

# 道路网多用户偏好 Top- $k$ 天际线查询方法

李 松 宾婷亮 郝晓红 张丽平 郝忠孝

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)

(lisongbeifen@163.com)

## Multi-User Preference Top- $k$ Skyline Query Method Based on Road Network

Li Song, Bin Tingliang, Hao Xiaohong, Zhang Liping, and Hao Zhongxiao

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080)

**Abstract** Existing Skyline query most focus on single-user scenarios, and calculate Skyline results based on single-user model. But less consideration is given to multi-user model in road network environment. The existing methods cannot solve the Top- $k$  Skyline query problem that comprehensively considers multi-user preference and weight in road network environment. Therefore, we propose a Top- $k$  Skyline query method MUP-TKS, based on multi-user preference in road network environment. In this environment, the different preference and weight of multi-user are considered for Skyline calculation. The result set which conforms to the preference and weight of the query user group can be obtained quickly to make better decision. Firstly, through the proposed algorithm G\_DBC, the position relation of data points and query points in the road network, and the new index structure Vor-R\*-DHash are used for pruning the data points. Thus the optimal distance set is obtained. Then taking advantage of the invariable property of the static Skyline set to precompute and save the set. KPRD algorithm is performed on  $S$  set, the union of the optimal distance set and static Skyline set. Finally, TK\_DC algorithm is used to score the candidate set. According to the score of the data points, the Top- $k$  of the sorted set are returned to the query user group. Theoretical studies and experiments show that the proposed method is efficient and reliable.

**Key words** Skyline query; Top- $k$  Skyline query; multi-user preference; road network; user weight

**摘要** 已有的天际线 (Skyline) 查询主要聚焦于单用户场景，并基于单用户模型进行 Skyline 计算，而较少考虑道路网环境下多用户情况。为了弥补已有方法无法解决道路网络环境下多用户偏好和权重 Top- $k$  Skyline 查询问题的不足，提出了一种基于道路网环境下多用户偏好 Top- $k$  Skyline (multi-user preference Top- $k$  Skyline, MUP-TKS) 查询方法。在道路网环境下考虑多用户的不同偏好和权重进行 Skyline 查询，可以快速得到符合查询用户群偏好和权重的结果集，提供用户群更好的决策支持。MUP-TKS 首先通过所提的 G\_DBC 算法，利用道路网中数据点与查询点之间的位置关系和新的索引结构 Vor-R\*-DHash 剪枝、过滤数据点，从而得到距离较优集；再利用静态 Skyline 集不变的性质，预先计算、保存该集合；然后通过所提的新支配关系对距离较优集与静态 Skyline 集取并集后的集合  $S$  进行放松支配；最后利用所提 TK\_DC 算法对经过放松支配后的候选结果集打分，依据数据点得分情况，排序输出 Top- $k$  个结果集返回用户群。理论研究与实验表明，所提方法具有较好的效率与可靠性。

收稿日期: 2022-06-09; 修回日期: 2022-12-09

基金项目: 国家自然科学基金项目 (61872105, 62072136); 黑龙江省自然科学基金项目 (LH2023F031); 国家重点研发计划项目 (2020YFB1710200)

This work was supported by the National Natural Science Foundation of China (61872105, 62072136), the Natural Science Foundation of Heilongjiang Province (LH2023F031), and the National Key Research and Development Program of China (2020YFB1710200).

关键词 Skyline 查询; Top- $k$  Skyline 查询; 多用户偏好; 道路网; 用户权重

中图法分类号 TP391

天际线(Skyline)查询<sup>[1]</sup>作为多目标决策、兴趣点发现、推荐系统等领域关键问题的一种解决途径,在2001年被提出,自此受到研究学者的广泛关注与研究。近些年, Skyline 查询研究拓展到不确定数据 Skyline 查询<sup>[2]</sup>、数据流 Skyline 查询<sup>[3]</sup>、动态 Skyline 查询<sup>[4]</sup>、反 Skyline 查询<sup>[5]</sup>、偏好 Skyline 查询等方面,其中偏好 Skyline 查询可以返回满足用户偏好需求的结果集。针对因用户偏好不同导致属性的重要性不同问题,研究者们提出了新的支配关系与算法。但已有研究主要集中在非道路网的用户偏好 Skyline 查询或者道路网单用户偏好 Skyline 查询方面,没有考虑道路网多用户偏好和权重的 Top- $k$  Skyline 查询。

传统偏好 Skyline 查询算法主要存在3点局限性:1)偏好 Skyline 查询需要确定属性的重要性程度,由于不同用户权重与偏好不同,因此不同属性的重要性程度也不一致,而已有研究中较少有提出将用户偏好和权重综合考虑,得到对用户群统一的属性重要程度次序处理方法;2)传统偏好 Skyline 查询算法大多未考虑道路网环境下的距离维度,只考虑静态维度;3)传统偏好 Skyline 查询算法返回的结果集过大、无序,不能给用户提供有效的决策支持。

因此,针对道路网多用户偏好 Top- $k$  Skyline 查询问题,本文提出满足多用户不同权重和偏好需求的查询方法。

本文的主要贡献有3点:

1)针对道路网存在大量数据点以及多查询用户场景,需要计算数据点到各个查询用户的道路网距离,从而产生的很大距离计算开销,为了提升距离计算效率,本文根据所提的 Vor-R\*-DHash 索引结构以及数据点与查询用户群的空间位置关系,提前剪枝在距离维度被支配的大量数据点。

2)针对在道路网 Top- $k$  Skyline 查询处理时未综合考虑多用户不同权重和偏好以及返回的结果集数量不可控的问题,本文首先提出整体属性权重值的概念,综合考虑用户权重和偏好;并进一步提出用户群权重偏好次序,并基于此次序提出一种新的支配,即  $K$ -准放松支配;接着根据偏好次序进行逐次放松支配,使返回结果集大小可控;同时当  $k$  值改变时,动态调整放松轮次即可获取候选结果集  $CS$ ,而无需重新计算距离、偏好次序等,减少了查询计算开销。

3)针对 Skyline 查询返回结果集无序的问题,本

文基于  $z$ -整体属性权重值,提出了选取 Top- $k$  个结果集的打分函数,对候选结果集  $CS$  打分排序,返回有序结果集。

## 1 相关工作

Skyline 查询主要分为集中式查询和分布式查询。其中集中式查询主要分为使用索引结构和不使用索引结构。使用索引结构的算法常用 R-tree 等索引结构,例如文献 [6] 利用最近邻(nearest neighbor, NN)算法和 R-tree 索引查找 Skyline 点,基于 R-tree 可以快速判断数据点是否为 Skyline 点,接着利用数据点进行子集合的划分,递归查找 Skyline 点。不使用索引结构的 Skyline 查询算法主要有基于排序的 SFS(sort-filter Skyline)算法<sup>[7]</sup>。而 Skyline 查询在不断发展过程中又产生了许多变种问题,例如  $K$ -支配空间 Skyline 查询<sup>[8]</sup>、连续 Skyline 查询<sup>[9]</sup>、针对推荐系统的范围障碍空间连续 Skyline 查询<sup>[10]</sup>、概率 Skyline 查询<sup>[11]</sup>以及 Top- $k$  Skyline 查询等<sup>[12-13]</sup>。

在集中式计算环境下,文献 [14] 根据用户不同偏好提出了维度不确定的定义,根据维度特征划分数据,进行 Skyline 概率支配测试,同时利用阈值处理大规模数据集 Skyline 查询问题。文献 [15] 提出一种高效偏序域 Skyline 查询处理方法,利用倒排索引进行 Skyline 查询。在并行计算环境下,文献 [16] 提出了不完全数据集的偏好 Skyline 查询算法 SPQ(Skyline preference query)。文献 [17] 根据用户的偏好,基于 Voronoi 图将数据对象划分到不同网格中,并行计算所有对象组合,获取动态 Skyline 结果。文献 [18] 提出了 MapReduce 下 Top- $k$  Skyline 偏好查询。

