计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20140602
52(12):2750-2763, 2015

道路网络上基于网络 Voronoi 图的隐私保护算法

潘 晓 吴 雷 胡朝君
(石家庄铁道大学 石家庄 054003)
(smallpx@163.com)

A Privacy Protection Algorithm Based on Network Voronoi Graph over Road Networks

Pan Xiao, Wu Lei, and Hu Zhaojun
(*Shijiazhuang Tiedao University*, *Shijiazhuang* 054003)

Abstract With advances in wireless communication and mobile positioning technologies, location-based services (LBSs) have seen wide-spread adoption. The academia and industry have pay attention to location privacy preserving in LBSs for about ten years. Most existing location cloaking algorithms over road networks are based on DFS-based or BFS-based graph traversal algorithms. However, the main drawback of these approaches is the repetitive global road network traversal, which results in the low efficiency and high communication cost. In order to resolve the problem, we propose a network Voronoi-based location anonymization algorithm NVLA on road networks. Under the help of network Voronoi diagram (NVD), the road network is divided into several network Voronoi cells offline. Then, we reduce the problem of anonymization over the global road network into finding cloaking sets in local network Voronoi cells (NVC). Next, according to the number of road segments and the number of users in a cell, network Voronoi cells are classified to unsafe, safe-medium and safe-large cells. Finally, according to the features of the different network Voronoi cells, different cloaking algorithms are proposed. We design and conduct a series of experiments, and the experimental results validate the efficiency and effectiveness of the proposed algorithm. The average cloaking time is only 0.4 ms and the cloaking success rate is about 100%. On the other hand, only 0.01% of the query cost is sacrificed.

Key words location privacy; network Voronoi diagram (NVD); road networks; location-based services (LBSs); mobile computing

摘 要 基于位置服务(location-based services, LBSs)中的不可信服务提供商不断收集用户个人数据,为用户隐私带来威胁.因此,LBSs 中的位置隐私保护研究已在学术界和工业界受到广泛关注.现有道路网络中的位置隐私保护方法大多是基于深度或广度图遍历的算法,需重复扫描道路网络的全局拓扑信息,匿名效率较低.针对这一问题,利用网络 Voronoi 图(network Voronoi diagram, NVD)将道路网络事先划分为独立的网络 Voronoi 单元,将传统方法中的多次遍历全局道路网络转化为了访问网络 Voronoi 单元中的局部路网信息.根据网络 Voronoi 单元覆盖的移动用户数和路段数,将网络 Voronoi 单元分为了不安全单元、安全-中单元和安全-大单元 3 类,提出了适应不同类型网络 Voronoi 单元特点的高效位置匿名算法.最后,通过在真实数据集上进行大量实验,验证了提出算法在仅比传统算法多牺

收稿日期:2014-07-05;**修回日期**:2015-05-27
基金项目:国家自然科学基金项目(61303017);河北省自然科学基金项目(F2014210068);国家级大学生创新创业训练计划项目(201410107003);国家留学基金资助出国留学项目(201408130042)

牲 0.01%的查询代价的前提下,保证了 100%的匿名成功率和 0.34 ms 的高效匿名时间,在隐私保护强度和算法性能方面取得了较好的平衡.

关键词　位置隐私;网络 Voronoi 图;道路网络;基于位置服务;移动计算

中图法分类号　TP311.13

伴随着无线网络和移动定位技术的发展,基于位置服务(location-based services, LBSs)得到了广泛应用. LBSs 为人们生活和工作带来了巨大便利,但同时不可信服务提供商不断收集用户个人数据,为用户的隐私带来了威胁[1-2]. 研究者们最早关注的是欧氏空间中的位置隐私保护. 随着技术的发展和研究的深入,越来越多的学者[3-10]开始关注更具有实际意义的道路网络上的位置隐私保护问题.

一般情况下,基于路网的位置匿名算法生成的匿名区域有 3 种形式:矩形、路段集和不规则形状. Ku 等人[3]直接利用欧氏空间中 Casper 算法[4]在路网上生成矩形匿名区域,但这样的做法会造成隐私泄露的问题. 文献[5-8]以移动用户所在的路段集合作为匿名区域发布. 由于路段集保留了路网拓扑结构,在位置服务器端产生的查询代价较小. 所以,路段集是目前最常用的匿名区域发布形式,被称为匿名路段集. 文献[9-10]提出了使用 mix-zone 技术生成不规则匿名区域,保护用户位置隐私. 但是该方法仅能保证用户在不规则区域内的位置信息,当用户处于不规则区域外时,需要发布确切位置. 本文采用匿名路段集作为匿名区域.

现有发布匿名路段集的隐私保护方法大多属于基于图遍历的位置匿名法. 此类方法的主要缺点是需要重复对路网进行全局扫描. 众所周知,完整的路网结构数据庞大,每次对路网进行全局扫描为网络传输和匿名处理带来巨大压力. 为解决此问题,本文提出了一种道路网络上基于网络 Voronoi 图的高效位置匿名算法 NVLA. NVLA 的基本出发点是利用网络 Voronoi 图(network Voronoi diagram, NVD)将道路网络划分为独立的网络 Voronoi 单元(network Voronoi cells, NVC),将原来在整个路网的匿名问题转化为了在局部 NVC 中进行位置匿名. 图 1 是一个 NVD 的例子,包括 3 个 NVC$\{NVC(p_1), NVC(p_2), NVC(p_3)\}$(由虚线隔开),其中利用感兴趣点(points of interests, POIs)$\{p_1, p_2, p_3\}$作为种子①. 如果移动用户正好位于 $NVC(p_1)$,同时用户的隐私需求得到满足,则直接发布 $NVC(p_1)$中覆盖的所有路段作为匿名路段集即可.

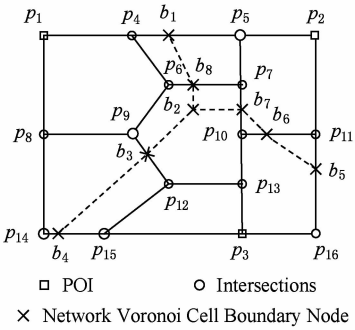


Fig. 1　Network Voronoi diagram.

图 1　网络 Voronoi 图

在路网上基于 NVD 进行位置匿名所面临的主要挑战是: NVC 大小不一,用户分布不均匀,致使适用于某种 NVC 的匿名算法在其他 NVC 中未必有效. 为解决此问题,本文将 NVC 分为不安全单元(unsafe cells)、安全-中单元(safe-medium cells)和安全-大单元(safe-large cells). 基于不同类型 NVC,提出了不同的匿名方法. 具体来讲,如果用户位于安全-中单元,则直接将此单元中包含的所有用户作为匿名集;如果用户位于安全-大单元,则基于用户密度,分别采用基于用户分组或路段分组的匿名算法寻找匿名集;如果用户位于不安全单元,则将该单元与邻近单元合并直至形成安全-大单元或安全-中单元为止.

本文贡献总结如下:

1) 提出了一种在道路网络上基于网络 Voronoi 图的高效位置匿名算法 NVLA,实现了在局部网络 Voronoi 单元中寻找匿名集,避免了路网的全局重复扫描.

2) 在网络 Voronoi 安全-大单元中,根据用户密度,分别提出了基于路段分组的位置匿名算法 SCA 和基于用户分组的位置匿名算法 UCA.

3) 利用真实数据进行了大量实验,验证了提出算法在隐私保护强度和算法性能方面取得了较好的平衡,实现了牺牲较少查询代价的前提下提供较安全的隐私保护强度.

① 利用 POIs 作为种子的原因是 POIs 位置相对静态且公开.

1 相关工作

道路网络上现有的位置隐私保护技术可以总结为4类:基于树型索引的隐私保护技术、基于图遍历的隐私保护技术、基于mix-zone的隐私保护技术和基于PIR的隐私保护技术.

1) 基于树型索引的隐私保护技术[11]. 其主要思想是在道路网络上建立一个树型索引,当用户提出查询请求时,首先确定用户所在路段.从路段索引中找到包含该路段的第1层非叶子结点,若该结点满足用户隐私需求,则将该结点下覆盖的路段组成匿名路段集,路段上的用户组成匿名集合.若用户隐私需求没有得到满足,则递归地检查同双亲结点的兄弟结点.如果与兄弟结点的组合满足用户隐私需求,则将该组合覆盖的所有用户组成匿名集.如果依然有用户的隐私需求未满足,则重复这一步,直至满足用户隐私需求或递归到根结点.

