

项目编号：

# “筹政基金”

## 大学生进习研修计划结题报告

课题名称 道路网络中带关键字的  $k$  近邻查询

申 请 人 陈炜栋

院 系 计算机科学与技术

导 师 孙未未

填表日期 2014 年 4 月 18 日

复旦大学教务处制表

## 莒政项目结题须知

- 每学年春季学期第 12 周，已经中期报告通过的同学，可以申请结题。如果在本科毕业之前没有结题，视作课题中止，将不授予“莒政学者”称号。
- 申请结题的学生，应在 FDUROP 网站（<http://www.fdrop.fudan.edu.cn>）下载结题表格，填写完整后，并附《FDUROP 课题财务日志》（须有导师和学生本人手写签名）和课题感言，书面版交至光华楼西辅楼 703 室门口信箱；电子版结题报告（报告正文部分不得多于 12 页，请自行删减，但格式必须按照要求，否则不收入结题报告集）、感言、《FDUROP 课题财务日志》、以及不少于一张的个人生活照及与导师合影照片上传至 <ftp://10.99.7.52> (用户名密码都是 junzheng)。所有书面和电子版材料提交完成后，方正式接受同学的结题申请。两种材料递交不全者，不作评审；本说明及附录页可不递交。
- 学生应同时在 FDUROP 网站更新个人课题信息，提交结题相关资料。
- 如果您的课题或者后续研究的成果已经发表，也请您填写网站上的“成果发表情况”，为后来者树立榜样。
- 结题报告提交后 2 至 4 周，教务处实践教学办公室将在 BBS 和 FDUROP 网站发布结题评审结果，同时将结题材料归入莒政学者个人档案。
- 每年 6 月份最后一个星期四下午二点，在美国研究中心谢希德报告厅举行 FDUROP 结题典礼，届时将向结题合格学生及其导师颁发证书，并发放年度结题报告集。
- 我们希望“莒政学者”今后继续关注 FDUROP 并且参加我们的活动，给后来者以指导。

## 一、课题信息

课题名称	道路网络中带关键字的 $k$ 近邻查询					
申请人	姓 名	陈炜栋	性别	男	学 号	11300240057
	院 系	计算机科学与技术			专 业	计算机科学与技术
	寝室电话	18721469533			手 机	18511870893
导 师	姓 名	孙未未			工 号	25206
	院系单位(部门)	计算机科学与技术				
	办公室电话				手 机	13701964101
	电子邮箱	sun@fudan.edu.cn				
研究成果	成果形式	<input checked="" type="checkbox"/> 论文 <input type="checkbox"/> 报告 <input type="checkbox"/> 其他				
	成果是否发表	<input type="checkbox"/> 是 <input checked="" type="checkbox"/> 否				
	请填写成果发表情况（包括发表论文题目、期刊名称、期刊类别、作者排序、发表日期、卷次和页码），同时将发表文章的纸质版和电子版作为结题材料一并递交，以供评审时参考。					

备注：1.个人信息如有变更，请及时登陆 UROP 网站修改，以便联系；

2.期刊类别指：SCI(Science Citation Index, 科学引文索引)、EI(Engineering Index, 工程索引)、ISTP (Index to Scientific & Technical Proceedings, 科技会议录索引)、SSCI(Social Sciences Citation Index, 社会科学引用索引)等。

## 二、结题报告

### 道路网络中带关键字匹配的 $k$ 近邻查询

计算机科学与技术学院 陈炜栋  
指导老师 孙未未

**摘要**移动设备的普及使得更多的基于位置信息的查询成为可能,对该领域的研究也得到了越来越多的重视。这其中最基础的就是  $k$  近邻查询,  $k$  近邻查询返回兴趣点集合中距离查询点距离最近的前  $k$  个点。随着现在的兴趣点(如餐厅、旅馆等)都被标记了描述性的关键字信息,传统的路网  $k$  近邻查询仅仅依靠兴趣点与查询点的距离作为度量,无法满足用户对于关键字匹配的需求。至今为止,已产生了大量的关于附带关键字要求的近邻查询问题的研究,但是大多都是在欧氏空间中展开的。现实生活中,人们往往需要按照既定的道路(即道路网络)来行走,所以在道路网络环境中展开这些问题的研究就显得更为实际和重要。本文提出了在道路网络环境下的带关键字匹配的  $k$  近邻查询 (*Boolean  $k$  Nearest Neighbor Query,  $BkQ$* ),该查询返回兴趣点集合中覆盖所有查询关键字并且距离查询点距离前  $k$  近的兴趣点。为了有效得处理该查询,本文提出了基于距离优先(Distance First, DF)、基于关键字优先(Keyword First, KF)和基于网格索引(Grid-based, GB)的三种方法。此外,基于真实和模拟数据集,我们进行了充分的实验来验证本文所提算法的有效性。

**关键词**空间数据库; 道路网络;  $k$  近邻; 关键字匹配; 关键字匹配的  $k$  近邻查询;

### $k$ Nearest Neighbor Search with Keyword Matching in Road Networks

**Abstract** The popularity of mobile devices makes more and more location-based queries come true, and there is an increasing interest in this area. One of the most basic queries is  $k$  nearest neighbor ( $kNN$ ) query, which returns the  $k$  nearest points to the query point from an interesting point set. Nowadays interesting points (e.g. restaurants) are usually marked with descriptive keywords, traditional  $kNN$  queries can no longer meet the users' requirements of the keyword matching. The users may want to get the  $k$  nearest neighbors which meet their keyword demands. There are a lot of study focus on both spatial and keyword requirements, while most of them are in Euclidean space. In real life, people move in the constrained roads (i.e. road networks), it's more practical and important to conduct the queries in road networks. We propose Boolean  $k$  Nearest Neighbor Query ( $BkQ$ ) in road network, which returns  $k$  nearest interesting points to the query point and all these  $k$  interesting points must match the query keywords. We develop three algorithms for this query, which are Distance First Algorithm (DF), Keyword First Algorithm (KF) and Grid-based Algorithm (GB). Finally, we perform an experimental evaluation under the real and synthetic dataset which shows the efficiency of our approaches.

**Keywords** spatial databases; road network;  $k$  nearest neighbor; keyword matching;

## 1. 引言

基于地理位置信息的查询在我们的日常生活中扮演着越来越重要的角色。根据美国研究公司 ABI Research 的研究,基于地理位置信息的查询在 2012 年创造的全球利润接近 40 亿美元。这一巨大并且仍在快速发展的市场已受到了学术界和产业界的关注。

在本文中,我们研究在道路网络环境下,带关键字匹配的  $k$  近邻查询。对于用户给定的

关键字集合，查询需要返回所有与查询关键字集合完全匹配的兴趣点中距离用户位置前  $k$  近的兴趣点。由于我们只考虑兴趣点是否存在用户需求的關鍵字，所以我们将查询命名为  $BkQ$  (*Boolean  $k$  Nearest Neighbor Query*)。

与本文工作较为相近的是[14]研究的在道路网络中的  $Top-k$  查询，其假设每个兴趣点关联一个文本，文本中的关键字带有权值信息，查询同时考虑距离与文本相似度。与本文工作不同的是，[14]将距离与文本相似度聚合成一个新的度量标准，并用该标准对兴趣点进行排序。这样，对于一个关键字并未完全匹配的兴趣点，其有可能因为距离查询点距离较近而被作为结果返回给用户。

然而在某些情况下，用户需要返回的兴趣点必须完全包括查询的关键字。比如用户希望搜索距离其最近的带有关键字“无线网咖啡”的兴趣点，此时，对于两个关键字的完全匹配是用户的真实且基本的需求，即使一个较近且“咖啡”关键字权值很高但不具有“无线网”关键字的兴趣点也不应该作为查询的结果返回给用户。