道路网 Skyline 查询近年来也受到越来越多的关注。道路网 Skyline 查询既考虑数据点的路网空间属性,又考虑非空间属性。文献 [19] 提出了基于范围的移动对象连续 Skyline 查询处理方法,利用 Voronoi 图组织道路网中的数据点,通过所提的 3 种算法减少道路网产生的相交节点数和距离计算开销。文献 [20] 提出了道路网环境下综合考虑空间距离和社交距离的 Skyline 组用户查询方法。

Top- $k$  Skyline 查询在多目标决策中往往更具优势,因为它可以控制返回的结果集数量。文献 [21] 提出基于安全区域技术解决连续 Top- $k$  Skyline 查询结

果更新问题,提出了结合 Top- $k$  查询和 Skyline 查询的安全区域构建算法.文献[22]提出了MapReduce 环境下 Top- $k$  Skyline 处理方法.文献[23]将  $K$ -Skyband 查询与 Top- $k$  Skyline 查询结合处理大数据集的 Top- $k$  Skyline 查询.

目前道路网环境下 Top- $k$  Skyline 查询研究大多集中在单用户场景,较少考虑多用户偏好和权重不同的场景.针对已有方法的不足,本文利用查询点与数据点的位置关系剪枝数据集,利用所提的  $K$ -准放松支配控制结果集数量;利用所提的打分函数返回有序结果集,在理论论证和分析基础上提出了道路网多用户偏好 Top- $k$  Skyline 查询方法.

## 2 主要定义

设道路网环境下数据集  $P=\{p_1, p_2, \dots, p_n\}$ , 查询用户群  $G=\{q_1, q_2, \dots, q_m\}$ .

**定义 1.** 道路网距离支配.给定查询用户群  $G$ 、数据点  $p_1$ 、数据点  $p_2$ , 数据点之间的距离为  $Dist$ , 当且仅当  $Dist(p_1, q_i) \leq Dist(p_2, q_i)$ ,  $1 \leq i \leq m$ ; 且存在  $Dist(p_1, q_i) < Dist(p_2, q_i)$ ,  $1 \leq i \leq m$ , 称  $p_1$  道路网距离支配  $p_2$ , 记作  $p_1 \blacktriangleright p_2$ . 本文距离如不特殊说明,则为道路网距离.

**定义 2.** 整体属性权重.给定查询用户群  $G$ , 用户权重  $w=\{w_1, w_2, \dots, w_m\}$ , 用户  $q_i$  的查询关键字  $keys=\{C_1, C_2\}$ ,  $C_1$  为优先考虑的属性集合,  $C_2$  为一般偏好的属性集合, 任意维度  $d_j$  的整体属性权重  $W_j$  如式(1):

$$W_j = \sum_{i=1}^m s_i \cdot w_i, \quad (1)$$

其中  $s_i$  代表属性  $d_j$  对于用户  $q_i$  的重要性得分.

在属性的重要性程度计分时,将属性偏好分为 3 类:优先考虑、一般偏好和未考虑.不同类别分数不同,例如  $C_1$  中的属性被赋予 2 分,  $C_2$  中的属性被赋予 1 分,未考虑的属性被赋予 0 分.

**定义 3.** 用户群权重偏好次序.指针对查询用户群属性的有序集合  $GP=\{d_1, d_2, \dots, d_r\}$ , 其中  $d_i$  代表任意属性,  $GP$  中属性对用户群的重要性程度呈非递增排列.用户群权重偏好次序综合考虑用户的偏好和权重.

**定义 4.**  $K$ -准放松支配(KPRD).设  $P$  为数据集,数据维度空间为  $D$ ,  $d_j$  为任意维度, 总维度数为  $r$ ,  $\theta=(\theta_1, \theta_2, \dots, \theta_K)$  是  $D$  上  $K$  个维度的无差异阈值.数据点  $p_i, p_j \in P, p_i$   $K$ -准放松支配  $p_j$ , 记作  $p_i \mathcal{C} p_j$ , 当且仅当:

$$\begin{cases} |\{j|p_i[d_j] - p_j[d_j] > \theta_j\}| = 0, \\ |\{j|p_i[d_j] - p_j[d_j] > 0\}| < |\{j|p_i[d_j] - p_j[d_j] > 0\}|, \end{cases} \quad (2)$$

其中  $1 \leq j \leq K$ .

**定义 5.** 道路网多用户偏好 Top- $k$  Skyline 查询.给定道路网路段集  $R$ 、查询用户群  $G$ 、数据集  $P$ 、用户的查询关键字集合  $keys$  和用户权重集合  $w$ , 道路网多用户偏好 Top- $k$  Skyline 查询返回  $P$  的一个子集.该子集中数据点在道路网的距离维度和静态维度都不能被  $P$  中任意其他数据点支配,并且是根据用户群偏好和权重排序的 Top- $k$  个数据点.本文将道路网多用户偏好 Top- $k$  Skyline 查询方法记作 MUP-TKS.

## 3 道路网多用户偏好 Top- $k$ Skyline 查询

本文提出的道路网多用户偏好 Top- $k$  Skyline 查询方法主要分为 3 个部分: 距离较优集选取、 $K$ -准放松支配和 Top- $k$  个数据点选取.

### 3.1 道路网距离较优集选取方法

**定义 6.** Mindist 距离<sup>[24]</sup>.  $r$  维欧氏空间中, 点  $p$  到同一空间内某矩形  $N$  的最小距离为  $Mindist(N, p)$ .

**定义 7.** Edist 距离.设查询用户群的最小外接矩形(minimum bounding rectangle, MBR)为  $Q$ , 数据点  $p$  的 MBR 为  $N$ , 则  $\min\{Mindist(p, Q)\}$  为  $(Q, N)$  最小欧氏距离, 记作  $Edist_{\min}$ ;  $\max\{Mindist(p, Q)\}$  为  $(Q, N)$  最大欧氏距离, 记作  $Edist_{\max}$ .

**定义 8.** Ndist 距离.设查询用户群的 MBR 为  $Q$ , 数据点  $p$  的 MBR 为  $N$ , 有  $\min\{Ndist(p, Q)\}$  为  $(Q, N)$  最小网络距离, 记作  $Ndist_{\min}$ ;  $\max\{Ndist(p, Q)\}$  为  $(Q, N)$  最大网络距离, 记作  $Ndist_{\max}$ , 其中  $p$  为  $N$  中的任意数据点,  $Ndist(p, Q)$  为  $p$  到  $Q$  的网络距离.

**定理 1.** 设查询用户群的 MBR 为  $Q$ , 道路网中数据点构成的 2 个中间节点分别为  $N_1, N_2$ , 若  $DE_1=Edist_{\min}(Q, N_2)$ ,  $DE_2=Edist_{\max}(Q, N_1)$ ,  $DN_1=Ndist_{\max}(Q, N_1)$ , 并且  $DE_1 > DE_2$ ,  $DE_1 > DN_1$ , 则  $N_1 \blacktriangleright N_2$ , 且  $N_2$  中任意数据点都被  $N_1$  中数据点距离支配.

