

# 空间查询和路径搜索的集成处理策略\*

吴京 景宁 陈华

(国防科学技术大学电子工程学院 长沙 410073)

**摘要** 在数据库研究中,路径搜索和空间查询处理被认为是两个互不相关的领域,然而在处理具有空间约束的路径查询时,需要数据库系统提供路径计算和空间查询处理两方面的功能.为了处理路径计算中的空间约束,考虑了两类处理策略:(1)空间运算是否在路径计算之前预处理;(2)空间对象是否在路径计算之前预选取.基于这两类策略,应用现有的空间连接、R树空间搜索和空间对象聚类技术,提出4种集成的空间路径查询处理方法.

**关键词** 空间查询,路径搜索,查询优化,查询处理.

**中图法分类号** TP311

路径查询处理是在很多应用中都具有的功能,如地理信息系统、导航系统、城市规划等.虽然简单路径查询如“找出从A到B的最短路径”是这类应用中必须提供的功能,但是,先进的应用却越来越需要提供具有空间约束的路径查询处理.本文研究具有约束的空间路径查询,这类空间路径查询的例子如下:

Q1:“找出从A到B的仅经过海拔低于1000米区域的最短路径”;

Q2:“找出从A到B的不经过洪水淹没区域的最短路径”;

Q3:“找出从A到B的不经过布雷区域的最短路径”.

查询Q1是导航系统经常处理的空间查询,Q2是在紧急救助和灾难处理中可能遇到的查询,Q3为军队调动提供重要支持.

处理这一类空间路径查询需要集成路径计算(搜索)和空间查询技术,所谓“集成”是指在空间路径处理时,为了确定其路段(一个路段是指在一条道路中两个相邻交叉路口的道路.基于图论,我们用节点模拟交叉路口,用边模拟路段)是否满足空间约束,空间查询处理必须是路径计算的一部分.本文研究在给定条件下,将空间查询和路径搜索集成于空间路径查询处理的优化策略.

为了处理空间路径查询,需要研究下面可能的两类处理策略.首先,在进行路径搜索之前预处理空间交运算,从而获得与区域相交路段的实例化关系;与此相反的是,在路径搜索时再动态验证路段与区域的相交性,我们称这种方法为“路段预处理”策略.同样地,我们也可以在预处理时过滤掉不满足约束的区域,从而为后续的空间查询处理缩小搜索空间;与此相反的是,空间查询处理在整个搜索空间进行,我们称这种方法为“区域预处理”策略.

本文基于“路段预处理”和“区域预处理”的组合策略,提出各种空间路径查询处理优化策略.综合应用空间连接<sup>[1]</sup>,R-树<sup>[2]</sup>、空间聚类<sup>[3]</sup>和路径搜索<sup>[4]</sup>技术,我们提出有效实现空间路径查询处理策略的技术.

相关的研究集中于路径搜索<sup>[3,4]</sup>和空间处理<sup>[1,2,5]</sup>两方面.本文的目的是集成这两个领域的已有技术(如空间连接、R-树、空间聚类等等)来有效地解决空间路径查询处理问题.与我们的初步工作<sup>[6]</sup>相比,本文在查询处理策略的完整性和形式化两方面有较大的改进.

\* 本文研究得到国防科学技术大学预研项目基金和教育部优秀青年教师基金资助.作者吴京,女,1968年生,讲师,主要研究领域为系统仿真,信息处理.景宁,1963年生,博士,教授,博士生导师,主要研究领域为地理信息系统,数据库技术.陈华,1973年生,博士生,主要研究领域为数据库技术.

