

文章编号 1000-5013(2001)02-139-04

交通问路系统中最短路径的新算法

傅冬绵

(华侨大学经济管理学院, 泉州 362011)

摘要 研究交通线路中的最短路径算法,应用图论中的广度优先搜索思想,给出优先队列的进出队方式,提供一种有效的最短路径的新算法.该算法简洁、运算速度快、效果好,并在实际应用中取得了证明.

关键词 最短路径, 优先队列, 算法

中图分类号 U 491.2 : O 157.5

文献标识码 A

随着计算机的普及和发展,人工智能(Artificial Intelligent 简称 AI)显示出巨大的实用价值.许多 AI 的应用基础是问题求解(Problem solving)的方法,如交通路线中的最佳路径问题,若通过数学公式来求解,既复杂有时又无法用有限形式表示,因而通常采用搜索求解的方法.在寻找最佳路径问题时,面对搜索空间中的庞大结点数和复杂路径,需有一个优秀的解.根据图论知识,搜索的速度取决于搜索过程中所经历的路径长度和遍历过的结点个数.若采用穷尽搜索(Exhaustive Search)方法,虽然求出来的解是最优的,但当搜索空间每增加一个结点,搜索路径增加的数目远大于一倍.当结点增加到一定数目时,其解又相当复杂难于实现.实践证明,采用类似于 AI 的探索方法是一种行之有效的方法.对此,国内外有着一定的搜索技术性能评价.这些性能评价针对不同的搜索情况有不同的要求,并非只着重最优解,而是重视在一定条件下的非劣解.其两个重要的评价指标:(1)解的算法实现,要求快且容易;(2)解的质量,能对给定的问题找出令人满意的解.根据这两点,我们对交通路线中的最佳路径问题,给出了一种搜索最短路径的新算法.与 Dijkstra^[1]算法相比较,其解的质量是令人满意的.该算法成功地在计算机上得以实现,并在泉州公交路线管理系统的问路子系统中得到了充分证明.

1 求单源结点的最短路径算法

1.1 问题的提出

在交通路线中寻找两结点间最短路径,可归结成求单源结点的最短路径问题^[1].它可用模型表示:给定一个赋权无向图 $G, G = (V, A)$, 其中 V 为顶点集, A 为路径集.对每一条弧 α

$= (V_i, V_j)$, 相应地有权 $W(\alpha) = W_{ij}$, 又给 G 中的两个顶点 V_s, V_t . 设 P 是 G 中从 V_s 到 V_t 的一条路, 定义路 P 的权是 P 中所有弧的权之和, 记为 $W(P)$. 最短路径问题就是要在所有从 V_s 至 V_t 的路中, 求一条权最小的路, 即求一条从 V_s 到 V_t 的路 P_0 , 使

$$W(P_0) = \min_P \{W(P)\}, P \in G.$$

我们把满足上式的 P_0 叫做从 V_s 到 V_t 的一条最短路径.

1.2 算法的基本思想

对上述问题的求解, Dijkstra^[1] 曾提出一个按路径长度递增的次序产生最短路径的经典算法, 该算法是目前多数系统解决最短路径问题采用的理论基础^[1]. 但是, 在下面两个问题上有它的不足. (1) 要同时算出源点到其它所有顶点的最短路径. (2) 求出的结果只是最短路径顶点的集合, 要知道最短路径还需根据最短路径进行排序. 为克服上述缺陷, 我们直接从求出任意两个顶点之间的最短路径序列出发, 提出一个新的算法. 我们的基本思想, 是根据图的广度优先搜索, 采用的队列不是一般的先进先出队列, 而是优先队列, 即按路径长度进行排序入队. 路径短的顶点先出队列, 且当顶点出队时才作出访问的标记. 该算法明显地避免走回头路, 可以在同层范围内找出更短的路径来, 从而尽量地避免了从死结点回溯, 大大地提高了运算速度. 从编出的程序来看, 它比经典的算法更加简单、有效. 其算法, 可用图 1 所示的结构流程图表示.

