最短路径算法分析及其在公交查询的应用

陈箫枫 蔡秀云 唐德强

（华南理工大学，广州 510641）

摘 要：通过对常见的最短路径及其算法的分析，指出以往的最短路径算法不能实现公交路线的查询，提出更适合公交查询的最短路径算法以及广州市公交线路查询系统的实现。

关 键 词：最短路径；公交查询

中图分类号：TB 23

文献标识码：A **文 章 编 号**：1003-0158(2001)03-0020-05

1 常见的最短路径问题及算法

1.1 在实际中常见的最短路径问题

例如，如果司机用汽车运输货物从 A 城到 B 城，他就会考虑走路程最短或者时间最少的道路。这是考虑路程或时间的最短路径问题。

又例如，要在 A 地到 B 地之间铺设煤气管道。在 A，B 地间的长方形地区划分格子点，根据地形、土壤、是否经过江河、泥塘、农田、村庄、郊区、城镇、公路、铁路等各种情况，将从一格子点至另一格子点的铺设费用估计出来。求：由 A 地到 B 地铺设煤气管道经过哪些格子点使总的造价最小。这是考虑费用的最短路径问题。

另外在线路安排、设备更新、厂区布局、城市规划、电子导航、交通转车等方面均需要考虑到最短路径问题。

1.2 赋权图的最短路径

设有图 G，对 G 中的每一条边 (V_i, V_j)，相应地有一个数 $L(V_i, V_j)$ 称为边的权。图 G 连同在它边上的权被称为赋权图。一条边的权也说成它的长。一条道路 $u=V_1, V_2, \cdots, V_n$ 的长是 u 上所有长的和，即 $L(V_1, V_2)+L(V_2, V_3)+\cdots+L(V_{n-1}, V_n)$。

在赋权图中给定一个始点 V_i 及终点 V_j。所谓最短路径问题就是在 (V_i, V_j) 道路集合 $\{P_{ij}\}$ 中，寻求长为最小的路径，这样的路径称为从 V_i 到 V_j 的最短路径。从 V_i 到 V_j 的最短路径长度即最短距离记作 $d(V_i, V_j)$。

赋权图中的权可以表示两个顶点间的距离，或者途中所经的时间，或者交通费用等。此时路径长度的度量不是路径上边的数目，而是路径上边的权（距离、时间、费用等）之和。

收稿日期：2001-03-06

作者简介：陈箫枫（1974-），男，广西浦北人，硕士研究生，主要研究领域为计算机图形学。

例如图 1，每个顶点表示城市，两个顶点构成的边表示两城市间的道路，边上的数字也就是上面说的权表示两个城市之间的距离（公里），如果用汽车运输货物从 A 城到 H 城，司机就会考虑走路程最短的道路，那么最短路径是哪一条呢？应该是 $A \rightarrow B \rightarrow D \rightarrow H$。而且最短距离 $d(A, H) = L(A, B) + L(B, D) + L(D, H) = 100 + 100 + 100 = 300$ 公里。

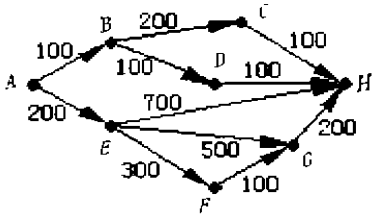

图 1　赋权图的最短路径

1.3　迪杰斯特拉(Dijkstra)最短路径算法

寻找两顶点间的最短路径的算法很多，目前公认最好的算法是迪杰斯特拉（Dijkstra）在 1959 年提出的，它不仅求出从始点到终点的最短路径，而且最后所得到的实际上是始点到各顶点的最短路径。

对 Dijkstra 算法进行补充得出的步骤如下：

第一步　初始化。$V=\{1, 2, \cdots, N\}$，$S=\{F\}$，$D[I]=L[F, I]$，$Y[I]=F$，其中 $I=1, 2, \cdots, N$。

F 表示路径的始点，I 表示某一顶点，N 表示网络中所有顶点的数目，V 是所有顶点的集合，$L[F, I]$表示从 F 点到 I 点的距离，S 是顶点的集合，D 为 N 个元素的数组用来存储顶点 F 到其它顶点的最短距离，Y 为 N 个元素的数组用来存储最短路径中在顶点 I 之前经过的最近顶点。

第二步　从 $V-S$ 集合中找一个顶点 T 使得 $D[T]$ 是最小值，并将 T 加入到 S 集合中。如果 $V-S$ 是空集合则结束运算。

第三步　调整 Y、D 数组中的值：在 $V-S$ 集合中对于顶点 T 的邻接各顶点 I，如果 $D[I] > D[T]+L[I, T]$，那么令 $Y[I]=T$，$D[I]=D[T]+L[I, T]$。

继续执行第二步。

2　公交线路的最短路径算法

2.1　复杂公交线路查询不能采用 Dijkstra 最短路径算法

Dijkstra 最短路径算法由于其稳定性、能适应网络拓扑的变化，同时对系统的内存空间占用少，因而在计算机网络拓扑路径选择以及 GIS 中得到广泛的应用。但是对公交线路来说，Dijkstra 算法所采用的数据结构及其实现方法总体上说是比较复杂的，其缺点也是明显的，难以应付公交线路的网络拓扑中的复杂性。主要表现如下：