本文通讯联系人:吴京,长沙 410073,国防科学技术大学电子工程学院

本文 1998-11-30 收到原稿,1999-03-15 收到修改稿

本文第 1 节介绍相关研究工作,第 2 节介绍空间查询和路径搜索,第 3 节提出 4 种空间路径查询处理策略,最后总结全文。

### 1 相关研究

在数据库研究中,路径问题抽象为无约束的传递闭包(transitive closure)计算<sup>[4]</sup>,文献[3]提出的空间聚类优化适用于有圈(cyclic)图。

对于空间查询,R-树<sup>[2]</sup>是多维空间数据存取方法,文献[7]提出了优化的 R-树算法。文献[1,5]提出了各种空间连接技术,本文的研究重点不在于提出独立的用于路径搜索或空间查询的新算法,而是集成这两个领域已有的技术来解决空间路径查询的优化问题。

### 2 空间查询和路径搜索

#### 2.1 空间地图数据表示

为了处理如 Q1 这类空间路径查询问题,数据库必须存储两类空间对象:(1)全部边的空间和属性数据;(2)全部区域的空间和属性数据。下面来建立表示边和区域的数据结构。

**LINK 表。** LINK 表存储地图上的全部边,每一个 LINK 元组表示边的属性。用  $(i, j, MBR, \&O_e)$  来表示 LINK 元组,其中  $i$  和  $j$  分别是边的起始节点和终止节点,  $w$  是边的长度,  $MBR$  是该条边的最小包围矩形,  $\&O_e$  表示对描述该边的空间数据的引用。由于边的空间数据可能很大,并且是变长数据,因此,它不直接存放在 LINK 表内。

**AREA 表。** AREA 表存储地图上的全部区域,每一个 AREA 元组表示区域的属性。用  $(a, h, MBR, \&O_p)$  来表示 AREA 元组,其中  $a$  是区域标识,  $h$  是区域高度,  $MBR$  是该区域的最小包围矩形,  $\&O_p$  表示对描述该区域的空间数据的引用。

**ARC 文件。** ARC 文件存储所有边的空间数据,每条边用一个整数和多个空间点矢量表示,其中整数表示空间点数,每个点矢量的坐标用 X-和 Y-浮点数表示。

**POLYGON 文件。** POLYGON 文件存储全部区域的空间描述,空间区域形状的描述类似于边的形状描述,不同之处是空间区域的第 1 个空间点和最后一个空间点具有相同坐标。

图 1 说明 LINK 表与 ARC 文件以及 AREA 表与 POLYGON 文件之间的关系。

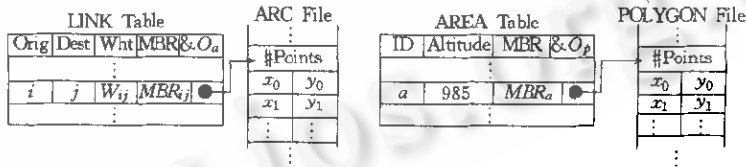


Fig. 1 Map data representation

图1 地图数据表示

#### 2.2 路径搜索

我们用 LINK 表来模拟地图中各条边连接信息的拓扑结构,路径搜索是在 LINK 表的元组中进行递归处理,用  $SP_{ij}$  表示从节点  $i$  到节点  $j$  的最短路径。空间路径查询基于最短路径计算,可以应用很多已有的路径搜索算法,如著名的 Dijkstra 算法。

#### 2.3 空间查询

下面 3 种空间查询技术用于完成空间查询。

**空间相交。** 空间相交运算  $\otimes$  用于判断一条边是否与某一区域相交,一般采用 plane-sweep 技术<sup>[1]</sup>来完成这一运算。 $\otimes$  是二元算子,其参数分别是 LINK 表中元组的空间属性  $\&O_e$  和 AREA 表中元组的空间属性  $\&O_p$ 。如果边与区域相交, $\otimes$  返回 TRUE,否则返回 FALSE。

**空间索引。** 为了有效地搜索与某一边相交的区域,我们采用 R-树索引<sup>[2]</sup>技术,R-树是目前应用最广泛的空

间索引。

空间连接. 文献[1]提出的空间连接技术进行了 CPU 和 I/O 两方面的优化,我们选择该方法来实现空间相交处理。

### 3 空间路径查询处理

本节以开始部分中 Q1 的查询为例来讨论集成的空间路径查询处理策略。

#### 3.1 查询处理的集成设计

我们考虑两类空间路径处理的集成设计:(1) 是否进行空间相交查询的预处理来预先确定边和区域相交关系;(2) 是否过滤不满足空间约束的区域。

设计 1(空间相交预处理). 空间搜索(spatial search,简称 SS)与空间连接(spatial join,简称 SJ)。

在处理查询 Q1 中,对每条可能作为最短路径的边必须基于空间约束来进行有效性验证. 为了验证一条边的有效性,需要进行空间相交处理来选择所有与其相交的区域. 如果与这条边相交的全部区域满足约束“altitude<1000”(“低于海拔 1000 米”),则这条边是有效的. 如果这种空间相交查询处理在路径搜索时对每条经过的边进行,我们称这种方法为空间搜索。

