第32卷 第20期 计 算 机 工 程 2006年10月
Vol.32 № 20 Computer Engineering October 2006
·人工智能及识别技术· 文章编号：1000—3428(2006)20—0207—03 文献标识码：A 中图分类号：TP18

基于层次空间推理模型的交通网络最优路径算法

李建元[1,2]，师 军[1]
（1. 陕西师范大学计算机科学学院，西安 710062；2. 山西师范大学机械电子工程系，临汾 041000）

摘 要：在交通地理信息系统(GIS-T)和智能交通系统(ITS)中，单源最短路径问题是一个研究热点。该文总结了已有的层次空间推理过程，针对交通网络，运用了限制搜索区域策略和比较策略，提出了新的层次空间推理过程，寻找更为可靠的入口，以便搜索到更可靠的最优路径，保证层内最优路径搜索的高效率，进一步提高了最优路径算法的可靠性，并对算法的效率进行了必要的分析。

关键词：层次空间推理；交通网络；最优路径；平面算法；入口

Optimum Path Algorithm in Traffic Network Based on Model of Hierarchical Spatial Reasoning

LI Jianyuan[1,2], SHI Jun[1]
(1. School of Computer Science, Shanxi Normal University, Xi'an 710062;
2. Department of Mechanical and Electronic Engineering, Shanxi Teachers University, Linfen 041000)

【Abstract】 This paper summarizes the known hierarchical spatial reasoning procedure, for traffic network, adopts the strategy narrowing search region and the strategy of comparison, puts forward a new hierarchical spatial reasoning recursive procedure, improves the reliability of optimum path algorithm, and gives a necessary analysis about the efficiency of the optimum path algorithm.

【Key words】 Hierarchical spatial reasoning; Traffic network; Optimum path; Plat algorithm; Entrance

在交通地理信息系统(GIS-T0 和智能交通系统(ITS0 中，单源最短路径问题是一个研究热点，Dijkstra算法是GIS系统中最短路径搜索的首选算法，但由于经典Dijkstra算法求解单源最短路径的时间复杂度为O($n^2$0，对海量GIS数据的处理效率较低，许多专家学者提出了Dijkstra改进算法，其中有双向Dijkstra算法、基于邻接表的Dijkstra算法、基于k叉堆优先级队列的Dijkstra算法等，文献[1]提出的基于邻接表和四叉堆的优先级队列的改进型Dijkstra算法的时间复杂度为O(nlogn0, 显著提高了经典算法的搜索效率，完全能满足城市交通网络最短路径的快速搜索，但是若面对多个省多个城市的几百万平面数据，即使是时间复杂度为O(nlogn)的算法搜索路径的效率也比较低。而且，交通网络中“最短”并不代表“最优”，最短路径中可能包括很多窄小拥挤的小巷，大大降低行车的速度，而最优路径则要尽量包含高速路和主干道，少包含小路，针对上述问题，文献[2]提出了层次空间推理过程，既可以查找到“最优”路径，又可以提高道路搜索效率，但Adrijana Car的推理过程存在一定的缺陷，导致运用Adrijana Car推理过程的最优路径算法具有不可靠性，这种不可靠性主要表现在从低层到高层的过渡可能会造成整个路径的舍近求远，针对这种不可靠性，本文利用了新的层次空间推理策略，目的是寻找更为可靠的入口，以便搜索到更可靠的最优路径，而在层内则利用文献[1]提出的基于四叉堆的优先级队列的改进型Dijkstra算法，这样既可以保证层内最优路径搜索的高效率，又进一步保证了其可靠性。