（1）数据结构复杂。网络在教学和计算领域被抽象为图，所以其基础是图的存储表示。一般而言，无向图可以用邻接矩阵和十字链表表示。公交线路网络拓扑，很难用现有的数据结构加以完整的表示。如果采用现有的最短路径算法分析，其建立的公交线路网络图的数据结构模型将非常复杂。

（2）算法时间长。以 Dijkstra 算法来计算公交路线最短路径，在大数据量的情况下，计算速度会慢得让人难以忍受。系统设计中要求公交转车的查询必须在较短的时间内完成，Dijkstra 算法难以实现。

（3）Dijkstra 最短路径算法对于网络拓扑图要求简捷，对于复杂的广州公交网络拓扑，必须对其进行复杂的抽象、合并成简捷的网络拓扑图，这无疑增加了程序的复杂性。

（4）公交转车中的特殊性并不一定要求用 Dijkstra 算法算出一条最短路径。求乘客从 A 站到 B 站的最短路径，将每个公交站点均看作网络上的顶点，每相邻站点间的路段看作一条边，假设乘客每到一个公交站点都考虑转车，才可用 Dijkstra 算法计算最短路径。用 Dijkstra 算法计算出来的结果可能是：从 A 站到 B 站需要转好几次车或十几次车才能到达。这样的计算结果是没有什么意义的。

故根据广州交通的实际情况，笔者认为广州市的公交线路查询不能采取 Dijkstra 算法。

2.2 一种公交线路的最短路径算法

一般地，从 A 站乘公交车到 B 站，会先看经过 A 站的车有直接到 B 站的吗？若有，马上得到直达车路线（图 2(a)）。若无，则再看 B 站有什么车经过，经过 A 站的车和经过 B 站的车有交叉点吗？若有则可考虑在交叉点 C 转车（图 2(b)）。若无，则乘坐经过 A 站的车到某一站如 C 站下车，经过 C 站的车与经过 B 站的车有交叉点 D 吗？若有再在交叉点 D 转车，两次转车可到达 B（图 2(c)）。若无，两次转车不成功。

若只有一种转车方法，没有其它选择机会，则好办。若有几种转车方法（图 2(d)），这几种转车方法的路程可能不同，则需要确定那种转车方法的路程是最短的。

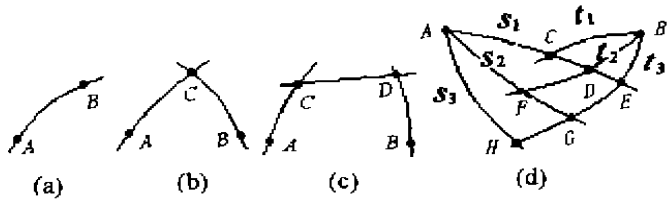

图 2 公交线路的最短路径算法

还是图 2(d)，经过 A 站的三路车 s_1，s_2，s_3 与经过 B 站的三路车 t_1，t_2，t_3 分别有交叉点，s_1 交 t_1 于 C，s_1 交 t_2 于 D，s_1 交 t_3 于 E，s_2 交 t_2 于 F，s_2 交 t_3 于 G，s_3 交 t_3 于 H。这说明从 A 站乘 s_1 路公交车到 C 站后，转乘 t_1 路公交车可以到达 B 站，所经过的路程为 $L(A, C)+L(C, D)$，其它同理。如此看来，从 A 到 B 有 6 种转车方法，设所经过的路程分别为 L_1，L_2，L_3，L_4，L_5，L_6。如果 L_1 均小于 L_2，L_3，L_4，L_5，L_6，那么 L_1 为最短路程，$A \rightarrow C \rightarrow B$ 为最短路径。

根据公交线路的实际，可以认为两次之内的转车是比较合理的，超过两次的转车是无意义的，不予考虑。笔者提出公交转车最短路径算法的步骤，如下所示。

（1）输入乘车始点 A、终点 B。

（2）求经过 A 或其附近的线路 $s(I)$（I=1，2，…，m）（m 为正整数），及经过 B 或其附近的线路 $t(J)$（J=1，2，…，n）（n 为正整数）。

（3）有 $s(I)=t(J)$ 吗？

若有，满足此条件的线路 $s(I)$ 也即 $t(J)$ 为 A 到 B 的直达车线路，输出结果，结束运算。

若没有，往下执行。

（4）求线路 $s(I)$ 的站点 $E(I, U)$（U=1，2，…，p，p 为正整数），及线路 $t(J)$ 的站点 $F(J, V)$（V=1，2，…，q，q 为正整数）。

（5）有 $E(I, U)=F(J, V)$ 吗？

若有，满足此条件的线路 $s(I)$，$t(J)$（可能不止一种）即为一次转车的线路，计算各种一次转车方法的乘车路程，乘车路程最短转车线路，再求转车地点，输出结果，结束运算。