如图 1 所示，五角星点为查询点，黑色实心正方形为道路节点，白色圆形为兴趣点，每个兴趣点都有关键字描述信息。若查询  $k$  值为 2，关键字要求为“pizza steak”，则  $BkQ$  查询返回的结果是  $p_1$  和  $p_7$ ，且  $p_1$  和  $p_7$  是全网仅有的两个与查询关键字集合完全匹配的兴趣点。然而对于  $Top-k$  查询来说，其返回的可能是  $p_1$  和  $p_2$ ，因为  $p_2$  较  $p_7$  来说，即使关键字并未完全匹配，但是其较近的距离可能使其在聚合的度量标准下获得一个较大的权值。

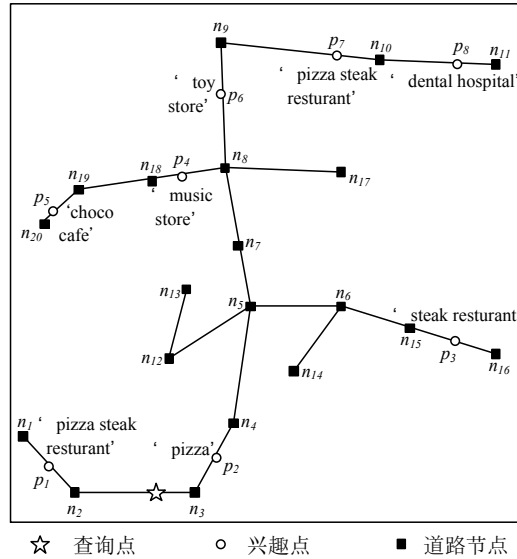


图 1  $BkQ$  查询的图例

据我们所知，本文是对道路网络中单纯带关键字匹配的  $k$  近邻查询的首次尝试。在本文中，我们给出该查询的正式定义。同时，我们提出了基于路径优先和基于关键字优先的两种基础算法。为了提升算法的性能，我们基于网格索引，提出了一个更为高效的算法来处理该查询。

总的来说，本文贡献如下：

- 本文给出了道路网络中带关键字匹配的  $k$  近邻查询 ( $BkQ$ ) 的正式定义。
- 本文提出了基于距离优先和基于关键字优先的两个基础算法来处理道路网络中带关键字匹配的  $k$  近邻查询。
- 为了提升算法的性能，本文结合网格索引来减少算法的搜索空间。
- 最后，我们利用真实道路网络以及关键字数据对本文提出的算法的有效性进行了验证。

本文接下来的内容组织如下。第 2 节介绍关于路网  $k$  近邻查询以及空间关键字查询的目前研究进展。第 3 节介绍道路网络的模型并给出道路网络中带有关键字匹配的  $k$  近邻查询的正式定义。第 4 节介绍我们提出的三种算法。第 5 节展示基于真实道路网络数据集的实验结果。第 6 节总结本文工作并简单介绍未来的工作方向。

## 2. 相关工作

### 2.1 路网 $k$ 近邻查询

单纯的道路网络中的  $k$  近邻查询已经受到了广泛的研究。[1]首次研究道路网络中的  $k$  近邻查询，提出了道路网络的存储方法并提出了两种解决  $k$  近邻查询的算法 IER 和 INE，然而 IER 性能并不稳定，INE 在兴趣点比较稀疏的情况下，性能退化显著。[2]将 Voronoi 图应用于道路网络中的计算，根据 Voronoi 图的性质能够快速求出最近邻兴趣点。[3]通过挑取部分关联度较大的道路节点并预先计算并存储指定数量的最近邻来加速查询的运算。[4]提出了利用路径覆盖的想法来减少搜索的空间从而加速了计算的过程。

### 2.2 空间关键字查询

关于空间关键字的查询[5]已有了很多成果。[6]首次提出空间关键字查询，其用 R 树来索引空间信息，用比特表来索引关键字信息，构造出称为  $IR^2$  树的新索引来处理该查询。[6]中的关键字信息只有“有”和“无”两种情况，[7]提出每个兴趣点关联一个文本，其中的每个关键字都附带权值信息，并且提出 DIR 树和 CIR 树以高效完成该查询。[16]以及[17]都是围绕该问题展开的进一步工作，其分别提出被称为 IR 树和 BR 树的索引结构来支持高效的查询。[9]在查询中加入角度要求，使得返回的兴趣点不仅要满足关键字要求并且距离查询点距离较近，还需在该角度之中。[9]研究查询点移动情况下的带关键字的  $k$  近邻查询，其同样利用构造安全区域的思想，但是文中安全区域的计算同时考虑了空间距离和文本相似度。

关于空间关键字查询还有很多，[10]研究对于用户给出的  $m$  个关键字，如何找出一个直径最小且大小为  $m$  的兴趣点集合，使得该集合所含有的关键字能覆盖用户的要求。[11]研究如何找出一个兴趣点集，使得该集合所含有的关键字覆盖查询给出的关键字，同时该集合与查询点的距离以及该集合的直径都较小。[12]研究如何找出把查询点当作同时考虑空间距离和文本距离的  $k$  近邻的兴趣点。[13]研究兴趣区域的查询，要求返回的兴趣区域在空间和文本距离上都要满足给定的阈值。

然而，以上所有的工作都是在欧氏空间中进行的，无法应用于道路网络的环境下。[14]在道路网络中求解带关键字的 Top- $k$  查询。文中提出了三种用于解决该问题的算法，并设计了相应的数据结构来解决道路信息与文本信息的存储于查询。与本文不同的是，由于其将路网距离与文本相似度整合成了一个新的权值，这就使得距离较近的兴趣点即使关键字不完全匹配也可能出现在结果集中。[18]研究的是范围查询与关键字的结合，查询返回距离查询点给定距离内的满足关键字相似度约束的所有兴趣点，对于欧氏空间与道路网络分别设计了算法。[19]返回给定起点和终点的一条最优路径，该路径要覆盖查询关键字并且满足一个给定的限制。[20]返回给定起点和终点的一条最短路径，该路径上的每个兴趣点都必须至少满足一个关键字要求且路径上的所有兴趣点必须完整得满足查询关键字要求。[21]从聚集点集合中挑出一个聚集点并为查询点集合中各查询点提供一条到该聚集点的路径，使得所有路径上经过的兴趣点所带关键字的并能覆盖查询要求，并且能使这些路径中长度最长的路径最短。

## 3. 基础知识

本节首先介绍道路网络的模型，接下来简单介绍关键字匹配，最后给出  $BkQ$  的正式定义。

**道路网络：**我们把道路网络  $G$  定义为由一个节点集合  $N$  和一个边集合  $E$  组成的带权图，

即  $G = (N, E)$ 。一个节点  $n \in N$  代表一个道路路口，一条边  $(n, n') \in E$  代表连接节点  $n$  和  $n'$  的路段。 $w(n, n')$  表示相应边的权值，该权值可以表示路段长度或者是通过时间，为简单起见，本文使用路段长度作为权值并且假设所有长度都是正数。一条路径  $P(u, v)$  代表连接节点  $u$  和节点  $v$  的一系列边的集合，其长度  $|P(u, v)| = \sum_{(n, n') \in P(u, v)} w(n, n')$ 。在所有连接节点  $u$  和节点  $v$  的路径中，长度最短的那条被定义为连接  $u$  和  $v$  的最短路径  $SP(u, v)$ ，本文使用  $\|u, v\|$  来表示  $|SP(u, v)|$ 。图 1 为一个道路网络的实例，其中有 20 个节点和 19 条边。

