城市智能交通系统车辆路径规划算法优化研究

◆ 金 欢

摘要：本文提出基于双向搜索的改进算法，适用于城市智能交通。目前常用的最短路径算法为Dijkstra算法，时间复杂度为但由于城市交通网中结点众多，该算法解决速度慢，而且时间复杂度高。针对该问题，本文从算法设计入手，提出改进的双向搜索算法。实验证明，优化后的算法较好地改善搜索效率，符合城市智能交通的发展需求。

关键词：智能交通；路径规划；双向搜索

一、算法的优化原理

1.1 所谓双向搜索，指的是两个方向搜索同时进行。正向搜索：搜索是通过初始结点向目标结点进行；逆向搜索：搜索是通过目标结点向初始结点进行；当出现新结点时，要求对本队列之中的所有结点进行判定是否重复，同时还要求与对方队列中的所有时间判定是否重复，若存有相同的结点 ，那么就发生了相遇事件，此时搜索结束，两方向搜索步数之和为搜索频数，这样生成的搜索树为菱形，搜索结点的数据得到有效的减少，提高搜索效率。通过实验得出，与单向搜索比较，双向搜索结点数可以减少 1/2 以上。

1.2 算法优化。搜索时，初始结点和目标结点与搜索到的每一个节点连线投影，初始结点与正向搜索的新结点投影与目标结点与逆向搜索的新结点投影，计算两距离和，若距离和比初始结点到目标结点的距离长，则判定双向搜索无重合，即可停止某一方向的搜索，而另一方向继续。选择从哪个方向开展搜索，主要因素为哪个方向结点个数较少则向该方向扩展[1]。众所周知，“两边之和大于第三边”，所以如寻找结点在同一边上，则此边就是所需要找的最短路径，否则，连线夹角最小的边是最短路径的可能性最大。

二、路径规划算法设计

2.1 算法实现。设置两个队列 c:array[0..1,1..maxn] of jid，这两队列分别为正向搜索和逆向搜索的扩展队列；设置两个头指针 h:array[0..1] of integer，这两个头指针分别为正向搜索和逆向搜索的头指针；设置两个尾指针 t:array[0..1] of integer ，这两个指针分别为正向搜索和逆向搜索的尾指针。

主程序：选择节点数较少且队列未空、未满的方向先扩展

if（t[0]<=t[1]）and not（（h[0]>=t [0]）or（t [0]>=maxn））then expand（0）;

if（t[1]<=t [0]）and not（（h [1]>=t [1]）or（t [1]>=maxn））then expand（1）;

当某一个方向搜索终止时，另一方向搜索继续，直至两个方向都终止

if not（（h[0]>=t[0]）or（t[0]>=maxn））then expand（0）;

if not（（h[1]>=t[1]）or（t[1]>=maxn））then expand（1）;

Until（（h[0]>=t[0]）or（t[0]>=maxn））And（（h[1]>=t[1]）or（t[1]>=maxn））

实验证明，双向搜索与单向搜索相比扩展的结点数减少了至少 1/2，本文中的双向搜索算法采用邻接表存储，不但减少了存储空间还有效地降低了算法的时间复杂度。

2.2 实验及结果分析。算法可行性分析，采用经典的 Dijstra 算法与优化的双向搜索算法对南昌市区部分道路进行最短路径搜索实验[2]。实验过程中，主要在市区的主干道进行实验，共存储 186 个结点烽 548 条弧。对同一起点与同一终点进行最短路径搜索，搜索结果如表 1 所示。

实验次数	Dijstra 算法		改近的双向搜索	
	顶点总数	时间 /s	顶点总数	时间 /s
1	82	0.24	21	0.09
2	105	0.45	58	0.12

三、结论

本文提出的路规划算法采用邻接表的存储方法，对双向搜索算法进行优化，对算法中因双向结点不相遇使得时间复杂度增加的问题，采用停止一方搜索的办法，使该算法符合城市智能交通的要求。实验证明，该算法大多数情况下可得到最优解，克服了以往算法中的不足，符合城市智能交通发展的需求。

参考文献

[1] 周兴 . 面向 Internet 的动态路径规划算法研究与应用系统设计 [D]. 广州 : 华南理工大学 ,2011.

[2] 底园园 , 苏小会 . 交通诱导系统中动态路径诱导算法的研究 [J]. 计算机与数字工程 ,2011,39(2):33-35.

（作者单位：江西科技学院信息工程学院）