1 路网层次空间推理模型

1.1 层次空间数据模型

人类在认识客观世界时，往往具有明显的空间层次特征，每一个层次都包含着解决特定问题的信息，假设在城市内搜索最优路径，希望所得的路径中主干道相对多一些，小巷和胡同相对少一些，这样才更符合驾车出行者的愿望，于是可以将所有主干道抽象为概化道路模型，而包含小巷和胡同的原始地图可以看作详细道路模型，概化道路就好像一棵树的主干和粗枝，而详细道路则不仅包含主干和粗枝还包括所有的细枝和树叶，这样认识 GIS 路网便具有了层次性，详细道路网可视为低层，概化道路网可视为高层，高层节点集合是低层节点集合的真子集，高层路段集合是低层路段集合的真子集，图 1 所示是一般采用的 3 个层次的模型，该模型对道路进行了分类：高速路的类标识为 0，一般公路的类标识为 1，小巷和胡同等的类标识为 2，第 0 层只包含类标识不大于 0 的路段和节点，第 1 层只包含类标识不大于 1 的路段和节点，最 2 层则包含类标识不大于 2 的路段和节点(即所有路段和节点)，这种层次空间数据模型是与平面数据模型相对应的，平面数据模型只相当于层次模型的最低层，把在平面数据模型下用来搜索最短路径的算法统称为平面算法。

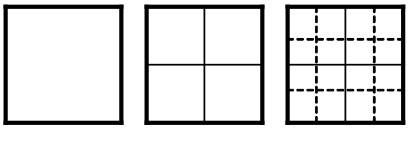


图 1 道路的层次空间模型

1.2 层次空间推理过程

如果要将平面算法运用到层次路网中，则须定义一些规

基金项目：国家自然科学基金资助项目(40471102)
作者简介：李建元(1979－)，男，硕士生，主研方向：智能信息处理；师 军，副教授
收稿日期：2005-11-03 **E-mail：**ljy@stu.snnu.edu.cn

则，该规则必须明确两点：(1)最短路径算法所针对的数据集；(2)在不同的层次之间如何有效地切换。这是由于不同的层次空间，平面算法所对应的数据集是不同的。另外，在层次间切换时，怎样寻找合理的入口是求取最优路径的关键问题，其中入口是指从低层进入高层的过渡节点 。以下是文献[2]中 Adrijana Car 的层次空间推理的递归过程：

假设A_i和B_j分别为起始节点和目标节点，i和j分别表示包含A、B两个节点所在的最高层(不失一般性，假设i<j)。

(1)决定起算节点(选择层次较低的节点，即B_j)；

(2)运用平面算法搜索从起算节点B_j到上一层(第k层 0 最近的入口节点；

(3)存储在第 j 层搜索到的这部分最短路径；

(4)创建一个子图(即第 k 层)；

(5)将B_k 当作新的起算节点；

(6)如果A_i 和 B_k属于不同的层或者A_i 和 B_k不属于最高层，则转至第(2)步；否则，运用平面算法在这个子图中搜索另外一部分最短路径。

(7)将部分最短路径拼接在一起，构成全局最优路径。

1.3 新的层次空间推理过程

上述层次空间模型是利用规则格网来分析的，然而现实的路网非常复杂，现实路网的复杂性利用上述算法往往会得到“绕行”的道路，因为从低层到高层的入口节点不一定是离起算点最近的那一个高层节点，也就是说最近的入口未必是最好的，如图 2 所示，A 为起始点，B 为目标点，由 Adrijana Car 递归过程得到的入口是 C，最优路径是 $A \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow B$，而实际中期望的入口却可能是 E(这里假设 AC<AE0，期望的最优路径是 $A \rightarrow E \rightarrow F \rightarrow G \rightarrow B$。当然，并不是仅仅考虑后一条路径比前一条路径的实际距离短，合理的比较条件是两条路所用的时间，而计算行车时间要根据驾车者的期望速度和道路的限速来决定。