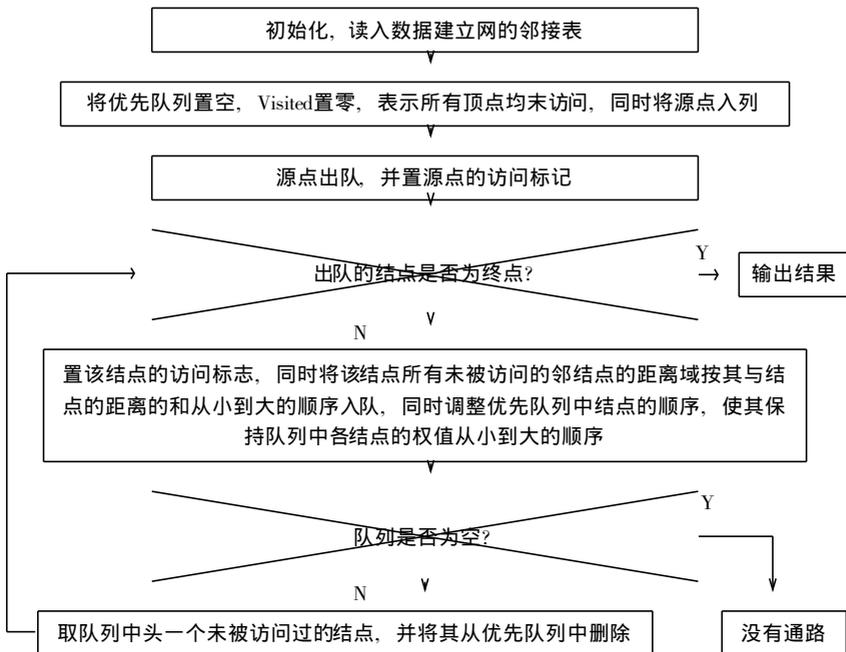


图1 算法的结构流程图

1.3 数据的存贮结构

我们用邻接表来表示给定的网络图. 在该邻接表中对图中的每一个顶点建立一个单链表, 第 i 个单链表中的结点单元表示依附于顶点 V_i 的边. 每个结点由 3 个域组成, 其中邻接点域指示与顶点 V_i 邻接的点在图中的位置, 链域指示下一条边的结点, 数据域存贮和边有关

的信息(如权值). 每个链表附设一个表头结点. 在表头结点中,除了设有链域指向链表的第一个结点外,还设有存贮顶点 V_i 的名称、编号数据域. 这些表头结点采用顺序结构的形式存贮,以便能随机访问任一顶点的链表. 同时设立优先队列,即用环形链表存贮,以方便按结点的

结点	权值	前趋结点	后继结点
----	----	------	------

图 2 环形链表结点的存贮结构

的数据域大小进行插入、删除等功能. 该结点的存贮结构,如图 2 所示. 为了提高运算速度,引入一辅助数组 $Visited[1, N]$: (1) 用以判断顶点 i 是否访问过; (2) 指出各条路径的走向. 一旦找到最短路径,便可以从其终端反过来不断求出其直接的前趋结点,直到起点为止. 其中, $Visited[i] = 0$, 顶点 i 未访问过; 或者 $Visited[i] = j$, 顶点 i 已访问过,且 j 为路径序列中 i 的直接前趋结点编号.

