

# 道路网环境下 $K$ -支配空间 Skyline 查询方法

李松<sup>1</sup> 窦雅男<sup>1</sup> 郝晓红<sup>1</sup> 张丽平<sup>1</sup> 郝忠孝<sup>1,2</sup>

<sup>1</sup>(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)

<sup>2</sup>(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

(lisongbeifen@163.com)

## The Method of the $K$ -Dominant Space Skyline Query in Road Network

Li Song<sup>1</sup>, Dou Yanan<sup>1</sup>, Hao Xiaohong<sup>1</sup>, Zhang Liping<sup>1</sup>, and Hao Zhongxiao<sup>1,2</sup>

<sup>1</sup>(College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080)

<sup>2</sup>(College of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

**Abstract** In order to make up for the shortcomings of the existing research results in dealing with the  $K$ -dominant space Skyline query problem in the road network environment, the method of the  $K$ -dominant space Skyline query in road network based on the network Voronoi diagram is proposed. This method applies  $K$ -dominant to the Skyline query of road network to deal with the multi-attribute data objects and can be used to solve the multi-objective decision problems in practical applications in the road network. The method mainly includes reduction data set process and  $K$ -dominant checking process in road network. Firstly, the Voronoi diagram is constructed based on spatial data points, and the query convex hull is built for query points. The Voronoi diagram of the data and the positional relationship of the query area are used to cut the data set. Thus the data set is optimized and the phenomenon of repeated search of query points is effectively reduced. Then, the refined set is obtained by  $K$ -dominant checking on the non-spatial attributes of the candidate set. Finally, the final spatial Skyline set is obtained by dominating the refined set. Theoretical research and experiments show that the proposed method has higher efficiency and can handle the  $K$ -dominant space Skyline query problem in the road network better.

**Key words** spatial Skyline query; multi-objective decision-making; Voronoi diagram; road network; spatial attribute

**摘要** 为了弥补已有的研究成果无法直接处理道路网环境下  $K$ -支配空间 Skyline 查询问题的不足,提出了基于网络 Voronoi 图的道路网环境下  $K$ -支配空间 Skyline 查询方法。该方法将  $K$ -支配应用到道路网 Skyline 查询中以处理多属性数据对象,在实际应用中可以用来解决道路网环境下多目标查询和决策问题。方法主要包括道路网中约减数据集过程和  $K$ -支配检查过程。首先基于空间数据点构建网络 Voronoi 图,并对查询点建立查询凸包,通过网络 Voronoi 图的性质与查询区域的位置关系对数据集约减,从而优化数据集并且有效地减少查询点重复搜索的现象;然后对候选集的非空间属性进行  $K$ -支配

收稿日期:2019-01-14;修回日期:2019-05-08

基金项目:国家自然科学基金项目(61872105);黑龙江省留学归国人员科学基金项目(LC2018030);黑龙江省教育厅科学技术研究项目(12531z004)

This work was supported by the National Natural Science Foundation of China (61872105), the Scientific Research Foundation for Returned Scholars Abroad of Heilongjiang Province of China (LC2018030), and the Science and Technology Research Project of Heilongjiang Provincial Education Department (12531z004).

检查得到道路网精炼集合;最后对精炼集合进行支配检查得到最终的空间 Skyline 集合.理论研究和实验表明所提出的方法具有较高的效率,可较好地处理道路网环境下  $K$ -支配空间 Skyline 查询问题.

**关键词** 空间 Skyline 查询;多目标决策;Voronoi 图;道路网;空间属性

**中图分类号** TP311.13

随着数据智能和数据处理技术的发展及广泛应用,多目标优化查询问题成为一个热点和难点问题.多目标优化查询帮助用户在一定约束条件下获得在多个目标对象下的查询结果以达到最优的结果集. Skyline 查询是多目标优化查询中一类重要的查询问题. Skyline 查询方法由 Borzsony 等人<sup>[1]</sup>提出,通过 Skyline 点的性质缩小结果集,提高并优化查询效率.

目前, Skyline 查询方法的研究已经拓展到无线传感网络中的动态 Skyline 查询<sup>[2]</sup>、不确定数据 Skyline 查询<sup>[3]</sup>、移动  $K$ -支配最近邻查询<sup>[4]</sup>、反 Skyline 查询<sup>[5]</sup>、数据流 Skyline 查询<sup>[6]</sup>、子空间 Skyline 查询<sup>[7]</sup>、连续 Skyline 查询<sup>[8]</sup>、概率 Skyline 查询<sup>[9]</sup>、Top- $k$  Skyline 查询<sup>[10]</sup>、基于 MapReduce 的 Skyline 查询<sup>[11]</sup>、障碍空间中的 Skyline 查询<sup>[12]</sup>等方面.取得的研究成果解决了 Skyline 查询领域及延伸领域内的一系列重要的问题.上述研究的空间条件均没有考虑路网环境,但实际生活中人们很多时候均活动在道路网中,因此,道路网 Skyline 查询是一类重要的 Skyline 查询问题,道路网 Skyline 查询问题具有重要的应用价值.

相对于理想空间,道路网的空间拓扑结构更加复杂,数据对象之间的距离计算复杂度更高,传统的距离计算已不能很好地适用于道路网环境中.近年来针对道路网多目标优化查询的研究已经取得一些重要研究成果,潘晓等人<sup>[13]</sup>提出了在道路网络上基于时空相似性的连续查询隐私保护算法,该算法通过分组策略和共享机制防止连续查询攻击和位置依赖攻击. Bao 等人<sup>[14]</sup>提出一种基于将道路边缘划分为子区间的基本算法,并通过检查子区间的端点来找到最佳位置.这些算法很好地处理了路网边缘区域的点,但是却忽略了数据点之间的空间支配关系,导致非边缘的数据对象之间的距离也需要通过复杂的计算得到结果.文献<sup>[15]</sup>对各个区域的数据点与查询点的位置关系均进行了处理,但是算法需要重复检查数据点之间的支配性,从而影响了算法查询效率.

Huang<sup>[16]</sup>出于用户隐私保护提出了基于位置

范围的 Skyline 查询方法,设计了数据对象与道路网间的数据结构,将查询对象模糊化为一个位置范围用以保护用户隐私.该方法计算数据点与查询对象的网络距离时,只需计算数据点与位置范围边界交点的网络距离,再重新利用 Dijkstra 算法来计算距离,不再需要重复查询,因而可以节省实时计算开销. Miao 等人<sup>[17]</sup>针对查询点推理确定更有潜力的 Skyline 点,方法具有一定的创新,但是算法对每个查询点的计算效率有待进一步提升.文献<sup>[18-19]</sup>提出了安全区域的概念,在安全区域内计算查询对象与目标对象之间的最短网络距离.但是安全区域需要对区域更新和维护.

对于  $K$ -支配 Skyline 查询问题, Lee 等人<sup>[20]</sup>提出了一种在计算期间减少选择候选者和支配误报的方法. Miao 等人<sup>[21]</sup>研究了不完整数据上  $K$ -支配 Skyline 查询的问题,提出利用位图索引缩小搜索空间的方法.这些算法可以有效减少数据点的  $k$  维属性支配判断的比较次数和减少 I/O 代价次数,具有较好的性能.但忽略了数据对象的空间属性对查询结果的影响,当数据对象的空间位置属性改变时,则需要重新计算  $k$  维的结果.文献<sup>[22]</sup>考虑到数据对象的属性具有相关性,通过查询预处理对空间属性必支配的属性进行处理使得查询效率提升.文献<sup>[23]</sup>基于 MapReduce 平台提出了一个有效的并行算法来处理  $K$ -支配 Skyline 查询问题.文献<sup>[24]</sup>提出了一种基于索引的  $K$ -支配 Skyline 查询算法,通过为数据集建立 2 个索引,算法可以高效地进行计算,在时间、空间和渐进性上具有一定的性能优势.

已有的道路网 Skyline 查询一般只有一个查询对象,但是查询对象往往是不唯一的,查询对象之间的相互影响可能导致数据对象的属性值发生变化.因此,使用单一的查询点会导致查询结果不符合实际需求.并且查询方法若不考虑数据点的非空间属性对查询的影响,则可能导致查询的结果不准确.在实际应用中,数据点的空间位置和非空间属性值都对用户的选择有影响,这就需要结合空间属性和非空间属性  $k$  维的支配关系来选择.如何将  $K$ -支配 Skyline 查询问题应用于道路网环境,特别当查询点

的空间位置发生动态改变时还能准确提供最优化查询结果成为研究的难点,针对这些问题,本文将  $K$ -支配应用到道路网 Skyline 查询中以处理多属性数据对象,利用  $K$ -支配解决了道路网查询支配的问题.本文利用道路网 Voronoi 图的定义与相关性质对数据点进行约减和支配检查.在理论论证和分析的基础上提出了道路网  $K$ -支配空间 Skyline 查询方法.