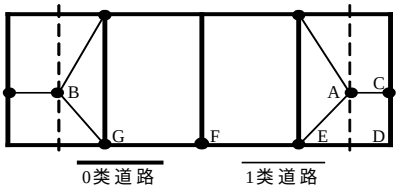


图 2 缺陷之一

针对图 2 的绕行缺陷，在B_j所在的网格中搜索入口时，引入限制搜索区域的方法，为简单起见，用起始点A和目标点B构造边界(如图 2 虚线所示)，假设A的坐标为(xA,yA)，B的坐标是(xB,yB)，在界内便可能搜索到期望的入口，而不至于绕行，其中边界分为垂直边界和水平边界，图 2 采用垂直边界，那么如何选择边界呢？我们定义，当$|xA - xB| > |yA - yB|$时选择垂直边界，反之则选择水平边界。但同时也不放弃搜索Adrijana Car提出的离A最近的入口，也就是说要搜索两个入口(当然也可能这两个点会是同一个点)，并对两个入口对应的路径进行比较，选择代价最小的一条作为最优路径。

另一方面，文献[3]中提出了当起始点和目标点所在的网格相邻或相间(两个网格被另外一个网格分开)时，直接抽取相关的网格利用平面算法直接获得最短路径，但是在复杂的道路网中，这样获得的最短路径并不一定是最优路径，如图 3，利用平面算法获得的最短路径是路径 1，利用分层算法获得的最优路径是路径 2，若根据行车所需的时间计算，后者可能优于前者。针对这一缺陷，当起始点和目标点所在的网格相同、相邻或相间时，单纯地对相关的网格采用平面算法获得的最短路径是不可靠的，如果把路径 1 与分层算法获得的路径 2 进行比较择优选取，便可以保证最优路径的可靠性。

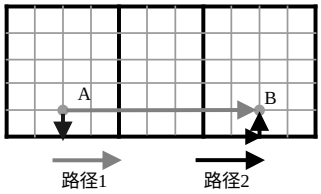


图 3 缺陷之二

以下是新的层次推理递归过程：

(1)决定起算节点(选择层次较低的节点，即B_j)；

(2)运用平面算法搜索从起算节点B_j到其上一层(第k层)最近的入口节点B_{k1}和界内最近入口B_{k2}；

(3)存储在第 j 层搜索到的这两部分最短路径；

(4)创建一个子图(即第 k 层)；

(5)将B_{k1}和B_{k2}当作新的起算节点；

(6)如果A_i 和B_{k1}或B_{k2}属于不同的层或者A_i 和B_{k1}或B_{k2}不属于最高层，则转至第(2)步；否则，运用平面算法在这个子图中搜索另外两条部分最短路径；

(7)将部分最短路径拼接在一起，构成两条全局路径，选择时间代价较小的一条作为最优路径；

(8)若A_i 和B_j所在的网格相同、相邻或相间，则直接在相关网格内利用平面算法求取最短路径，并将该路径与第(7)步的最优路径比较，择优选之。

2 分层路网最优路径算法

2.1 采用的平面算法和数据结构

无论是寻找入口还是寻找层内的最短路径都必须利用某种平面算法，最常用的便是A*搜索算法和Dijkstra算法，由于A*搜索算法的时间复杂度与启发函数有关，只有选择非常好的启发函数，才能提高搜索效率，对于复杂的路网，寻找合理的启发函数较为困难，因此不使用A*算法。另一方面，传统的基于邻接矩阵的Dijkstra算法具有庞大的网络数据存储需求和较高的空间复杂度，时间复杂度无法降低，因此不能采用。基于邻接表的网络存储结构虽不能改变其时间复杂度$O(n^2)$，但可改善其存储需求和空间复杂度，而且为降低时间复杂度提供了可能，邻接表被证明是目前最适合最短路径搜索的数据结构。在众多的Dijkstra变种算法中，文献[1]提出的基于邻接表和四叉堆优先级队列的改进型Dijkstra算法的时间复杂度为O(nlogn)，显著提高了经典算法的搜索效率。

根据层次空间推理的需求，将各层的数据分开存储，并把每一层的数据存储为邻接表，平面算法采用这种方式：基于邻接表结构并结合四叉堆优先级队列存储 Dijkstra 算法中扩展出的向量 D。