2) 基于图遍历的隐私保护技术[5-8]. 该技术是从某点(城市中心、敏感位置等)或边(如用户位置所在路段)开始,以特定目标函数最优为依据(如包含的用户数最多、查询代价最小等),在路网中进行宽度或广度优先遍历,遍历过的边构成一个连图子图.子图中包含的边组成匿名路段集,包含的用户组成匿名集发布.因为匿名路段集保留了路网拓扑,并具有可共享查询结果的特点,较其他方法查询代价较小,所以基于图遍历的隐私保护技术是截至到目前为止使用最为广泛的方法.本文提出的NVLA算法也属于基于图遍历的匿名算法.然而,与现有算法不同的是,NVLA从局部NVC中寻找匿名集,无需遍历整个路网.

3) 基于mix-zone的隐私保护技术[9-10]. 该技术即以路网交叉口为中心,沿着路网出口的方向建立具有自适应特点的非矩形mix zone.每当用户进入到该区域(zone)时,不作任何位置更新,出区域后更换假名.由于用户在区域中位置信息被隐藏同时其对应假名也发生了变更,增加了将同一个用户前后使用的假名关联起来的难度,从而达到了保护用户标识的目的.该方法的缺点如引言所述.

4) 基于PIR的隐私保护技术[12]. 该技术在服务提供商的服务器上路网和POI以明文形式存在,并被划分为一些块序列(sequential blocks).一个被称为安全同步处理器(secure co-processor, SCP)的黑盒子安装在服务器上.移动用户将查询请求发给SCP,SCP从服务提供商的数据库中抽取包含真实结果的结果集块(block)并安全地转发给用户.攻击者知道SCP在服务器中请求了数据,但不知道服务器返回的是哪个数据块,进而保护了用户隐私.基于PIR的隐私保护技术是一种强隐私保护机制,没有任何信息的泄露,较其他3种方法最安全.但是,由于算法代价高,当数据量较大时,较难在实际中应用.

2 系统结构

系统结构采用目前最流行的中心服务器结构[3-8],如图2所示,包括移动用户、可信匿名服务器(trusted anonymizing proxy, TAP)和不可信服务提供商.采用中心服务器结构的好处在于匿名服务器具有全局用户信息,可达到较好的隐私保护效果.

移动用户通过加密的方式将查询提交给TAP. TAP包含4个部件:查询分类器(query classifier)、知识库(repository)、匿名引擎(cloaked engine)和查询结果求精引擎(answer refined engine).查询分类器将提交来的查询分成最近邻查询和非最近邻查询;知识库中存储着离线建立的NVD;匿名引擎负责为提出查询的用户寻找匿名集;查询求精引擎将位置服务器返回的候选结果求精,把真实的查询结果返回给相应的用户.

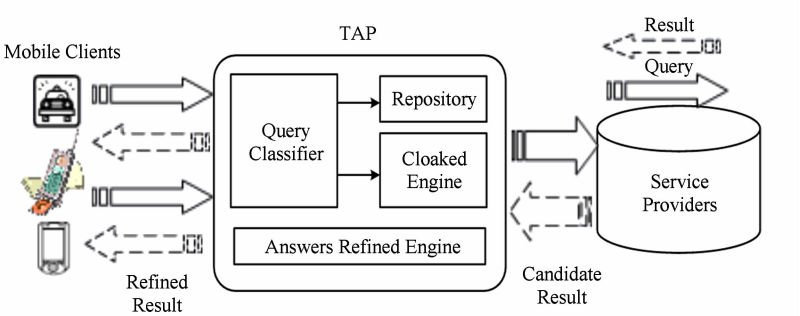


Fig. 2 Centralized system architecture.

图2 中心服务器系统结构

本文基于这样的假设:道路网络和 POIs(如机场、故宫、天安门等)信息是公开且相对静态的. TAP 用 POIs 作为种子离线生成 NVD. 如此,道路网络信息以 NVC 的形式存储于知识库中. 离线划分道路网络具有 3 个优点:

1) 利用 NVD 信息,TAP 可以直接回答最近邻查询而无需询问位置服务器,节省了网络带宽. 表面上看,回答最近邻查询貌似增加了 TAP 的工作负担. 实际上,事实正好相反. 利用 NVC,TAP 从原来为该查询寻找匿名集和后期的查询结果求精工作中解放了出来.

2) 节省了网络带宽,网络传输效率提高. TAP 直接响应最近邻查询,省去了原本 TAP 与服务提供商之间的信息传输.

3) 提高了用户体验. 在用户提出最近邻查询时,无需等待匿名和查询结果求精即可直接获得查询结果. 获得查询结果的用户较早地离开系统,释放相应资源,提高了其他用户的处理效率.

系统处理流程如下:移动用户提出查询请求,提交给 TAP;TAP 中的查询分类器辨别查询类型,如果是最近邻查询,则直接交由知识库判断该用户所在单元;将所在单元的种子作为查询结果返回给用户;如果用户提出的不是最近邻查询,则由匿名引擎启动位置匿名算法,为用户寻找匿名集;在匿名成功后将匿名结果发送给服务提供商进行查询处理. 最后,TAP 中的查询结果求精引擎根据用户的真实位置将求精后查询结果返回给相应用户.

3　预备知识

3.1　网络 Voronoi 图

采用无向图 $G(V,E)$表示路网,其中 V 是点集,包括交叉点($d(v)\geqslant 3$)、终点($0<d(v)\leqslant 2$)和感兴趣点 POIs($d(v)$表示点 v 的度);E 是边集,代表 2 点间的路段组成的集合①. 如果一个点 p 在边 e 上,则表示为 $p\in e$. 图 1 中显示了一个路网实例.

NVD 是 Voronoi 图的特例. 在 NVD 中,移动对象的运动被限制在图中连接 2 点的边上. 在给出 NVD 具体定义前,首先给出一些相关定义. 设图 $G(V,E)$的点集 $V=\{p_1,p_2,\cdots,p_n,p_{n+1},\cdots,p_o\}$. V 中包含 o 个点,其中前 n 个点表示 NVD 的种子. 设图 G 的边数 $|E|=m$,则点的支配区域和边界点定义如下[13].

定义 1. 支配区域(dominance region). $\forall i,j\in \mathbb{N}^+$ 并且 $j\neq i$,$p_i,p_j\in V$,则点 p_i 相对于点 p_j 的支配区域为

$$Dom(p_i,p_j)=\{p\mid p\in\bigcup_{o=1}^{m}e_o,\ d(p,p_i)\leqslant d(p,p_j)\},$$

其中,$d(p,p_i)$是点 p 与点 p_i 之间的网络距离.

定义 2. 边界点(border points). 点 p_i 和点 p_j 之间的边界点为

$$b(p_i,p_j)=\{p\mid p\in\bigcup_{o=1}^{k}e_o,d(p,p_i)=d(p,p_j)\},$$

称 $b(p_i,p_j)$为在点 p_i 和点 p_j 之间的边界点集合. $b(p_i,p_j)$的物理含义是,在边集合 E 上且属于 $b(p_i,p_j)$的点,距离 p_i 和 p_j 的距离相同.

定义 3. Voronoi 边集合. 与点 p_i 相关的 Voronoi 边集合满足:

$$V_{\text{edge}}(p_i)=\bigcap_{j\in\mathbb{N}^+\setminus\{i\}}Dom(p_i,p_j).$$

Voronoi 边集合是指位于边集合 E 中某一边上的点,到种子点 p_i 的距离比到其他任何一个种子点的距离都近.

基于定义 3,定义 NVD 如下.

定义 4. NVD 定义为

$$NVD(P)=\{V_{\text{edge}}(p_1),V_{\text{edge}}(p_2),\cdots,V_{\text{edge}}(p_n)\},$$

其中 $P=\{p_1,p_2,\cdots,p_n\}$是种子集合. NVD 中的元素除边界点外,相互独立不交叉. 如果将边界点 a 和种子点 p_i 连接,并且连接过程中不与路网上的边交叉,则称由连接虚线组成的区域为网络 Voronoi 单元. NVD 中,不同的网络 Voronoi 单元由种子标识,如种子点 p_i 的网络 Voronoi 单元,记作 $NVC(p_i)$.

例如$\{p_1,p_2,p_3\}$是图 1 所示路网的种子集合. 该路网被划分为 3 个 NVC,$NVD(P)=\{V_{\text{edge}}(p_1),V_{\text{edge}}(p_2),V_{\text{edge}}(p_3)\}$. 与种子点 p_2 相关的 Voronoi 边集合 $V_{\text{edge}}(p_2)=\{\langle p_2,p_5\rangle,\langle p_2,p_{11}\rangle,\langle p_5,p_7\rangle,\langle p_5,b_1\rangle,\langle p_7,b_8\rangle,\langle p_7,b_7\rangle,\langle p_{11},b_6\rangle,\langle p_{11},b_5\rangle\}$. 边界点包含 $b_1\sim b_8$,其中$\{b_1,b_2,b_8\}$是 p_1 和 p_2 之间的边界点. 很明显,每一个 NVC 包含的点和边组成了道路网络的一个子图.

