文章编号:2095-7386(2020)01-0057-05
DOI:10.3969/j.issn.2095-7386.2020.01.011

用路由矩阵搜索法实现智能公交系统的快速查询

王防修
(武汉轻工大学 数学与计算机学院,湖北 武汉 430023)

摘 要:给出了在网络环境下实现智能公交系统快速查询的方法。首先,设计了用 Dijkstra 算法实现从源站点到目的站点的最优乘车方案的查询。然后,通过对 Dijkstra 算法进行改进,在一定程度上提高了公交系统的查询速度。最后,设计了路由矩阵搜索法,通过它极大地提高公交系统的查询速度。算法分析表明,路由矩阵搜索法的时间复杂度小于 Dijkstra 算法。系统测试表明,在进行同一乘车路线的查询时,路由矩阵搜索法比 Dijkstra 算法及其改进算法所花的时间要少得多。与 Dijkstra 算法及其改进算法相比,路由矩阵搜索法能大大提高智能公交系统的查询速度。

关键词:Dijkstra 算法;改进的 Dijkstra 算法;路由矩阵搜索法
中图分类号:U 495 **文献标识码**:A

Realizing quick query of intelligent public transportation system by routing matrix search method

WANG Fang-xiu
(School of Mathematics and Computer Science, Wuhan Polytechnic University, Wuhan 430023, China)

Abstract: This paper presents a method for realizing quick query of intelligent public transportation system in network environment. Firstly, the Dijkstra algorithm is used to query the optimal ride plan from the source site to the destination site. Then, by improving the Dijkstra algorithm, the query speed of the public transportation system is improved to some extent. Finally, the routing matrix search method is designed, which greatly improves the query speed of the public transportation system. Algorithm analysis shows that the time complexity of the routing matrix search method is smaller than the Dijkstra algorithm. Systematic tests have shown that the routing matrix search method takes much less time than the Dijkstra algorithm and its improved algorithm when querying the same ride route. Compared with the Dijkstra algorithm and its improved algorithm, the routing matrix search method can greatly improve the query speed of the intelligent public transportation system.

Key words: Dijkstra algorithm; improved Dijkstra algorithm; routing matrix search method

1 引言

为方便市民出行,每个城市都有与之配套的公交系统[1-7]。对于三四线城市,由于城市规模小,其公交系统主要由公汽构成,且其线路少,故对智能网络公交查询系统的要求不是很高。但是,对于一二

收稿日期:2019-12-16.
作者简介:王防修(1973-),男,副教授,E-mail:wfx323@126.com.
基金项目:中国食品公益专项研究基金(201513004-3).

线城市而言,由于城市规模大,人口众多,故其公交线路非常复杂。因此,我国一二线城市的公交系统主要由公交和地铁组成。市民在乘公交出行前,如果对到达目的地的路线不是很熟悉,一般需要通过网上智能公交查询系统来查询乘车路线[8-13]。所以,智能公交系统必须能够快速地将查询结果告诉市民。

在计算机硬件一定的情况下,要想进一步提高公交系统查询速度,关键是需要一个好的查询方法。鉴于用户要查询的内容千差万别,不可能将所有可能的结果保存于数据库文件,然后静态查询。也就是说,公交系统查询是一个动态过程,对用户提交的查询任务是当场完成。因此,系统对用户输入的任务必须在尽可能短的时间内做出回应。在本文中,笔者设计了三种不同的方法来实现公交系统的快速查询。通过比较从中选择速度最快的查询方法。