相反地,空间相交在路径搜索之前进行的方法称为空间连接. 在空间连接中,所有边和区域相交关系通过 LINK 表和 AREA 表的空间连接运算生成一个实例化的相交关系表. 在路径搜索时,要验证经过的边的有效性,仅需在相交关系表中进行检索,而不需要进行任何空间处理。

设计 2(过滤不满足空间约束区域). 区域过滤(filtering,简称 F)与非区域过滤(no filtering,简称 NF)。

在计算查询 Q1 时需要对其某条边搜索与其相交的区域,如果搜索是在全部区域集合上进行,我们称这种方法为非区域过滤(no filtering)。相反地,我们也可以先过滤掉不满足约束“altitude<1000”的区域,将满足约束的区域存入另一个表中. 这样,空间相交运算就可以在一个较小的、满足约束的区域表上进行,这种方法称为区域过滤。

组合以上两类设计,我们导出 4 种空间路径查询处理策略:

- 空间搜索和非区域过滤——SSNF(spatial search and no filtering);
- 空间搜索和区域过滤——SSF(spatial search and filtering);
- 空间连接和非区域过滤——SJNF(spatial join and no filtering);
- 空间连接和区域过滤——SJF(spatial join and filtering)。

#### 3.2 空间搜索和非区域过滤——SSNF

SSNF 策略在处理空间搜索时,对每条经过的边在整个 AREA 表上进行空间相交运算. SSNF 策略处理查询 Q1 的步骤如图 2 所示。

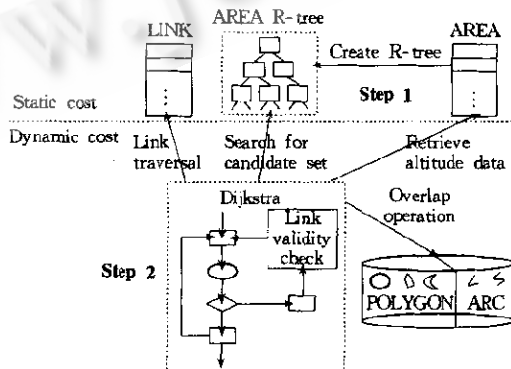


Fig. 2 Spatial search and no filtering strategy  
图2 空间搜索和非区域过滤策略

Step 1. 对 AREA 表中的区域建立空间索引,即 R-树.这样,边和区域的空间相交运算先在 R-树上搜索候选区域,避免直接对路径计算中经过的每条边进行空间相交运算(边和区域的空间相交运算需要读取存放在 ARC 和 POLYGON 文件中的空间矢量数据,并且需要应用 plane-sweep<sup>[1]</sup>技术来判别它们的空间相交性,空间相交运算的 I/O 和 CPU 开销均较大).

Step 2. 在路径计算经过边  $\langle i, j, w, MBR_e, \&O_e \rangle$  时,用  $MBR_e$  在 R-树中搜索与其相交区域,生成候选的相交区域集合.对每一候选区域(假设  $a$ ),SSNF 完成如下检查过程:从 AREA 表中检索元组  $\langle a, h, MBR_p, \&O_p \rangle$ ;如果  $h < 1000$ ,完成  $\&O_e$  和  $\&O_p$  的空间相交运算  $\otimes$ ;如果区域  $\langle a, h, MBR_p, \&O_p \rangle$  与边  $\langle i, j, w, MBR_e, \&O_e \rangle$  相交,则边  $\langle i, j, w, MBR_e, \&O_e \rangle$  是最短路径的边,否则终止该边的路径搜索.