2.2 搜索入口的算法描述

搜索边界内入口的算法描述如下：

```
D[v0]= 0; final[v0]=TRUE;
Make-heap (A); //建立四叉堆
for(i=0;i< v0.adj.num ;i++ ) //将v0 邻接节点对应D 值插入堆中，
//其中v0.adj.num表示邻接点总数，而v0.adj[w ]表示第w个邻接点
heap-insert (A ,D [v0.adj[w ]]) ;
for (i= 1;i<n; i+ + ) {
heap-extract-min (A , v ) ;
//从堆中抽出最小 D 值，标识为已处理节点
final[v]=TRUE;
if (v.highestlayer>v0.highestlayer){
//如果v在v0的高层，设置边界，得到一个入口
```

```
if ( |t.x - v0.x| >= |t.y - v0.y| ) //t 表示目标点
{if(v.x>v0.x&&v.x<t0.x ‖ v.x<v0.x&&v.x>t0.x) return(v);break;}
else {if(v.y>v0.y&&v.y<t0.y ‖ v.y<v0.y&&v.y>t0.y) return(v);
break ;}
}
for(w=0;w<v.adj.num;w++) {
if (!final[v.adj[w]]&&(D[v]+c[v ][v.adj[w ]<D[v.adj[w]]]) {
if (D[v.adj[w]]= = INFINITY ) {  //未插入堆中的节点
D[v.adj[w ]]=D[v]+c[v][v.adj[w]];
Heap-insert(A ,D[v.adj[w]]);}
else {  //已插入堆中但需松弛节点
D[v.adj[w]]=D[v]+c[v][v.adj[w];
Heap-decrease-key (A ,pos,D[v.adj[w]]) ; } } }
}
```

若要搜索最近的入口则将上述 if 条件语句改为

```
if (v.highestlayer>v0.highestlayer){return(v);break;}
```

2.3 算法效率和最优路径可靠性分析

在平面数据模型下，采用文献[1]提出的基于四叉堆优先级队列及邻接表的改进型Dijkstra算法，其算法的时间复杂度为O(nlogn)，对一条长度为m的路径，若以m为子网半径，则子网内包括m^2个节点，算法的时间复杂度为O(m^2logm)。

对于层次推理递归过程的网格，假设层次网格宽度比例为k，共有k^2个节点，于是搜索的复杂度为O(k^2logk)，这是一个常数，而且在每个递归层次内都是O(k^2logk)，若递归的次数为L，则时间复杂度为O(L)，而递归次数是路径长度m的对数，故层次策略将非分层模式下的时间复杂度O(m^2logm)降低到O($\log_k$ m)，将路径搜索的效率大大提高，而且节点和路段越多，效率提高的倍数就越高。

根据图 2，可以看出采取限制搜索区域策略可以有效地避免“绕行”，通过两条路径的比较，选择一条更为合理的道路，这样能更符合出行者的意愿，对图 3 中网格相同、相邻或相间的情况采取比较策略也是必要的，新的策略并没有增加算法的时间复杂度，但获得的最优路径的可靠性明显提高。

3 结束语

针对复杂的海量空间数据，层次空间推理的运用是必要的，比起单纯基于平面数据搜索，得到的最优路径更能为驾车出行的人所接受，在层次空间推理过程中，搜索到合理的入口是至关重要的，若只单纯地寻找最近的入口，往往会造成行车方向与目的地的局部背离，造成全局的绕行，因此要在搜索入口的时候限制搜索的区域，但若单纯地使用这种限制，虽不会产生驾车出行的局部背离，也可能造成全局路径的欠佳。所以采用两个入口相比较的办法更为合理，更能提高结果的可靠性，对于网格相同、相邻或相间的情况也应采用比较策略。

参考文献

1 陆 锋, 卢冬梅, 崔伟宏. 基于四叉堆优先级队列及逆邻接表的改进型 Dijkstra 最短路径算法[J]. 中国图像图形学报, 1999, 4 (12): 1039-1045.