**关键字匹配：**由于关键字是一种文本信息，通常用字符串来表示。对于查询关键字集合  $Q_K = \{k_1, k_2, \dots, k_n\}$ ，关键字集合  $A = \{k'_1, k'_2, \dots, k'_m\}$  与  $Q_K$  完全匹配当且仅当  $Q_K \subseteq A$ 。

下面给出 BkQ 的正式定义。

**定义 1 (BkQ)：**我们将道路网络中的带关键字匹配的  $k$  近邻查询定义为  $Q_B = \langle q.l, q.d, q.k \rangle$ ，其中  $q.l$  是查询点的空间位置（例如坐标）， $q.d$  是查询关键字集合， $q.k$  是查询需返回的兴趣点数量。对于给定道路网络  $G$  和一个带有关键字标记的兴趣点集合  $P$ ，查询  $Q_B$  返回所有匹配所有查询关键字  $q.d$  的兴趣点中，与查询点  $q.l$  最短路径距离最近的  $q.k$  个兴趣点。

为了简单起见，本文假设查询点位于道路节点上。对于查询点位于路段中间的情况，可以通过对该查询点所在路段的两个端点就行查询，并与所在路段上存在的兴趣点做比较即可，这对算法的有效性和效率不构成影响。

## 4. 算法

本节介绍所提出三个用于处理 BkQ 的算法：基于距离优先，基于关键字优先以及基于网格索引的方法。

关于道路网络，由于本文所有的网格索引整合了道路网络和兴趣点的信息，所以我们在预处理阶段把所有兴趣点插入到道路网络中构成新的节点，并把兴趣点所在边分为两段。

关于关键字的存储，假设地图中不同关键字的个数为  $n$ ，通过为每一个兴趣点维护一个长度为  $n$  的 Bitmap 来支持关键字的存储与访问通常来说代价太大。比如，若地图中有  $10^4$  个不同的关键字，则每个兴趣点需要约 1KB 的空间。然而，在实际道路网络中，每个兴趣点拥有的关键字很少，比如[14]中列出的三个数据集（London, Australia, British Isles）中，每个兴趣点平均仅含有 3 个关键字。所以，我们在预处理阶段为每个关键字添加从 0 开始的整数作为标识，这样每个兴趣点仅需维护极少数个整数以达到关键字的存储，这样大大减少了存储空间。

### 4.1 基于距离优先的算法（DF）

基于路径优先的算法先考虑兴趣点与查询点的距离关系，再考虑关键字匹配的问题。其过程类似 Dijkstra 算法[22]，从查询点开始，按照距离查询点距离从近到远，不断访问当前节点的邻居节点，每访问到一个兴趣点，检查其关键字与查询关键字是否完全匹配，若匹配则将其加入结果集。当找到第  $k$  个满足关键字匹配的兴趣点时，算法结束。关于关键字的索引，只需在存储道路网络信息的邻接表中，为每个节点多保存一份相应的关键字信息即可。

**Algorithm 1: Distance First Algorithm**

**Input:**  $G(N, E), Q_B = \langle q.l, q.d, q.k \rangle$ .

**Output:** Return the nearest  $q.k$  interesting points to  $q.l$  that match  $q.d$ .

```
1  ResultSet  $R \leftarrow \emptyset$ 

2  MinHeap  $H \leftarrow \emptyset$ 

3   $H.enheap(q, ||q, q|| = 0)$ 

4  while ( $H \neq \emptyset$ ) and ( $|R| < q.k$ ) do

5    ( $u, ||u, q||$ )  $\leftarrow H.deheap()$ 

6    if ( $u$  is an interesting point) then
7      if ( $u.d$  matches  $q.d$ ) then
8        insert  $u$  into  $R$ 
9      endif
10     end if
11   for each non-visited adjacent node  $v$  of  $u$  do
12      $H.enheap(v, ||u, q|| + w(u, v))$ 
13   end for
14   mark  $u$  as visited
15 end while
16 return  $R$ 
```

Algorithm 1 展示了基于距离优先的算法。算法将查询  $Q_B$  作为输入，返回距离查询点  $q.l$  距离前  $q.k$  近的匹配关键字  $q.d$  的兴趣点。算法通过一个最小堆  $H$  来维护各节点与查询点的距离。首先，算法初始化最小堆  $H$ ，将查询点  $q$  伴随其与查询点的距离（此时为 0）入堆（第 3 行）。接着，算法不断判断出堆顶点  $u$  是否为兴趣点，若是则检查其关键字  $u.d$  与查询关键字  $q.d$  是否匹配，若匹配则将  $u$  与其到查询点的距离  $||q.l, u.l||$  保存到结果集  $R$  中（第 5-8 行）。算法不断访问当前处理节点  $u$  的邻居节点（第 9 行），直到地道路网络已经被搜索完全或者已经找到  $q.k$  个需要的兴趣点（第 4 行）。

## 4.2 基于关键字优先的算法（KF）

本节将介绍基于关键字优先的算法。不同于 DF 算法的是，该算法优先考虑关键字匹配的问题，再计算兴趣点与查询点的距离。算法首先遍历查询的关键字集合，对于每一个查询关键字，查找包含该关键字的兴趣点，之后对每个关键字对应的兴趣点集合求交集，得到候选兴趣点集合。最后，算法对候选兴趣点集合中的每个点求其与查询点的最短路径距离，并得出距离查询点距离前  $k$  近的查询点。



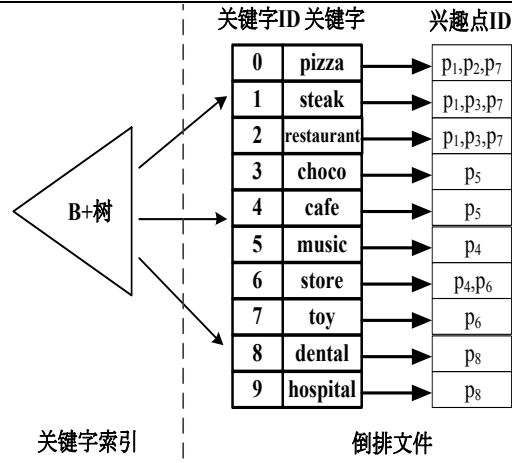


图 2 倒排文件

本算法采用倒排文件（Inverted File, IF）对关键字进行索引，其维护一个关键字列表，每个关键字元素维护一个链表，用于存储含有该关键字的兴趣点信息。图 2 是一个倒排文件的实例，存储的是图 1 中兴趣点的关键字信息。如我们第 4 节开头所讨论的，我们为每个关键字设定标志性的整数 ID，如图 2 中，关键字“pizza”的 ID 为 0，其关联的链表中存储的兴趣点为  $p_1, p_2$  和  $p_7$ 。我们为关键字列表建立 B+树索引，用于根据关键字 ID 快速访问到相关联的兴趣点信息。

Algorithm 2 展示了基于关键字优先的算法。算法输入需要一个额外的数据结构，即倒排文件  $IF$ 。对于查询的关键字集合  $q.d$  中的每个关键字，算法利用倒排文件获取含有该关键字的兴趣点集合（第 4-6 行）。之后对上述所有兴趣点集合求交集，得到一个候选兴趣点集合  $C$ （第 7 行）。很明显的，候选集合  $C$  中的每个兴趣点的关键字信息都与查询的关键字完全匹配。在达到关键字要求后，算法继续求查询点与每个候选兴趣点的距离并选出其中距离最小的  $q.k$  个兴趣点作为结果返回。算法维护一个变量  $\epsilon$ （第 3 行），其保存当前结果集  $R$  中距离查询点第  $q.k$  近的兴趣点与查询点的距离，若  $R$  中兴趣点数量不足  $q.k$ ，则始终保持为  $\infty$ 。