为方便第 4 节算法描述,定义 r-路径邻居和开放点如下.

定义 5. r-路径邻居(r-path neighbor). 设 $G_1(V_1,E_1)$和 $G_2(V_2,E_2)$是图 $G(V,E)$的子图,其中

① 本文设无向图的边权为 1 个标准单位. 如果实际中存在一条较长路段,则根据标准单位长度将该路段分解为多条边.

$V_1, V_2 \subset V, E_1, E_2 \subset E$. 设在图 G 中存在一条从 v_s 到 v_d 的路径，记为 $path_{v_s-v_d}$；路径长度是从 v_s 到达 v_d 经过的边数，记为 $length(path_{v_s-v_d})$. 特殊情况，若 $v_s=v_d$，则 $length(path_{v_s-v_d})=0$. 如果：

$$\min_{\forall v_s \in V_1, \forall v_d \in V_2} length(path_{v_s-v_d}) = r,$$

则称 G_1 和 G_2 是 r-路径邻居($r \geqslant 0$)(当 $v_s=v_d$ 时，等号成立)，$path_{v_s-v_d}$ 称为 r-路径.

图 3 中 G_1, G_2, G_3 是图 G 的 3 个子图. 子图中的边用实线表示，在图 G 中但没有在任何一个子图中的边用虚线表示. G_1 和 G_2 通过 r-路径$\langle p_4, p_5 \rangle$构成了 1-路径邻居；G_1 和 G_3 由于共享点 p_9，构成 0-路径邻居. 很明显，2 个互为 r-路径邻居的子图的 r-路径不唯一.

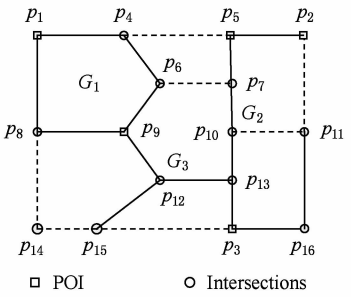


Fig. 3 r-path neighbor.

图 3 r-路径邻居

定义 6. 开放点(open vertices). 设 $G_1(V_1, E_1)$ 是图 $G(V, E)$ 的子图，其中 $V_1 \subset V, E_1 \subset E$. G_1 的开放点 V_{open} 满足：

$$V_{open} = \{v \mid \exists v' \in V - V_1, v \in V_1, \langle v, v' \rangle \in E - E_1\}.$$

根据定义 6，图 3 中 G_1 的 $V_{open}=\{p_4, p_6, p_8, p_9\}$.

3.2 隐私模型

为了保护用户在道路网络中的位置隐私，本文采用位置 K-匿名模型和空间粒度作为隐私标准. 因此，定义(K, L)-位置共享隐私模型如下.

定义 7. (K, L)-位置共享隐私模型. 设 S_C 是用户集合，S_R 是 S_C 中用户发布的匿名路段集，如果 S_C 和 S_R 同时满足以下条件：

1) $|S_C| \geqslant K$，其中 $|S_C|$ 是 S_C 中的用户数；

2) $|S_R| \geqslant L$，其中 $|S_R|$ 是 S_R 中的路段数；

3) $\forall u \in S_C, \exists e \in S, u \in e$；

4) $\forall u \in S_C$，u 发布 S_R 作为匿名区域.

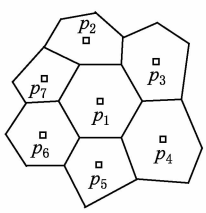
则 S_C 中的用户满足(K, L)-位置共享隐私模型.

条件 1 说明满足(K, L)-共享位置隐私模型的用户也满足位置 K-匿名模型；条件 2 确保了匿名区域中至少包含 L 条不同路段，反映空间粒度隐私标准；条件 3 说明 S_C 中的用户所在路段一定被包含在 S_R 中；条件 4 确保了 S_C 中的用户满足 K-共享(K-sharing)性质.

结合(K, L)-位置共享隐私模型，很容易将 NVC 分为安全单元和不安全单元 2 种. 如果直接将安全单元中的路段作为匿名路段集发布，其路段数不宜过多. 所以根据 NVC 覆盖的路段数($2L$ 为界)进一步将安全单元分为安全-中单元和安全-大单元. 具体定义如下.

定义 8. 不安全单元、安全-中单元、安全-大单元. 设 c_{cell} 是 1 个 NVC，c_{cell} 中包含的路段数(用户数)记为 $num\_edge(num\_user)$. 如果 $num\_edge < L$ 或者 $num\_user < K$，则 c_{cell} 是不安全单元；如果 $L \leqslant num\_edge < 2L$ 且 $num\_user \geqslant K$，则 c_{cell} 是安全-中单元；如果 $num\_edge \geqslant 2L$ 且 $num\_user \geqslant K$，则 c_{cell} 是安全-大单元.

图 4(a)是 1 个 NVD 的例子，图 4(b)是存储 NVD 的邻接表. 邻接表中的列 1 存储种子信息以及该单元覆盖的用户数(括号中数 1)和路段数(括号中数 2). 假设 $K=3, L=5$，则 $NVC(p_7)$ 是安全-大单元，$NVC(p_6)$ 是安全-中单元，$NVC(p_3)$ 是不安全单元.



(a) Sample of NVD

$p_1(3,4)$	$p_2, p_3, p_4, p_5, p_6, p_7$
$p_2(6,8)$	p_1, p_3, p_7
$p_3(5,2)$	p_1, p_2, p_4
$p_4(1,8)$	p_1, p_3, p_5
$p_5(7,10)$	p_1, p_4, p_6
$p_6(4,5)$	p_1, p_5, p_7
$p_7(3,10)$	p_1, p_2, p_6

(b) Sample of an adjacent list

Fig. 4 Network Voronoi graph.

图 4 网络 Voronoi 图

4 道路网络上基于 NVD 的位置匿名算法

4.1 NVD 的构建

本文采用并行 Dijkstra 算法[13]生成 1 阶 NVD，POIs 作为种子点，邻接表存储 NVD. 所有单元存储于 1 个 1 维数组中. 数组每一个元素指向 1 个链表，链表中存储了该单元在 NVD 中所有的邻接单元. 如果 2 个 NVC 共享边界点，则称这 2 个单元为邻接单元. 如图 4(b)所示，$NVC(p_2)$ 的邻接单元是 $NVC(p_1)$，$NVC(p_3)$ 和 $NVC(p_7)$. 为了快速定位移动用户所在单元，本文采用 NVC 的最小边界矩形(minimum bounding rectangle, MBR)近似表示每

一个单元,利用 R-树索引所有的 MBR. 如果移动用户位于 2 个 NVC 的边界处,则随机选取其中 1 个作为其所在单元.

4.2　NVLA 主要思想

定义 8 中将 NVC 分成为了不安全单元、安全-中单元和安全-大单元 3 种. 很明显,对于后 2 种安全单元,可以通过直接访问 NVC 中覆盖的局部路网成功寻找匿名集(具体细节参见 4.3 节). 如果可以将不安全单元转换成安全单元,则可采用统一策略. 所以 NVLA 的基本思想是:以(num_user,num_edge)作为关键字,为所有单元建立小顶堆. 取出堆顶元素 top_c. 如果 top_c 不安全,则将 top_c 与邻接单元合并重新插入堆. 如果 top_c 是安全单元,则从该安全单元中直接寻找匿名集. 重复这个过程,直至堆为空.

算法 1 显示了 NVLA 的具体步骤. 首先以(num_user,num_edge)作为关键字建立小顶堆 h(行①). 当堆不为空时,取堆顶 top_c(行③). 如果 top_c 包含的用户数小于 K,则从存储 NVD 的邻接表中找到 top_c 的邻接单元集合 $list$. 从 $list$ 中随机选取 1 个单元 $rand_c$. 组合 top_c 与 $rand_c$ 形成 new_c,将 new_c 重新插入小顶堆 h(行⑤～⑨). 如果 top_c 包含的路段数小于 L,则从 NVC 覆盖的连通子图中找到开放点集合,随机选取(L-top_c.num_edge)条与开放点邻接但却不包含在 top_c 中的路段. 随机选取的路段和原 top_c 覆盖的路段组成匿名路段集,top_c 覆盖的用户作为匿名集合(行⑫～⑭). 如果 top_c 是安全单元,则应用算法 4 寻找匿名集(行⑯). 重复执行行②～⑯,直至堆空.

算法 1. 基于网络 Voronoi 图的位置匿名算法 NVLA.

输入:NVCs c_{cells},匿名度需求 K,路段差异性性需求 L;

输出:匿名集合.

