithms: taxonomy and advance in research [J]. ACTA GEODAETICA et CARTOGRAPHICA SINICA, 2001, 30(3): 269 ~ 275. [陆锋. 最短路径算法: 分类体系与研究进展 [J]. 测绘学报, 2001, 30(3): 269 ~ 275.]

7 Lu Feng, Lu Dong-mei, Cui Wei-hong. Improved Dijkstra algorithm based on quad-heap priority queue and inverse adjacent list [J]. Journal of Image and Graphics, 1999, 4A(12): 1044 ~ 1050. [陆锋, 卢冬梅, 崔伟宏. 基于四叉堆优先级队列及逆邻接表的改进型 Dijkstra 算法 [J]. 中国图象图形学报, 1999, 4A(12): 1044 ~ 1050.]

8 Yue Yang, Gong Jian-ya. An efficient implementation of shortest path algorithm based on Dijkstra's algorithm [J]. Journal of Wuhan Technical University of Surveying and Mapping, 1999, 24(3): 209 ~ 212. [乐阳, 龚健雅. Dijkstra 最短路径算法的一种高效率实现 [J]. 武汉测绘科技大学学报, 1999, 24(3): 209 ~ 212.]

9 Wang Kai-yi, Zhao Chun-jiang, Xu Gui-xian, et al. A high-efficiency realization way of the shortest path search problem in GIS field [J]. Journal of Image and Graphics, 2003, 8A(8): 951 ~ 956. [王开义, 赵春江, 胥桂仙等. GIS 领域最短路径搜索问题的一种高效实现 [J]. 中国图象图形学报, 2003, 8A(8): 951 ~ 956.]

10 Pan Jin-gui, GU Tie-cheng, Zeng Jian, et al. Common data structure and algorithms of modern computer [M]. Nanjing: Nanjing University Press, 1994. [潘金贵, 顾铁成, 曾俭等编译. 现代计算机常用数据结构 and 算法 [M]. 南京: 南京大学出版社, 1994.]

11 Bertsekas D P. A simple and fast label correcting algorithm for shortest paths [J]. Network, 1993, 23: 703 ~ 709.

12 Glover F, Glover R, Klingman D. Computational study of an improved shortest path algorithm [J]. Networks, 1984, 14: 25 ~ 36.

13 Goldberg A V, Radzik T. A heuristic improvement of the Bellman-Ford algorithm [J]. Applied Mathematics Letters, 1993, 6(3): 3 ~ 6.

14 Xie Jin-xing, Xing Wen-xun. Network optimization [M]. Beijing: Tsinghua University Press, 2000. [谢金星, 邢文训编著. 网络优化 [M]. 北京: 清华大学出版社, 2000.]