对每一个求出距离的兴趣点，算法尝试利用其距离去更新结果集与  $\epsilon$ （第 8-13 行）。

#### Algorithm2: Keyword First Algorithm

**Input:**  $G(N, E)$ ,  $Q_B = \langle q.l, q.d, q.k \rangle$ , Inverted File  $IF$ .

**Output:** Return the nearest  $q.k$  interesting points to  $q.l$  that match  $q.d$ .

```

1  ResultSet  $R \leftarrow \emptyset$ 

2  CandidateSet  $C \leftarrow \emptyset$ 

3   $\epsilon \leftarrow \infty$ 

4  for each keyword  $w_i \in q.d$  do

5     $C_i \leftarrow IF.getInterestingPoints(w_i)$ 

6  end for

7   $C \leftarrow \bigcap C_i$ 

```

---

```

8   for each interesting point  $p_i \in C$  do
9     if  $\|q, p_i\| < \epsilon$  then
10      update  $R$  by  $p_i$ 
11      update  $\epsilon$  by  $\|q, p_i\|$ 
12    end if
13  end for
14  return  $R$ 

```

---

### 4.3 基于网格的算法（GB）

基于距离优先的算法（DF）可以保证待判断的兴趣点总是距离查询点较近。但是当兴趣点分布比较稀疏或  $k$  值较大时，算法需要扩展地图的很大部分才能得到结果，尤其是在当地图中只有少数兴趣点与查询关键字完全匹配时，算法性能退化明显。基于关键字优先的算法（KF）则较少得受到  $k$  值的影响，并且在地图中仅有少量兴趣点与查询关键字完全匹配时，其效率提升显著。然而，当查询关键字数量较少或者地图中有大量兴趣点与查询关键字完全匹配时，算法的性能会有所下降。

基于距离优先的算法（DF）以及基于关键字优先的算法（KF）均是把两个要求分开来单独处理，本小节接下来提出一种基于网格的算法（Grid-based, GB），试图将空间信息与关键字信息整合起来，以解决上述两个算法在某些情况下的缺陷。

#### 4.3.1 网格划分

对于道路网络的网格划分可分为两类，一类是基于边来划分，此时一条边仅属于一个网格中；另一类是基于顶点来划分，此时一个顶点完全属于一个网格中，而存在一些边穿过多个网格。

我们基于顶点对道路网络进行划分，得到网格集合  $Grid = \{G_1, G_2, \dots, G_m\}$ ，对于网格  $G_i$ ，我们定义  $G_i.N$  为该网格所包含的顶点，如此则有  $\bigcup G_i.N = G.N$  以及  $G_i.N \cap G_j.N = \emptyset$  ( $i \neq j$ )，其中  $G.N$  为道路网络的顶点集合。

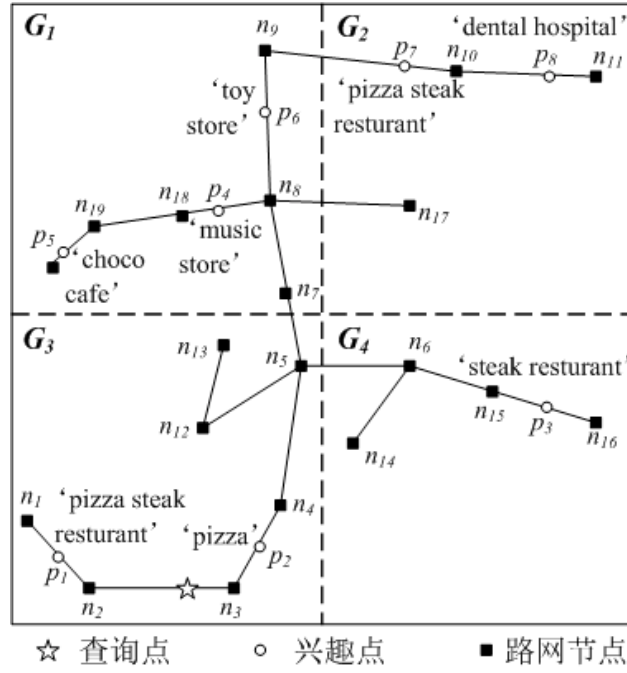


图 3 一个网格划分的实例

图 3 是对图 1 中的道路网络进行划分的一个实例，道路网络被简单划分成一个  $2 \times 2$  的网格，分别被标记为  $G_1, G_2, G_3$  和  $G_4$ ，每个网格大小相等、形状相同。需要注意的是,可能存在顶点落在划分线（图 3 中的虚线）上的情况,此时我们可以定义每个网格的左边界和上边界属于该网格即可。

在网格中，有一类特殊的节点，它们至少存在一个邻居节点与其位于不同的网格中，即节点  $n_i$  为其所在网格  $G_i$  的边界点当且仅当存在一条边  $\langle n_i, n_j \rangle$ ，使得  $n_j \in G_j$  并且  $G_i \neq G_j$ 。

如图 3 中的节点  $n_7$ ，其位于网格  $G_1$  中，其邻居节点  $n_5$  位于网格  $G_3$  中，所以  $n_7$  和  $n_5$  分别为网格  $G_1$  和  $G_3$  的边界点。

#### 4.3.2 网格的存储

对于每个网格，我们需要在预处理阶段计算并存储一些额外的距离和关键字信息来支持高效的查询操作。

对于一个网格  $G_i$ ，我们定义  $G_i.B$  为该网格所含的边界点集合。在一个网格内的所有边界点，我们需要计算并存储其两两之间的距离。例如图 3 中，对于网格  $G_1$ ，其有三个边界点  $n_7, n_8$  和  $n_9$ ，则需要计算  $n_7$  和  $n_8, n_8$  和  $n_9, n_7$  和  $n_8$  之间的距离。为了减少预处理阶段的计算量，对于两个边界点之间的距离，我们仅计算通过所在网格内部的边所能达到的最短路径距离。这一步的时间和空间代价都是可以接受的[2]。

除了距离信息之外，我们还需要对每个网格存储关键字信息。网格的关键字为其包含的所有兴趣点的关键字的并集。我们定义  $G_i.d$  为该网格的关键字信息，则  $G_i.d = \bigcup_{n_i \in G_i.N} (n_i.d)$ 。

### 4.3.3 查询处理

由 4.3.1 中边界点的定义可知，对于网格 $G_i$ 内的节点 $n_i$ ， $G_i$ 外的节点与 $n_i$ 之间的路径必然先经过网格 $G_i$ 的边界点。所以，一个边界点可以看作是其所在网络的“入口”，当查询搜索到一个邻居网络的边界点时，我们可以利用存储的网格的关键字来与查询做匹配，若不匹配，则可以跳过对该网格的搜索。