证明.假设  $DN_2=Ndist_{\min}(Q, N_2)$ , 因为欧氏距离值一定小于等于道路网距离值, 所以当  $DE_1 > DE_2$  且  $DE_1 > DN_1$  时一定有  $DN_2 \geq DE_1$ , 可得  $DN_2 > DN_1$ , 即  $N_2$  中数据点到  $Q$  的最小网络距离大于  $N_1$  中数据点到  $Q$  的最大网络距离, 进而可得  $N_2$  中任意数据点到  $Q$  的网络距离都大于  $N_1$  中任意数据点到  $Q$  的网络距离.因此  $N_1 \blacktriangleright N_2$ , 且  $N_2$  中任意数据点被  $N_1$  中任意数据点道路网距离支配.证毕.

**剪枝规则 1.** 设数据点构成的 MBR 分别为  $N_1, N_2$ , 查询用户群的 MBR 为  $Q$ , 如果满足:  $Edist_{\max}(Q, N_1) \leq Edist_{\min}(Q, N_2)$ , 并且  $Ndist_{\max}(Q, N_1) < Edist_{\min}(Q, N_2)$ , 则

节点  $N_2$  可被剪枝.

**定义 9.** 道路网最大距离的最小值. 给定数据点  $p_1, p_2$ , 查询用户群  $G$ , 数据点  $p$  到查询点  $q$  的道路网距离为  $Ndist(p, q)$ . 若有  $DN_1 = \max\{Ndist(p_1, q_i)\}$ ,  $DN_2 = \max\{Ndist(p_2, q_i)\}$  ( $1 \leq i \leq m$ ), 并且  $DN_1 < DN_2$ , 则当前道路网最大距离的最小值为  $DN_1$ , 记作  $DN\_MaxMin$ . 对应的数据点为  $p_1$ .

**定理 2.** 若节点  $N$  的  $Edist_{min}(Q, N) > DN\_MaxMin$ , 则节点  $N$  可被剪枝.

证明. 因为  $Edist_{min}(Q, N) > \max\{Ndist(p, q_i)\}$  ( $1 \leq i \leq m$ ), 所以  $Edist_{min}(Q, N) > \max\{Ndist(p, q_i)\}$ , 即  $p \triangleright N$ , 且  $N$  中数据点也被  $p$  距离支配. 证毕.

**剪枝规则 2.** 若  $Edist_{min}(Q, N) \geq DN\_MaxMin$ , 则节点  $N$  被支配, 即  $N$  和  $N$  中数据点  $\{p_1, p_2, \dots, p_i\}$  被剪枝.

如图 1 所示, 数据点  $p_1, p_2$  到查询用户群  $\{q_1, q_2, q_3\}$  的最大网络距离分别为  $DN_1, DN_2$ , 有  $DN_1 > DN_2$ , 则  $DN\_MaxMin = DN_2$ . 数据点  $\{p_3, p_4, p_5, p_6, p_7, p_8\}$  构成的 MBR 为  $N_1$ ; 若  $Edist_{min}(Q, N_1) > DN\_MaxMin$ , 可得  $N_1$  中数据点到各查询用户的网络距离大于  $DN\_MaxMin$ , 因为  $Edist_{min}(Q, N_1) > DN\_MaxMin$ , 且有  $\min\{Ndist(p_2, q_i)\} \geq Edist_{min}(Q, N_1)$  ( $1 \leq i \leq 3$ ), 所以  $p_2 \triangleright N_1, N_1$  可被剪枝.

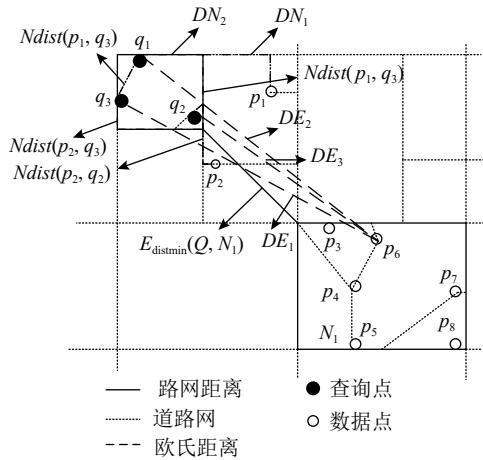


Fig. 1 Example of pruning rule 2

图 1 剪枝规则 2 示例

**定理 3.** 设  $DE$  为数据点  $p_i$  到查询用户  $q_j$  的欧氏距离, 若  $\min\{DE(p_i, q_j)\} > DN\_MaxMin$  ( $1 \leq j \leq m$ ), 则  $p_i$  被剪枝.

证明. 假设  $p_1$  为  $DN\_MaxMin$  对应的数据点, 若  $\min\{DE(p_1, q_j)\} > DN\_MaxMin$ , 则有  $Ndist(p_1, q_j) > DN\_MaxMin$  ( $1 \leq j \leq m$ ), 即数据点  $p_1 \triangleright p, p$  可被剪枝. 证毕.

**剪枝规则 3.** 假设数据点  $p_1$  为  $DN\_MaxMin$  对应的数据点, 若存在  $DN\_MaxMin < \min\{DE(p_i, q_j)\}$  ( $1 \leq j \leq m$ ), 则  $p_1$  可被剪枝.

$p_1 \triangleright p_i$ , 可将  $p_i$  从候选集中删除, 其中  $p_i$  为任意不为  $p_1$  的数据点.

为了减少计算, 在剪枝前基于路网数据点的网络 Voronoi 图构建 Vor-R\*-DHash 索引结构, 如图 2 所示.

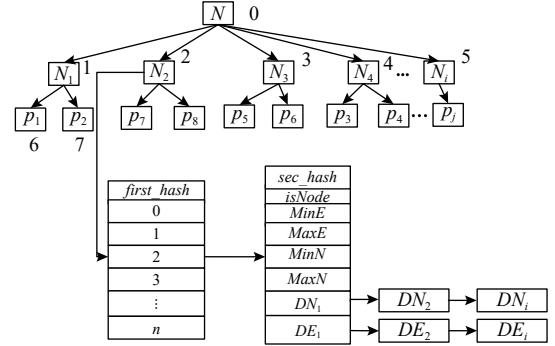


Fig. 2 Index structure of Vor-R\*-DHash

图 2 Vor-R\*-DHash 索引结构

Vor-R\*-DHash 索引结构构造过程有 3 步:

- 1) 构建路网所有数据点的网络 Voronoi 图.
- 2) 创建 R\*-tree. 从 R\*-tree 的根部开始, 从上至下、从左至右给每个节点编号, 从 0 开始编号.
- 3) 构建 2 级 HashMap 结构, 第 1 级 HashMap 为  $first\_hash$ 、key 为 R\*-tree 中每个节点编号; 第 2 级 HashMap 为  $sec\_hash$ 、key 为后续剪枝处理需要的值, 包括  $isNode$  (非数据点的节点)、 $MinE$  (节点到  $Q$  的最小欧氏距离值)、 $MaxE$  (节点到  $Q$  的最大欧氏距离值)、 $MinN$  (节点到  $Q$  的最小网络距离值)、 $MaxN$  (节点到  $Q$  的最大网络距离值)、 $\{DN_1, DN_2, \dots, DN_i\}$  (数据点到各查询用户的网络距离)、 $\{DE_1, DE_2, \dots, DE_i\}$  (数据点到各查询用户的欧氏距离).

2 级 key 对应的 value 值初始都为空, 若数据点根据剪枝规则提前被剪枝, 则这些值无需计算  $DE_i, DN_i$  的值也是后续需要使用才被计算, 并存入  $sec\_hash$ .

基于剪枝规则 1~3 和 Vor-R\*-DHash 索引结构, 进一步给出距离较优集选取方法, 如算法 1 所示.

**算法 1.** 距离较优集选取方法 G\_DBC.

输入: 查询用户群  $G$ , 道路网路段集  $R$ , 数据集  $P$ ;

输出: 距离维度不被支配的距离较优集  $DBC$ .