## 1 定义与性质

道路网  $K$ -支配空间 Skyline 查询是道路网 Skyline 查询的变种问题,该问题涉及 4 个组成对象:数据点集、查询区域范围、查询属性范围以及道路网.本文假定数据点集  $P$  是一组具有  $k$  维属性的点的集合,其中  $k-1$  维为静态的非空间属性, $k$  维属性为数据点在道路网中位置的坐标.

本节给出相关的定义与性质.

**定义 1.** 非空间属性支配<sup>[22]</sup>. 给定 2 个数据点  $p_1, p_2$ , 考虑所有的非空间属性, 如果  $p_1$  能被  $p_2$  支配, 称  $p_2$  非空间属性支配  $p_1$ , 记为  $p_2 < p_1$ . 所有非空间属性支配  $p_1$  点的集合称为  $p_1$  的非空间属性支配集, 记为  $Dom(p_1)$ .

**定义 2.** 道路网支配<sup>[16]</sup>. 在道路网环境下给定一个查询点  $q$  和 2 个数据点  $p_1, p_2$ ,  $d(q, p_1)$  表示数据点  $p_1$  距离查询点  $q$  的网络距离. 当且仅当满足 2 个条件时, 称  $p_2$  道路网支配  $p_1$  (简称“支配”), 记为  $p_2 < p_1$ :

1)  $p_2$  在  $k-1$  维的空间中支配点  $p_1$ , 即  $p_2 <_{k-1} p_1$ ;

2)  $d(p_2, q) < d(p_1, q)$ , 在道路网中,  $p_2$  与点  $q$  的网路距离比  $p_1$  与点  $q$  的网路距离更小.

**定义 3.** 空间支配<sup>[7]</sup>. 空间中存在一个数据点  $p \in P$ , 查询点  $q_i$ , 如果满足条件: 对于  $P$  中任意一个数据点  $p' \in P$ , 对于查询点  $q_i$  存在  $d(q_i, p) \leq d(q_i, p')$  成立, 则数据点  $p$  空间支配数据点  $p'$ .

**定义 4.** 网络 Voronoi 图 (network Voronoi diagram, NVD)<sup>[25]</sup>. NVD 与基本 Voronoi 图的区别在于 NVD 采用的是路网距离, 而并非欧氏距离. 在 NVD 中, 由路网中的边和虚线所围成的区域称为网络 Voronoi 区域 (network Voronoi polygon, NVP). 由  $p_i$  生成的 NVP 记为  $V_{NVP}(p_i)$ . 若给定生成点集合  $P = \{p_1, p_2, \dots, p_n\}$ , 边的集合  $E = \{e_1, e_2, \dots, e_k\}$ .  $p_j$  为不同于  $p_i$  的生成点, 则  $V_{NVP}(p_i)$  可以形式化定义为

$$V_{NVP}(p_i) = \{p \mid p \in \bigcup_{i=1}^k e_i, d(p, p_i) \leq d(p, p_j)\}. \quad (1)$$

**性质 1**<sup>[25]</sup>. 如果点  $q$  在网络 Voronoi 多边形  $V_{NVP}(p_i)$  中, 则  $q$  到生成点  $p_i$  的距离小于  $q$  到其他生成点的距离.

**性质 2**<sup>[25]</sup>. 距离  $p_i$  点最近的生成点一定在与  $V_{NVP}(p_i)$  有公共 Voronoi 边的 Voronoi 多边形中.

**定义 5.** 道路网  $K$ -支配空间 Skyline 查询. 给定一个查询点  $q$  和一个  $n$  维空间的对象集  $P$ , 其中道路网  $K$ -支配空间 Skyline 查询结果返回  $P$  的一个子集, 在此子集中的数据点都不能被  $P$  中的任何其他点在用户偏好的  $k-1$  个维度上非空间属性支配, 并且在第  $k$  维空间属性上在道路网环境中也未被其他数据点空间支配, 这样的数据点被称为道路网  $K$ -支配空间 Skyline 点, 记为  $K-PNS(q, P)$ . 道路网  $K$ -支配空间 Skyline 查询本文简称为  $K-PNS$  查询.

表 1 表示一个宾馆数据对象的非空间属性表. 非空间属性是数据点的非空间地理位置的数据点自身存在的一些属性. 这些信息不是地理位置上的空间坐标的位置, 是非空间上数据点本身自有的属性. 本文中的非空间属性是在一定范围内随机取得的模拟值. 表 1 中, 数据点  $p_1, p_2, p_3, p_4, p_5$  为数据对象集中 5 个具有 5 维属性的宾馆, 其中前 4 维为非空间属性, 对应宾馆的价格、开业时间、用户评价和清洁情况. 最后一维属性为宾馆距离查询点的道路网距离, 如图 1 所示, 图 1 给出了 5 个宾馆对象在道路网上的位置分布以及查询区域, 需要查找的是在 5 维属性中数据点在 4 维属性维度上不被支配的查询结果, 即道路网 4-支配空间 Skyline 查询.

Table 1 Example of Non-spatial Attributes of Hotel Data Object

表 1 宾馆数据对象的非空间属性示例

Hotel	Price	User Rating	Opening Hours	Cleanliness
$p_1$	175	3	2	4
$p_2$	226	4	3	5
$p_3$	198	4	5	4
$p_4$	300	5	1	3
$p_5$	156	3	3	3

首先对空间属性进行支配查询, 图 1 中圆弧线  $a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8$  表示数据点  $p_2, p_3, p_4$  相对查询点的支配圆域, 由图 1 可以看出在支配圆域内的点支配生成该圆域的点, 因此, 数据点  $p_2, p_3, p_4$  在道路网环境空间支配点  $p_1$  和  $p_5$ , 即数据

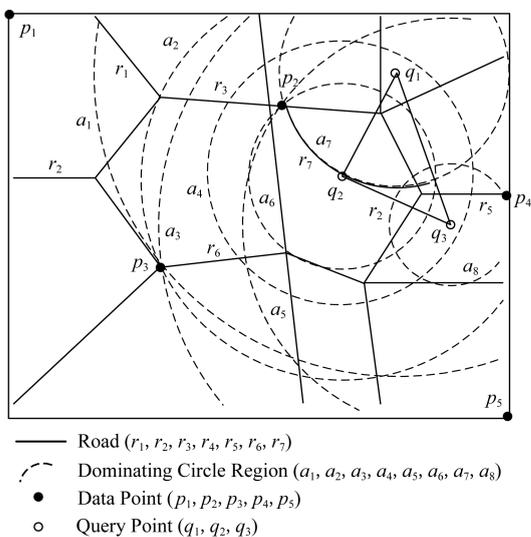


Fig. 1 Example of spatial attributes of hotel data object

图 1 宾馆数据对象的空间属性的示例

点在空间属性上不被支配的查询结果为  $\{p_2, p_3, p_4\}$ . 进一步对非空间属性进行 3 维支配的查询, 基于查询区域的道路网 Skyline 查询需要查找对象集中既便宜, 且用户评价好又距离用户当前位置近的宾馆. 由表 1 可以看到在非空间属性中, 在价格方面数据点  $p_4$  和点  $p_2$  被点  $p_3$  支配, 因此对于数据点在道路网 4 支配空间 Skyline 查询的结果为  $\{p_3\}$ . 即该例子中的道路网  $K$ -支配空间 Skyline 查询的结果为  $\{p_3\}$ .

## 2 道路网 $K$ -支配空间 Skyline 查询

本文提出的道路网  $K$ -支配空间 Skyline 查询方法主要分为 2 个主要的过程: 道路网中约减数据集过程和  $K$ -支配检查过程. 首先通过约减大量的非候选数据点得到道路网环境中的候选集合, 然后对候选集的非空间属性进行  $K$ -支配检查得到道路网精炼集合, 最后对精炼集合进行支配检查得到最终的空间 Skyline 集合. 本节优先处理数据点空间属性的道路网支配关系, 由于数据点非空间属性往往较多, 从而使得非空间属性间支配比较次数增加, 如果优先处理非空间属性间的支配关系, 在数据点数量较大时, 算法的查询效率会降低.