① 以(num_user, num_edge)为排序关键字将 c_{cells} 插入小顶堆 h 中;
② while h 不为空 {
③　top_c=从 h 中取堆顶;
④　if top_c.num_user<K 或者 top_c.num_edge<L then {
⑤　　if top_c.num_user<K {
⑥　　$list$=top_c 的邻居单元;
⑦　　$rand_c$=从 $list$ 中随机取一个单元;
⑧　　将 $rand_c$ 和 top_c 合并为 new_c;
⑨　　将 new_c 插入 h 并从 h 中删除 $rand_c$;
⑩　　重建堆 h; }
⑪　else {
⑫　　V_{open}=NVC 开放点;
⑬　　随机选取 L-top_c.num_edge 条与 V_{open} 邻接的路段插入 NVC;
⑭　　return top_c 中的用户作为匿名集;} }
⑮　else
⑯　　{利用算法 4 寻找匿名集;} }

以图 4 中的 NVCs 作为运行例子. 设 K=3,L=5,最初 top_c 是 $NVC(p_4)$. $NVC(p_4)$违反了(3,5)-位置共享隐私模型. 通过邻接表获得 $NVC(p_4)$ 的邻接单元包括{$NVC(p_1)$,$NVC(p_3)$,$NVC(p_5)$}. 设 $NVC(p_1)$被选中,合并 $NVC(p_1)$和 $NVC(p_4)$为新单元 $NVC(p_1p_4)$(包含 4 个用户),重新插入堆. 此时,堆顶元素更新为 $NVC(p_7)$,$NVC(p_7)$是安全-大单元,应用 4.3 节中的算法 4 寻找匿名集.

4.3　安全单元中的位置隐私保护方法

安全单元包括安全-中单元和安全-大单元. 对于安全-中单元,单元中包含的用户数和路段数恰好刚满足用户隐私需求,所以直接将单元中的所有用户作为匿名集、单元下覆盖的路段组成匿名路段集返回. 对于安全-大单元,由于其覆盖的路段数过多,如果直接把覆盖的所有路段作为匿名路段集发布将造成较高的查询处理代价. 另一种简单的方法是从提出查询的用户所在位置开始,在道路网络上重复寻找用户的 K 最近邻. 然而,众所周知,在道路网络上寻找最近邻是一项很耗时的工作.

根据安全-大单元中用户分布的密度,本文提出了基于用户分组的匿名方法和基于路段分组的匿名方法,用户密度用$\frac{num_user}{num_edge}$评价.

定义 9. 稀疏单元、非稀疏单元. 对于任意 NVC c_{cell},num_user 是 c_{cell} 下覆盖的移动用户数,num_edge 是 c_{cell} 下覆盖的路段数,如果$\frac{num_user}{num_edge}<\delta$,则 c_{cell} 是稀疏单元,否则是非稀疏单元.

基于用户分组的位置匿名算法 UCA :在稀疏单元中存在很多没有任何用户的空边. 为了尽量不扫描空边,本文提出了基于用户分组的匿名算法 UCA,主要思想是首先将所有的非空路段抽出来,组成新图 g'. 很明显,g'是非连通图. g'中的所有连通分量组成集合 $component\_set$. 针对每一个连通分量,若该连通分量覆盖的用户数和边数不满足(K,

L)-位置共享隐私模型,则将该连同分量与其 r-路径邻居合并,组成新连通分量.重复该合并过程,直至每一个连同分量均满足(K,L)-位置共享隐私模型.

具体算法参见算法 2.针对每一个稀疏单元 c_{cell},所有非空边组成非连通图 g'(行①).从 g'中找到所有连通分量,并组成集合 *component_set*(行②).针对 *component_set* 中所有连通分量,以覆盖的用户数为关键字按非降序排序.设 *component_set* 中首元素是 *component*.如果 *component* 中包含的用户数小于 K,则寻找 *component* 的 r-路径邻居,最初 $r=1$.如果 r-路径邻居存在多个,则选择包含用户数最小的连通分量(行⑥～⑨).如果 *component* 没有 r-路径邻居,则将 $r+1$,重复执行行⑦～⑨,直到找到 r-路径邻居 r_n 为止.将 r_n、r-路径 $path_{v_s-v_d}$ 与连通分量 *component* 合并(行⑩～⑬).如果首元素 *component* 包含的用户数不小于 K,则 *component_set* 中的其他项包含的用户数也不小于 K.因此,当 *component.num_user*$\geqslant K$ 时,可从行⑤～⑮的循环跳出.最后,判断 *component_set* 中的每一个连通分量覆盖的路段数是否小于 L,若是则通过开放点寻找邻接边加入到连通分量中(行⑯～⑳).

算法 2. 基于用户分组的位置匿名算法(UCA).

输入:稀疏单元 c_{cell},匿名度需求 K,路段差异性需求 L;

输出:匿名集合.

① c_{cell}中的非空边组成图 g';
② *component_set*=g'的连通分量;
③ 以*num_user* 为关键字对 *component_set* 中的元素非降序排序;
④ *component*=*component_set* 中第 1 个元素;
⑤ while *component.num_user*<K do {
⑥ for 每一个点 v *in component* do {
⑦ n_b =从点 v 开始从 c_{cell}中寻找 *component* 的一个 r-路径邻居;
⑧ if $r\_n.num\_user > n\_b.num\_user$ then
⑨ {$r\_n=n\_b$;} }
⑩ if *r_path* 存在 then {
⑪ 将 *r_path*,r_n 合并至 *component*;
⑫ 从 *component_set* 中删除 r_n 并重排序;
⑬ *component*=*component_set* 中第一项;}
⑭ else
⑮ {r++;} }
⑯ for 每一个 *component in component_set* do {
⑰ if *component.num_edge*<L then {
⑱ V_{open}=在 *component* 中与其他连同分量邻接的点;
⑲ 随机选取 *L-component.num_edges* 条与 V_{open}邻接的路段插入 *component*;}
⑳ return *component* 作为匿名集; }

继续 4.2 节中的例子,$top_c=NVC(p_7)$.设 $\delta=1$,$NVC(p_7)$覆盖的局部路网结构如图 5 所示.很明显,$NVC(p_7)$是稀疏单元.根据算法 2,g'包括 2 条非空边$\langle o_1,p_7\rangle$和$\langle o_4,b_6\rangle$,分别组成 2 个连通分量 G_1 和 G_2.G_1 和 G_2 通过 1-路径$\langle o_4,p_7\rangle$构成 1-路径邻居.将 G_1,G_2 以及 1-路径$\langle o_4,p_7\rangle$合并为新 *component*,合并后的 *component* 包含 3 条边{$\langle o_1,p_7\rangle$,$\langle o_4,b_6\rangle$,$\langle o_4,p_7\rangle$}和 3 个用户.开放点集 V_{open}(*component*)={o_1,p_7,o_4,b_6}.假设 o_1 被选中,与 o_1 邻接的边中{$\langle b_1,o_1\rangle$,$\langle b_2,o_1\rangle$,$\langle o_1,b_3\rangle$}不在 *component* 中.设边$\langle b_1,o_1\rangle$,$\langle b_2,o_1\rangle$被插入 *component* 中.最终 3 个用户组成匿名集,{$\langle o_1,p_7\rangle$,$\langle o_4,b_6\rangle$,$\langle o_4,p_7\rangle$,$\langle b_1,o_1\rangle$,$\langle b_2,o_1\rangle$}是发布的匿名路段集.

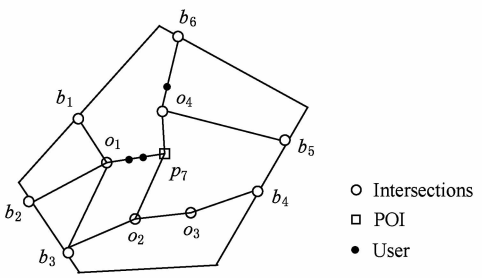


Fig. 5 Road network in $NVC(p_7)$.

图 5 $NVC(p_7)$中的局部路网拓扑

基于路段分组的位置匿名算法 SCA:对于非稀疏单元,几乎每一个路段上都包含用户.针对这种情况,受文献[5]启发,提出基于路段分组的位置匿名算法 SCA.SCA 基本思想是首先遍历所有的边,将所有边划分为 $\lfloor\frac{num_edge}{L}\rfloor$个组.每一组至少包含 L 条路段,最后 1 组包含 $2L-1$ 条路段.然后,检查每一组中包含的用户数,如果包含的用户数不小于 K,则返回组内用户为匿名集,组中覆盖路段为匿名路段集;否则,将该组与另一组合并,直至包含的用户数不小于 K.具体算法参见算法 3.

算法 3. 基于边的位置匿名算法 SCA.

输入:非稀疏 NVC c_{cell},匿名度需求 K,路段差异性需求 L;

输出:匿名集合.