若没有，往下执行。

（6）求经过 $E(I, U)$ 的线路 $r(K)$（K=1，2，…，g）（g 为正整数），求线路 $r(K)$ 的站点 $G(K, W)$（W=1，2，…，h）（h 为正整数）

（7）有 $G(K, W)=F(J, V)$ 吗？

若有，满足此条件的线路 $s(I)$，$t(J)$，$r(K)$（可能不止一种）即为两次转车的线路，计算各种两次转车方法的乘车路程，乘车路程最短的线路就是最佳转车线路，再求转车地点，输出结果，结束运算。

若没有，表明两次转车不成功，不予考虑。结束运算。

3　广州公交线路查询系统

3.1　广州市公交线路的特点及其处理方法

广州是个老城市，其道路体系复杂，旧城区街道狭窄，公交线路的规划就不得不考虑这些特点，从而导致公交网络的复杂性。如下是广州公交线路的特点及其处理方法：

（1）公交车线路繁多，约有好几百条线路。将每一线路、各站名及其站点间的路程均存入数据库中，查询时由程序从数据库内读取线路、站名、路程，并据查询条件按照公交线路的最短路径算法计算得出所需的线路。

（2）公交车的往返路线往往不同。由于市区一部分街道采取单行线路，或交通管制的原因，同一路公交车次其往返的路线并不完全一样。为解决这一问题，对于每一条公交线路，在数据库中都把它当作是两条有方向的线路看待。这样做的结果是增加了数据库的容量，同时增加了电子地图绘制的工作量，但与公交线路查询的准确性相比，这点牺牲是值得的。另外这种解决方法令人可喜的是数据库访问时间并没有太明显的增加。

（3）相同的车站名有的出现在不同的地点。解决这一问题的方法是对于地图上的每一个站点，都赋予它一个“别名”（alias），（通常是给每个站点编号）这样重名现象就不会出现了。

（4）同一线路同一地点的往返车站有的相隔很远，或者具有不同的车站名。这种现象在广州公交规划中特别明显。应该说这是公交规划中不好的一面，但由于历史的原因，这一现象一直没有得到更正。对于这个问题，人们在车站搜寻的过程中采用“面积”搜索方式而不是采用“点”搜索方式来解决。具体地说，就是人们在转车时，往往并不是下车后就在下车的那个车站转车，常常需要到另一个车站（比如说对面车站、下一个车站等）去转车。因而搜寻转车点时给出一个搜寻的范围，以下车点为中心搜索一定范围内所有的可转车站的可转车次，而不是只搜寻下车点的可转车次。

3.2　广州公交线路查询在“广州之窗”多媒体地理信息系统得到应用

按照以上思路，笔者采用 VB 实现了公交查询系统的最佳路径模块，应用在“广州之窗”多媒体地理信息系统中。“广州之窗”多媒体地理信息系统采用先进的 GIS 软件技术，同时结合当今流行的浏览器编程方式，实现 GIS 与多媒体的无缝连接，在大量的历史文字资料和图片的基础上，建立一个关于广州市商贸旅游的社会经济信息系统，从而进一步推动广州信息化建设，起到宣传广州，促进广州市的进一步对外开放的作用。“广州之窗”多媒体地理信息系统已经完成并发行。

3.3 广州公交线路网上查询系统的实现

近年，随着 Internet/Intranet 环境在世界范围内的不断扩大，根据用户请求实现动态数据交换更是成为不可缺少的应用，如网上商品购物系统、数据录入、数据查询等。广州公交线路的查询也可以在网上实现，可采用的方法是 ASP(Active Server Pages)，数据库用 SQL Server 或 Access。

与其它的方式相比，ASP 方式具有明显的优势，因为 ASP 用寥寥数行代码就可以写出很好的数据库程序，而且它不需要编译，语言简洁，与 HTML，VBScript，JavaScript 等能很好地融合在一起。数据查询是 ASP 的强项，数据运算是 VBScript，JavaScript 的强项，可以将它们与 SQL Server 或 Access 数据库完美结合，运用公交线路的最短路径算法实现广州公交线路网上查询系统。

参 考 文 献

[1] 徐孝凯. 数学结构简明教程[M]. 北京. 清华大学出版社, 1995.

[2] 王朝瑞. 图论及其应用[M] 北京· 北京理工大学出版社, 1995

[3] 余 波. 动态 Web 应用高级开发指南[M] 北京 人民邮电出版社, 2000

[4] 汪晓平, 吴勇强, 张宏林, 等 ASP网络开发技术[M]. 北京 人民邮电出版社, 2000.

Shortest Path Algorithm Analysis and Its Application to Bus Route Query

CHEN Xiao-feng CAI Xiu-yun TANG De-qiang

(South China University of Technology, Guangzhou 510641, China)

Abstract: This paper presents the normal shortest path and its algorithm, explains the former shortest path algorithm cannot realize the bus route query, puts forward the algorithm which is fit to the bus route query, and explains how to realize the Guangzhou city bus route query system.

Key words: shortest path; bus route query