### 2.1 约减数据集

约减数据集过程主要是对数据点进行过滤处理, 通过道路网 Voronoi 图与查询点集合建立影响的查询区域间的空间位置关系来剪枝掉大量的被支配的数据点, 用于获得精确的候选点集合. 首先对空

间数据点构建道路网 Voronoi 图, 再对查询点建立一个查询的凸包. 然后利用道路网 Voronoi 图及其查询区域形成的位置关系以及数据点之间的支配关系来约减大量被支配的数据点, 从而缩小查询范围, 剩下的数据点构成候选点集合. 在此过程中通过对查询点集建立查询区域与建立数据点的道路网 Voronoi 图的方法来约减数据集, 能够优化处理数据集并且有效地减少查询点重复搜索的现象, 约减掉大量的非候选点, 较大程度地缩小了查询的范围.

本文所研究的数据点与查询点之间的距离均采用道路网络中的网络距离. 本节以定理的形式给出约减方法. 假定数据点前  $n$  维的非空间属性是固定不变的, 本节为了简化道路网非空间属性对空间支配的影响, 用数据对象的非空间属性值的总和来表示非空间属性的支配能力, 结合数据点的空间属性, 查询结果得到的不被空间支配的候选集是固定的, 使得查询效率更快. 由于数据点为道路网上的静态点, 故其道路网 Skyline 查询结果的变化是由查询位置与数据点之间的距离变化引起的.

**定理 1.** 在由数据点构造的道路网 Voronoi 图中, 若查询凸包  $CH(Q)$  内有一数据点  $p$ , 则数据点  $p$  支配  $CH(Q)$  与该点所成圆相切区域外的数据点, 则该区域外的数据点均被约减, 区域内的数据点可加入候选集中.

证明. 已知在道路网 Voronoi 图中的数据点  $p$  与各查询点所成圆域外的一点数据点, 2 点所连接的线段的垂直平分线将 2 点分割为 2 个区域,  $p$  点与查询区域  $CH(Q)$  在同一边, 根据垂直平分线性质,  $p$  点与各查询点的距离更近, 则数据点  $p$  支配  $CH(Q)$  与该点所成圆相切区域外的数据点. 在图 2 中, 任意选取 1 组数据点和查询点, 例如查询点集为  $Q = \{q_1, q_2, q_3\}$ , 数据点集为  $P = \{p_1, p_2, p_3, p_4, p_5\}$ . 查询点组成的查询区域  $CH(Q)$  内任意位置有一数据点  $p$ , 分别以各查询点  $q$  为圆心、以数据点  $p$  与各查询点间的距离为半径做圆. 图 2 中的矩形区域表示数据点  $p$  与各查询点形成的圆 ( $circle(p, q)$ ) 相切的数据点  $p$  的支配区域, 圆弧线  $l_1, l_2, l_3$  表示数据点相对不同查询点的支配圆域. 如图 2 所示, 数据点  $p_2$  是数据点  $p$  与各查询点形成的圆相切矩形区域外的任意一个数据点. 从图 2 可以求得数据点  $p$  与查询点  $q_1$  的道路网环境中的网络距离关系为:  $d(q_1, p) < d(q_1, p_2)$ , 同理可得数据点  $p$  与另外 2 个查询点  $q_1, q_2$  的网络距离关系为:  $d(q_2, p) < d(q_2, p_2)$ ,  $d(q_3, p) < d(q_3, p_2)$ . 根据空间数

据点支配的定义可得,点  $p$  在空间上支配区域外的点  $p_2$ ,则数据点  $p$  支配  $CH(Q)$  与该数据点所成圆相切区域外的点,则该区域外的点均被约减。证毕。

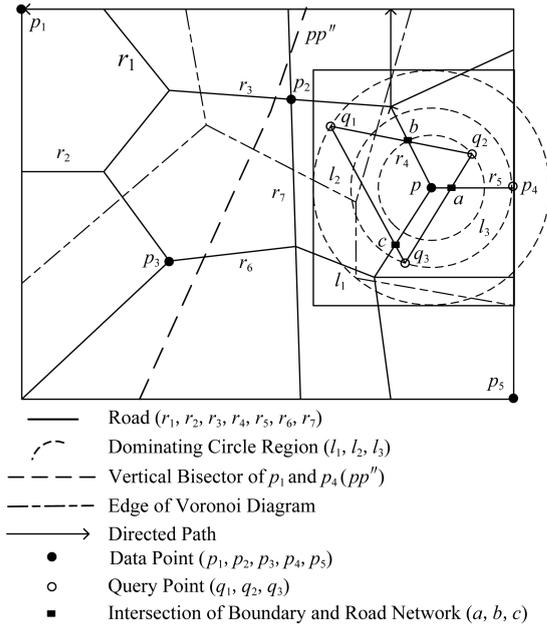


Fig. 2 Example based on theorem 1,2 and 3

图 2 基于定理 1,2,3 的示例

**定理 2.** 已知数据点  $p \in P$ , 如果查询区域  $CH(Q)$  与该数据点所在的道路网 Voronoi 单元  $VC(p)$  相交, 若数据点  $q$  为与数据点  $p$  所在道路网上任意一条不相交的道路网上的一个数据点, 则数据点  $p$  支配点  $q$ , 数据点  $q$  被剪枝。

证明. 任意连接与查询区域有关联的数据点  $p$  和与查询区域不相交的 Voronoi 单元内的数据点, 则该 2 个数据点构成的线段的垂直平分线将与查询区域相交的 Voronoi 单元的数据点与查询点划分在一边. 根据 Voronoi 图性质可知, 在 Voronoi 单元内的数据对象点与形成该 Voronoi 单元的生成点之间的距离最短, 因此与查询区域相交的数据点支配未与查询区域相交的数据点. 如图 2 所示, 任意选取数据点如  $p_4$ , 数据点  $p_4 \in P$ , 数据点  $p_4$  所在的 Voronoi 单元与查询区域  $CH(Q)$  相交, 作线段  $pp''$  为数据点  $p_1$  和数据点  $p_4$  线段的垂直平分线, 根据垂直平分线性质的可知, 同属一边的 2 点距离更近, 由图 2 可以知道线段  $pp''$  与查询区域  $CH(Q)$  不相交, 且该垂直平分线将数据点  $p_4$  与  $CH(Q)$  划分到同一侧, 已知垂直平分线上点到线段的两端的距离相等, 则对于  $q_i \in CH(Q)$ , 一定有  $d(q_i, p_4) < d(q_i, p_1)$ . 因此, 数据点  $p_4$  支配点  $p_1$ , 故数据点  $p_1$

被剪枝. 因为与查询点的距离更近, 所以圆域内的数据点支配圆域外的数据点, 如数据点  $p$ . 同样作线段  $pp_4$  的垂直平分线, 如图 2 所示, 可以看到此垂直平分线与  $CH(Q)$  相交, 则  $CH(Q)$  中距离  $p$  点更近的查询点之间的距离比  $p_4$  更近, 与  $p_4$  同一侧的  $CH(Q)$  内的点之间的距离比与  $p$  点距离更近, 即  $p$  点与  $p_4$  点之间互补支配, 故可以将 2 点都加入候选点集中进行下一步的精炼, 从而得到最终的空间道路网候选集。证毕。

**定理 3.** 对于给定的查询凸包区域  $CH(Q)$ , 若数据点所在的道路网与  $CH(Q)$  的边界相交, 则该数据点与查询各点的网络距离更近, 该数据点可加入到候选集中。

证明. 对于查询区域  $CH(Q)$  内任意查询点  $q$ , 从  $q$  出发, 要到达查询区域  $CH(Q)$  外的任意对象点  $p_i$ , 其最短路径必定经过查询区域  $CH(Q)$  与道路网的交点. 由 2 点间线段最短定理可知, 数据点  $p_i$  与查询点  $q$  间的网络距离一定大于查询点  $q$  与相交路上数据点的距离, 故数据点  $p_i$  被剪枝. 如图 2 所示, 要从对象数据点  $p_1, p_2, p_3, p_4$  到查询区域  $CH(Q)$  中任意一个查询点  $q$ , 必定经过查询区域  $CH(Q)$  的边界与道路网的交叉点  $a, b, c$  这 3 个点之中的某一个. 因此, 对于数据点  $p_1$ , 存在有  $d(q_1, p_1) = d(q, b) + d(b, p_1)$ . 可见数据点  $p_1$  到查询区域内点  $q$  必经过查询区域的交点  $b$ , 由此可知数据点所在的道路网与  $CH(Q)$  的边界相交, 则该数据点与查询各点的网络距离更近, 将该数据点加入到候选集中。证毕。