2 快速查询方法的设计

当在网络查询系统中输入了起始站和终点站后,系统必须从线路库中找到满足用户需求的最佳线路组合。

2.1 用 Dijkstra 算法搜索起始站和终点站间的最佳线路

2.1.1 搜索起始站和终点站间的乘车时间最短的公交线路

(1)从始站点出发,搜索到其他所有站点的所化时间最短的公交线路。

(2)由终站点出发,通过使用栈的方法搜索到从始点站到终点站的时间最短公交线路。

2.1.2 搜索起始站和终点站间的换乘次数最少的公交线路

(1)从始站点出发,搜索到其他所有站点的换乘次数最少的公交线路。

(2)由终站点出发,通过使用栈的方法搜索到从始点站到终点站的换乘次数最少公交线路。

2.2 用改进的 Dijkstra 算法搜索起始站和终点站间的最佳线路

2.2.1 搜索起始站和终点站间的乘车时间最短的公交线路

(1)从始站点出发,搜索到其他所有站点的所化时间最短的公交线路,一旦搜索到终站点的时间最短的公交线路,就停止搜索。

(2)由终站点出发,通过使用栈的方法搜索到从始点站到终点站的时间最短公交线路。

2.2.2 搜索起始站和终点站间的换乘次数最少的公交线路

(1)从始站点出发,搜索到其他所有站点的换乘次数最少的公交线路,一旦搜索到终站点的换乘次数最少的公交线路,就停止搜索。

(2)由终站点出发,通过使用栈的方法搜索到从始点站到终点站的换乘次数最少的公交线路。

2.3 用路由矩阵搜索算法搜索起始站和终点站间的最佳线路

2.3.1 搜索起始站和终点站间的乘车时间最短的公交线路

(1)先用递归搜索路由矩阵的方式得到从始站点到终站点最省时的公交线路,将搜索的站点信息保存在队列中。

(2)通过队列出队的方式输出从始站点到终站点的公交路线。

2.3.2 搜索起始站和终点站间的换乘次数最少的公交线路

(1)先用递归搜索路由矩阵的方式得到从始站点到终站点换乘次数最少的公交线路,将搜索的站点信息保存在队列中。

(2)通过队列出队的方式输出从始站点到终站点的公交路线。

3 算法实现

3.1 用 Dijkstra 算法搜索起始站和终点站间的最佳线路的算法实现

用 Dijkstra 算法求出从始站点到终站点最省时的公交路线过程如下:

(1)$d_j = C_{begin,j}, i = 1,2,\cdots,begin-1,begin+1,\cdots,n$。

(2)如果 $d_j \neq \infty$,则始站点 $begin$ 是站点 i 的前驱,即 $pre_i = begin$;否则,站点没有前驱,即 $pre_i = 0$。其中,$i = 1,2,\cdots,begin-1,begin+1,\cdots,n$。

(3)令 $s_i = 0, i = 1,2,\cdots,n$。表示目前所有的站点都没有被选择。

(4)令 $d_p = 0$ 和 $s_p = 1$。表示站点 v_i 被访问。

(5)令 $i=1$。

(6)令 $tp=\infty$。

(7)如果 $\exists q\in\{1,2,\cdots,n\}$ 和 $s_q=0$ 并且满足 $s_q=min\{d_j\}$,那么令 $s_q=1$。

(8)如果 $j\in\{1,2,\cdots,n\}$ 且 $s_j=0, c_{qj}<\infty, s_q=0, d_p=0, d_q+c_{qj}<d_j$,则 $d_j=c_{qj}+d_q$ 和 $pre_j=0$。

(9)执行 $i=i+1$。

(10)如果 $i<n$,则返回步骤(6)重复计算。

(11)通过从终站点 *end* 出发,通过搜索前驱,将整个线路中的站点保存在栈中。

(12)通过访问栈的方式,得到满足要求的从始站点 *begin* 到终站点的最优线路。

从以上的步骤可以看出,该方法可以求出始站点到其它任意站点的最优线路。因此,该方法可以求出 $n-1$ 条始点站为 *begin* 而终点站各不相同的最佳线路,而此处需要得到的只是其中的一条终点站为的线路而已。

3.2　用改进的 Dijkstra 算法搜索起始站和终点站间的最佳线路的算法实现

从算法 3.1 中可以看出,在搜索从 begin 到 end 的最优线路过程中,还花了时间求 begin 到其它站点的最优路线。一方面,为了求 begin 到 end 的最优线路,必须先求出到其它部分站点的最优线路。另一方面,一般也并不需要求出 begin 到其它所有站点的最优线路。因此,算法改进如下:

(1)$d_j=C_{begin,j}, i=1,2,\cdots,begin-1,begin+1,\cdots,n$。

(2)如果 $d_j\neq\infty$,则始站点 *begin* 是站点 i 的前驱,即 $pre_i=begin$;否则,站点没有前驱,即 $pre_i=0$。其中,$i=1,2,\cdots,begin-1,begin+1,\cdots,n$。

(3)令 $s_i=0, i=1,2,\cdots,n$。表示目前所有的站点都没有被选择。

(4)令 $d_p=0$ 和 $s_p=1$。表示站点 v_i 被访问。

(5)令 $i=1$。

(6)令 $tp=\infty$。

(7)如果 $\exists q\in\{1,2,\cdots,n\}$ 和 $s_q=0$ 并且满足 $s_q=min\{d_j\}$,那么令 $s_q=1$。

(8)如果 $j\in\{1,2,\cdots,n\}$ 且 $s_j=0, c_{qj}<\infty, s_q=0, d_p=0, d_q+c_{qj}<d_j$,则 $d_j=c_{qj}+d_q$ 和 $pre_j=0$。