- ① 以  $P$  中数据点、道路网路段集  $R$  构建 Vor-R\*-DHash 索引;
- ② 构建查询用户群的最小外包矩形  $Q$ ;
- ③ 初始化  $DBC \leftarrow \emptyset$ ;
- ④ 根据索引找到距离查询用户最近的点  $point$ ;
- ⑤ 将  $point$  加入  $DBC$  中;
- ⑥ 计算数据点到各查询用户的网络距离  $Ndist(p, q_i)$ 、欧氏距离  $DE(p, q_i)$ ;

```

⑦ 将数据点网络距离、欧氏距离存入  $sec\_hash$ ;
⑧ 找到数据点到查询用户的网络距离最大值的
    最小值;
⑨  $DN\_MaxMin \leftarrow \min\{Ndist(p, q_i)\}$ ; /*将最小值
    赋给  $DN\_MaxMin$ */
⑩ 将数据点父节点  $N_i$  加入队列  $queue$  中;
⑪ 计算  $N_i$  到  $Q$  的最小、最大欧氏距离和最
    小、最大网络距离, 存至  $sec\_hash$ ;
⑫  $N_i \leftarrow \min\{MaxN\}$ ; /*将当前支配能力最强的
    节点赋值给  $N_i$ */
⑬ for  $node$  in  $queue$  do
⑭     if  $node$  的孩子节点都被访问过 then
⑮         将  $node$  的父节点加入  $queue$  中; /*向上
    一层访问*/
⑯     end if
⑰     if  $node$  的孩子节点  $N$  为非叶子节点 then
⑱         计算  $N$  到  $Q$  的欧氏距离  $DE_i$ ;
⑲         if  $DE_i > DN\_MaxMin$  then
⑳             Cut  $N$ ; /*将  $N$  剪枝, 剪枝规则 2*/
㉑         else if  $MaxE(N_i) < MinE(N)$  且
㉒              $MaxN(N_i) < MinE(N)$  then
㉓                 Cut  $N$ ; /*剪枝规则 1*/
㉔         else
㉕             将  $N$  加入队列  $queue$ ;
㉖             计算  $N$  到  $Q$  的最小、最大网络距离, 并
                存至  $sec\_hash$ ;
㉗             更新  $N_i \leftarrow \min\{MaxN\}$ ; /*当前支配能力强
                的节点赋给  $N_i$ */
㉘         end if
㉙     end if
㉚     if  $node$  的孩子节点  $N$  为叶子节点 then
㉛         计算数据点到各查询用户欧氏距离
             $DE(p, q_i)$ ;
㉜         if  $\min\{DE(p, q_i)\} > DN\_MaxMin$  then
㉝             Cut  $N$ ; /*剪枝规则 3*/
㉞         else
㉟             计算  $N$  到各查询用户网络距离  $DN$ ;
㉟             if  $\min\{DN\} > DN\_MaxMin$  then
㉟                 Cut  $N$ ;
㉟             else
㉟                 将  $N$  与  $DBC$  中数据点支配比较;
㉟                 if  $N$  被支配 then
㉟                     Delete  $N$ ;
㉟                 else

```

- ④② 将  $N$  加入  $DBC$  中;
- ④③ 更新  $DN\_MaxMin \leftarrow \min\{DN\}$ ;
- ④④ end if
- ④⑤ end if
- ④⑥ end if
- ④⑦ end if
- ④⑧ end for
- ④⑨ return  $DBC$ .

算法 1 首先构建 Vor-R<sup>\*</sup>-DHash 索引和查询用户群最小外接矩形  $Q$ , 可快速得到距离查询点最近的数据点  $point$ , 计算并保存  $sec\_hash$  所需数据. 将  $point$  加入距离较优集  $DBC$ , 并初始化  $DN\_MaxMin$ . 接着将  $point$  父节点加入队列  $queue$  中, 计算并保存  $sec\_hash$  所需数据, 并初始化  $N_i$ . 每次取出队头节点处理, 依据剪枝规则 1~3 进行节点的剪枝或者将节点加入  $DBC$ , 并判断是否需要更新  $N_i$ ,  $DN\_MaxMin$  等值, 直至队列为空, 循环结束. 最后返回距离较优集  $DBC$ .

### 3.2 数据集的放松支配过程

### 3.2.1 获取用户群权重偏好次序

首先初始化整体属性权重集合  $W = \{W_1, W_2, \dots, W_i\} = \{0, 0, \dots, 0\}$ ; 接着计算每个属性的整体属性权重值得到  $W$ ; 最后对整体属性权重值不为 0 的属性降序排列, 得到属性的重要性次序, 即用户群权重偏好次序.

在获取用户群权重偏好次序时,为了减小计算开销,利用  $HMap_1$ ,  $HMap_2$  分别保存优先考虑的属性和一般偏好的属性.当用户发起查询时,将  $C_1$  中属性作为键,对应的用户权重作为值保存到  $HMap_1$ ;将  $C_2$  中属性作为键,对应的用户权重作为值保存到  $HMap_2$ .

进一步给出获取用户群权重偏好次序算法 CDW, 如算法 2 所示.

### 算法 2. 获取用户群权重偏好次序算法 CDW.

输入: 用户群  $G$ , 用户查询关键字  $keys$ , 用户权重  $w$ , 维度空间  $D$ ;

输出: 用户群权重偏好次序  $GP$ .

- ① 初始化  $W$  为 0; /\*大小为数据集维度数\*/
  - ② 根据  $keys, w$  创建  $HMap_1, HMap_2$ ;
  - ③ for  $d_j \in D$  do
  - ④     基于  $HMap_1, HMap_2$  和式(1)得到  $W_j$ ;
  - ⑤ end for
  - ⑥ 根据  $W$  降序得到用户群权重偏好次序  $GP$ ;
  - ⑦ return  $GP$ . /\*返回用户群权重偏好次序\*/

### 3.2.2 基于用户群权重偏好次序的 $K$ -准放松支配

获取用户群偏好次序后,基于该次序进行放松支配处理.本文中  $K$  为整体属性权重值不为 0 的维度

数. 放松支配过程的处理对象为  $DBC$  与静态 Skyline 集取并集后的集合  $S$ . 经  $K$ -准放松支配后得到数量可控的候选结果集  $CS$ .

**定理 4.** 任意 2 个数据点  $p_i, p_j \in P$ , 若第  $i (i > 0)$  轮在  $K$  个维度上  $p_i \mathbb{C} p_j$ , 则数据点  $p_i$  必定在前  $K-i$  维支配数据点  $p_j$ .

证明. 若在第  $i$  轮  $p_i \mathbb{C} p_j$ , 可知该轮的无差异阈值为  $(0, 0, \dots, 0, \theta_{K-i+1}, \dots, \theta_K)$ , 进而可得前  $K-i$  维使用的无差异阈值为  $(0, 0, \dots, 0)$ , 所以前  $K-i$  维为严格支配比较, 即数据点  $p_i$  必定在前  $K-i$  维支配数据点  $p_j$ . 证毕.

**定理 5.** 数据集  $P$  经过第  $i (i > 1)$  轮放松支配后所得结果集  $S_i$  一定是第  $i-1$  轮放松支配后所得结果集  $S_{i-1}$  的子集.

证明. 设第  $i$  轮放松的维度为第  $(K-i+1) \sim K$  维, 第  $i-1$  轮放松的维度为第  $(K-i+2) \sim K$  维, 其余维度使用严格支配. 可知第  $i$  轮的无差异阈值为  $(0, 0, \dots, 0, \theta_{K-i+1}, \theta_{K-i+2}, \dots, \theta_K)$ , 第  $i-1$  轮的无差异阈值为  $(0, 0, \dots, 0, \theta_{K-i+2}, \dots, \theta_K)$ , 进而可知第  $i-1$  轮在前  $K-i+1$  个维度为严格支配比较, 即在前  $K-i+1$  个维度的无差异阈值为  $(0, 0, \dots, 0)$ . 第  $i$  轮不同于第  $i-1$  轮之处在于对第  $K-i+1$  维进行了放松支配, 即在前  $K-i+1$  个维度无差异阈值为  $(0, 0, \dots, 0, \theta_{K-i+1})$ , 所以有  $S_i \subseteq S_{i-1}$ . 证毕.