① 随机选取起始点 v 在 c_{cell} 中进行深度或宽度优先遍历;
② 以遍历顺序为边标号;
③ 边序号从 $(i-1)L+1$ 到 $i\times L$ 的边为一组;
④ for 每一组 g_p do {
⑤　if $g\_p.num\_user \geqslant K$ then
⑥　　{return g_p 作为匿名集;}
⑦　else
⑧　　{g_set=随机选取 g_p 的 0-路径邻居;}
⑨　将 g_p 与 g_set 合并;}

从 NVC c_{cell} 中随机选取 1 点 v,从 v 开始对 c_{cell} 覆盖下的子图进行深度或宽度优先遍历. 与传统遍历算法不同的是,这里每一条边仅被访问 1 次而非每个点(行①). 按照访问顺序给每一条边分配 1 个访问序号(行②). 接着,每 L 个序号连续的路段分为一组(行③). 具体来说,序号为 1 到 L 的边为一组,序号为 $L+1$ 到 $2L$ 的边为一组,以此类推. 一般情况,第 i 组包含的边序号为 $(i-1)L+1$ 至 $i\times L$,其中 $0<i\leqslant\left\lfloor\frac{num\_edge}{L}\right\rfloor$. 在完成组的划分后,针对每一组 g_p,如果 g_p 中包含的用户数少于 K,g_p 随机与它的 1 个 0-路径邻居合并(行⑨);否则 g_p 中的用户组成匿名集,g_p 下覆盖的路段作为匿名路段集(行⑥).

以 $NVC(p_5)$ 为例,该 NVC 下的局部路网拓扑如图 6 所示. 设从 b_4 开始,执行深度优先遍历,并为每一条边标号,图 6 中边上带括号的数字如 (i) 即边序号. 如此,$NVC(p_5)$ 下覆盖的路段被分成 3 组,不同组在图 6 中用不同类型的线表示. 3 组均满足 (3,5)-位置共享隐私模型.

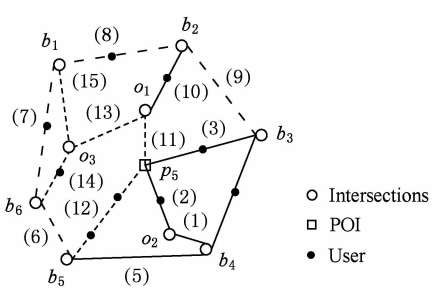


Fig. 6　$NVC(p_5)$.

图 6　$NVC(p_5)$ 中的局部路网拓扑

综上所述,针对每一个安全单元 c_{cell},如果 c_{cell} 是安全-中单元,c_{cell} 中包含的用户组成匿名集,c_{cell} 下覆盖的路段组成匿名位置(行①和行②). 如果 c_{cell} 是安全-大单元,进一步判断 c_{cell} 中移动用户的密度:如果是稀疏单元,则应用 UCA 寻找匿名集(行⑥);否则应用 SCA 寻找匿名集(行⑧). 具体参见算法 4.

算法 4. 安全单元中基于密度的位置匿名算法.

输入:NVC c_{cell},匿名度需求 K,路段差异性需求 L;

输出:匿名集合.

① if $c_{cell}.num\_edge\geqslant L$ and $c_{cell}.num\_edge<2L$ then
②　{return c_{cell} 为匿名集;}
③ else
④　$\{density=\frac{num\_user}{num\_edge}$;
⑤　　if $density<\delta$ then
⑥　　　{应用算法 2;}
⑦　　else
⑧　　　{应用算法 3;} }

安全分析:由于 NVLA 产生的匿名集符合 (K,L)-位置共享隐私模型,所以匿名集同时满足位置 K 匿名和降低位置信息粒度的隐私标准. 位置 K 匿名模型可以成功的保护用户标识;降低位置粒度可以有效的保护了用户位置信息,用户确切位置隐藏于 L 个不同的路段中. 此外,算法采用的 K-共享和随机策略均可保证相同的输入产生不同的匿名集结果,进而防止道路网络上特有的回放攻击(replay attack)[8]. 随机选取策略在 NVLA 中体现在 4 处:

1) 当用户位于 2 个 NVC 的边界时,用户所在单元是随机选取的;

2) 不安全单元在与其邻接单元合并时,合并单元是随机选取的;

3) 当路段集合包含的路段数小于 L 时,从开放点集合中随机选取 1 点并随机选取邻接边加入;

4) 在 SCA 中,为某个分组选取 0-路径邻居时,当存在多个 0-路径邻居,合并对象随机选取.

5　实验结果与分析

5.1　实验设置

实验中比较了 TreeCA,GTCA,NVLA 三个算法,其中 TreeCA 是文献[11]中基于树型索引的隐私保护方法;GTCA 是从文献[5]中基于深度优先遍历的匿名算法修改而来. 选用基于深度优先遍历的算法因为文献[5]证明并验证了该算法在所有遍历算法中具有最小查询处理代价. 由于在文献[5]中忽略了路段差异性,所以当匿名集中覆盖的路段数

小于 L 时,GTCA 采用算法 1 中行⑫～⑬的随机策略选取合适的邻接边加入匿名路段集.所有算法采用 C++实现,在双核 AMD 780 MHz 处理器和 2 GB 内存的台式机上运行.

实验采用美国加利福尼亚真实公路网络[14],数据集包含 21 048 个顶点、21 693 条边,在该公路网络上附着 32 399 个真实的感兴趣点,本文采用 1%的感兴趣点作为种子生成 1 阶 NVD.利用所有的感兴趣点模拟提出非最近邻查询的移动用户.默认情况下,匿名度需求 K 和路段差异需求 L 均设置为 10,用户密度阈值 $\delta=1$,默认系统设置如表 1 所示:

Table 1 Default System Settings

表 1 实验参数及默认取值

Parameter	Default Value
Number of users	32 399
Number of nodes	21 048
Number of edges	21 693
K	10
L	10
δ	1

5.2 评价标准

采用的评价标准包括匿名成功率、信息熵、相对匿名度、相对路段差异性、匿名时间、查询处理代价.

1) 匿名成功率是指成功匿名的移动用户在所有提出查询的移动用户中所占比例.匿名成功率越高,说明匿名算法对查询响应能力越好.

2) 信息熵[8]反映匿名算法为移动用户提供的保护强度,用概率分布的信息熵衡量.信息熵越大,用户被猜出在匿名路段集中的具体位置的概率就越小,也就是说保护强度越好.

3) 相对匿名度是匿名集中实际包含的用户数与用户提出的匿名度需求的比值.该比值一定大于等于 1,比值越大说明匿名集中包含的移动用户数越多,用户标识被泄露的可能性越小.

4) 相对路段差异性与相对匿名度类似,是匿名路段集中包含的实际路段数与用户提出的路段差异性需求的比值.该比值也不小于 1,比值越大说明匿名路段集中覆盖的路段数越多,用户的确切位置被保护得越好.

5) 匿名时间指的是一定规模移动用户的查询请求在多长时间内可以得到匿名处理.这是反映匿名算法好坏的重要指标之一.匿名时间越短越好,说明了匿名算法的高效性.

6) 查询处理代价是指服务提供商处理具有匿名位置的查询产生的额外代价.实验利用文献[7]中的查询代价模型,使用匿名路段集中包含的路段长度和开放点个数评价查询代价.查询代价越小越好.

5.3 匿名度需求

此节评价了匿名度 K 对匿名算法性能的影响.匿名度 K 从 10 增加到 100,随着匿名度 K 的增加,意味着每一个匿名集需要包含更多用户.从图 7(a)观察到,匿名需求的变化对 3 个匿名算法的成功率无影响,均保证 100%的匿名成功率.从图 7(b)观察到,3 个算法的信息熵均随着匿名度 K 的增加而逐渐增加,说明随着匿名需求逐渐严格,用户获得了更安全的隐私保护.其中 TreeCA 的信息熵最高,NVLA 次之.说明 TreeCA 较其他 2 个算法提供了较高的隐私保护强度,这一点亦可从图 7(c)(d)看出.但另一方面,这种高隐私保护度是牺牲了较高的查询处理代价和较长的匿名时间为前提的(如图 7(e)(f)所示).

图 7(c)显示了 3 个算法的相对匿名度随着 K 增长的变化情况.随着 K 的增加,TreeCA 匿名集覆盖的用户数与匿名度需求更接近,所以相对匿名度逐渐降低.然而,即使当匿名度需求增加到 100 时,TreeCA 的相对匿名度是 NVLA 的 58 倍,是 GTCA 的 96 倍.由于差距较大,从图 7(c)中较难分辨 NVLA 和 GTCA.在所有的匿名度设置上,NVLA 的相对匿名度比 GTCA 高.NVLA 的相对匿名度随着 K 的增加而降低,因为随着用户匿名度要求的严格,匿名集中包含的用户数量更接近用户要求.然而,GTCA 的相对匿名度无论在何种设置下均为 1,因为 GTCA 是直接根据匿名度为用户分组的.