---

**Algorithm3:Grid-based Algorithm**

---

**Input:**  $G(N,E)$ ,  $Q_B = \langle q.l, q.d, q.k \rangle$ ,  $Grid$

**Output:** Return the nearest  $q.k$  interesting points to  $q.l$  that match  $q.d$ .

```
1  ResultSet  $R \leftarrow \emptyset$ 
2  MinHeap  $H \leftarrow \emptyset$ 
3   $gid \leftarrow locate(q)$ 
4   $H.enheap(q, ||q,q||=0, gid)$ 
5  while ( $H \neq \emptyset$ ) and ( $|R| < q.k$ ) do
6    ( $u, ||u,q||, gid$ )  $\leftarrow H.deheap()$ 
7    Line 6-10 of Algorithm 1
8    if  $u$  is a border point then
9       $gid' \leftarrow locate(u)$ 
10     if  $gid == -1$  then
11        $expandOutsideGrid(u, ||u,q||)$ 
12     else if  $gid \neq gid'$  then
13        $expandInsideGrid(u, ||u,q||, Grid)$ 
14     else
15       for each non-visited adjacent node  $v$  do
16          $H.enheap(v, ||u,q||+w(u,v), gid')$ 
17       end for
18       mark  $u$  as visited
19     end if
20   else
21     Line 15-18 of Algorithm 3
22   end if
23 end while
24 return  $R$ 
```

---

基于路径优先的算法维护<节点编号，与查询点的距离>这样的二元组，该二元组表示从查询点到该节点的一条路径的长度。对于基于网格的算法，我们需要多维护一个信息，用

于记录该二元组所代表的路径中，该节点的上一个节点所在的网格编号，即该节点在入最小堆时是由哪个网格的点扩展得到的。我们把这一额外信息附加在二元组中构成三元组，并同样由最小堆进行维护。当访问的节点为边界点时，我们检查其三元组中上一节点的网格编号，若与自己网格编号相同，表示当前搜索是在一个网格内部进行的，只需继续按原图邻接信息进行搜索即可。若与自己网格编号不同，表示网格外部节点欲通过该边界点进入网格内部，此时可利用网格的关键字与查询做匹配，若匹配，则需在网格内部进行搜索，与前一种情况相同；若不匹配，则可以利用存储的边界点之间的距离跳过对网格内部的搜索，此外由该步扩展到的边界点的三元组的最后一项设置为-1。所以当访问的边界点对应三元组的最后一项为-1 时，只需向所在网格外扩展即可。由于我们假设网格编号为非负整数，这里的-1 就充当一个无效网格编号的作用以区别于正规网格编号。

Algorithm 3 展示了基于网格的算法。在算法的初始化阶段（1-4 行），算法通过 *locate* 操作获得查询点所在网格编号，并将其放入查询点对应的三元组中，这与三元组第三项的定义不符，但这是唯一的一个特例，因为根据前文的规定，可以通过这一设定使查询点出堆时，执行的操作是在所在网格内按原图邻接信息进行搜索。对于每一个访问的节点，算法需要判断其是否需要加入结果集（第 7 行）。若当前访问的节点为边界点，算法首先获得该点所属网格的编号（第 9 行），并根据前文分的三种情况对道路网络进行扩展（10-17 行），其中的 *expandOutsideGrid* 和 *expandInsideGrid* 两个函数将在下文进行说明。若当前访问的节点不是边界点，则只需按照原图的邻接信息进行扩展即可（第 20 行）。

---

#### Algorithm 4: *expandOutsideGrid*

---

Input:  $u, ||u, q||$

```

1   $gid \rightarrow locate(u)$ 
2  for each non-visited adjacent node  $v$  of  $u$  do
3     $gid' \rightarrow locate(v)$ 
4    if  $gid \neq gid'$  then
5       $H.enheap(v, ||u, q|| + w(u, v), gid)$ 
6    endif
7  endfor
8  mark  $u$  as visited
```

---

Algorithm 4 展示了 *expandOutsideGrid* 函数，其对应三元组最后一项为-1 时应进行的操作。函数有两个参数，分别为当前访问的节点和其与查询点的距离。算法首先获得该节点所在网格编号，之后遍历其在原图中邻接的节点，只有当邻接的节点与其不在同一网格时，才将该点对应的三元组放入最小堆。

---

#### Algorithm 5: *expandInsideGrid*

---

Input:  $u, ||u, q||, Grid$

```

1   $gid \leftarrow locate(u)$ 
```

---

```

2  keywords  $\leftarrow$  Grid.getKeywords(gid)

3  if keywords match q.d then
4    Line 15-18 of Algorithm 3
5  else
6    adjLists  $\leftarrow$  Grid.getAdjLists(gid,u)

7    for each  $\langle u, b \rangle \in$  adjLists do

8      H.enheap(b, ||q,u|| + ||u,b||, -1)

9    end for
10  end if
11  mark u as visited

```

---

Algorithm 5 展示了 *expandInsideGrid* 函数，其对应三元组最后一项与当前访问节点所在网格编号不同的情况。此时需要检查网格关键字与查询是否匹配，在不匹配时，利用存储的边界点之间的距离来跳过对该网格的搜索。其中 *Grid.getKeywords(gid)* 返回编号为 *gid* 的网格的关键字信息，*Grid.getAdjLists(gid,u)* 返回编号为 *gid* 的网格中，边界点 *u* 与其它边界点 *b* 的之间邻接信息，简单起见，其之间的距离也记为  $||u,b||$ 。

接下来我们将基于网格的算法应用于图 3 的实例中，设查询 *k* 值为 2，查询关键字为“pizza steak”。查询首先在网格  $G_3$  内部按原图的邻接信息进行搜索，将兴趣点  $p_1$  放入结果集。当访问完边界点  $n_5$  时开始对邻居网格进行访问，不妨设先访问到边界点  $n_7$ ，由于  $n_7$  所在网格为  $G_1$ ，而其三元组第三项，即  $n_5$  所在网格为  $G_3$ ，所以需要用到  $G_1$  的关键字与查询作匹配。匹配失败后，直接利用  $G_1$  内部存储的边界点之间的距离将  $n_8, n_9$  入最小堆，此时其三元组第三项为 -1， $G_1$  内部的其它节点都无需再访问。同理，当访问到  $n_6$  时，可判断出  $G_4$  内的节点也无需再访问。算法最后进入  $G_2$  并找到第 2 个结果  $p_7$ ，至此算法结束，返回结果集  $\{p_1, p_7\}$ 。

## 5. 实验

### 5.1 实验配置

数据采用真实路网数据,分别对 Dublin、London、British 三个数量级大小的数据集进行测试。

数据集	Dublin	London	British
点数	68K	219.2K	3.7M
边数	87K	292.7K	4.8M
兴趣点	5K	10K	484

实验机器:

Inter(R) core(TM) i7-3770 CPU @ 3.40GHz

Main Memmory:32GB

## 5.2 性能比较与分析

### 5.2.1 网格划分的影响

这部分实验中,对于 British 数据集,对于相同的询问集合,按照网格划分的粒度和算法 DF,IF 的运行时间进行比较。图 4 中我们可以看出,本文提出的网格划分算法优势明显,在该数据集下  $20 \times 20$  的粒度最佳。

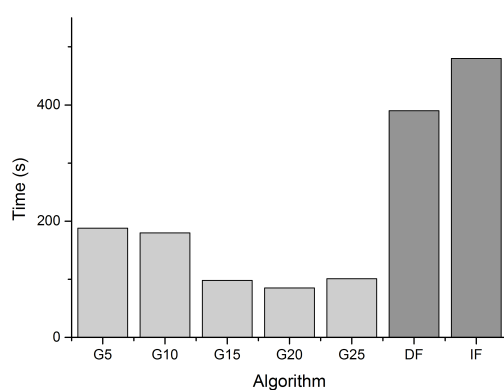


图 4 网格划分对时间性能的影响

### 5.2.2 查询关键字个数 N 的影响

这部分实验中,对于相同的询问集合,以查询关键字的数量为变量进行运行时间的比较,图 5,6,7 分别对应 British,London,Dublin 三个数据集。实验结果表明,随着查询关键字数量增加,算法运行时间随之增加。