由定理 4、定理 5 可直接得出定理 6.

**定理 6.** 给定数据集  $S$ , 结果集数量随着每一轮放松而呈单调非递增趋势, 即

$$|KPRD(i-1, D, S)| \leq |KPRD(i, D, S)|. \quad (3)$$

为使返回的结果集更符合用户群偏好, 并保证数量可控, 基于定理 4~6 进行逐次放松支配. 逐次放松支配过程中,  $\theta$  是  $D$  上  $K$  个维度的无差异阈值,  $\theta = (\theta_1, \theta_2, \dots, \theta_K)$ . 假定当前放松轮次为第  $i$  轮 ( $1 \leq i \leq K$ ), 无差异阈值  $\theta = (0, 0, \dots, 0, \theta_{K-i+1}, \dots, \theta_K)$ . 位于  $d_i$  前面的维度重要性都要高于  $d_i$ , 因此该轮放松支配维度  $d_1 \sim d_{i-1}$  都使用严格支配比较. 放松支配从对用户群而言最不重要的属性开始, 并预先将数据点按照用户群权重偏好次序非递增排序, 距离维度值用数据点到查询用户群网络距离的最大值表示.

基于以上讨论, 进一步给出基于用户群权重偏好次序的  $K$ -准放松支配算法 KPRD, 如算法 3 所示.

**算法 3.** 基于用户群权重偏好次序的  $K$ -准放松支配算法 KPRD.

输入: 用户群  $G$ , 无差异阈值  $\theta$ , 并集  $S$ , 数据维度空间  $D$ ,  $k$  值, 用户查询关键字  $keys$ , 用户权重  $w$ ;

输出: 候选结果集  $CS$ .

①  $GP \leftarrow \text{call } CDW(G, keys, w, D); /* \text{调用算法 2}$

获取用户群权重偏好次序  $GP*/$   
②  $K \leftarrow |GP|; /* \text{整体属性权重值大于 0 的维度数}*/$   
③ 根据  $GP$  调整  $S$  中数据点;  
④ 初始化  $CurS \leftarrow S; /* CurS \text{ 为每轮放松支配后的结果集}*/$   
⑤ 初始化  $oldCount \leftarrow |S|; /* \text{保存上一轮结果集个数}*/$   
⑥ 初始化  $curCount \leftarrow |CurS|; /* \text{保存本轮结果集个数}*/$   
⑦ for  $j = K$  to 1 do /\* 进行最多  $K$  轮放松支配 \*/  
⑧ for every  $p_i, p_j \in CurS$  do  
⑨ if  $p_i \mathbb{C} p_j$ , then  
⑩ 将  $p_j$  从  $CurS$  删除;  
⑪  $curCount = curCount - 1$ ;  
⑫ end if  
⑬ end for  
⑭ if  $oldCount \geq k$  且  $curCount < k$  then  
⑮  $CS \leftarrow S$ ;  
⑯ return  $CS$ ; /\* 返回上一轮的结果集 \*/  
⑰ else  
⑱ 将  $CurS$  结果集保存至文件;  
⑲  $S \leftarrow CurS$ ; /\* 更新  $S$  \*/  
⑳  $oldCount \leftarrow |S|; /* \text{更新 oldCount}*/$   
㉑ end if  
㉒ end for  
㉓  $CS \leftarrow CurS$ ;  
㉔ return  $CS$ .

### 3.3 Top- $k$ 个数据点选取方法

通过放松支配处理后可有效控制返回用户群的结果集大小, 本节进一步给出 Top- $k$  个数据点选取策略, 使返回结果集有序. 利用  $z$ -整体属性权重值的打分函数选取 Top- $k$  个数据点, 处理对象为候选结果集  $CS$ .

**定义 10.** 单调打分函数  $F^{[25]}$ . 数据集中数据点作为输入域将数据点映射到实数范围.  $F$  由  $r$  个单调函数构成,  $F = \{f_1, f_2, \dots, f_r\}$ . 对于数据集中任意数据点, 有  $F = \sum_{j=1}^r f_j(p[d_j])$ , 其中  $f_j(p[d_j])$  为数据点在  $d_j$  维度的单调函数.

**定理 7.** 假设数据集  $P$  的单调打分函数为  $F$ , 若数据集中任意一个元组有最高的分数, 那么它一定是 Skyline 点.

证明. 以反证法进行证明. 假设有  $p_1, p_2 \in P$ ,  $p_1$  的得分  $F(p_1)$  为数据集的最高得分,  $F(p_1) > F(p_2)$ ,  $p_1$  不

是 Skyline 点,  $p_1$  支配  $p_2$ ,  $p_1[d_j] \leq p_2[d_j]$  ( $1 \leq j \leq r$ ), 则可得  $\sum_{j=1}^r f_j(p_1[d_j]) \leq \sum_{j=1}^r f_j(p_2[d_j])$ , 即  $F(p_1) \leq F(p_2)$ , 与假设矛盾. 证毕.

**定理 8.** 数据集  $P$  根据任意单调打分函数所得数据点顺序是 Skyline 支配的拓扑顺序.

证明. 以反证法进行证明. 假设存在 2 个数据点  $p_1, p_2 \in P$ , 单调打分函数为  $F$ ,  $p_1$  支配  $p_2$ ,  $F(p_1) < F(p_2)$ , 根据定理 7 可知,  $p_1$  支配  $p_2$ , 则有  $F(p_1) \geq F(p_2)$ , 与假设矛盾. 所以如果  $F(p_2) > F(p_1)$ , 可能有  $p_2$  支配  $p_1$ , 但可以确定  $p_1$  不可能支配  $p_2$ . 如果  $F(p_1) = F(p_2)$ , 则  $p_1$  支配  $p_2$  或  $p_2$  支配  $p_1$ (这两者是等价的, 会根据属性的映射关系排序), 或者  $p_1$  和  $p_2$  之间不具备支配关系. 因此依据打分函数  $F$  所得数据点顺序是按照 Skyline 支配关系的一个拓扑顺序. 证毕.

**定义 11.** 线性打分函数<sup>[25]</sup>. 给定线性打分函数  $L$ , 一般化形式为  $L(p) = \sum_{j=1}^r \omega_j \cdot p[d_j]$ , 其中  $\omega_j$  为实常数,  $p[d_j]$  为数据点在  $d_j$  维度的取值.

**定义 12.**  $z$ -整体属性权重值. 给定数据集  $P$ , 数据点  $p_i \in P$ ,  $p_i$  在  $d_j$  维度的  $z$ -整体属性权重值为

$$\varphi_{i,j} = \frac{(V_{i,j} - \mu)}{\sigma} \cdot W_j \cdot \zeta_j, \quad (4)$$

其中,  $\frac{(V_{i,j} - \mu)}{\sigma}$  为数据点  $p_i$  在维度  $d_j$  的  $z$  值,  $W_j$  为  $d_j$  的整体属性权重值,  $\zeta_j$  为  $d_j$  的维度优劣值,  $\zeta_j = 1$  或  $\zeta_j = -1$ . 由定义 10~12 可知,  $f_j(p[d_j]) = \varphi_{i,j} = \omega_j \cdot \zeta_j$ ,  $\omega_j = W_j \cdot \zeta_j$ .

