

提高大型交通网络最短路搜索效率研究

高虹霓, 杨建军, 曹泽阳
(空军工程大学 导弹学院, 陕西 三原 713800)

摘要:针对大型交通网络节点多、道路等级差别大等特点,提出了综合提高大型交通网络搜索效率,避免发生“维数灾难”问题的有效途径。

关键词:交通网络;最短路;搜索效率

中图分类号: O22 **文献标识码:** A **文章编号:** 1009-3516(2003)01-0054-03

在物资供应道路选择^[1]辅助决策支持系统(DSS)中,如果保障地域为一个地区或数个地区,其道路交通网节点数少则数千多则上万。实践证明,对于如此大型运输交通网络如果仅仅通过一般的图搜索算法^[2]进行最短路搜索,容易发生“维数灾难”问题,即由于网络节点过多,导致程序搜索时间过长甚至死机,无法满足实际应用的需要。此外,由于搜索效率低,占用空间大,程序对运行的硬件环境也会提出更高的要求。所以,如何提高搜索效率、减少搜索时间是解决大型交通网络搜索问题的关键。

1 预先节点处理,缩小搜索空间

1.1 节点合并法

在交通网络中,常常会遇到如图1所示的情况:有些网络节点只与两个边相联(节点度数为2),当搜索到节点的其中一个边后只能从另一个边离开。如图1中的 V_5 、 V_6 和 V_7 节点等。铁路网络节点就具有这个特点,绝大部分车站都是只和两个边相连(即没有叉路),连接三个边及更多边的车站只占百分之几。如果采用节点合并法就能极大地提高搜索速度。

节点合并法是指忽略中间点(节点度数为2),形成只有几个顶点(节点度数大于2)的简单网络,如图1可以简化为图2。这样,原始节点数据库就分解为简化后的节点数据库和被合并的节点数据库两个数据库,在搜索过程中首先判断起点、终点等是否存在于简化后的节点数据库中,若存在则直接在简化的节点网络中进行最短路求解;若不存在,则在被合并的节点数据库中找到该节点数据并临时恢复到简化的节点数据库中,然后再进行最短路搜索求解。

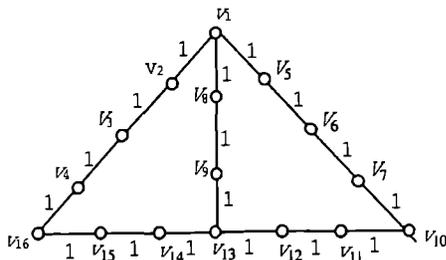


图1 节点合并前示意图

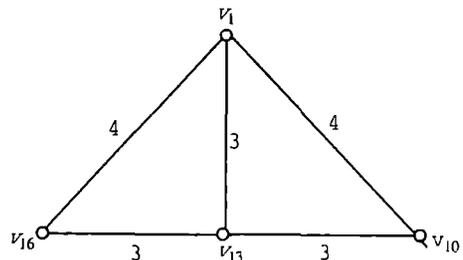


图2 节点合并后示意图

收稿日期:2001-12-13

作者简介:高虹霓(1966-),男,陕西宝鸡人,讲师,硕士,主要从事管理科学与工程研究;
杨建军(1955-),男,河北阜平人,教授,主要从事空军武器装备综合保障研究。

在对网络作了如上处理后,即可在简化的网络上计算从起点至终点的最短路径。这种简化适用于所有的交通网络,尤其是铁路网络,效果更为显著。经验表明,对公路交通网合并后网络节点数据量可减少 5%~8%,铁路交通网减少量更高。

1.2 节点裁剪法

大型道路交通网络不仅节点多,而且道路等级差别大,一般情况下低等级道路较多,高等级公路相对较少。对于运输任务而言,交通图上标绘的许多道路不能满足运输要求,实际可供选择的道路并不是很多。因此,对整个交通网络在一定规则下进行裁剪,可达到简化网络数据,提高搜索效率的目的。