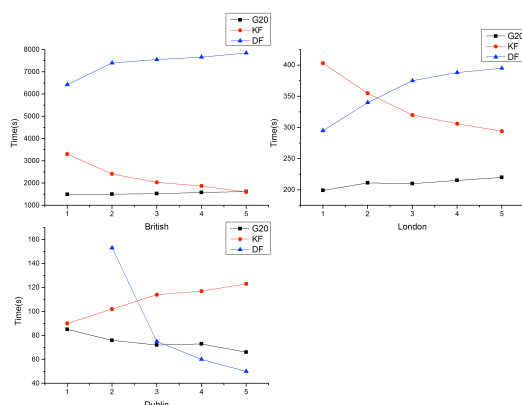


图 5 查询近邻个数对系统性能的影响

理论上来说查询关键字数量增加导致匹配目标点减少,从而搜索空间增大。实验结果符合预期理论分析。从三个数据集都能看出网格划分方法效率高,高稳定性。

### 5.2.3 查询近邻个数 $k$ 的影响

这部分实验中，对于相同的询问集合，以查询近邻的个数为变量进行运行时间的比较，图 8,9,10 分别对应 British,London,Dublin 三个数据集。实验结果表明，随着  $k$  增加，算法运行时间随之增加。

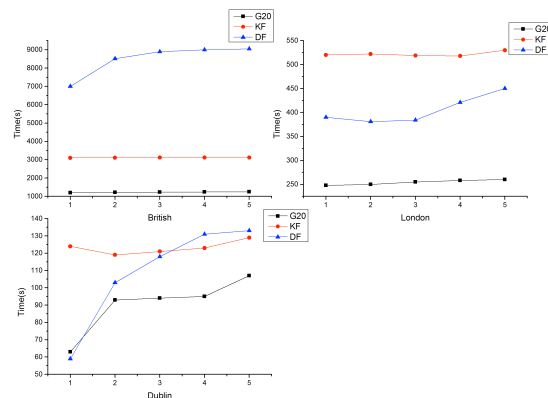


图 6 查询近邻个数对系统性能的影响

结果集近邻数增加使得搜索空间增大，实验与理论相符。从三个数据集都能看出网格划分方法效率高，高稳定性。

### 5.2.4 空间复杂度

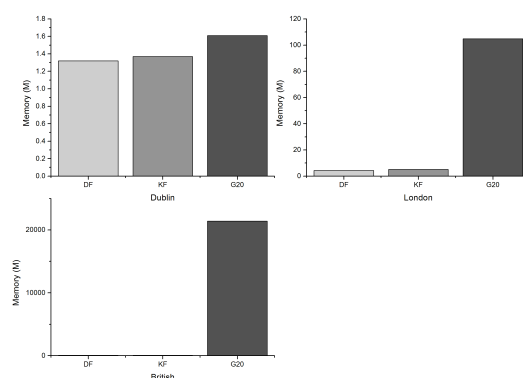


图 6 三个数据集各算法的空间复杂度

由于 G20 将路网划分成  $20 \times 20$  的网格,网格预计算了两两边界点间的最短路,该部分空间开销巨大。

## 6. 总结

本文研究了在道路网络环境下的带关键字匹配的  $k$  近邻查询。我们提出了基于距离优先以及基于关键字优先的两种基础算法。此外，为了弥补这两种算法在特定情况下的缺陷，并结合这两种算法各自的优点，我们提出了基于网格索引的算法，该算法通过预先计算并存储网格内边界点之间的最短距离，可以在查询处理过程中，跳过对不含所需关键字网格



的搜索，从而加速查询处理。真实数据集上的实验结果表明，基于网格索引的算法在查询性能上有着一定的优势。如本文一二节所述，单纯路网中  $k$  近邻问题以及单纯的关键字匹配查询或欧氏空间中的带关键字近邻查询，已有相关的成果。但是道路网络中考虑关键字属性匹配的查询由于其复杂性，还未有成果。本文做出一些探索，得出了初步的成果。

### 参考文献

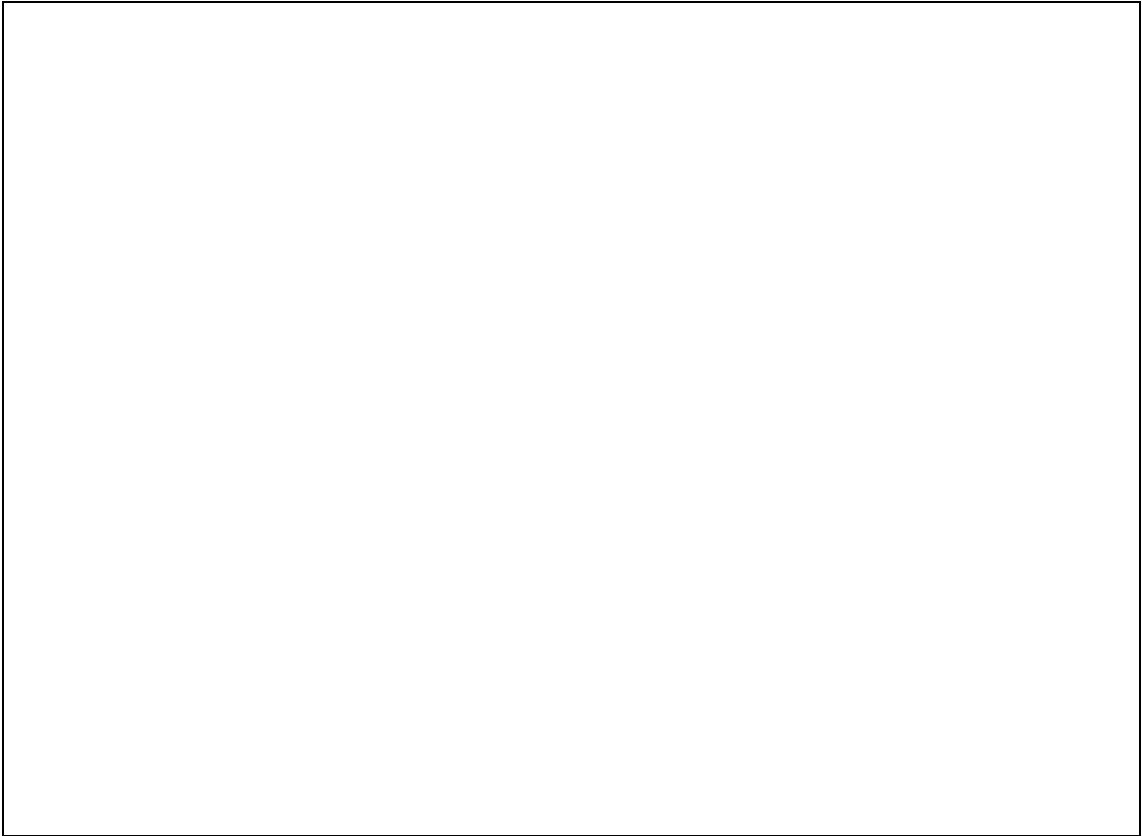
- [1] D. Papadias, J. Zhang, N. Mamoulis and Y. Tao. Query Processing in Spatial Network Databases. In *VLDB*, pages 802-813, 2003.
- [2] M. Kolahdouzan, C. Shahabi. Voronoi-Based K Nearest Neighbor Search for Network Databases. In *VLDB*, pages 840-851, 2004.
- [3] H. Cho and C. Chung. An Efficient and Scalable Approach to CNN Queries in a Road Network. In *VLDB*, pages 865-876, 2005.
- [4] K. C. K. Lee, W. Lee, B. Zhang and Y. Tian. Road: A New Spatial Object Search Framework for Road Networks. In *TKDE*, pages 547-560, 2012.
- [5] X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu and M. L. Yiu. Spatial Keyword Querying. In *ER*, pages 16-29, 2012.
- [6] I. D. Felipe, V. Hristidis and N. Rishe. Keyword Search on Spatial Databases. In *ICDE*, pages 656-665, 2008.
- [7] G. Cong, C. S. Jensen and D. Wu. Efficient Retrieval of the Top-K Most Relevant Spatial Web Objects. In *VLDB*, pages 337-348, 2009.
- [8] G. Li, J. Feng and J. Xu. Desks: Direction-aware spatial keyword search. In *ICDE*, pages 474-485, 2012.
- [9] D. Wu, M. L. Yiu, C. S. Jensen and G. Cong. Efficient Continuous Moving Top-K Spatial Keyword Query Processing. In *ICDE*, pages 541-552, 2011.
- [10] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung and M. Kitsuregawa. Keyword Search in Spatial Databases: Towards Searching by Document. In *ICDE*, pages 688-699, 2009.
- [11] X. Cao, G. Cong, C. S. Jensen and B. C. Ooi. Collective Spatial Keyword Querying. In *SIGMOD*, pages 373-384, 2011.
- [12] J. Lu, Y. Lu and G. Cong. Reverse spatial and textual k nearest neighbor search. In *SIGMOD*, pages 349-360, 2011.
- [13] J. Fan, G. Li, L. Zhou, S. Chen and J. Hu. Seal: spatio-textual similarity search. In *VLDB*, pages 824-835, 2012.
- [14] J. B. Rocha-Junior and K. Nørsvåg. Top-k spatial keyword queries on road networks. In *EDBT*, pages 168-179, 2012.
- [15] L. Chen, G. Cong, C. S. Jensen and D. Wu. Spatial Keyword Query Processing: An Experimental Evaluation. In *VLDB*, pages 217-228, 2013.
- [16] Z. Li, K. C. K. Lee, B. Zheng, W. Lee, D. L. Lee and X. Wang. IR-Tree: An Efficient Index of Geographic Document Search. In *TKDE*, pages 585-599, 2011.
- [17] G. Li, J. Xu, J. Feng. Keyword-based k-nearest neighbor search in spatial databases. In *CIKM*, pages 2144-2148, 2012.
- [18] F. Li, B. Yao, M. Tang and M. Hadjieleftheriou. Spatial Approximate String Search. In *TKDE*, 2012.
- [19] X. Cao, L. Chen, G. Cong and X. Xiao. Keyword-aware optimal route search. In *VLDB*, pages 1136-1147, 2012.
- [20] B. Yao, M. Tang and F. Li. Multi-approximate-keyword routing in GIS data. In *GIS*, pages 201-210, 2011.
- [21] K. Chen, W. Sun, C. Tu, C. C. and Y. Huang. Aggregate keyword routing in spatial database. In *GIS*, pages 430-433, 2012.
- [22] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, pages 269-271, 1959.