2 Car A. Hierarchical Spatial Reasoning: Theoretical Consideration and Its Application to Modeling Way Finding[D]. Vienna: Department of Geoinformation, Technical University of Vienna, 1997: 101-107.

3 Car A, Frank A. General Principles of Hierarchical Spatial Reasoning the Case of Wayfinding[C]. Proceedings of the 6th International Symposium on Spatial Data Handling, Scotand, Edinburgh, 1994: 646-664.

4 Drozdek A. 陈曙辉译. 数据结构与算法 C++版 [M]. 北京: 清华大学出版社, 2003: 237-241.

5 陆 锋, 卢冬梅, 崔伟宏. 交通网络限制搜索区域最短路径算法[J]. 中国图像图形学报, 1999, 4(12): 849-853.

(上接第 192 页)

表 4 规则表

规则	覆盖度,置信度	规则	覆盖度，置信度
(b = 3) =>(类别 = 1)	[75%,100%]	(e = 3) & (h = 1) => (类别= 2)	[50%,100%]
(a = 2) & (c = 1) => (类别=1)	[50%,100%]	(a = 2) & (f = 1) => (类别= 3)	[50%,100%]
(c = 1) & (g = 3) => (类别=1)	[75%,100%]	(c = 3) & (f = 1) => (类别= 3)	[50%,100%]
(d = 1) & (f = 3) => (类别=1)	[100%,100%]	(d = 2) & (f = 1) => (类别= 3)	[50%,100%]
(f = 3) & (g = 3) => (类别=1)	[75%,100%]	(d = 1) & (f = 1) => (类别= 3)	[50%,100%]
(d = 3) => (类别 = 2)	[50%,100%]	(e = 1) & (f = 1) => (类别= 3)	[50%,100%]
(a = 3) & (b = 1) => (类别=2)	[50%,100%]	(e = 2) & (f = 1) => (类别= 3)	[50%,100%]
(a = 3) & (c = 2) => (类别=2)	[75%,100%]	…	

表 5 规则验证

地点	阜新	霍林河	晋城
a	7.5(2)	0.2(1)	11.2(3)
b	2 750(2)	3 163(3)	2 563(1)
c	540(2)	381.5(1)	650(2)
d	3 400(2)	-	4 000(2)
e	30(1)	-	-
f	500(1)	-	-
g	1 746(3)	1 543.8(2)	1 558(3)
h	3.7(3)	-	-
规则	10,1214	1	7，8
类别	3	1	2

4 结论

本文把自组织人工神经网络自组织自学习的优点和粗糙集有效约简数据冗余属性的特点相结合，设计了一种分类规则的提取方法。本方法能有效地解决连续性数据离散后信息流失的问题。方法利用自组织人工神经网络的分类结果验证离散的可行性，使两次分类结果相同或者达到满意的要求，保证了提取规则的一致性。利用粗糙集约简冗余属性，大大提高分类的精确性和分类效率。通过对煤矿废弃地分类提取的规则可以看出，该方法提取的规则较为合理，不仅可以有效地解决知识系统规则瓶颈，提取的规则还可以对不完整信息进行分类。

由于不同的离散方法并不能保证分类的结果一致，因此如何选择适当的离散方法，使得离散前后的分类结果一致将是下一步研究的主要问题之一。

参考文献

1 何月娟, 谢铭培. 神经网络在提取规则中的应用[J]. 计算机工程 1995, 21(5).

2 黎 明, 张化光. 基于粗糙集的神经网络建模方法研究[J]. 自动化学报, 2002, 28(1): 27-33.

3 谭天乐, 宋执环, 李 平. 信息系统数据清洗、规则提取的矩阵算法[J]. 信息与控制, 2003, 32(8): 289-294.

4 Renpu L, Zhengou W. Mining Classification Rules Using Rough Set and Neural Networks[J]. European Journal of Operation Research, 2004, 157(2): 439-448.

5 Gafiychuk V V. Analysis of Data Clusters Obtained by Self-organizing Methods [J]. Physica, 2004, 341(10): 547-555.