图 7(d)展示了 TreeCA,NVLA 和 GTCA 算法的相对路段差异性的变化情况.随着匿名度需求的增加,TreeCA 的相对路段差异性先呈现增加势态,因为匿名集为满足更高的匿名度需求覆盖了更多边.但是当匿名度继续增加时,相对路段差异性震荡于 700～750 之间.也就是说,当匿名度需求超过一定值时,TreeCA 递归至树的一定层次后由于覆盖的用户路段数和用户数过多,均可满足后面增大的匿名度需求.NVLA 和 GTCA 的相对路段差异性均随着匿名度的增加而增加,其原因是随着 K 的增加匿名集中包含了更多用户.而这些更多的用户散落在不同的边上,致使匿名路段集中的路段数增加.在相对路段差异性方面,TreeCA 再次展现了较高的保护强度,NVLA 次之.

图 7(e)显示 3 个算法的匿名时间均对 K 的变化不敏感. NVLA 的平均匿名时间随着 K 的增长变化幅度很小,从 0.34 ms 缓慢增长到了 0.37 ms. 随着 K 的增加,NVLA 更多的时间消耗在了单元合并上. GTCA 的匿名时间随着 K 的增加逐渐变小,因为 K 的增加致使用户分组数变少. TreeCA 随着 K 的增加保持不变. 在默认设置下,TreeCA 的匿名时间是 NVLA 的 5 倍,是 GTCA 的 2.7 倍. NVLA 在算法效率方面表现最好. 这是因为,NVLA 在局部 NVC 中寻找匿名集;GTCA 需要重复遍历整个道路网络;TreeCA 的时间主要消耗在扫描全局路网信息以及树索引的建立和更新上.

如图 7(f)显示,TreeCA,GTCA 和 NVLA 的查询代价均随着 K 的增加而增加,其中 TreeCA 的查询代价最高,GTCA 最小. 在自底向上的递归过程中,TreeCA 每次直接增加兄弟结点下覆盖的路段,使其匿名路段集包含的路段数过多. 随着匿名度需求的增加,平均查询代价恶化. 当 $K=100$ 时,TreeCA 是 NVLA 的 32 倍,是 GTCA 的近 55 倍. 相比之下,在 $K=10$ 时,NVLA 仅比 GTCA 多花费了 0.01% 的查询代价;当 K 增加到 100 时,这种额外的查询处理代价仅增长了 0.07%. 显然,NVLA 比 GTCA 提供了更好的隐私保护强度,比 TreeCA 更高效、查询代价少,在隐私保护和服务质量之间取得了较好平衡.

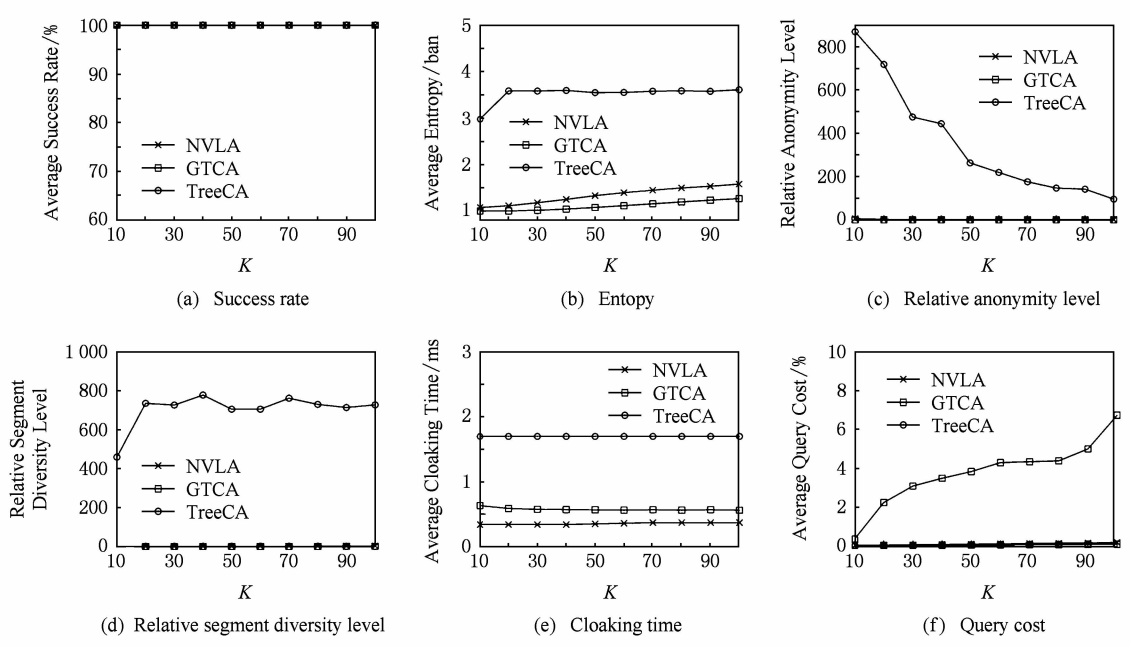


Fig. 7　Evaluation of algorithms using six evaluation metircs over different anonymity requirement level K.

图 7　3 个算法在 6 个评价标准上随匿名度需求 K 变化比较

5.4　路段差异性

路段差异性需求 L 增加意味着用户的位置隐私需求更加严格,L 从 10 增加到 100. 从图 8(a)看出,3 个算法的平均成功率并没有随着 L 的增加发生变化,均可保证 100% 的匿名成功率. 同时,随着 L 的增加,匿名集为用户提供了更安全的保护,这一点可从图 8(b)看出. 随着 L 增加,TreeCA,NVLA 和 GTCA 的信息熵均逐渐增加. 与图 7(b)类似,TreeCA 的信息熵最高.

图 8(c)展示了 3 个算法的相对匿名度随着路段差异性变化的变化情况. TreeCA 比 GTCA 和 NVLA 提供了更高的相对匿名度. 由于差距较大,从图 8(c)中较难分辨 NVLA 和 GTCA. 当 L 从 10 增长到 20 时,TreeCA 相对匿名度逐渐增加,但是当 L 超过 20 时,相对匿名度基本保持不变(将近 1 500),且远远超出用户的隐私需求. 这是因为 TreeCA 从第 1 层非叶子结点到根结点逐层检查,每次从孩子结点递归到父结点时最多增加 $2^{\lfloor \mathrm{lb}\, n \rfloor - i}$ 条路段,其中 n 是移动用户总数,i 为当前扫描结点的深度. 增加的路段数以及这些路段下覆盖的用户远远超过用户的隐私需求,这一点亦可从图 8(d)中得到印证.

在所有设置上,NVLA 的相对匿名度比 GTCA 高. 随着路段差异性需求的增加,NVLA 的相对匿名度也随之增加. 在 NVLA 中,L 增加导致安全-中单元和不安全单元的数量增加. 这 2 种单元数量的增加均导致匿名路段集中包含的路段数和用户数增多. 所以当 L 从 10 增长到 30 时,相对匿名度增加较

快. 之后 L 继续增加,对匿名集中用户数的影响并不明显,只会造成由于路段差异性需求不满足,添加邻接边的情况. 因此,当 $L>30$ 时,相对匿名度增加不明显. 相比之下,GTCA 的相对匿名度保持在 1,刚刚满足用户的隐私需求.

图 8(d)中显示了 3 个算法的相对路段差异性随着 L 增加的变化情况. TreeCA 的相对路段差异性随着位置差异性需求的增加而降低. 这一点容易理解,因为实际覆盖的路段数更接近用户位置差异性需求. 但是即使 L 增加到 100,TreeCA 的相对路段差异性依然是其他 2 个算法的近 73 倍. 随着 L 的增加,NVLA 的相对路段差异性逐渐减少. L 增加致使更多的 NVCs 在算法初始被界定为安全-中单元和不安全单元,匿名集中覆盖的路段数更接近用户的路段差异性需求. 但是 L 增加对 GTCA 的相对路段差异性无影响. 因为 GTCA 根据用户匿名度需求分组,匿名集覆盖的匿名路段集严格按照路段差异性需求寻找,致使匿名路段集中覆盖的路段数恰等于用户所需,所以相对路段差异性始终为 1.