### 后记（致谢）：

时光飞逝，转眼就到了暮政项目结题的时候。我非常享受暮政项目的过程，思考，尝试，修改，周而复始。回顾一年多时光，有很多要感谢的人。在遇到困难时，实验室的师兄师姐总是能给出及时的帮助。在困惑时，孙老师有一语中的的建议。

通过一年半的科学实践，我完整地体验了课题的构思发展和结题。深刻体会到科研是一件实践和理论紧密结合的创造。一次次地尝试和修正，让我对科学工作者产生深深敬佩之情。从文献阅读，构思，实验，到结题时自己整理论文，通过亲身体验科研工作的流程。提高了我的科研技能，增强对计算机理论研究的兴趣。

非常感谢 FDUROP 给我这次机会，这是一次难忘的经历！



### 三、指导老师评语（请您手写）

陈炜栋同学进入课题组开始做思政课题到现在已经有一年多的时间了，在这段时间里，能够广泛阅读，积极思考，相信他从中学习很多东西。一年多以来取得了很大的进步。这一段经历会让他的未来科研和学习受益匪浅。

指导教师：

年 月 日





五、课题财务日志

项目类型	<input type="checkbox"/> 菽政 <input type="checkbox"/> 望道 <input type="checkbox"/> 曦源 <input type="checkbox"/> 上海大学生创新活动计划 <input type="checkbox"/> 国家大学生创新性实验计划			项目编号	
日期	费用事项	费 用 信 息		费用	实际费用
	合计				

注：不够可加页。

学生签字：

导师签字：

日 期：

# 附录 1、FDUROP 项目财务制度

2010 年 7 月 16 日修订

## 一. 经费额度及发放

1. 自 2010 年 3 月 ( 即 2010 年春季学期第 3 周 ) 起 , 薯政、望道项目资助额度为 6000 元 (在此之前入选的项目资助额度仍为 4000 元), 分两次发放:
  - a) 课题正式立项后, 向课题导师发放第一笔假期补贴 500 元, 注入导师工资卡中; 向学生发放第一笔假期补贴 500 元, 注入学生校园卡中; 以校内财务转帐支票的形式发放课题经费 2000 元。
  - b) 课题中期考核通过后, 向课题导师发放第二笔假期补贴费 500 元, 注入导师工资卡中; 向学生发放第二笔假期补贴 500 元, 注入学生校园卡中; 以校内财务转帐支票的形式发放课题经费 2000 元。
2. 薯政、望道项目结题审核通过后, 自动入选“国家大学生创新实验计划”, 该计划最高有 5000 元的补足经费:
  - a) I 类: 入选“国家大学生创新性实验计划”的课题均可获得 1000 元研究费用; 其中导师和学生各 500 元, 分别注入导师工资卡和学生校园卡中;
  - b) II 类: 需田野考察或实验室研究的课题凭《FDUROP 项目财务日志》可申请最高 2000 元的资助;
  - c) III 类: 课题如参加学术会议, 或研究成果获得发表, 可在结题后一年内申请最高 2000 元的资助。II 类和 III 类经费以校内财务转帐支票的形式发放, 具体额度根据《FDUROP 项目财务日志》核算。

## 二. 课题经费使用

1. 课题经费的使用由导师负责。可用于以下与课题相关的开支: (1)书籍 (包括导师为课题所购买的书籍费用); (2)文献信息费; (3)打印费用; (4)实验材料费用; (5)使用实验仪器设备费用, 实验测试、分析费用; (6)软件或数据库费用; (7)调研所产生的交通费用; (8)访谈或与导师交流所产生的茶点费用; (9)参加会议、差旅费; (10)发表文章版面费用; (11)其它费用 (需另外具体注明)。
2. 校内财务转帐支票使用。每笔转帐支票发放后, 由导师负责保管。导师可将该课题经费加入其个人经费本中, 以支付课题相关开销; 或加入导师所在院系的 FDUROP 经费本 (见附件 2) 中, 之后根据课题开支报销; 也可凭转帐支票及课题研究发生费用的发票 (等于或多于 2000 元) 到学校财务处一次性报销。