(9)如果 $q=end$,则跳到步骤(12)。

(10)执行 $i=i+1$。

(11)如果 $i<n$,则返回步骤(6)重复计算。

(12)通过从终站点 *end* 出发,通过搜索前驱,将整个线路中的站点保存在栈中。

(13)通过访问栈的方式,得到满足要求的从始站点 *begin* 到终站点 *end* 的最优线路。

从上述改进算法可以看出,步骤(9)可以减少算法的计算量。一旦得到终点站为的最优解,立即停止搜索。原因很简单,接下来的搜索与本次的最优解搜索没有任何关系。因此,改进方法可以一点程度上减少最优解的搜索时间,从而提高了查询速度。

3.3　用路由矩阵搜索算法搜索起始站和终点站间的最佳线路

设路由矩阵搜索法中的递归函数为 $find(r,i,j)$,则搜索过程如下:

步骤 1　如果 $r_{ij}\neq0$,则函数 $find(r,i,j)$ 的递归过程如下:

(1)$k=r_{ij}$。

(2)递归执行 $find(r,i,k)$。

(3)将站点 k 保存在队列中。

(4)递归执行 $find(r,k,j)$。

步骤 2　队列中的站点,即可得到最优线路。

从上述方法可以看出,其比较次数都在 10 次以下,从而大大提高查询速度。如果寻找换乘次数最优,其比较次数一般不会超过 4 次。如果寻找乘车时间最优,一般不会超过 8 次。

4　算法测试

例　公交线路见 2007 年数学建模 B 题,已知该公交系统中共有 3 957 个站点。用上面介绍的 3 种方法进行查询速度的测试。测试的笔记本电脑运行环境:①软件环境:WindowV7 旗舰版(版本);32 位操作系统(系统类型)。②硬件配置:Intel(R) Core(TM) i3, M380 @ 253GHZ(CPU);2.00G(内存)。

(1)用 3 种方法测试乘车时间最短线路的查询速度,其查询时间如表 1 所示。

表 1 不同方法搜索相同乘车时间最短线路所花时间比较

起点	终点	乘车时间最短线路	总站数	Dijkstra 算法搜索时间/μs	改进 Dijkstra 算法搜索时间/μs	路由矩阵算法搜索时间/μs
V3359	V1828	V3359[L015|分段计价|下行|1 站] V2903[L201|分段计价|上行|2 站] V1327[L328|单一票价 1 元|上行|5 站]V525[L103|分段计价|上行|2 站] V73[L480|单一票价 1 元|下行|1 站] V2704[L027|分段计价|循环|1 站] V1784[L167|分段计价|下行|1 站]:	13	217 077.2	81 392.5	3.646 855
V3652	V489	V3652[L149|单一票价 1 元|下行|1 站][L223|分段计价|下行|5 站]V1415 [L323|分段计价|上行|4 站]V2387 [L287|分段计价|下行|4 站]V2295 [L015|分段计价|下行|5 站]V477 [L131|分段计价|下行|1 站]V490 [L074|单一票价 1 元|往返|1 站]V489	21	211 242.5	25 722.7	4.052 108
v1775	v2451	V1775[L117|单一票价 1 元|下行|1 站]V400[L447|分段计价|上行|3 站] V143[L058|分段计价|下行|1 站] V2683[L347|分段计价|上行|2 站] V2962[L385|单一票价 1 元|上行|1 站]V2323[L175|分段计价|上行|2 站] V751[L118|分段计价|下行|1 站]V752 [L286|分段计价|下行|4 站]V2451	18	219 590.2	161 788.1	6.078 163

(2)用 3 种方法测试换乘次数最少线路的查询速度,其查询时间如表 2 所示。

从表 1 和表 2 中的算法测试结果可以看出,在最优公交线路查询时,改进的 Dijkstra 算法比 Dijkstra算法在查询速度上有一定程度的提高。然而,路由矩阵搜索法与 Dijkstra 算法及其改进算法相比,所花的时间简直可以忽略不计。

表 2 不同方法搜索相同换乘次数最少线路所花时间比较

起点	终点	换乘次数最少线路	总站数	Dijkstra 算法搜索时间/μs	改进 Dijkstra 算法搜索时间/μs	路由矩阵算法搜索时间/μs
V3359	V1828	v3359[L436|分段计价|下行|31 站] v1784[L167|分段计价|下行|1 站] v1828	32	219 029.2	141 924.2	3.646 855
V3652	V489	v3652[L149|单一票价 1 元|上行|5 站] v154[L185|单一票价 1 元|下行|19 站] v3264[L074|单一票价 1 元|往返|4 站] v489	28	214 362.2	46 225.2	1.620 843
v1775	v2451	v1775[L474|分段计价|上行|11 站] v605[L286|分段计价|上行|24 站] v2451	35	227 407.1	107 462.3	2.431 265