只有当一个数据点在空间属性与非空间属性上均支配另一个数据点时, 该数据点才可能是道路网中  $K$ -支配 Skyline 点, 应该将这类数据点加入到候选集中. 由于可能存在数据点的某些非空间属性值较大, 影响  $K$ -支配查询的效率, 因此在道路网环境中用数据点的非空间属性值的总和代表数据点非空间属性的支配能力, 总体的支配能力可以初步约减掉部分非空间属性值较大的情况, 便于约减非空间属性支配能力差的数据点。

**约减策略 1.** 数据点在查询区域内或者查询点所在的道路网 Voronoi 单元与查询区域相交, 则将查询点加入到候选集中。

在数据点构造的道路网 Voronoi 图中, 若查询凸包  $CH(Q)$  内有一个数据点  $p$ , 则数据点  $p$  支配  $CH(Q)$  与该点所成圆的相切区域外的数据点, 则该区域外的数据点均被约减, 区域内的数据点可加入

候选集中.任意连接与查询区域有交集的数据点  $p$  和未与查询区域相交的 Voronoi 单元内的数据点,则该 2 个数据点构成的线段的垂直平分线将与查询区域相交的 Voronoi 单元内的数据点与查询点分在一边.根据 Voronoi 图性质,在 Voronoi 单元内的点与形成该 Voronoi 单元的点之间的距离最短,因此,与查询区域相交的数据点支配未与查询区域相交的数据点.

**约减策略 2.** 数据点所在的道路网如果与查询区域相交,则将该数据点加入到候选集中.

任意连接与查询区域有关联的数据点  $p$  和未与查询区域相交的 Voronoi 单元的数据点,则该两个数据点够成的线段的垂直平分线将与查询区域相交的 Voronoi 单元的数据点与查询点化分在一边.根据 Voronoi 图性质可知在 Voronoi 单元内的点与形成该 Voronoi 单元的点之间的距离最短,因此各查询点与有关联的  $V_{NVP}(p)$  内的数据点  $p$  的空间距离更短,即与查询区域相交的数据点支配未与查询区域相交的数据点.对于查询区域  $CH(Q)$  内的任意查询点  $q$ ,从  $q$  开始到达查询区域  $CH(Q)$  外任意对象点  $p_i$ ,其最短路径必定经过查询区域  $CH(Q)$  与道路网的交点.因此,数据点  $p_i$  与查询点  $q$  间的网络距离一定大于查询点  $q$  与相交路网上数据点的距离,故数据点  $p_i$  被剪枝.

**约减策略 3.** 对于候选集中数据点的非空间属性的支配能力进行比较,支配能力更强的数据点留在候选集,否则从候选集中删除.

用数据点的非空间属性值的总和来表示数据对象的非空间属性的支配能力,只有当一个数据点在空间属性与非空间属性上均支配另一个数据点时,该数据点才可能是道路网中  $K$ -支配 Skyline 点,应该将这类数据点加入到候选集中.

基于定理 1、定理 2 和定理 3 及约减策略,本节进一步给出算法  $R$ -PNS,如算法 1 所示.

**算法 1.**  $R$ -PNS( $Q, P, R$ ).

输入:查询区域  $Q$ 、道路网路段集合  $R$ 、数据集  $P$ ;

输出:基于查询范围  $Q$  的道路网 Skyline 查询候选集  $resultSP$ .

- ① 以  $P$  中的数据点和道路网路段为生成点创建道路网 Voronoi 图;
- ②  $a[p_i] \leftarrow \emptyset$ ;
- ③ for  $i=0$  to  $n$  do
- ④ 计算  $p$  点的前  $n$  维非空间属性值的和;

- ⑤ 将结果按升序结果存入  $a[p_i]$  中;
- ⑥ end for
- ⑦  $resultSP \leftarrow \emptyset$ ;
- ⑧ 连接查询点组成最大的凸多边形查询区域  $CH(Q)$ ;
- ⑨ if  $p$  在  $CH(Q)$  内 then
- ⑩  $SR(p, Q) \leftarrow circle(p, q_1) \cup \dots \cup circle(p, q_m)$ ; /\* 以  $q_i$  为圆心,  $p, q_i$  之间的距离为半径做圆,  $m$  为查询点的数量 \*/
- ⑪  $BR = BR \cap MBR(SR(p, Q))$ ;
- /\* 获得内切矩形 \*/
- ⑫ if  $p$  在  $BR$  内 then
- ⑬  $resultSP \leftarrow resultSP + p$ ;
- /\* 定理 1 \*/
- ⑭ end if
- ⑮ end if
- ⑯ for  $j=1$  to  $m$  do
- ⑰ if  $q_j$  与  $V_{NVP}(p)$  相交 then
- ⑱ if  $p$  的路段与  $p$  所在的路段集合  $R$  相交 then
- ⑲  $resultSP \leftarrow resultSP + p$ ;
- /\* 定理 2 \*/
- ⑳ end if
- ㉑ else if 不相交 then
- ㉒ 找出查询区域  $CH(Q)$  与道路网集合  $R$  的相交的路段,记为  $r$ ;
- ㉓ if  $p$  在路段  $r$  上 then
- ㉔  $resultSP \leftarrow resultSP + p$ ;
- /\* 满足定理 3, 加入候选集中 \*/
- ㉕ end if
- ㉖ end if
- ㉗ end if
- ㉘ end for
- ㉙ for  $i=0$  to  $|resultSP|$  do
- ㉚ if  $a[p] > a[p_i]$  then
- ㉛  $resultSP \leftarrow resultSP - p$ ;
- ㉜ else if  $a[p] \leq a[p_i]$  then
- ㉝  $resultSP \leftarrow resultSP - p_i$ ;
- ㉞ end if
- ㉟ end for
- ㊱ return  $resultSP$ .

约减数据集由算法 1 实现.算法 1 首先对数据集建立道路网 Voronoi 图并对数据点的非空间属性值进行计算,总和的值越小表示数据点的非空间

属性的支配能力越强.再对查询点建立查询区域  $CH(Q)$ ,根据道路网 Voronoi 图与查询区域  $CH(Q)$  的位置关系,由定理 1 判断数据点是否在查询区域  $CH(Q)$  内,如果数据点在查询区域内,则直接将该数据点加入候选集中.再进行判断与查询区域有交集的道路网 Voronoi 单元的数据点,根据定理 2 将这些数据点加入候选集中,这些点在空间上支配那些所在的道路网 Voronoi 单元不与查询区域相交的数据点.再判断数据点所在的道路网路段与查询区域是否相交.根据定理 3 将道路网上相交的数据点加入候选集中,将不符合条件的数据点约减,减少计算大量非候选数据点.根据道路网 Voronoi 图查询区域  $CH(Q)$  与道路网集  $R$  路段的交点,将每一个交点所在路段上的数据点加入到候选集,再将所有的候选点进行合并.最后对  $resultSP$  集合中数据点的非空间属性值进行支配能力检查,值越小的数据点支配另一个数据点.在道路网环境中用数据点的非空间属性值的总和代表数据点的非空间属性的支配能力,总体的支配能力可以初步约减掉部分非空间属性值较大的情况,约减非空间属性支配能力差的数据点,获得最终的候选集.

## 2.2 精炼候选集

精炼候选集过程的主要处理对象是在约减过程中所得到的候选集  $resultSP$ .本节首先根据数据点  $K$ -支配的定义及支配关系给出定理 4 和定理 5,再根据定理对候选集中的数据点进行非空间属性支配判断,若满足条件则加入候选集中,从而构成精炼候选集.本文中数据点的非空间属性是由数据点的前  $n$  维随机生成的,非空间属性值越小越好.

**定理 4**<sup>[24]</sup>. 对于任意 2 个数据点  $p_i$  和  $p_j$ ,若数据点  $p_i$  在  $k$  维非空间属性上支配数据点  $p_j$ ,则数据点  $p_i$  必定  $k-1$  维支配数据点  $p_j$ .

本节将数据点的非空间维数属性值的范围定为相同的,并且属性值越小它的优势越大.数据点在  $k$  维上的最好和最差的支配能力分别为

$$bestK(p_i) = \sum_{i=1}^{k-1} \min(p, i), \quad (2)$$

$$worstK(p_i) = \sum_{i=1}^{k-1} \max(p, i). \quad (3)$$

**定理 5**<sup>[24]</sup>. 如果对于数据集  $d$  维中的任意 2 个数据点  $p_1$  和  $p_2$ ,若在  $k$  维上存在  $worstK(p_1) \geq bestK(p_2)$ ,则数据点  $p_2$  一定不在  $k-1$  维非空间属性上支配数据点  $p_1$ .