## 三. 财务日志填写及审核

1. FDUROP 项目自正式立项之日起, 即需根据实际开支情况 (可少于或超过课题资助总额), 按日期记录课题研究所产生的每笔费用。

2. 费用事项如上文所述。费用信息需逐项具体写明，如：(1)书籍，注明所购书名、作者、出版社、单价、数量；(2)文献信息费，注明所用电子数据库及费用；(3)打印费用；(4)实验材料费用，包括实验所需原材料、试剂药品、易耗实验器材，需注明名称、单价及数量；(5)使用实验仪器设备费用，注明仪器设备名称、价值，使用次数及时间；实验测试、分析费用，注明测试分析项目及费用；(6)软件或数据库费用，注明所购软件或数据库名称、版本、价格；(7)调研所产生的交通费用，注明往返地点、所乘交通工具及费用；(8)访谈或与导师交流所产生的茶点费用，访谈或交流必须与课题研究相关，并注明每次访谈或交流时间及发生茶点费用；(9)参加会议、差旅费，注明会议名称、地点、参加形式（报告、海报以及是否正式发表论文）；(10)发表文章版面费用，注明发表论文题目、期刊名称、作者排序、发表日期、卷次和页码、期刊类别；(11)其它费用（需另外具体注明）。
3. 仪器测试、分析费用有实际费用和虚拟费用之分，实际费用是指测试实际所花费用，虚拟费用是指如果是一个外校人员的测试费用（有些院系的大型仪器测试费用校内校外价格不同，虚拟价格是指校外测试费用；有些实验室自己有一些测试仪器，实际使用过程中不需要支付费用，也请估计一个如果是校外样品来测试的费用）。
4. 会议差旅制度参照学校财务处制度（交通、住宿、生活费等）实行。
5. 版面费必须是由该学生作为第一作者的论文（导师第一作者的，学生可以是第二作者）。
6. **课题经费不得用于购买实验设备仪器或电脑及其配件。**
7. 最后的合计费用，实际费用就是将各类合计即可，虚拟费用是指测试分析费用的虚拟费用加上其他各类费用。
8. **学生在参加中期报告前一周，须将中期报告前的财务日志电子版发送至 [fdurop@fudan.edu.cn](mailto:fdurop@fudan.edu.cn)；结题时，须将项目全程的财务日志与其他结题材料一起上交。课题财务日志将作为结题考核指标中的一项内容。**

#### 四. 其他

1. 所购买书籍在项目结束之后将统一捐献给所属院系资料室\*，文献、专利信息可由指导老师负责处理。
2. **若课题研究过程中课题受到其他项目资助，则要求说明该课题曾经接受“复旦大学本科生学术研究资助计划（FDUROP, Fudan's Undergraduate Research Opportunities Program）”资助情况。**

教务处实践 教学办公室

---

\* 请同学们将所购书籍书脊上贴统一制作的、有 FDUROP 字样的标签，并在封三贴学者信息，如学者姓名，课题名称，简介，指导教师等信息。



## 附录 2、薏政项目结题报告格式要求

结题报告一般分两个部分，表格部分和论文部分，此外还可以根据自己的需要提供各种形式的附录。表格部分请在网站上下载“薏政项目结题报告”，根据要求认真填写。结题报告的论文部分必须包含题目、作者姓名和院系、指导教师姓名和单位、中英文摘要、关键词、引言（前言）、正文、注释、参考文献、后记（致谢）和导师评语。具体要求如下：

1. 页面设置：保留 word 默认设置（A4，页边距左 3.17，右 3.17，上 2.54，下 2.54，装订线 0，页眉 1.5，页脚 1.75）
2. 段落：在格式选项下的段落设置选项中，“缩进”选 0 字符，“间距”选 0 行，“行距”固定值 20 磅，“特殊格式”选（无）。
3. 字体和字号：中文字体一般用**宋体**（有特殊要求的除外），英文用 **Times New Roman 字体**  
论文题目字号为小二号，**加粗**，居中；若有副标题，则用小三号，**加粗**，居中。  
作者院系、姓名为小四号，**加粗**，居中；  
指导教师姓名为小四号，**加粗**，居中；  
摘要、关键词字号为五号，“摘要”和“关键词”**加粗**；（中文首行缩进两个中文字符；英文则顶格）  
引言（前言）标题字号为四号，**加粗**，居中；其他文字字号为小四号。  
正文各类标题（包括“参考文献”标题）**加粗**；一级标题用四号字体；二级标题、三级标题用小四号字体；正文其他为小四号；  
参考文献、后记（致谢）、指导教师评语为小四号；标题**加粗**，字号为小四号；指导教师评语放在表格中。
4. 附图和附表：图表名为小四号，**加粗**；附表的表头应写在表的上面，居中；附图的标题应写在图的下面，**加粗**，居中。按表、图、公式在论文中出现的先后顺序分别编号。**每个图表尽量放于一页内。**
5. 引文：较长的引文可以不加引号，单独成段，用**楷体**，字号为小四号。
6. 注释：论文一律用脚注（在 word 中“插入”—“引用”—“脚注和尾注”设置）。脚注（如引用文献），用小五号字，严格按以下顺序：序号（用纯阿拉伯数字）、作者姓名、书名（或文章名）、出版社（或期刊名）、出版或发表时间。
7. 参考文献按照正文格式，标题用小四号，居中。（标题**加粗**）  
内容及次序如下：著作为[序号]作者或编者、文献题名、出版地、出版社、出版年份、起止页码；论文为[序号]作者、文献题名、期刊名、出版时间、卷号或期号、起止页码；参考文献根据作者或编者姓名的第一个字按字母排序；若超过十篇，按著作和论文分类。
8. 致谢：为小四号，感谢基金的时候，统一用薏字(上竹下君，输入方法见 FDUROP 网站)。

9. 其他注意事项:

- 全文尽量不分栏（即只有一栏）。
- 不设置页眉页脚，全文尽量不插入各种分隔符。

10. 论文或报告限制在 12 页以内。

11. “项目编号”请在网站课题列表中的课题信息中查看

12. 样张（见下页）。

# 国有企业“错位”与非国有化问题

标题小二号，加粗

——以\*\*为案例

副标题小三号，加

经济学院 张某某  
指导教师 李某某

**摘要：**本文通过对国有企业职能的重新定位，阐述了当今国企改革的最佳途径，……。

**关键词：**国有企业<sup>1</sup> 非国有企业<sup>2</sup>。

空一行

**Abstract:** This paper relocates enterprise of country's function……

**Keywords:**

## 引言（前言）

国有企业……

（正文标题如下）

### 1 一级标题

四号加粗

#### 1.1 二级标题

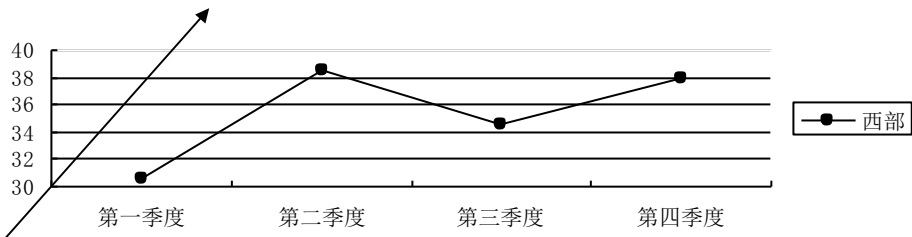
小四号加粗，无缩进

##### 1.1.1 三级标题

国有企业作为生产关系的一种组织形式，……。

表1 95年—98年统计表

	95年	96年	97年	98年
项 目				



空一行

图1 增长趋势图

国有经济的产业分布既受产业性质所约束，又受国有企业的目标及制度优势所指

<sup>1</sup> 国有企业……

<sup>2</sup> 非国有企业……

导，……

空一行

### 参考文献

空两行

[1] 李桂陵：《国有企业的规模及其演变轨迹》，中国经济出版社，1998 年，第 100 页。

[2] 王平：《中国国有企业改革》，中国经济出版社，1999 年，第 20 页。

后记（致谢）：感谢……

空两行

### 指导教师评语

该课题……