2 算法应用的举例说明

我们以 6 个结点为例,作为解释该算法的可行性(图 3). 如图 3(a) 所示,现有 6 个顶点的

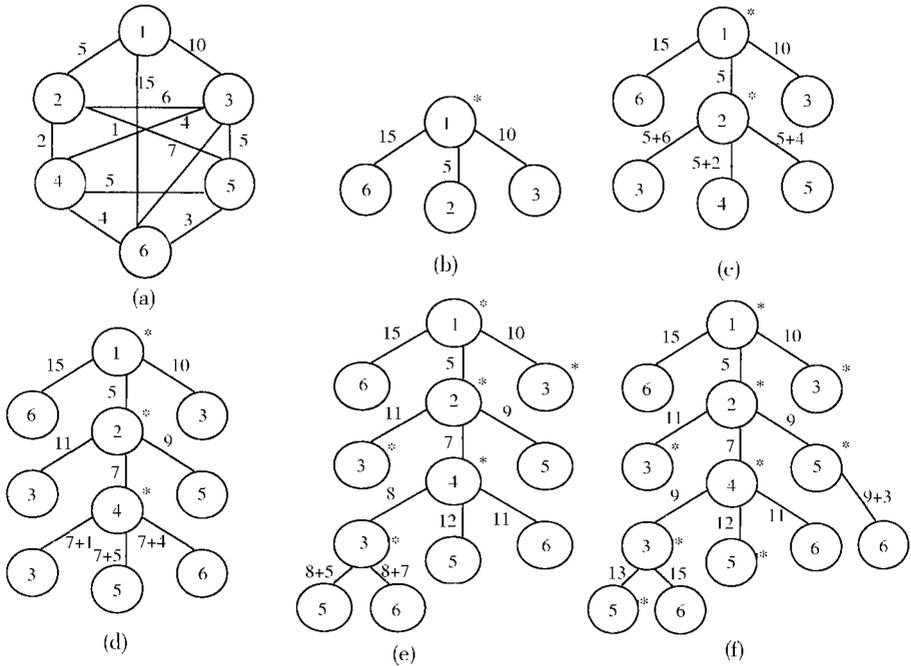


图 3 应用实例

交通图以及它们的权值(两个顶点之间的距离),设要求出顶点 1 到顶点 6 之间的最短路径序列. 按照该算法求解的具体步骤,有如下 5 个方面. (1) 如图 3(b) 所示,顶点 1 先进入优先队列,且置顶点 1 的距离域为 0. 由于顶点 1 的距离域为 0 最小,所以它先出队. 判断它不是终点 6,就置顶点 1 的访问标志(*). 同时,将顶点 1 的所有邻结点 2, 3, 6 的距离域 5, 10, 15 和顶点 1 的距离域 0 相加. 然后,按从小到大的顺序,即 2(5) 3(10) 6(15) 的顺序进入优先队列. 括号内是结点的距离域. (2) 如图 3(c) 所示,现选择优先队列中距离域最小的结点即顶点 2 出发. 当顶点 2 出队后,由于它不是终点 6,将其置过访问标志. 然后,将其所有的邻结点 3, 4, 5 入队(顶点 1 已置完访问标志,不能再访问,所以不应将它入队). 同时,调整优先队列中的顺

序. 经调整后, 优先队列为 4(7) 5(9) 3(10) 6(15). 括号内是它们到顶点 1 的权值和. (3) 如图 3(d) 所示, 现从顶点 4 出发, 由于它不是终点 6, 将其置过访问标志. 然后, 将其所有的邻结点 3, 5, 6 入队. 这时, 优先队列变成为 3(8) 5(9) 6(11). (4) 如图 3(e) 所示, 现从顶点 3 出发, 由于它不是终点 6, 将其置过访问标志. 然后, 将其所有的邻结点 5, 6 入队. 这时, 优先队列变成为 5(9) 6(11). (5) 如图 3(f), 这时从搜索树上看, 结点 5 出队后, 由于除了结点 6 以外的所有结点都已访问过, 所以没有结点进队. 这样, 当结点 6 出队后, 判断它是终点, 结束搜索. 最终得到最短的搜索路径为 1 2 4 6, 且路径的全部距离长度为 11.

3 算法分析

按 Dijkstra 算法, 要具体求出每一顶点到源点的最短距离. 同时, 在每次求出一顶点的最短路径后, 又要修改其它各点到源点的当前最短距离. 所以, 时间复杂度为 $O(n^2)^{[4]}$. 本算法在一般情况下, 优先队列的长度与顶点数 n 无关, 每一个顶点的邻接点的平均数也是常数 C . 因此, 本算法时间复杂度为 $O(n)$. 这样, 该算法成功地把搜索单源结点的最短路径时间复杂度从 $O(n^2)$, 缩小到 $O(n)$ 量级. 在实例操作上, 我们应用到泉州公交线路管理系统. 在拥有近 300 个站点的泉州公交无向图中, 本算法实施在寻求任意给出的两站点的最短路径时, 体现出优异快捷的运算速度, 收到了良好的效果. 实例证明它是可行的.

最短路径问题是交通线路中普遍存在的问题, 也可引申涉及到其他的度量, 如时间、费用、线路容量等问题. 在 DSS(Decision Support System) 中也有着广泛的应用. 所以对该问题的研究, 有理论和实际上的重要意义. 借助于计算机科学理论的发展与提高, 本文提出的新算法该将是一种有效的途径.

参 考 文 献

- 1 严蔚敏, 吴伟民. 数据结构[M]. 第 2 版. 北京: 清华大学出版社, 1992. 188 ~ 193
- 2 甘应爱, 田 丰. 运筹学[M]. 北京: 清华大学出版社, 1996. 264 ~ 271
- 3 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
- 4 严寒冰, 刘迎春. 基于 GIS 的城市道路网最短路径算法探讨[J]. 计算机学报, 2000, 23(2): 210 ~ 215

A New Algorithm of Shortest Path in Traffic Inquiring System

Fu Dongmian

(College of Econ. Manag., Huaqiao Univ., 362011, Quanzhou)

Abstract A study is made on the algorithm of shortest path in traffic line. By applying the idea of scope first search in graph theory, the author gives the way of get in and out the queue in priority queue and offers an effective new algorithm of shortest path. This concise algorithm is confirmed by practical application to be fast in operation and good in effect.

Keywords shortest path, priority queue, algorithm