可以将候选点与各查询点间的距离总和作为空

间属性的值,如果数据点在非空间和空间属性上均不被其他数据点支配,则将该点加入到精炼集合中.将数据点与查询点的空间距离总和作为空间属性与非空间属性进行  $K$ -支配判断,得到精炼集合  $Sk$ .由于数据点的空间属性同样需要检查  $K$ -支配,影响  $K$ -支配查询的效率,因此在道路网环境中用数据点与各查询点的空间距离总和作为空间属性代表数据点空间属性的支配能力,总体的支配能力可以约减掉部分空间属性值较大的情况,便于精炼数据点.

我们进一步给出道路网数据点  $K$ -支配的  $I$ -PNS 算法如算法 2 所示.

**算法 2.**  $I$ -PNS( $resultSP, Q, t$ ).

输入:候选集  $resultSP$ 、查询集合  $Q$ 、用户感兴趣的非空间属性值个数  $t$ ;

输出: $resultSP$  中的  $K$ -支配 Skyline 数据精炼候选集  $Sk$ .

①  $Sk \leftarrow \emptyset$ ;

② for every point  $p \in resultSP$  do

③ 计算点  $p$  最好的支配能力值  $bestK(p)$  和最差的支配能力值  $worstK(p)$ ;

④  $p$  初始化标记为  $p.isDominat = false$ ;

⑤ 对  $resultSP$  中的数据点  $p$  的最好的支配能力值  $bestK(p)$  进行升序排序;

⑥ 将  $bestK(p)$  升序排序结果存储到数组  $AI[1, 2, \dots, t]$  中; /\* 数组  $AI[1, 2, \dots, t]$  构成  $A$  表 \*/

⑦ 对  $resultSP$  中的数据点  $p$  最差的支配能力值  $worstK(p)$  进行升序排序;

⑧ 将  $worstK(p)$  升序排序结果存储到数组  $BI[1, 2, \dots, t]$  中; /\* 数组  $BI[1, 2, \dots, t]$  构成  $B$  表 \*/

⑨ end for

⑩ for  $i=1$  to  $t$  do /\*  $t$  是数组  $A$  和数组  $B$  中数据点的个数 \*/

⑪  $q_i \leftarrow AI[i]$ ; /\*  $p, q$  是数组中的点 \*/

⑫ if  $p.isDominat = false$  then

⑬  $p_i \leftarrow BI[i]$ ;

⑭ if  $worstK(p_i) < bestK(q_i)$  then

⑮  $q_i < p_i$ ;

⑯  $Sk \leftarrow Sk + q_i$ ;

⑰ else if  $worstK(p_i) \geq bestK(q_i)$  then

⑱  $Sk \leftarrow Sk + p_i$ ;

⑲ end if

⑳ end if

```

① end for
② for  $i=0$  to  $|Sk|$  do
③   for  $j=0$  to  $m$  do
④     count  $d(p_i, q_j)$ ;
⑤      $d[p_i]=d(p_i, q_j)$ ;
⑥   end for
⑦   if  $d[p_i]>d[p_i+1]$  then
⑧      $Sk \leftarrow Sk - p_i$ ;
⑨   else if  $d[p_i] \leq d[p_i+1]$  then
⑩      $Sk \leftarrow Sk - p_{i+1}$ ;
⑪   end if
⑫ end for
⑬ return  $Sk$ .
    
```

算法 2 首先建立 2 个索引表来存储数据点的  $k$  维的最好和最坏支配能力值:  $AI$  表和  $BI$  表, 根据对算法 1 中所得的候选集优化精炼的基本思想, 首先要对候选集中的数据点计算其最好支配能力值和最差支配能力值, 将结果按升序存储在表中, 并对每一个数据点初始标记为未被支配. 再根据定理 4 和 5 对表  $AI$  和  $BI$  中的数据点进行比较. 如果  $worstK(p_i) < bestK(q_i)$ , 则点  $q_i$  支配点  $p_i$ , 将点  $q_i$  加入精炼候选集  $Sk$  中. 再计算数据点空间支配的影响, 被支配的数据点从候选集中删除, 剩余的数据点组成最后的精炼集合, 将数据点与各查询点的空间距离总和作为空间属性与非空间属性进行  $K$ -支配判断, 最后得到精炼集合  $Sk$ .

### 2.3 查询算法

本节根据道路网  $K$ -支配空间 Skyline 查询的不同情况分别调用道路网空间 Skyline 情况下的  $R$ - $PNS$  算法和  $K$ -支配空间 Skyline 查询情况下的  $I$ - $PNS$  算法进行查询.

**定理 6.** 确定数据点的支配判定圆区域的范围, 数据点  $p$  与查询点  $q_i$  之间通过道路网相互连接, 当数据点  $p \in Sk$  时, 则以查询点  $q_i$  为圆心、查询点  $q_i$  与数据点  $p$  间的网络距离  $d(q_i, p)$  为半径做圆, 若在多个圆域的相交区域中存在数据点  $p_k$ , 则数据点  $p_k$  空间上支配数据点  $p$ , 则将数据点  $p$  剪枝.

证明. 当数据点  $p \in Sk$  时, 以查询点  $q_i$  为圆心、以  $q_i$  与数据点  $p$  之间的网络距离  $d(q_i, p)$  为半径作圆, 若在多个圆域的相交区域内存在数据点  $p_k$ , 则对于查询点  $q_i$ , 一定存在  $d(q_i, p_k) < d(q_i, p)$ , 根据空间数据点支配定义可知, 在空间上数据点  $p_k$  支配数据点  $p$ , 则数据点  $p$  就不是查询点  $q$  的空间 Skyline 点, 故将数据点  $p$  删除. 证毕.

图 3 中, 以  $q_i$  点为中心、以候选集内的任意一数据点  $p$  之间的网络距离为半径做圆, 圆弧线  $d_1, d_2, d_3$  表示数据点相对不同查询点的支配圆域. 若该圆域相关的相交区域内存在其他数据点  $p'$ , 则相交区域内的数据点  $p'$  支配数据点  $p$ . 根据定理 2 和 3,  $\{p_1, p_2, p_4, p'\}$  可加入候选集. 以  $\{q_1, q_2, q_3\}$  为圆心、以各候选集内的数据点的网络距离为半径做圆 (如图 3 所示), 点  $p'$  即处于这些圆域的相交区域中, 根据定理 6 可知  $p_2$  支配点  $p'$ . 由此可知, 数据点中不被空间支配的查询结果为  $Sk = \{p_4, p'\}$ .

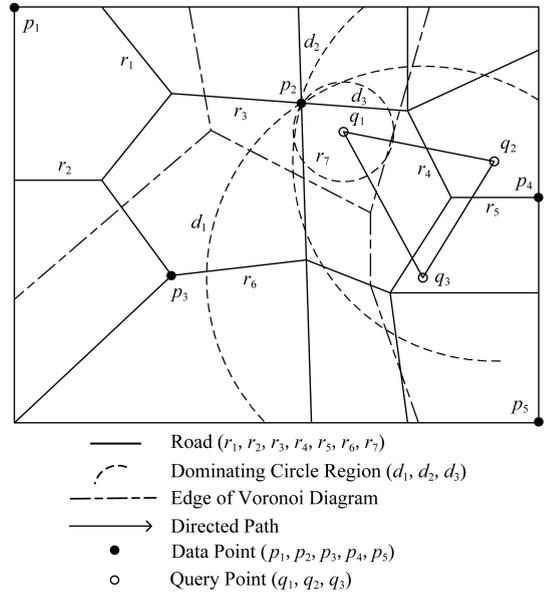


Fig. 3 Example based on theorem 6

图 3 基于定理 6 的示例

本节查询算法的主要思想为: 首先调用算法 1 得到初步约减数据集后的候选集, 再调用算法 2 对候选集进行非空间属性  $K$ -支配精炼处理, 获得更精确的候选集. 然后分别求出候选集中的点  $p$  到  $q_i$  的网络距离  $d(p, q_i)$ . 若满足  $d(q_i, p_1) < d(q_i, p)$  时, 候选点  $p$  被数据点  $p_1$  支配, 则从候选集中删除数据点  $p$ . 最后剩下的候选点就构成查询结果集.

本节进一步给出  $K$ - $PNS$  查询算法, 如算法 3 所示.