可见,实际的交通拓扑网络和理论上的拓扑图有很大的区别,实际的网络有它的一些具体特征,对这些特征进行分析后,提出具体的裁剪方法,将简化整个网络的数据量。对于一个具体的运输路线,在实施节点裁剪时可考虑如下裁剪条件:提高道路通行的等级,使可供选择的路段数急剧减少;增加桥梁载重的限制;增加道路坡度、最大转弯半径的限制。

由于实际的交通网络有很强的特殊性,其分叉道路等级一般较低,这些分叉部分就可以裁剪,所以图 3 可以简化为图 4,从而达到提高搜索效率的目的。实际证明,在某一地区交通网络图上采用裁剪法后,网络节点数可减少 7% 以上。

裁剪法简化步骤如下:

- 1) 读取网络数据,找出度数为 1 的节点;
- 2) 删除该节点及相应的路段里程数据并更新其父节点度数,得到被裁剪后的简化的节点数据库;
- 3) 将被删除的节点及原始数据存入另一节点数据库——被裁剪掉的节点数据库。

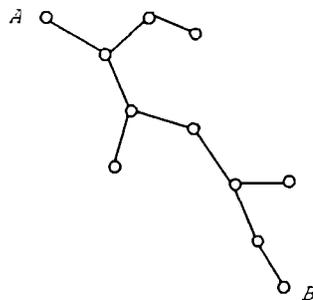


图 3 节点裁剪前示意图

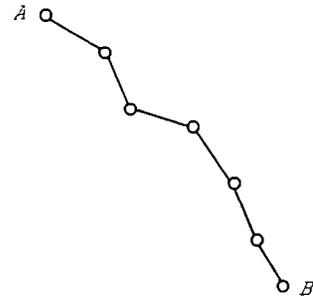


图 4 节点裁剪后示意图

于是,经过裁剪法简化的原始数据库可分解为被裁剪后的简化节点数据库和被裁剪掉的节点数据库。为避免搜索的起点和终点是被裁剪掉的节点,因此在每次搜索前应首先判断起点、终点等是否存在于简化节点数据库中,若存在则直接在简化节点网络中进行最短路求解;若不存在,则在被裁剪的节点数据库中找到该点数据并临时恢复到简化节点数据库中,然后再搜索最短路。

2 运用多种搜索技术,提高搜索效率

2.1 Dijkstra 标号搜索算法

Dijkstra 标号搜索算法是目前公认的求解最短路问题的高效经典算法,其显著特点是从起点到终点搜索的节点数少,搜索空间小,搜索效率高。基本思想是从起点出发,找出权值最小的节点标号并扩展;然后在所有当前已扩展节点中找出没有标号的权值最小的节点继续标号、扩展,直到目标节点也标上号为止。整个搜索过程最多经过网络总顶点数减 1 步,就可以求出从起点到终点的最短路。Dijkstra 标号搜索算法时间的复杂度为 $O(n^2)$, n 为网络节点个数。

2.2 启发式搜索技术

单纯采用标号搜索算法进行搜索,由于搜索存在盲目性,对于大型网络搜索效率仍显较低,严重时可能发生“维数灾难”。因此,需要在标号法基础上进一步借助启发信息进行启发式搜索^[3],即对待扩展节点按照估价函数进行排序,以确定扩展顺序。估价函数的一般形式是:

$$f(n) = g(n) + h(n)$$

其中, $g(n)$ 为从初始节点 S 到节点 n 的实际代价。 $h(n)$ 是从节点 n 到目标节点的估计代价,它代表搜索过程的启发信息,也就是启发函数。设 S 表示起始节点, G 表示目标节点, n 表示任一节点, $d(n, G)$ 表示节点 n 到目标节点 G 之间的直线长度。若启发函数 $h(n)$ 为节点 n 到目标节点 G 之间的直线长度。则估价函数为

$$f(n) = g(n) + h(n) = g(n) + d(n, G)$$

其中 $g(n)$ 表示从 S 到 n 已知路径上的长度(代价), 可以通过追踪指针得到。设 $h^*(n)$ 表示节点 n 到达目标集合的最小代价路径的代价值, 由于 $h(n) < h^*(n)$, 所以在该算法中, 以估价函数值作为代价进行搜索求解, 采用实际的后续最小代价路径的代价值的下界作为启发函数, 导引着搜索向最可能的方向进行, 从而提高了搜索效率。启发式搜索算法的时间复杂度为 $O(bd)$, b 为网络节点的平均度数, d 为从起始节点到目标节点的最短路搜索深度。