5　算法分析

从算法的设计可以看出,Dijkstra 算法及其改进算法的时间复杂度均为 $O(n^2)$。由于改进的 Dijkstra 算法只要访问了目的站点,则搜索过程立即停止。因此,在实际查询过程中,改进的 Dijkstra 算法比 Dijkstra 算法所花费的搜索时间要少。路由矩阵搜索算法的时间复杂度为 $O(1)$。在一个实际的智能公交查询系统中,要搜索乘车时间的最优解,一般比较的次数不会超过 10 个。如果要搜索换乘次数的最优解,则其比较次数不会超过 4 个。因此,路由矩阵搜索算法能够大大提高公交系统的查询速度。用路由矩阵搜索法实现公交系统快速查询的过程如图 1 所示。

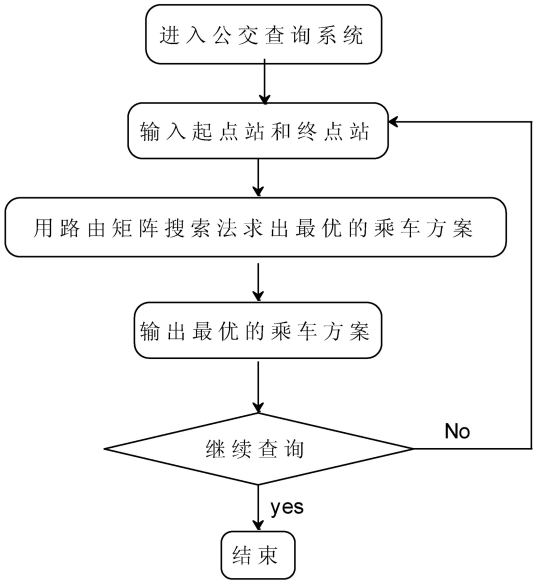

图 1　公交系统快速查询原理

6　结束语

为提高智能公交系统的查询速度,在本文中,笔者设计了一种递归搜索路由矩阵的方法:(1)该方法在软件代码方面对搜素过程进行了优化;(2)该方法的时间复杂度是 $O(1)$;(3)无论是搜索时间最短还是换乘次数最少的乘车方案,该方法所花费的时间都非常短。通过与 Dijkstra 算法及其改进算法相比较,此处设计的路由矩阵搜索法极大地提高了智能公交系统的查询速度。算例测试表明,路由矩阵搜索法能快速求出公交系统中任意两个站点间的时间最短或换乘次数最少的乘车方案。因此,本文中所设计的路由矩阵搜索法为智能公交系统提供了一种快速查询的方法。

参考文献:

[1] 曹柳芳,柏海舰. 弱客流地区公交线路选择与运营模式分析[J]. 交通科技与经济,2019,2101:33-37.

[2] 回贺. 城市智能公交查询系统平台的构建[J]. 自动化技术与应用,2018,(12):188-190.

[3] 李玉贞,丁贤勇. RFID 在智能公交系统中的应用研究[J]. 交通世界,2019(11):16-18.

[4] 蔡苗苗,费志鹏,张泽华,芮立. 基于大数据的城市智能公交管理系统方案设计[J]. 无线互联科技,2018(7):116-117 +120.

[5] 黄情,梁火层,韦秋雨,等. 新型智能公交车系统的设计[J]. 电子技术,2018,(11):71-73.

[6] 颜洲. 智能公交系统自组网路由算法研究与设计[J]. 新疆师范大学学报(自然科学版),2016,(4):49-56.

[7] 官俊涛,陈程俊,秦梦莹,等. 基于 ARM 的智能公交系统设计[J]. 应用科技,2014,41(1):78-83.

[8] 韩冬成,赵欣,李旸. 基于互联网 + 的新型不定线公交策略研究[J]. 武汉理工大学学报(交通科学与工程版),2019,(4):723-729.

[9] 陈君,杨东援,赵清梅. 公交线路选择的相关性规律和换乘总量计算模型[J]. 西安建筑科技大学学报(自然科学版),2017,49(6):835-840.

[10] 张俊丽. 公交线路选择的优化模型[J]. 价值工程,2015(28):206-207.

[11] 郭洪洋,张玺,刘澜,等. 信息影响下区分潜在类别的公交线路选择模型[J]. 计算机应用研究,2014(10):2937-2940.

[12] 钱萌,彭张节,程树林,等. 基于综合评价指数的城市公交线路选择优化模型[J]. 吉林大学学报(信息科学版),2008(2):180-185.

[13] 周文峰,李珍萍,刘洪伟,等. 最优公交线路选择问题的数学模型及算法[J]. 运筹与管理,2008(5):80-84.