**定理 9.** 数据点任意维度的  $f_j(p[d_j])$  是单调的.

证明. 因为  $\omega_j = W_j \cdot \zeta_j$ , 在打分阶段  $W_j$  为实常数, 所以可得  $\omega_j$  为实常数, 且随着数据点维度值变大, 它的  $z$  值也变大, 因此数据点的任意维度  $f_j(p[d_j])$  是单调的. 证毕.

**定义 13.** 基于  $z$ -整体属性权重值的打分函数. 数据点  $p_i$  各维度  $z$ -整体属性值之和为它的得分, 记作  $F(p_i)$ :

$$F(p_i) = \sum_{j=1}^r \varphi_{i,j}. \quad (5)$$

**定理 10.**  $F(p_i)$  是单调打分函数.

证明. 因为有  $F(p_i) = \sum_{j=1}^r f_j(p[d_j])$ , 根据定理 9 可知数据点的任意维度  $f_j(p[d_j])$  随着维度值变大单调递增, 它们具备相同的单调性, 因此  $F(p_i)$  也是单调的. 证毕.

进一步给出 Top- $k$  个数据点选取方法, 如算法 4 所示.

**算法 4.** Top- $k$  个数据点选取方法 TK\_DC.

输入: 候选结果集  $CS$ , 整体属性权重集合  $W$ , 维度优劣集合  $\zeta$ ;

输出: Top- $k$  Skyline 结果集.

- ① for  $p_i \in CS$  do
- ② 计算数据点的  $z$ -整体属性权重值; /\*根据式(4)\*/
- ③ 计算数据点得分; /\*根据式(5) \*/
- ④ end for
- ⑤ 根据  $F(p_i)$  降序排序;
- ⑥ return Top- $k$  个数据点.

算法 4 主要对经过算法 3 处理后的候选结果集  $CS$  打分, 并对行②③计算  $CS$  中各个数据点的得分, 基于行⑤⑥数据点的得分排序, 输出 Top- $k$  Skyline 结果集给用户群.

综合距离较优集选取、 $K$ -准放松支配和 Top- $k$  个数据点选取的处理过程, 进一步给出算法 5 MUP-TKS 的算法.

**算法 5.** 道路网多用户偏好 Top- $k$  Skyline 查询算法 MUP-TKS.

输入: 数据集  $P$ , 道路网路段集  $R$ , 用户群  $G$ , 用户查询关键字  $keys$ , 用户权重  $w$ , 无差异阈值  $\theta, k$ , 维度优劣集合  $\zeta$ ;

输出: Top- $k$  Skyline 结果集.

- ① 预先计算保存数据集的静态 Skyline 集;
- ② 距离较优集选取方法 G\_DBC; /\*调用算法 1\*/
- ③ 对距离较优集与静态 Skyline 集求并集  $S$ ;
- ④  $K$ -准放松支配算法 KPRD; /\*调用算法 3\*/
- ⑤ Top- $k$  个数据点选取方法 TK\_DC. /\*调用算法 4\*/

## 4 实验比较与分析

本节主要对 MUP-TKS 进行实验以及性能评估. 实验对比算法为道路网单用户偏好 Skyline 算法 UP-BPA<sup>[26]</sup>、 $K$  支配空间偏好 Skyline 算法 KSJQ<sup>[23]</sup> 以及基于时间道路网多用户偏好 Skyline 算法 DSAS<sup>[27]</sup>. UP-BPA 算法适用于道路网单用户, 为了更好地与本文所提 MUP-TKS 进行对比, 将其扩展, 对查询用户群的每个用户分别运行该算法; 再对子结果集取并集, 得到候选结果集  $CS$ ; 最后对候选结果集基于  $z$ -值的打分函数打分, 得到 Top- $k$  个数据点, 扩展后的算法称为 EUP-BPA. 将 KSJQ 算法扩展, 对每个用户单独执行该算法, 用户偏好对应它的  $K$  个子空间; 对每个

用户的结果集取并集后得到候选结果集; 对候选结果集  $CS$  基于  $z$ -值的打分函数打分, 得到 Top- $k$  个 Skyline 结果集, 扩展后的算法称为 EKSJQ. 将 DSAS 算法扩展, 对满足不同用户需求的数据点基于  $z$ -值打分函数打分, 按照数据点得分从高至低返回 Top- $k$  个 Skyline 结果集, 扩展后的算法称为 EDSAS.

#### 4.1 数据集及实验环境设置

实验使用真实道路网数据集. 道路网数据集<sup>①</sup>是北美  $2.5 \times 10^7 \text{ km}^2$  范围内的路段信息, 它包含 175 813 个节点和 179 179 条边. 兴趣点数据集<sup>②</sup>来自北美酒店及登记信息. 查询用户采用随机生成的方式. 本文使用 Vor-R\*-DHash 索引结构组织数据集. 实验参数取值范围如表 1 所示, 每个用户最大关注维度为 4. 每个实验采取单一变量原则, 其余变量为默认值, 实验结果取 30 次实验运行的平均值.

Table 1 Experimental Parameter Setting

表 1 实验参数设置

参数	取值范围
用户数量	5, 10, 15, 20, 25, 30, 35
数据集规模	$1 \times 10^4, 2 \times 10^4, 3 \times 10^4, 4 \times 10^4, 5 \times 10^4$
数据维度	5, 7, 9, 11, 13, 15, 17
无差异阈值 (标准差)	0.1 倍, 0.5 倍, 1 倍, 2 倍, 10 倍
获取数据点数量 $k$	2, 4, 6, 8, 10

注: 加粗数值表示参数默认取值.

实验环境为: Windows 10(64b), Core™ i6-5200U CPU @2.20 GHz 2.19 GHz 处理器, 12 GB 运行内存. 在 IntelliJ IDEA 开发平台上使用 Java 实现本文所提的算法 MUP-TKS 和对比算法 EUP-BPA, EKSJQ, EDSAS.

#### 4.2 算法对比实验

##### 1) 用户数量对算法性能的影响

为了分析用户数量对算法性能的影响, 本实验对不同用户数量下的 MUP-TKS, EKSJQ, EDSAS, EUP-BPA 算法进行测试, 观察算法在不同用户数量下的 CPU 运行时间、候选结果集  $CS$  数量的变化情况.

图 3 展示了 4 种算法在不同用户数量下 CPU 运行时间变化情况. 由图 3 可知, 随着用户数量的增加, 4 种算法的 CPU 运行时间都在增加. 因为用户数量增加导致不同用户的偏好情况增加, 从而需要更多时间处理用户偏好. MUP-TKS 的 CPU 运行时间增长趋势没有其他 3 种算法的增长趋势大, 主要原因是

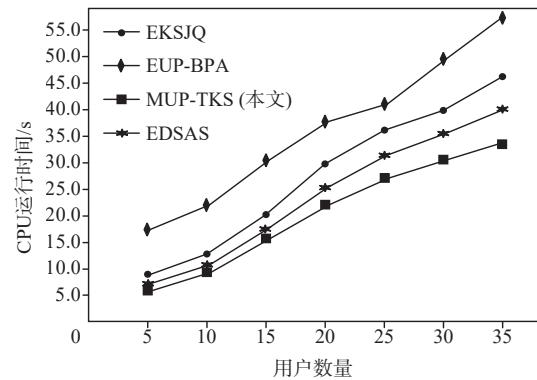


Fig. 3 Effect of user number on CPU execution time

图 3 用户数量对 CPU 运行时间的影响

MUP-TKS 将多用户的偏好转换成用户群权重偏好次序, 对数据集按照该次序预排序, 再进行  $K$ -准放松支配, 使用户数量增加对 CPU 运行时间的影响减小.