**算法 3.**  $K$ - $PNS(Q, k, P, R)$ .

输入: 查询集  $Q$ 、支配维数  $k$ 、数据集  $P$ 、道路网路段集  $R$ ;

输出: 道路网  $K$ -支配情况下的空间 Skyline 查询的结果集  $Sp$ .

- ①  $resultSp \leftarrow \emptyset, Sk \leftarrow \emptyset, Sp \leftarrow \emptyset$ ;
- ②  $resultSp \leftarrow call R-PNS(Q, P, R)$ ; / \* 调用算法 1, 计算约减候选集合 \* /

- ③  $S_k \leftarrow \text{call } I\text{-PNS}(\text{resultSp}, Q, k)$ ; /\* 调用算法 2, 计算精炼候选集合 \*/
- ④ if  $p$  在  $S_k$  中 then
- ⑤ for  $j=1$  to  $m$  do  
/\*  $m$  是查询点的个数 \*/
- ⑥  $C \leftarrow \text{Calculate\_Circle}(q_j, p_i)$ ; /\* 计算支配判定圆的范围 \*/
- ⑦ if  $p$  inside  $S_k$  并且  $p \cap C \neq \emptyset$  then
- ⑧  $S_p \leftarrow S_p - p$ ;
- ⑨ else if  $p$  不在其他点的支配圆域内 then
- ⑩  $S_p \leftarrow S_p + p$ ;
- ⑪ end if
- ⑫ end for
- ⑬ end if
- ⑭ return  $S_p$ .

算法 3 首先调用算法 1 求出 PNS 查询结果;在考虑数据点的  $k$  维属性时,则调用算法 2 得到精炼集.对精炼集中的数据点再进行支配检查,对数据点与查询点做判定圆区域检查,在区域内若无其他数据点则加入  $S_p$  集,否则删除.最后算法对  $S_p$  集中的点进行  $k$  维检查,不满足条件的数据点从  $S_p$  集中删除,最终留下的数据点就是道路网  $K$ -支配空间 Skyline 查询的结果集.

### 3 实验比较与分析

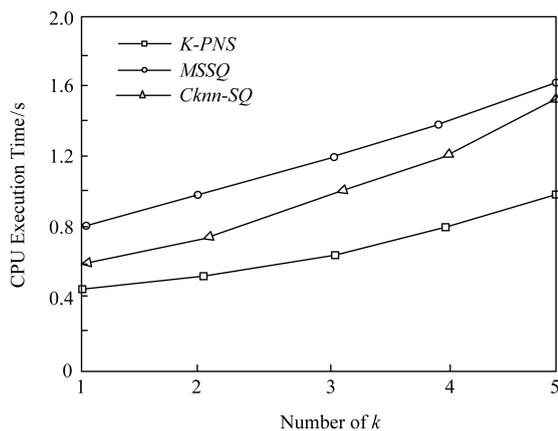
本节通过实验对算法进行性能评估.由于道路网中  $K$ -支配空间 Skyline 查询问题是本文研究的一个新问题,已有研究成果无法直接处理道路网中  $K$ -支配空间 Skyline 查询问题,因此,为了构造对比算法,我们对文献[26-27]的曼哈顿道路网中空间 Skyline 查询的 MSSQ 算法和道路网基于距离的连续 Skyline 查询的  $Cknn$ -SQ 算法进行了适当的修改,针对给定查询问题采用反复调用 MSSQ 算法和  $Cknn$ -SQ 算法的策略,从而逐步得到数据点  $k$  维属性上的 Skyline 查询结果,从而与本文提出的  $K$ -PNS 算法形成对比算法.得到的对比算法在本文称为 MSSQ 算法和  $Cknn$ -SQ 算法.MSSQ 算法分别对数据点集  $P$  和查询点集  $Q$  在一维上建立平衡二叉树,在其他维度上保存数据; $Cknn$ -SQ 查询算法从 Skyline 结果集中向用户返回  $s$  个 Skyline 点,这  $s$  个 Skyline 点在距离属性上要优于剩下的数据点.

本文使用的道路网数据集是来自美国加利福尼亚州空间信息网络,该网络中的数据集由地理信息

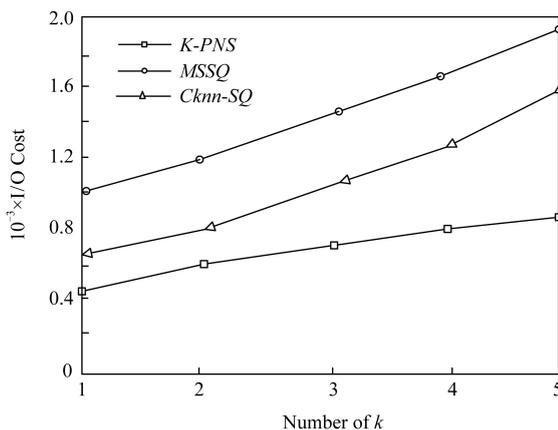
组成.我们从该数据集中抽取 10 000 个节点对象和 5 000 个线段对象(线段作为道路网线段).为便于实验,对数据对象信息进行了适当调整和改动.查询集是由数据集中随机抽取的 100 个点的集合,数据点的非空间属性以数值来表示,数值在  $[0, 20]$  上随机产生.

实验平台配置为:4 GB 内存、2.0 GHz CPU、500 GB 硬盘,Windows 10 操作系统上用 Microsoft Visual Studio 2013 实现.实验需要验证的是 3 个算法的查询性能,测试的指标是 CPU 时间和 I/O 执行次数,两者反映各个算法的查询性能.实验测试中采用的合成数据为 Skyline 查询的测试数据集,测试过程中取执行 30 次查询的平均值作为结果.实验分别测试非空间属性维数  $k$  值、数据点数量和道路网路段数量对 CPU 执行时间和 I/O 执行效率的影响.

如图 4 所示,实验测试了  $k$  值对算法的影响.在数据点的数量是 10 000、查询点数量为 100、道路



(a) Effect of  $k$  on CPU execution time



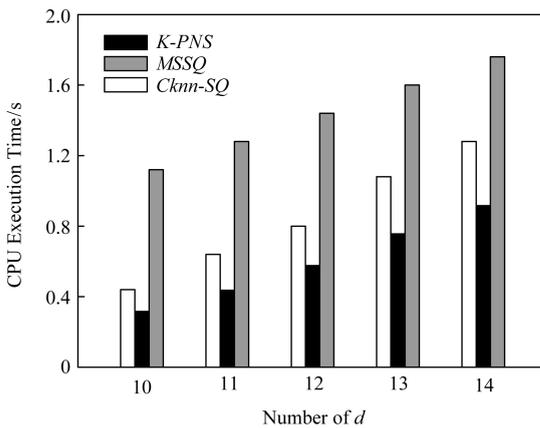
(b) Effect of  $k$  on I/O cost

Fig. 4 Effect of  $k$  value on the efficiency of the query algorithms

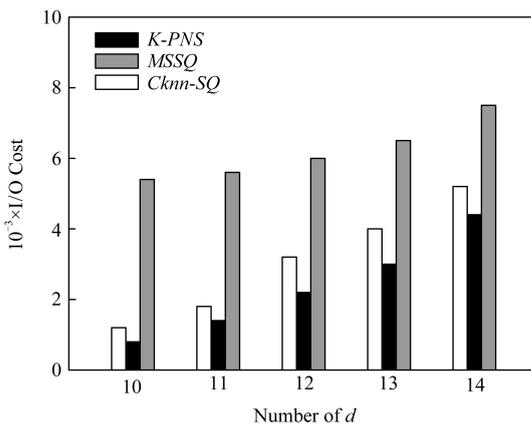
图 4  $k$  值对查询算法效率的影响

网线段数量为 1 000、数据点维数  $d = 10$ 、数据集的非空间属性维数  $k$  从 1 增加到 5 的情况下测试了 3 种查询算法的查询性能.从图 4(a)中可以看出,  $K$ - $PNS$  算法、 $MSSQ$  算法和  $Cknn$ - $SQ$  算法的 CPU 执行时间都是随着  $k$  值的增大而呈上升趋势,  $K$ - $PNS$  算法和  $Cknn$ - $SQ$  算法的查询效率均优于  $MSSQ$  算法.但是  $Cknn$ - $SQ$  算法上升趋势比  $K$ - $PNS$  算法更显著, 原因在于  $Cknn$ - $SQ$  算法需要重新调用算法对视作数据点的道路网线段重新处理.图 4(b)展示了  $k$  值增加对  $K$ - $PNS$  算法、 $MSSQ$  算法和  $Cknn$ - $SQ$  算法的 I/O 代价的影响情况.由图 4(b)可以看到随着  $k$  值的增加, 3 个算法的 I/O 代价都逐渐增大, 但  $K$ - $PNS$  算法随着  $k$  值的增长速率先增大后减小, I/O 代价的增幅最小.这是因为在数据点的非空间属性维数较小时, 数据点的非空间属性的相互支配较多, 所以 I/O 代价较高.

