基于物联网的智能交通系统车辆路径规划算法优化研究

王锐
（吉林工商学院，长春 130012）

摘要： 本文提出了适用于智能交通系统的基于双向搜索的改进算法。典型的最短路径算法被认为是 Dijkstra 算法,其时间复杂度是 O（n^2）。但一个城市的路网地图有很多节点,该算法的时间复杂度高和解决速度慢。为了改变这种情况,我们从算法的设计方面进行了讨论,提出了改进的双向搜索算法。实践证明,改进后的算法能够提高了搜索速度，适用于智能交通系统。

关键词： 物联网；智能交通；路径规划；双向搜索算法
中图分类号： U492.22 **文献标识码：** A **文章编号：** 1007-9599 (2012) 17-0140-02

1 引言

智能交通系统的核心即动态车辆的路径规划问题，如何能提高路径规划算法的速度是保证整个智能交通更好更快发展的前提。目前具有代表性的最短路径算法是 Dijkstra 算法，其时间复杂度为 O（n^2）。但因 Dijstra 算法是一个 NP 完备算法，面对城市交通路网的众多结点，此算法的时间复杂度高，很难满足导航系统中的实时性要求。本文从算法设计方面对现有的双向搜索算法进行优化，实验证明，能够达到提高算法效率的目的，使其适用于智能交通中的车辆导航系统。

2 算法的优化原理

所谓双向搜索指的是搜索沿两个方向同时进行，正向搜索：从初始结点向目标结点方向搜索；逆向搜索：从目标结点向初始结点方向搜索；每个新结点生成后，不仅要与本队列中的每个结点判重，还要和对方队列中的节点判重，如果有相同结点，即发生双向搜索相遇事件，搜索完成，搜索步数等于两个方向搜索步数之和，生成的搜索树是菱形的，极大的减少了搜索结点的数量，提高了搜索效率。实验表明,和单向搜索展开的结点数相比,双向搜索展开的结点数至少可以减少 1/2，搜索效率明显提高。

此算法的最优状态是正向和逆向的搜索在图中相遇，最不利的情况是正向搜索和逆向搜索没有相遇的结点，这样反而使算法的搜索时间增加了一倍。因此适当的放宽搜索终止条件才能真正缩短搜索时间。我们的优化就是在不增加算法的时间复杂度基础上，解决在双向搜索中结点没有相遇的情况。

算法的优化

在搜索过程中，将每次搜索到的新结点向源结点和目标结点的连线做投影，计算从正向搜索到的新结点的投影到源结点的距离和从逆向搜索到的新结点的投影到目标结点的距离，并计算这两个距离的和，如果距离之和比源结点和目标结点的连线的直线距离长，我们认为双向搜索不会有重合点，立刻停止某一方向的路径搜索，继续另一方向的结点搜索，双向搜索中选择某个方向继续进行搜索或某个方向停止搜索，主要取决于哪个方向的结点个数较少，就向那个方向进行扩展。众所周知，“两点之间直线距离最短”，如果我们所要寻找的结点恰好在同一条边上，这条边就是我们要找的最短路径，否则，和两点间连线的夹角最小的边是最短路径的可能性较大。如图 1 所示，在城市路网中搜索两点 A 和 E 之间的最短路径的优化后算法的步骤如下：

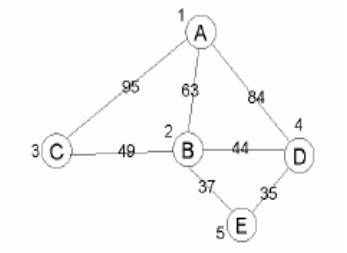

图 1 路网模拟图

（1）结点 A 为出发地，结点 E 为目的地。在结点 A 和结点 E 之间，建立一条连线 AE，若 AE 之间存在一条边和 AE 连线重合，则这条边就是我们所求的最短路径。

（2）若不满足上述条件，则我们从出发点 A 和目的地 E 同时向中部搜索与 AE 这条直线夹角最小的边，在搜索过程中，每次都选择结点个数较少的那个方向先扩展。

（3）重复步骤（2）直到两个方向的搜索能汇合于同一结点或同一条边。把两个方向搜索过的结点聚集，就能得出从 A 到 E 的完整最短路径。

（4）在搜索过程中，将双向搜索到的新结点向 AE 直线做投影，计算从 A 方向搜索到的结点的投影到 A 的距离和从 E 方向搜索到的结点的投影到 E 的距离，并计算这两个距离之和，如果距离之和比 AE 直线距离长，我们就认为双向搜索不会有重合点，立刻停止某一方向的路径搜索，继续向结点个数较少的那个方向进行扩展，搜索结果为单向扩展的结点为所求的最短路径。