从图 8(e)看出,GTCA 的匿名时间随着 L 的增加而增加. 这主要因为更多的时间被消耗在扫描整个路网结构,寻找邻接边以满足更严格的路段差异性需求. 然而,NVLA 的匿名时间先降低然后保持在 0.2 ms. 当 L 从 10 增加到 30 时,更多的 NVCs 界定为安全-中单元,无需更多处理即可通过访问 NVC 中的局部路网获得匿名集和匿名路段集,所以匿名时间降低. 当 L 继续增加时,更多的单元由于路段差异性不满足而变成了不安全单元,执行算法 1 中行⑫～⑬,在局部路网中通过开放点寻找邻接边加入匿名路段集. 从图 8(e)中看出当 $L>30$ 时,匿名时间在 0.2 ms 上下摆动. TreeCA 的匿名时间随着路段差异性需求的变化保持不变. 在默认设置下,TreeCA 的匿名时间是 NVLA 的 5 倍,是 GTCA 的 2.7 倍. 从总体情况看,在所有设置上,NVLA 的算法效率最高,TreeCA 的平均匿名时间表现最坏.

随着 L 增加,匿名路段集中包含路段数随之增加. 因此,如图8(f)所示,3个算法的查询代价均随着 L 的增加而增加. 随着 L 增加,TreeCA 平均查询代价急剧恶化. 在 $L=100$ 时,TreeCA 的平均查询处理代价是 NVLA 的近 19 倍. GTCA 再次展现了良好的查询处理代价. 由于 NVLA 为用户提供了更高的相对匿名度和相对路段差异性,所以 NVLA 的查询代价比 GTCA 高. 但随着 L 的增加,NVLA 与 GTCA 查询代价的差距增加缓慢.

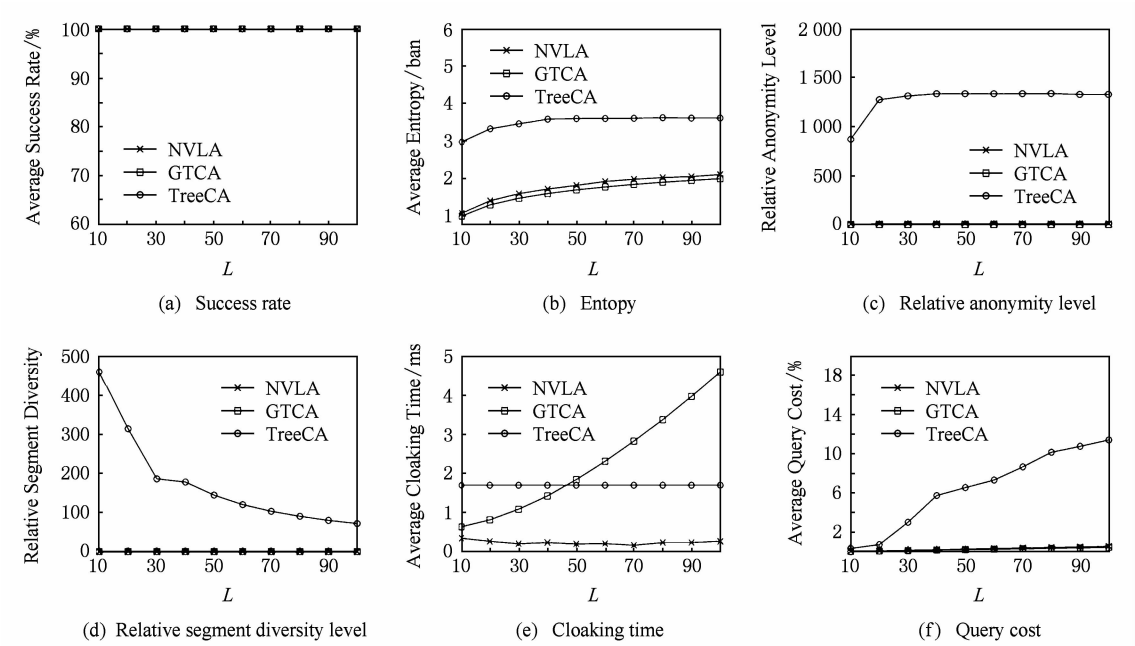


Fig. 8 Evaluation of algorithms using six evaluation metircs over different segment diversity levels L.

图 8 3 个算法在 6 个评价标准上随着路段差异性 L 变化的比较

5.5 算法可扩展性

本节通过增加系统中待匿名用户数量评价算法的可扩展性. 用户数量反映了服务区中的用户密度与系统工作量. 模拟生成了 10 000～100 000 的移动用户,这些模拟用户围绕 POIs 均匀分布在美国加利福尼亚公路网络上.

图 9(a)显示了 3 个算法在所有设置上均可达到 100%成功率. 图 9(b)显示随着用户数的增加,

TreeCA 平均信息熵逐渐降低.这是因为随着用户数量的增加,单位路段上覆盖的用户数亦增加,所以相同隐私需求设置下匿名集覆盖的路段数降低.然而,TreeCA 的平均信息熵在所有设置上依然大于 GTCA 和 NVLA.GTCA 的平均信息熵随着用户数的增加保持不变,这是因为 GTCA 严格按照用户的隐私需求(K,L)寻找匿名集,与待匿名用户数量无关.相比之下,NVLA 的平均信息熵先随着用户数的增加逐渐降低,当用户数大于 50 000 后,平均信息熵保持不变.这是因为随着用户密度的增加,NVLA 在较少的路段上即可找到满足用户匿名需求的用户数(此点亦可从图 9(c)得到证实).但是随着用户数量的继续增加,虽然更少的路段数量即可包含足够的匿名用户,但由于用户位置差异性需求的存在,使得 1 个匿名集中包含的平均路段数和用户数保持不变,进而信息熵亦不变.但是在所有设置上,NVLA 依然优于 GTCA.

图 9(c)和图 9(d)展示了用户数量对相对匿名度和相对路段差异性的影响.图 9(c)显示随着用户数量增加,TreeCA 相对匿名度逐渐降低.这是因为在单位路段上包含的用户数逐渐增多,TreeCA 自底向上寻找匿名集的层数也逐渐降低,覆盖的路段数亦减少.图 9(d)显示,随着用户密度的增加,TreeCA 的相对路段差异性逐渐减少,但当用户增加到 100 000 时,TreeCA 依然是 GTCA 和 NVLA 的 2 倍.由于用户匿名度(K)和路段差异性需求(L)不变,所以 GTCA 的相对匿名度和相对路段差异性没有随着用户数量的变化而变化.图 9(c)显示 NVLA 相对匿名度随着用户数量的增加而增强,这源于单位路段上用户数量的增加.如图 9(d)所示,NVLA 的相对路段差异度随着用户数量的增加先降低,当用户数量超过 50 000 时保持平稳,与图 9(b)中的平均信息熵的变化趋势一致.然而,无论在相对匿名度还是路段相对差异性方面,在所有设置上 NVLA 比 GTCA 均表现出了更高的保护强度.

图 9(e)展示了 3 个算法平均匿名时间的变情况.随着系统工作量的增加,NVLA 的平均匿名时间缓慢增加,当用户超过 60 000 时趋于稳定.当用户数量为 10 000 时,NVLA 的平均匿名时间为0.2 ms;当用户数量增加到 100 000 时,平均匿名时间仅为 0.45 ms.对于 GTCA 而言,无论移动用户数量是多少,遍历整个路网的时间是固定的.但是,当用户数量较少时,用户分布较稀疏.匿名时间主要消耗在寻找用户所在的路段上.随着用户密度的增大,单位路段上覆盖的用户数增加,提高了寻找用户所在路段效率,所以匿名时间减低.然而,如果用户密度继续增大,极少数的路段包含的用户数即可满足用户的匿名度需求,更多的时间消耗在了通过开放点寻找邻接边以满足位置差异性需求.TreeCA 的平均匿名时间随着用户数量的增加逐渐降低.本质上,TreeCA

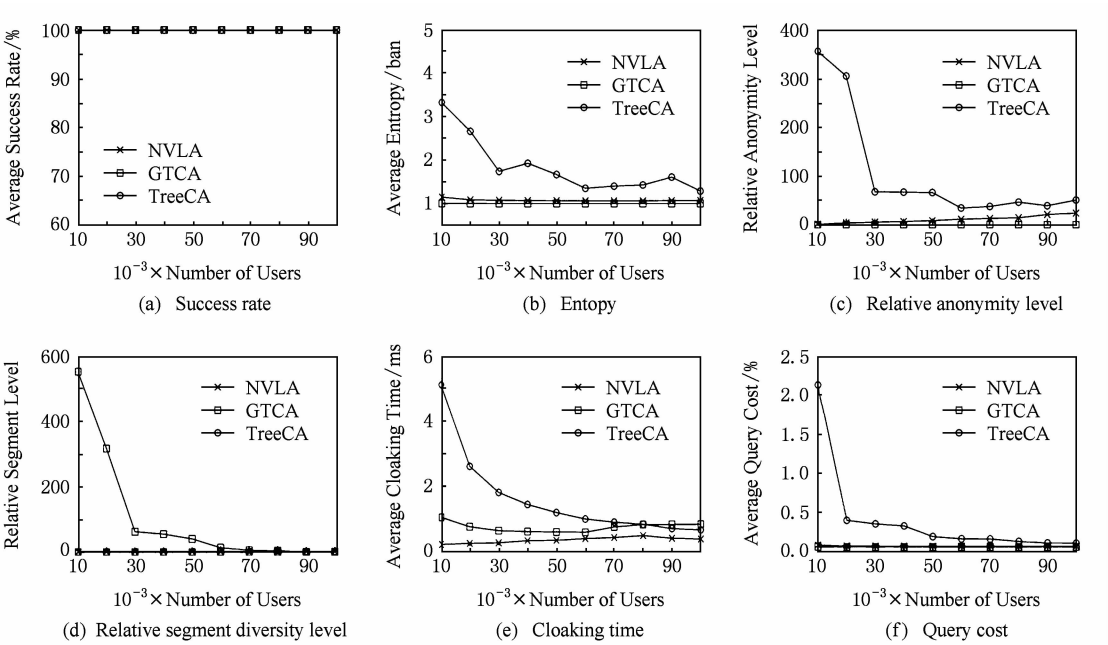


Fig. 9　Evaluation of algorithms using six evaluation metircs over different number of users in system.