图 5 测试的是关于数据集维数变化对算法查询



(a) Effect of dimension  $d$  on CPU execution time



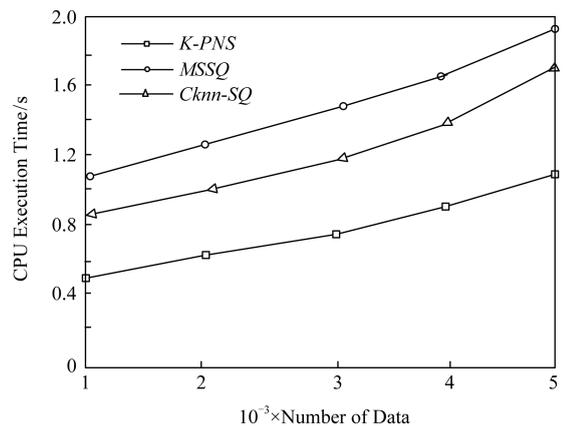
(b) Effect of dimension  $d$  on I/O cost

Fig. 5 Effect of dimension of data set on query efficiency

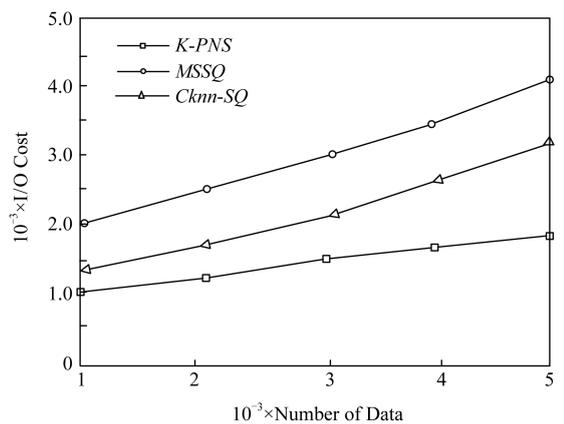
图 5 数据集维数对算法查询效率的影响

效率的影响.在数据点的数量是 10 000、查询点数量为 100、道路网线段数量为 1 000、 $k = 3$  的情况下测试了 3 种查询算法的查询性能.图 5 表示的是数据集维数  $d$  对算法的查询性能的影响.  $MSSQ$  算法采用平衡二叉树处理数据点间的道路网距离, 随着数据集维数的增大, 算法增加的趋势并不快, 但是算法原本需要花费较多的时间和 I/O 代价处理.  $Cknn$ - $SQ$  算法采用网格索引对道路网和道路网上的数据点进行索引, 维数的增多和距离增大对算法的查询性能有所降低.  $K$ - $PNS$  算法在约减数据集的过程中, 对空间属性和非空间属性分别进行了处理, 因此随着维数的增加, 算法的查询时间和 I/O 代价变化较小. 在数据集维数增加的情况下,  $K$ - $PNS$  算法和  $Cknn$ - $SQ$  算法优于  $MSSQ$  算法.

图 6 测试数据点数量对算法查询性能的影响. 设道路网线段数量为 1000、查询点为 100、数据点维数  $d = 10$ 、 $k = 3$ . 由图 6(a)可知,  $K$ - $PNS$  算法和  $Cknn$ - $SQ$  算法的时间耗费相对较少,  $MSSQ$  算法时



(a) Effect of data size on CPU execution time



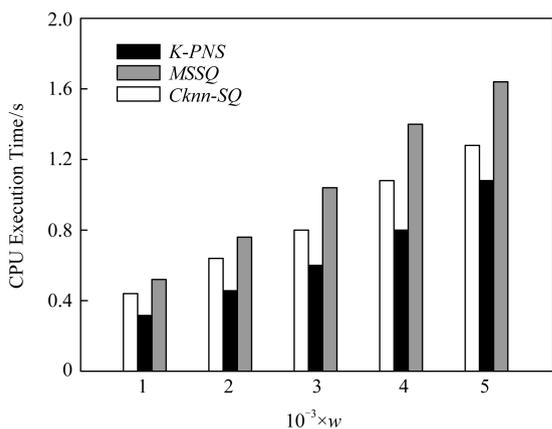
(b) Effect of data size on I/O cost

Fig. 6 Effect of data size on query performance

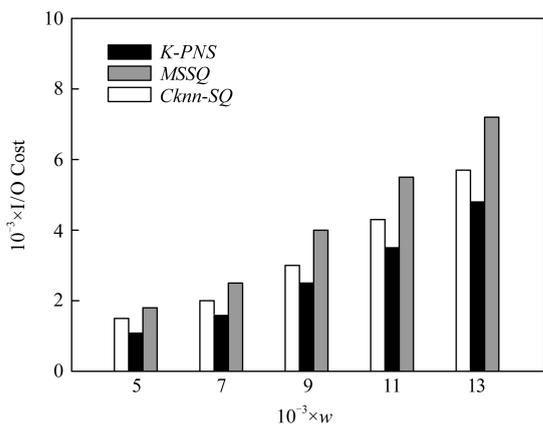
图 6 数据点数量对查询性能的影响

间耗费较多.因为  $MSSQ$  算法需要对数据点进行重复调用来计算数据点  $k$  维非空间属性,在数据点增加的过程中反复调用算法花费时间较多,而  $K-PNS$  算法已经对数据点的非空间属性进行了处理,所以随着数据点的增加,CPU 执行时间增加较小.从图 6(b)中可以看出,随着数据点的增加,3 个算法的 I/O 代价都增加,本文所提的  $K-PNS$  算法比  $Cknn-SQ$  算法和  $MSSQ$  算法的 CPU 时间和 I/O 开销都要小.随着道路条数的增多,道路连通性更加复杂, $Cknn-SQ$  算法计算全局 Skyline 点过程中,对道路网的访问以及对象之间的距离计算的复杂度都会增大.而  $K-PNS$  算法由于只需要维护固定对象序列的相邻对象点的距离关系,虽然其开销也会随着道路条数的增多而增大,但较  $MSSQ$  算法和  $Cknn-SQ$  算法的复杂度降低.

图 7 测试道路网线段对算法的查询性能的影响.设数据点数量为 10 000、查询点数量为 100、数据



(a) Effect of road network line segment number on CPU execution time



(b) Effect of road network line segment number on I/O cost

Fig. 7 Effect of road network line segment number on query performance

图 7 道路网线段数量对查询性能影响

点集维数  $d=10$ 、 $k=3$ .图 7 表示的是道路网线段数量对算法的查询性能的影响, $w$  表示道路网线段数量. $MSSQ$  算法采用平衡二叉树处理数据点间的道路网距离,随着道路网线段数量的增大,算法需要花费更多的时间和 I/O 代价处理,而  $Cknn-SQ$  算法采用网格索引对道路网和道路网上的数据点进行索引,随着道路网线段增多,数据点与查询点之间的距离关系会发生变化,算法的查询性能减慢.但是  $K-PNS$  算法中在约减数据集的过程中,对不被支配路段的点及其路段进行了约减,所以道路网路段的增加对 CPU 执行时间和 I/O 代价增长的趋势不大.因此在道路网线段增加的情况下, $K-PNS$  算法优于  $MSSQ$  和  $Cknn-SQ$  算法.

## 4 结束语

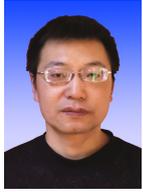
本文对道路网  $K$ -支配空间 Skyline 查询问题进行了系统研究.形式化定义了道路网环境下  $K$ -支配空间 Skyline 查询,提出了有效地处理道路网  $K$ -支配空间 Skyline 查询的算法,较大程度地减少了处理数据点和路段数量的时间,通过  $K$ -支配检查可得到精准的结果集.理论研究和实验表明,所提出的算法具有较好的性能.未来的研究重点主要集中在将障碍空间中的组反  $k$  最近邻查询技术<sup>[28]</sup>应用到障碍物的空间 Skyline 查询领域以解决复杂环境条件下的空间 Skyline 查询问题.