SSNF 策略可以表示为:

$$\{SP_e^* | (\forall \langle i, j, w, MBR_e, \&O_e \rangle \in SP_e^*) (\langle a, h, MBR_p, \&O_p \rangle \in AREA) \\ (\langle i, j, w, MBR_e, \&O_e \rangle \in LINK \wedge h < 1000 \wedge \&O_e \otimes \&O_p = True)\}$$

### 3.3 空间搜索和区域过滤——SSF

与 SSNF 策略不同,SSF 策略过滤 AREA 表(如图 3 所示),其查询处理步骤如下:

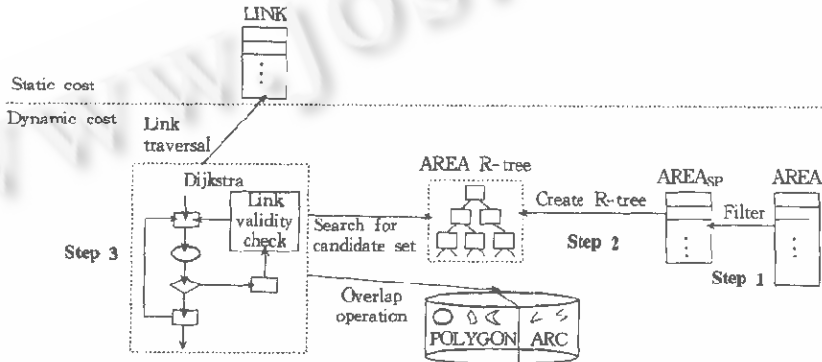


Fig. 3 Spatial search and filtering strategy  
图3 空间搜索和区域过滤策略

Step 1. 从 AREA 表中选取满足条件  $altitude < 1000$  的元组并存入临时表  $AREA_{SP}$  之中,这种过滤操作作为空间查询处理的一部分.

Step 2. 为  $AREA_{SP}$  表建立 R-树索引.

Step 3. 在路径搜索过程中经过边  $\langle i, j, w, MBR_e, \&O_e \rangle$  时,  $MBR_e$  用于在  $AREA_{SP}$  的 R-树中查找与其相交的候选区域.若 R-树中区域  $\langle a, h, MBR_p, \&O_p \rangle$  的最小包围矩形  $MBR_p$  与  $MBR_e$  相交,则进行  $\&O_e$  与  $\&O_p$  的空间相交运算,否则,该边无效.

SSF 策略可以表示为:

$$AREA_{SP} = \{ \langle a, h, MBR_p, \&O_p \rangle | \langle a, h, MBR_p, \&O_p \rangle \in AREA \wedge h < 1000 \}, \\ \{SP_e^* | (\forall \langle i, j, w, MBR_e, \&O_e \rangle \in SP_e^*) (\langle a, h, MBR_p, \&O_p \rangle \in AREA_{SP}), \\ (\langle i, j, w, MBR_e, \&O_e \rangle \in LINK \wedge \&O_e \otimes \&O_p = True)\}$$

### 3.4 空间连接和非区域过滤——SJNF

SJNF 策略(如图 4 所示)首先对 LINK 表和 AREA 表进行空间连接运算,建立连接索引.连接索引的元组如  $\langle i, j, a \rangle$  表示边  $\langle i, j \rangle$  与区域  $a$  相交.进行空间连接运算的优点是,在路径搜索时不必进行代价高的空间相交运算,而只需直接在连接索引上查找.生成的空间连接索引可以用于所有涉及边和区域相交的空间查询. SSNF 的处理步骤如下:

Step 1. 建立 LINK 表的 R-树索引.

Step 2. 建立 AREA 表的 R-树索引.