图 9　3 个算法在 6 个评价标准上随着系统用户数量的比较

的时间主要消耗在树的建立和路网扫描上,无论用户数量是多少,总的匿名时间变化不大.然而,由于用户数的增多,单位路段上覆盖的用户数增加,致使一次匿名为更多的用户提供了保护,进而平均到每位用户的匿名时间降低了.在默认设置下,TreeCA的匿名时间是NVLA的5倍,是GTCA的2.7倍.

图9(f)显示了3个算法查询代价的变化情况.从图9(f)看出,NVLA的查询代价先降低后保持平稳,原因与前面类似,用户分布松散造成1个匿名集中包含较多的路段数和边界点.随后,虽然用户密度增加,但由于隐私需求未发生变化,用户的匿名集包含的用户数、路段数等保持不变,所以查询代价维持1个稳定值.GTCA在不同用户数量上显示了相同查询代价.当用户数量为100 000时,NVLA的查询代价是0.07%,而GTCA是0.06%.换句话说,NVLA仅牺牲了约0.01%的查询代价得到了更安全和更高效的保护.虽然随着用户数增加,TreeCA的查询代价会减少,但是即使用户数到达100 000,TreeCA依然是NVLA的1.6倍,是GTCA的1.8倍.所以尽管TreeCA提供了更好的保护强度,但是由于匿名时间长和查询代价高,TreeCA实用性不高.

5.6 算法比较小结

图7至图9显示了当匿名度、路段差异性和用户数量变化时,TreeCA,NVLA,GTCA在匿名成功率、信息熵、相对匿名度、相对路段差异性、匿名时间和查询处理代价6个评价标准上的变化情况.

从总体情况来看,虽然TreeCA在信息熵、平均匿名度和相对匿名度3个方面展现良好性能,较其他2个算法提供了更好的隐私保护程度,但是其匿名时间和查询代价较NVLA和GTCA 2个算法很高.这种通过牺牲较多查询代价换取隐私保护的做法不实用.相比之下,NVLA比GTCA提供了安全的隐私保护,比TreeCA算法效率更高,牺牲的查询处理代价小.因此,NVLA在隐私保护强度和算法性能方面取得了较好的平衡,实现了牺牲较少查询代价的前提下提供较安全的隐私保护强度.基于5.2~5.5节实验结果,表2将NVLA,TreeCA,GTCA的优点和缺点进行了总结.

Table 2 Summary for the Experiments

表2 实验效果总结

Representative Method	Anonymization Technique	Good Points	Bad Points
TreeCA	Tree-type index-based cloaking algorithm	The entropy, the relative anonymity level, and the relative segment diversity level are higher.	The cloaking time and the query cost are both high.
NVLA	Graph traversal-based cloaking algorithm	The entropy, the relative anonymity level, and the relative segment diversity level are high. Moreover, the method is efficient	Query cost is acceptable
GTCA	Graph traversal-based cloaking algorithm	The method is more efficient, and the query cost is low.	The entropy, the relative anonymity level, and the relative segment diversity level are low.

6 结 论

本文提出了一种在道路网络上基于网络Voronoi图NVD的高效位置隐私保护算法.现有大部分道路网络上的位置匿名算法需要多次重复扫描全局网络,致使匿名效率低、网络传输代价高.为解决此问题,本文利用NVD先将道路网络划分成独立的网络Voronoi单元,针对不同单元的特点提出了不同的位置匿名算法,尤其针对安全-大单元分别提出了基于用户分组和基于路段分组的位置匿名算法.最后通过一系列实验,验证了提出算法的有效性;匿名算法可保证100%的匿名成功率,平均匿名时间仅为0.34 ms;在隐私保护强度和算法性能方面取得了较好的平衡.

参 考 文 献

[1] Terrovitis M, Liagouris J, Mamoulis N, et al. Privacy preservation by disassociation[J]. Proceedings of the VLDB Endowment, 2012, 5(10): 944-955

[2] Wang Lu, Meng Xiaofeng. Location privacy preservation in big data era: A survey[J]. Journal of Software, 2014, 25(4): 693-712 (in Chinese)
(王璐,孟小峰. 位置大数据隐私保护研究综述[J]. 软件学报, 2014, 25(4): 693-712)

[3] Ku W, Zimmermann R, Peng W C, et al. Privacy protected query processing on spatial networks [C] //Proc of the 23rd IEEE Int Conf on Data Engineering Workshop. Piscataway, NJ: IEEE, 2007: 215-220

[4] Mohamed F M, Chow C Y, Walid G A. The new Casper: Query processing for location services without compromising privacy [C] //Proc of the 32nd Int Conf on VLDB. New York: ACM, 2006: 763-774

[5] Mouratidis K, Yiu M L. Anonymous query processing in road networks [J]. IEEE Trans on Knowledge and Data Engineering, 2009, 22(1): 2-15

[6] Wang T, Liu L. Privacy aware mobile services over road networks [C] //Proc of the 35th Int Conf on VLDB. New York: ACM, 2009: 1042-1053

[7] Chow C, Mokbel M F, Bao J, et al. Query-aware location anonymization for road networks [J]. Geoinformatical, 2010, 15(3): 571-607

[8] Xue Jiao, Liu Xiangyu, Yang Xiaochun, et al. A location privacy preserving approach on road network [J]. Chinese Journal of Computers, 2011, 34(5): 865-878 (in Chinese)
(薛姣, 刘向宇, 杨晓春, 等. 一种面向公路网络的位置隐私保护方法[J]. 计算机学报, 2011, 34(5): 865-878)

[9] Palanisamy B, Liu L. MobiMix: Protecting location privacy with mix-zones over road networks [C] //Proc of the 27th IEEE Int Conf on Data Engineering (ICDE). Piscataway, NJ: IEEE, 2011: 494-505

[10] Palanisamy B, Liu L, Lee K, et al. Location privacy with road network mix-zones [C] //Proc of the 18th IEEE Int Conf on Mobile Ad-hoc and Sensor Networks. Piscataway, NJ: IEEE, 2012: 124-131

[11] Li P, Peng W C, Wang T W, et al. A cloaking algorithm based on spatial networks for location privacy [C] //Proc of the 2nd IEEE Int Conf on Sensor Networks, Ubiquitous and Trustworthy Computing. Piscataway, NJ: IEEE, 2008: 90-97

[12] Mouratidis K, Yiu M Y. Shortest path computation with no information leakage [J]. Proceedings of the VLDB Endowment, 2012, 5(8): 692-703

[13] Erwig M, Hagen F. The graph Voronoi diagram with applications [J]. Journal of Networks, 2000, 36(3): 156-163

[14] Li F, Cheng D, Hadjieleftheriou M, et al. On trip planning queries in spatial databases [G] //LNCS 3633: Proc of the 9th Int Symp Advances in Spatial and Temporal Databases. Berlin: Springer, 2005: 273-290

Pan Xiao, born in 1981. PhD. Assistant professor at Shijiazhuang Tiedao University. Member of China Computer Federation. Visiting scholar in the Department of Computer Science, University of Illinois at Chicago, USA. Her main research interests include mobile computing, social network, and privacy protection.

Wu Lei, born in 1980. Master. Lecturer at Shijiazhuang Tiedao University. Member of the Soft Science Research Institute on Engineering and Construction Management in Hebei province. His research interests include data management on moving objects and location based social networks (outhunder@126.com).

Hu Zhaojun, born 1993. A junior student at Shijiazhuang Tiedao University. Her research interests include privacy preservation computing (hzj18330108553@163.com).