图 4 展示了 4 种算法随着用户数量的变化, 候选结果集  $CS$  数量的变化情况. 由图 4 可知随着用户数量的增加,  $CS$  的数量变大. 但 MUP-TKS, EKSJQ, EDSAS 算法的变化趋势远没有 EUP-BPA 算法的变化趋势大, 主要因为 EUP-BPA 算法需要对每个用户进行偏好 Skyline 查询, 再合并各用户的偏好 Skyline 结果集.

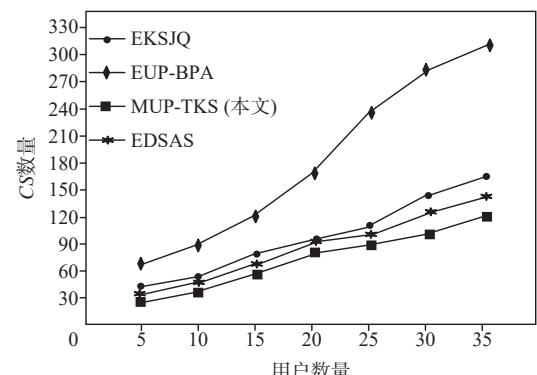


Fig. 4 Effect of user number on CS number

图 4 用户数量对  $CS$  数量的影响

##### 2) 数据规模对算法性能的影响

为了分析数据规模对 MUP-TKS 性能的影响, 本实验对不同数据规模下的 MUP-TKS, EKSJQ, EDSAS, EUP-BPA 算法进行测试, 观察 4 种算法在不同数据规模下 CPU 运行时间、 $CS$  数量的对比情况.

由图 5 可知, 随着数据集规模变大, CPU 运行时间不断增加, 因为当数据集规模变大时, 需要比较的元

① <http://www.cs.utah.edu/~lifeifei/SpatialDataset.htm>

② <https://www.ahla.com/>

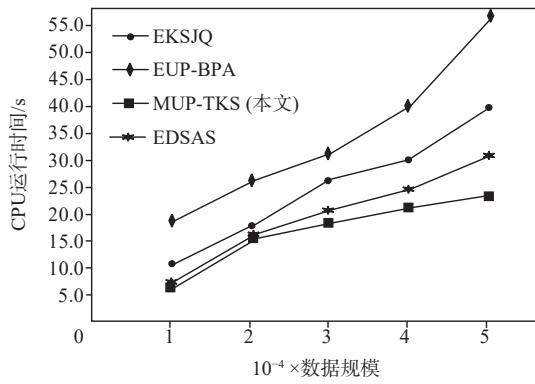


Fig. 5 Effect of data size on CPU execution time

图 5 数据规模对 CPU 运行时间的影响

组数量增加。而 MUP-TKS 的增长趋势比其他 3 种算法小, 主要因为 MUP-TKS 利用剪枝策略和 Vor-R<sup>\*</sup>-DHash 索引提前剪枝大量不可能成为 Skyline 的数据点, 减少了元组比较次数。

### 3) $k$ 值对算法性能的影响

图 6 展示了 4 种算法随着  $k$  值变化 CPU 运行时间变化的情况。随着  $k$  值变化, MUP-TKS 的 CPU 运行时间没有太大变化, 因为 MUP-TKS 在每一轮放松支配后会保存结果集, 当  $k$  值变化时, 可直接找到对应符合大小要求轮次的 CS 打分, 即可得到 Top- $k$  Skyline 结果集, 该过程时间消耗很小。而 EKSJQ, EUP-BPA 算法都需要重新计算, 时间消耗较大。

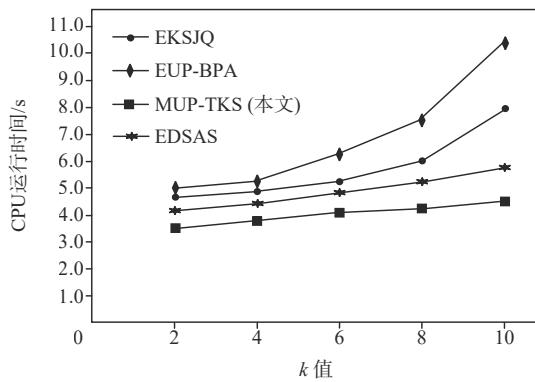
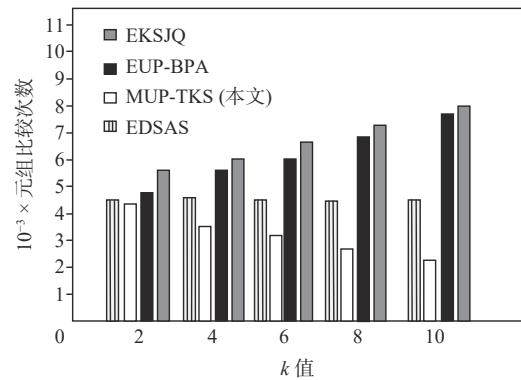
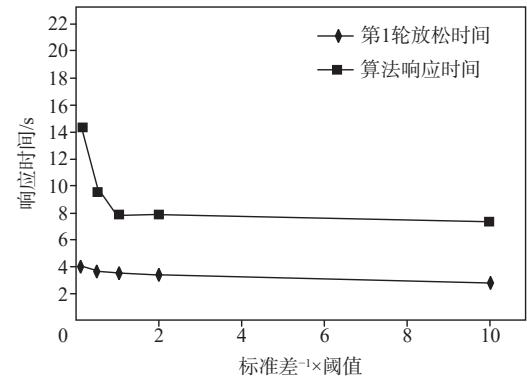
Fig. 6 Effect of  $k$  value on CPU execution time图 6  $k$  值对 CPU 运行时间的影响

图 7 展示了 4 种算法随着  $k$  值变化元组比较次数的变化情况。可以发现 MUP-TKS 随着  $k$  值增大, 元组比较次数减少, 因为当  $k$  值增大时, 放松支配的轮次减少。而随着  $k$  值增大, EKSJQ, EUP-BPA 算法的元组比较次数增多, 因为需要进行更多的支配比较找到 Top- $k$  个数据点。随着  $k$  值增大, EDSAS 算法的元组比较次数基本没有变化。

### 4) 无差异阈值对算法性能的影响

Fig. 7 Effect of  $k$  value on the number of tuple comparison图 7  $k$  值对元组比较次数的影响

本实验分析无差异阈值对 MUP-TKS 性能的影响。图 8 展示了 MUP-TKS 在不同无差异阈值下 CPU 运行时间的变化情况。由图 8 可知, 若只考虑第 1 轮放松时间, 无差异阈值变化对第 1 轮放松的 CPU 响应时间影响不大, 因为不同无差异阈值的初始数据集大小都是相同的, 处理相同数据集规模的时间差异不大。而算法总运行时间随着阈值增大而减小, 因为无差异阈值增大后, 放松支配时会删减更多被支配的元组。

Fig. 8 Effect of  $\theta$  on CPU execution time图 8  $\theta$  对 CPU 运行时间的影响

## 5 总 结

本文针对现实生活中道路网多用户场景的偏好 Top- $k$  Skyline 查询问题, 进行深入分析与研究。作为道路网上单用户偏好 Skyline 查询问题的补充, 提出了一种基于道路网环境下多用户偏好 Top- $k$  Skyline 查询方法。该方法利用剪枝规则和索引减少了距离计算开销, 并利用用户群权重偏好次序进行放松支配, 使结果集可控。实验结果表明, 本文方法能有效解决道路网多用户偏好查询问题, 返回的结果集可