以某一地区的交通网络(网络节点数 3 124 个)为例, 采用 Dijkstra 标号算法在有启发函数和无启发函数两种条件下进行最短路搜索求解, 搜索结果见表 1。可以看出, 借助启发函数, 扩展节点的次数和参加排序的节点数两项指标均小于无启发函数, 从而提高了大型网络的搜索效率。

表 1 标号法在有、无启发函数时最短路搜索效率比较

算法	搜索起点	搜索终点	里程/km	扩展节点次数	参加排序节点数
无启发函数	甲地	A 地	1 078	2 278	5 143
	乙地	B 地	783	1 876	3 884
有启发函数	甲地	A 地	1 078	1 996	4 212
	乙地	B 地	873	1 748	3 436

2.3 双向式搜索技术

双向式搜索, 是指除了象普通单向搜索那样从起点出发向终点搜索最优路线(这一过程称为前向搜索), 同时还由终点出发向起点进行反向搜索, 直至这两条路径在中途某处相交接为止^[5]。前向搜索和反向搜索均可采用“标号法+启发函数”算法从正反两个方向交替搜索, 正向搜索以 $d(n, G)$ 作为启发信息, 反向搜索以节点 n 到起始节点 S 之间的直线长度 $d(n, S)$ 作为启发信息, 一旦搜索中的正反两条最短路经上有相同的节点号时就停止搜索, 输出最短路径。

若 b 为网络节点的平均度数, d 为从起点到终点的最短路搜索深度, 则双向搜索算法的时间复杂度为 $O(bd/2)$ 。理论和实践表明, 双向搜索的搜索空间相对单向搜索最大可减少约 50%, 运行时间比单向搜索算法缩短, 进一步提高了大型交通网络的搜索效率。

综上所述, 综合提高大型交通网络搜索效率的算法可表述如图 5 所示下:

4 结束语

通过采用节点合并法和裁剪法可减少大型交通网络的搜索空间, 以标号法为基础, 综合运用启发函数和双向搜索可减小搜索的盲目性, 加快搜索速度, 提高大型交通网络的搜索效率, 此外,

建立合理的网络数据结构对提高搜索效率也具有一定的影响。通过在某物资供应道路选择辅助决策支持系统(DSS)课题中的实践证明, 上述算法对大型交通网络搜索效率的提高具有明显的效果。

参考文献:

- [1] 高虹霓. 基于模糊 AHP 的道路选优评价方法研究[J]. 空军工程大学学报(自然科学版)[J]. 2001, 2(2): 82-84.
- [2] 蔡自兴, 徐光佑. 人工智能及其应用[M]. 北京: 清华大学出版社, 2000.
- [3] 胡桐清. 人工智能军事应用教程[M]. 北京: 军事科学出版社, 1999.
- [4] 张可. 车辆自动导航的路线优化系统研究[J]. 系统工程, 2001, 19(2): 48-53.

(编辑: 田新华)

(下转第 70 页)

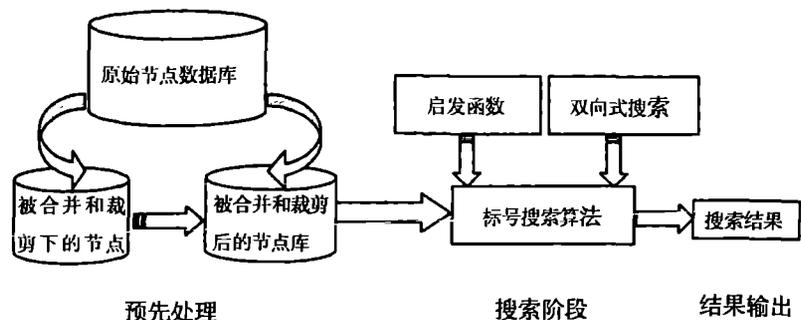


图 5 大型交通网络搜索算法流程图