3 路径规划算法设计

3.1 算法的实现。设置两个队列 c:array[0..1,1..maxn] of jid，分别表示正向搜索和逆向搜索的扩展队列；两个头指针 head:array[0..1] of integer 分别表示正向搜索和逆向搜索中当前将扩展结点的头指针；设置两个尾指针 tail:array[0..1] of integer 分别表示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 （tail[0]<=tail[1]） and not（（head[0]>=tail[0]）or（tail[0]>=maxn）） then expand（0）;

if （tail[1]<=tail[0]） and not（（head[1]>=tail[1]）or

（tail[1]>=maxn)） then expand（1）;

如果一方搜索终止，继续另一方的搜索，直到两个方向都终止

if not（(head[0]>=tail[0]）or（tail[0]>=maxn)） then expand（0）;

if not（(head[1]>=tail[1]）or（tail[1]>=maxn)） then expand（1）;

Until （(head[0]>=tail[0]）or（tail[0]>=maxn)） And（(head[1]>=tail[1]）or（tail[1]>=maxn))

通过实验表明，双向搜索和单向搜索展开的结点数相比至少减少了 1/2，在本项目中采用的邻接表存储并采用双向搜索算法，能大大减少存储空间并有效降低算法的时间复杂度。

3.2 实验及结果分析。为了验证算法的可行性，我们分别用经典的 Dijstra 算法和优化后的双向搜索算法对长春市区的部分街道进行最短路径的搜索。在搜索过程中，我们主要搜索的是长春市的主干道，忽略了一些小的胡同。如下部分长春市地图中共存储 189 个结点和 567 条弧。对同一起点和终点完成最短路径搜索，两种算法搜索的结点个数和花费的时间如表 1 所示。

表 1 最短路径搜索算法性能比较

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	79	0.27	23	0.08
2	103	0.43	56	0.12

4 结论

本文的路径规划算法采取邻接表的存储方式，并在此拓扑结构的基础上对双向搜索法进行优化，对双向搜索算法中双向结点搜索不相遇而导致的时间复杂度增加问题，通过停止一方搜索给出了解决的办法，使此算法适用于智能交通中对路径规划速度的要求，并给出了在此拓扑结构中交通管制带来的单向行驶问题在本算法中的解决方式。大部分的双向搜索都能在图中相遇，虽然个别没有在图中相遇的，此算法也能得到相应的解，通过实验表明此算法虽然不能每次都得到一个最优解，但最终的理论研究和案例分析的结果表明改进后的算法能够克服以往算法的不足，符合智能交通中路径规划的快速性要求，在实际应用中是可行的。

参考文献：

[1]赵亦林.车辆定位与导航系统[M].谭国真.北京:电子工业出版社,1999:110-132.

[2]Lu Ruqian.Artificial intelligence[M].Beijing:Science Press,1989

[基金项目]吉林省教育厅科研计划项目（吉教科[2012] 380）

（上接第 270 页）

在探讨问题，解决问题的过程中不断获取新知，最后由师生共同进行归纳总结。比如给学生一篇文章配图，要求大家对其排版，做成一篇图文并茂的美文。学生积极思考，充分发挥想象力，动手实践，有的同学的作品风格沉稳大气，有的活泼，有的隽秀，老师应充分肯定学生的成绩，鼓励大家通过独立思考进行创作，并在旁边给予适当的指导并答疑。作品完成后，选取 3——4 幅有代表性的作品，由学生和教师共同讨论点评，肯定长处，同时提出改进意见，并评出成绩，作为平时成绩记入成绩档案。这种教法使学生学习的内容变的非常具体、实际。通过这种师生之间、学生之间的互动，营造主动研究、探索，不断创新的课堂，从而有利于学生潜能的开发，综合素质的提高和创新能力的培养。

4 采用“走出去”的实践教学法

各个职业院校的毕业生都存在一个致命弱点，那就是动手能力有所欠缺。为解决这个普遍问题，教师一方面要鼓励学生动手，另一方面可以多举办一些技能比赛，再就是增加学生见习机会，这样既可以培养学生的动手能力和学习兴趣，同时还可以让学生尽快接触社会，接触岗位，为将来的工作奠定良好的素质基础。我们幼师专业的学生有去幼儿园见习的机会，能不能把多媒体引入幼儿园课堂，有没有更好的生动活泼的教学方式，这都值得我们幼师专业的学生思考。也许，当大家发现计算机能更好的辅助教学活动的时候，计算机学习能再上一个新台阶吧。

为培养合格的幼教人才，不断改善计算机课程的教学效果，教师必须在教学实践中不断更新教学理念，改善教学方法，激发学生对计算机的兴趣和热爱，这样才能在有限的教学时间内争取效率的最大化。