以满足多用户偏好与权重需求, 可以提供有效参考价值。下一步研究重点主要集中在对多查询用户移动情况下偏好 Top- $k$  Skyline 查询问题的处理。

**作者贡献声明:** 李松提出了方法思路和技术方案; 宾婷亮和郝晓红负责算法优化、完成部分实验并撰写论文; 张丽平完成部分实验; 郝忠孝提出指导意见并修改论文。

## 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, 2001: 421–430
- [2] Zeng Yifu, Chen Guo, Li Kenli, et al. M-Skyline: Taking sunk cost and alternative recommendation in consideration for Skyline query on uncertain data[J]. Knowledge-Based Systems, 2019, 163(1): 204–213
- [3] Ren Weilong, Lian Xiang, Ghazinour K. Skyline queries over incomplete data streams[J]. The Very Large Date Bases Journal, 2019, 28(6): 961–985
- [4] Wang Weiguo, Li Hui, Peng Yanguo, et al. Scale: An efficient framework for secure dynamic Skyline query processing in the cloud [C]//Proc of the 25th Int Conf on Database Systems for Advanced Applications. Berlin: Springer, 2020: 288–305
- [5] Yin Bo, Zhou Siwang, Zhang Shiwen, et al. On efficient processing of continuous reverse Skyline queries in wireless sensor networks[J]. KSII Transactions on Internet and Information Systems, 2017, 11(4): 1931–1953
- [6] Kossmann D, Ramsak F, Rost S. Shooting stars in the sky: An online algorithm for Skyline queries[C] //Proc of the 28th Int Conf on Very Large Databases(VLDB’02). San Francisco, CA: Morgan Kaufmann, 2002: 275–286
- [7] Chomicki J, Godfrey P, Gryz J, et al. Skyline with presorting[C] //Proc of the 19th Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2003: 717–719
- [8] Li Song, Dou Yanan, Hao Xiaohong, et al. The method of the  $K$ -dominant space Skyline query in road network[J]. Journal of Computer Research and Development, 2020, 57(1): 227–239 (in Chinese)  
(李松, 窦雅男, 郝晓红, 等. 道路网环境下  $K$ -支配空间 Skyline 查询方法[J]. 计算机研究与发展, 2020, 57(1): 227–239)
- [9] Fu Xiaoyi, Miao Xiaoye, Xu Jianliang, et al. Continuous range-based Skyline queries in road networks[J]. World Wide Web, 2017, 20(6): 1443–1467
- [10] Li Song, Wang Guanqun, Hao Xiaohong, et al. A multi-objective decision optimization algorithm for recommendation system [J]. Journal of xi'an Jiaotong University, 2022, 56(8): 104–112 (in Chinese)  
(李松, 王冠群, 郝晓红, 等. 面向推荐系统的多目标决策优化算法.
- 西安交通大学学报, 2022, 56(8): 104–112)
- [11] Zhang Kaiqi, Gao Hong, Han Xixian, et al. Probabilistic Skyline on incomplete data[C] //Proc of the 26th Int Conf on Information and Knowledge Management. New York: ACM, 2017: 427–436
- [12] Abdul Fattah H M, Azharul Hasan K M, Tsuji T. Indexed Top- $k$  dominating queries on highly incomplete data[C] //Proc of the 1st Int Conf on Big Data, IoT, and Machine Learning. Berlin: Springer, 2022: 231–241
- [13] Mouratidis K, Li Keming, Tang Bo. Marrying Top- $k$  with Skyline queries: Relaxing the preference input while producing output of controllable size[C] //Proc of the 47th Int Conf on Management of Data. New York: ACM, 2021: 1317–1330
- [14] Saad N H M, Ibrahim H, Sidi F, et al. Efficient Skyline computation on uncertain dimensions[J]. IEEE Access, 2021, 9: 96975–96994
- [15] Bai Mei, Wang Jinghui, Wang Xite, et al. PSP: An efficient Skyline computation method for partially ordered domains[J]. Journal of Hunan University: Natural Sciences, 2020, 47(8): 9–20 (in Chinese)  
(白梅, 王京徽, 王习特, 等. PSP: 一种高效的偏序域上 Skyline 查询处理方法[J]. 湖南大学学报: 自然科学版, 2020, 47(8): 9–20)
- [16] Wang Yan, Shi Zhan, Wang Junlu, et al. Skyline preference query based on massive and incomplete dataset[J]. IEEE Access, 2017, 5: 3183–3192
- [17] Tai Liangkuang, Wang E T, Chen A L P. Finding the most profitable candidate product by dynamic skyline and parallel processing[J]. Distributed and Parallel Databases, 2021, 39(4): 979–1008
- [18] Zheng Zhiyun, Zhang Minghao, Yu Mengyao, et al. User preference-based data partitioning Top- $k$  Skyline query processing algorithm [C]//Proc of the 2nd Int Conf on Industrial Application of Artificial Intelligence. Piscataway, NJ: IEEE, 2021: 436–444
- [19] Cai Zhi, Cui Xuerui, Su Xing, et al. Continuous road network-based Skyline query for moving objects[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22(12): 7383–7394
- [20] Li Qiyuan, Zhu Yuanyuan, Yu J X. Skyline cohesive group queries in large road-social networks[C]//Proc of the 36th Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2020: 397–408
- [21] Hidayat A, Cheema M A, Lin Xuemin, et al. Continuous monitoring of moving Skyline and Top- $k$  queries[J]. The VLDB Journal, 2022, 31(3): 459–482
- [22] Ding Linlin, Zhang Xiao, Sun Mingxin, et al. Efficient user preferences-based Top- $k$  Skyline using MapReduce[C] //Proc of the 4th Int Conf on Pioneering Computer Scientists, Engineers and Educators. Berlin: Springer, 2018: 74–87
- [23] 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 Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2017: 99–102
- [24] Zan C T, Yamana H. An improved symbolic aggregate approximation distance measure based on its statistical features[C] //Proc of the 18th Int Conf on Information Integration and Web-based Applications and Services. New York: ACM, 2016: 72–80
- [25] Chomicki J, Godfrey P, Gryz J, et al. Skyline with Presorting: Theory and Optimizations[C] //Intelligent Information Processing and Web

Mining. Berlin: Springer, 2005: 595–604

- [26] Zou Yonggui, Wang Zhi. User preference based spatial Skyline query method[J]. Applied Mechanics & Materials, 2014, 530: 832–838
- [27] John A, Singh S K, Adimoolam M, et al. Dynamic sorting and average Skyline method for query processing in spatial-temporal data[J]. International Journal of Data Science, 2021, 6(1): 1–18



**Li Song**, born in 1977. PhD, professor. Member of CCF. His main research interests include spatial database, data mining, and data information security.  
李 松, 1977 年生. 博士, 教授. CCF 会员. 主要研究方向为空间数据库、数据挖掘和数据信息安全.



**Bin Tingliang**, born in 1999. Master candidate. Her main research interests include spatial database and data mining.  
宾婷亮, 1999 年生. 硕士研究生. 主要研究方向为空间数据库和数据挖掘.



**Hao Xiaohong**, born in 1969. Master, senior experimentalist. Her main research interest includes spatial database.

郝晓红, 1969 年生. 硕士, 高级实验师. 主要研究方向为空间数据库.



**Zhang Liping**, born in 1976. Master, associate professor. Her main research interests include spatial database and data information security.

张丽平, 1976 年生. 硕士, 副教授. 主要研究方向为空间数据库和数据信息安全.



**Hao Zhongxiao**, born in 1940. PhD, professor. His main research interests include relational database, null database, acyclic database, active database, and spatial database.

郝忠孝, 1940 年生. 博士, 教授. 主要研究方向为关系数据库、空值数据库、非循环数据库、主动数据库和空间数据库.