Step 3. 进行 LINK 表的 R-树和 AREA 表的 R-树的空间连接运算,生成连接表 JOIN\_INDEX,其元组  $\langle i, j,$

a)表示边*(i,j)*与区域*a*相交。

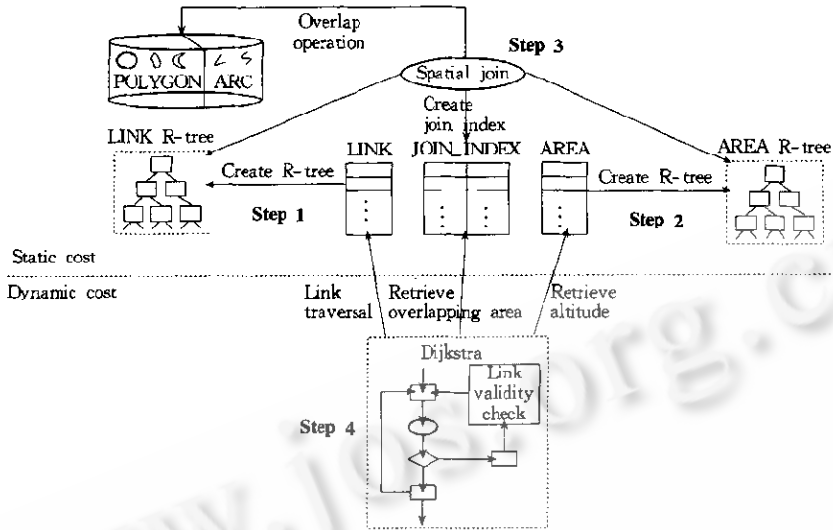


Fig. 4 Spatial join and no filtering  
图4 空间连接和非区域过滤策略

Step 4. 在路径搜索经过边*(i,j,w,MBR<sub>a</sub>,&O<sub>a</sub>)*时,*(i,j)*用于查询 JOIN\_INDEX 表,如果 JOIN\_INDEX 表中存在元组*(i,j,a)*,则用*a*在 AREA 表中搜索元组*(a,h,MBR<sub>p</sub>,&O<sub>p</sub>)*,如果  $h < 1000$ ,则边*(i,j,w,MBR<sub>a</sub>,&O<sub>a</sub>)*有效,否则该边无效。

SJNF 策略可以表达为:

$$\begin{aligned}
 JOIN\_INDEX = \{ & \langle i,j,a \rangle \mid \langle i,j,w,MBR_a,\&O_a \rangle \in LINK \wedge \\
 & \langle a,h,MBR_p,\&O_p \rangle \in AREA \wedge O_a \otimes O_p = True \}, \\
 (SP_a^2 \mid \forall & (\langle i,j,w,MBR_a,\&O_a \rangle \in SP_a^2) (\langle i,j,a \rangle \in JOIN\_INDEX) \wedge \\
 & \langle a,h,MBR_p,\&O_p \rangle \in AREA \wedge h < 1000 \}.
 \end{aligned}$$

### 3.5 空间连接和区域过滤——SJF

SJF 策略(如图 5 所示)结合空间连接方法和过滤方法,其处理步骤如下:

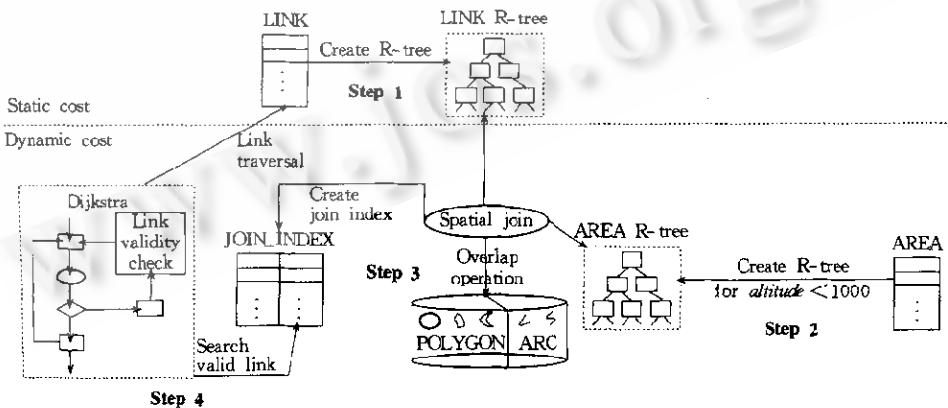


Fig. 5 Spatial join and filtering  
图5 空间连接和区域过滤策略

Step 1. 建立 LINK 表的 R-树索引。

Step 2. 从 AREA 表中选择满足条件  $altitude < 1000$  的元组  $\langle a,h,MBR_p,\&O_p \rangle$ ,然后对这些满足条件的元组建立 R-树索引。

Step 3. 完成 Step 1 和 Step 2 所建立的 R-树的空间连接运算,生成连接索引 JOIN-INDEX 表,其元组  $\langle i, j, a \rangle$  表示边  $\langle i, j \rangle$  与区域  $a$  相交,区域  $a$  满足  $altitude < 1000$ .