## 参 考 文 献

- [1] Borzsony S, Kossmann D, Stocker K. The Skyline operator [C] //Proc of the 17th Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2002: 421-430
- [2] Ahmed K, Nafi N, Gregory M. Enhanced distributed dynamic Skyline query for wireless sensor networks [J]. Journal of Sensor & Actuator Networks, 2016, 5(1): 97-110
- [3] Zhou Xu, Li Kenli, Zhou Yantao, et al. Adaptive processing for distributed Skyline queries over uncertain data [J]. IEEE Transactions on Knowledge & Data Engineering, 2016, 28(2): 371-384
- [4] Cui Ningning, Yang Xiaochun, Wang Bin. Research on authentication of moving  $K$ -dominant NN queries [J]. Chinese Journal of Computers, 2018, 41(8): 1780-1797 (in Chinese)  
(崔宁宁, 杨晓春, 王斌. 移动  $K$ -支配最近邻查询验证研究 [J]. 计算机学报, 2018, 41(8): 1780-1797)

- [5] Banaei-Kashani F, Ghaemi P, Movaqar B, et al. Efficient maximal reverse Skyline query processing [J]. *Geoinformatica*, 2017, 21(3): 549-572
- [6] Qi Yafei, Wang Yijie, Li Xiaoyong. A Skyline query method over Gaussian model uncertain data streams [J]. *Journal of Computer Research and Development*, 2012, 49(7): 1467-1473 (in Chinese)  
(祁亚斐, 王意洁, 李小勇. 基于高斯模型的不确定数据流 Skyline 查询方法 [J]. *计算机研究与发展*, 2012, 49(7): 1467-1473)
- [7] Wang Xiaoyi, Qin Xiaolin, Wang Ning, et al. Efficient algorithm for multiple subspace Skyline queries processing [J]. *Journal of Frontiers of Computer Science & Technology*, 2016, 10(5): 623-634
- [8] Zheng Jiping, Chen Jialiang, Wang Haixiang. Efficient geometric pruning strategies for continuous Skyline queries [J]. *ISPRS International Journal of Geo-Information*, 2017, 6(3): 91-117
- [9] Park Y, Min J K, Shim K. Processing of probabilistic Skyline queries using MapReduce [J]. *VLDB Endowment*, 2015, 8(12): 1406-1417
- [10] Yang Linqing, Li Zhan, Mou Yanchao, et al. Algorithm of parallel Top- $k$  Skyline queries for large data set [J]. *Journal of Frontiers of Computer Science & Technology*, 2015, 9(8): 182-193
- [11] Ding Linlin, Cui Ziqiang, Yin Xiankun, et al. Augmented dynamic Skyline query processing method based on MapReduce [J]. *Acta Electronica Sinica*, 2018, 46(5): 1062-1070 (in Chinese)  
(丁琳琳, 崔子强, 尹显坤, 等. 基于 MapReduce 的增广动态 Skyline 查询处理方法 [J]. *电子学报*, 2018, 46(5): 1062-1070)
- [12] Li Song, Li Shuang, Zhang Liping, et al. Spatial Skyline query method based on  $R^+$  tree for obstructed spaces [J]. *Journal of Frontiers of Computer Science and Technology*, 2017, 11(12): 1886-1896  
(李松, 李爽, 张丽平, 等. 障碍空间中基于  $R^+$  树的空间 Skyline 查询方法 [J]. *计算机科学与探索*, 2017, 11(12): 1886-1896)
- [13] Pan Xiao, Chen Weizhang, Sun Yige, et al. Continuous queries privacy algorithm based over road networks [J]. *Journal of Computer Research and Development*, 2017, 54(9): 2092-2101 (in Chinese)  
(潘晓, 谌伟璋, 孙一格, 等. 道路网络上基于时空相似性的连续查询隐私保护算法 [J]. *计算机研究与发展*, 2017, 54(9): 2092-2101)
- [14] Bao Jinling, Liu Kingshan, Zhou Rui, et al. Keyword-aware optimal location query in road network [C] //Proc of the 17th Int Conf on Web-Age Information Management. Berlin: Springer, 2016: 164-177
- [15] Deng Ke, Zhou Xiaofang, Shen Hengtao. Multi-source Skyline query processing in road networks [C] //Proc of the 23rd Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2007: 796-805
- [16] Huang Y K. Within Skyline query processing in dynamic road networks [J]. *ISPRS International Journal of Geo-Information*, 2017, 6(5): 137-152
- [17] Miao Xiaoye, Gao Yunjun, Guo Su, et al. On efficiently answering why-not range-based Skyline queries in road networks [J]. *IEEE Transactions on Knowledge & Data Engineering*, 2018, 30(9): 1697-1711
- [18] Fu Xiaoyi, Miao Xiaoye, Xu Jianliang, et al. Continuous range-based Skyline queries in road networks [J]. *World Wide Web-Internet & Web Information Systems*, 2017, 20(6): 1443-1467
- [19] Jiang Shunqing, Zheng Jiping, Chen Jialiang, et al. Efficient computation of continuous range Skyline queries in road networks [C] //Proc of the 12th Int Conf on Intelligent Computing Methodologies. Berlin: Springer, 2016: 520-532
- [20] Lee G, Lee Y H. An efficient method of computing the  $k$ -dominant Skyline efficiently by partition value [C] //Proc of the 3rd Int Conf on Information Management. Piscataway, NJ: IEEE, 2017: 416-420
- [21] Miao Xiaoye, Gao Yunjun, Chen Gang, et al.  $K$ -dominant Skyline queries on incomplete data [J]. *Information Sciences*, 2016, 367/368: 990-1011
- [22] Awasthi A, Bhattacharya A, Gupta S, et al.  $K$ -dominant Skyline join queries: Extending the Join Paradigm to  $K$ -dominant Skylines [C] //Proc of the 33rd IEEE Int Conf on Data Engineering. Los Alamitos, CA: IEEE Computer Society, 2017: 99-112
- [23] Siddique M, Tian H, Morimoto Y.  $K$ -dominant Skyline query computation in MapReduce environment [J]. *IEICE Transactions on Information and Systems*, 2015, 98(5): 1027-1034
- [24] Yin Jian, Yao Shuyu, Xue Shaoe, et al. An index based efficient  $k$ -dominant Skyline algorithm [J]. *Chinese Journal of Computers*, 2010, 33(7): 1236-1245 (in Chinese)  
(印鉴, 姚树宇, 薛少锴, 等. 一种基于索引的高效  $k$ -支配 Skyline 算法 [J]. *计算机学报*, 2010, 33(7): 1236-1245)
- [25] Kolahdouzan M R, Shahabi C. Voronoi-based  $K$  nearest neighbor search for spatial network databases [C] //Proc of the 30th Int Conf on Very Large Data Bases. San Francisco, CA: Morgan Kaufmann, 2004: 840-851
- [26] Son W B, Hwang S W, Ahn H K. MSSQ: Manhattan spatial Skyline queries [J]. *Information Systems*, 2014, 40(3): 67-83
- [27] Huang Y K, Chang C H, Lee C. Continuous distance-based Skyline queries in road networks [J]. *Information Systems*, 2012, 37(7): 611-633

- [28] Zhang Liping, Liu Lei, Hao Xiaohong, et al. Voronoi-based group reverse  $k$  nearest neighbor query in obstruct space [J]. Journal of Computer Research and Development, 2017, 54 (4): 861-871 (in Chinese)  
(张丽平, 刘蕾, 郝晓红, 等. 障碍空间中基于 Voronoi 图的组反  $k$  最近邻查询研究[J]. 计算机研究与发展, 2017, 54 (4): 861-871)



**Li Song**, born in 1977. Professor in the College of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China. Member of CCF. His research interests mainly focus on the spatial and spatio-temporal database, data mining, information data security.



**Dou Yanan**, born in 1992. Master. Her research interests mainly focus on the spatial database.



**Hao Xiaohong**, born in 1969. Master. She is currently senior experimentalist in Computer Center, Harbin University of Science and Technology, Harbin, China. Her research interests mainly focus on the spatial database.



**Zhang Liping**, born in 1976. Master. Associate professor in the College of Computer Science and Technology, Harbin University of Science and Technology. Her research interests mainly focus on the spatial and spatio-temporal database, information data security.



**Hao Zhongxiao**, born in 1940. Professor and PhD supervisor in the Harbin Institute of Technology and Harbin University of Science and Technology. His main research interests include relational database, null database, acyclic database, active database, spatial database and temporal database.