Step 4. 在路径搜索经过边  $\langle i, j, w, MBR_w, \&O_w \rangle$  时,  $\langle i, j \rangle$  用于查询 JOIN-INDEX 表;如果元组  $\langle i, j, a \rangle$  存在于 JOIN-INDEX 表中,则该边有效,否则该边无效.

SJF 策略可以表示为:

$$\begin{aligned}
 JOIN-INDEX = \{ \langle i, j, a \rangle \mid \langle i, j, w, MBR_w, \&O_w \rangle \in LINK \wedge \langle a, h, MBR_p, \&O_p \rangle \in \\
 AREA \wedge h < 1000 \wedge \&O_w \otimes \&O_p = True \}, \\
 \{ SP_a^A \mid \langle \forall \langle i, j, w, MBR_w, \&O_w \rangle \in SP_a^A \rangle \langle i, j, a \rangle \in JOIN-INDEX \}.
 \end{aligned}$$

### 4 总 结

本文研究具有约束条件的空间路径查询,提出了两种处理这类空间路径查询的优化方法:(1) 边和区域的空间连接预处理;(2) 空间约束条件的预处理.组合这两种优化方法,我们提出了 4 种空间路径查询处理优化策略.我们进行的初步实验<sup>[6]</sup>表明,各种策略在不同条件下各有其优点.本文的研究工作有助于理解复杂空间查询的处理过程,根据查询条件和数据空间,选择集成的查询优化处理策略.

### 参考文献

- 1 Brinkhoff T, Kriegel H, Seeger B. Efficient processing of spatial joins using R-trees. In: Peter Buneman, Sushil Jajodia eds. Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 1993. 237~246
- 2 Guttman A. R-tree: a dynamic index structure for spatial searching. In: Beatrice Yorrmak ed. Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 1984. 45~57
- 3 Huang Y W, Jing N, Rundensteiner E A. Effective graph clustering for path queries in digital map databases. In: Ken Barker, Tamer Oszu M eds. Proceedings of the 5th International Conference on Information and Knowledge Management. New York: ACM Press, 1996. 215~222
- 4 Agrawal R, Dar S, Jagadish H V. Direct transitive closure algorithms: design and performance evaluation. ACM Transactions on Database Systems, 1990, 15(3):427~458
- 5 Gunther O. Efficient computation of spatial joins. In: Proceedings of the IEEE International Conference on Data Engineering. Los Alamitos: IEEE Computer Society Press, 1993. 50~59
- 6 Huang Y W, Jing N, Rundensteiner E A. Integrated query processing strategies for spatial path queries. In: Proceedings of the IEEE International Conference on Data Engineering. Los Alamitos: IEEE Computer Society Press, 1997. 477~486
- 7 Bechmann N, Kriegel H, Schneider R et al. The R\*-tree: an efficient and robust access method for points and rectangles. In: Garcia-Molina Hector, Jagadish H V eds. Proceedings of the ACM SIGMOD International Conference on Management of Data. New York, ACM Press, 1990. 322~332

## Integrated Processing Strategies for Spatial Query and Path Search

WU Jing JING Ning CHEN Luo

(School of Electrical Engineering National University of Defense Technology Changsha 410073)

**Abstract** Path search and spatial query are considered as two independent topics in database research. The processing of path query with spatial constraint needs both path computation and spatial query. To provide path computation with spatial constraint, two processing strategies are considered: (1) whether pre-processing spatial operation before path computation, and (2) whether pre-filtering spatial objects before path computation. Based on these two strategies, four integrated spatial path query processing approaches are proposed in this paper.

**Key words** Spatial query, path search, query optimization, query processing.