《数据库系统原理》课程设计指导书

LTE 网络干扰分析系统

北京邮电大学计算机学院

2021 年 3 月

一. 引言

在 LTE 网络中，移动终端 UE 与基站 eNodeB 间通过无线链路传递上下行无线信号，实现“移动用户-基站-核心网元-移动用户”间的相互通信。LTE 网络整个覆盖区域由一系列小区/扇区组成。在每个小区/扇区内，由位于小区/扇区中心位置的基站 eNodeB 为小区/扇区内的移动用户 UE 提供无线通信服务。地理位置相邻的小区 Cell 或扇区 Sector 的覆盖范围将会有重叠，而这些相邻小区具有**邻区关系**。在通话过程中，邻小区的可能会对主小区的产生干扰，从而影响通话质量。主邻小区的信号强度的差值可以表示邻小区对主小区的干扰强度，通过对干扰的分析可以找出潜在的干扰源，从而采取相应措施。

本次课程设计要求学生以小组为单位（每组 3-4 人），根据课堂教学所讲授的数据库系统的设计过程和开发方法，在《数据库系统原理》课程实验已经建立的数据库基础上，采用采用：（1）GaussDB 数据库系统平台（如 openGauss, GaussDB(MySQL)，或 PostgreSQL、MySQL），（2）Java、Python、C、C++、C#语言，（3）现有 Web 开发框架和工具，如 Springboot + VUE、Springboot + freemarker 模板，或其它开发工具（如 Qt），设计开发具有（1）三层 B/S 结构、或（2）两层 C/S 架构的数据库应用系统——LTE 网络干扰分析系统，支持对数据库系统中 LTE 网络配置数据、KPI 指标数据和 PRB 干扰数据的查询，并通过 MRO 测量报告数据计算主邻小区间的干扰，集成网络分析软件分析网络干扰结构，对 XML 形式的 MRO 测量数据进行解析。

二. 数据库应用系统生命周期和设计过程

本课程设计所开发的 LTE 网络干扰分析系统属于数据库应用系统。

面向某一具体应用领域的数据库应用系统（Data Base Application System, DBAS）包括:

- 以数据库文件形式存在的数据库 DB
- 关系数据库管理系统 DBMS， 其核心是数据库引擎. 如 openGauss、MySQL、PostgreSQL、Oracle、DB2 等。DBMS 提供了查询处理与优化、存储管理、事务管理等关系数据库系统的核心数据管理功能
- 数据库应用程序. 例如, 数据库系统对外接口程序、位于客户端、对服务器端的数据库 DB 进行各种数据查询与更新的数据库应用程序等。
- 数据库应用系统用户。可以是数据库系统的操作人员或数据库系统管理员 DBA, 也可以是其它应用程序。

数据库应用系统 DBAS 对外提供了**数据存储管理**和**数据访问与处理**功能。因此，DBAS 的设计与实现

包括以下 2 个方面：

（1） 数据库 DB 所存储的面向具体应用领域**应用数据的存储设计与实现**。

（2） 对数据库数据进行访问和处理的各种**数据库事务和应用程序**的设计与实现。

注： 对比 程序 = 数据结构 + 算法

数据模型 = 数据定义 + 数据操作

数据库应用系统 DBAS 的设计过程和生命周期如图 1 所示，具体参见“附录 3-数据库应用系统生命周期模型”。DBAS 的设计需要经过概念设计、逻辑设计、物理设计等步骤，有关物理设计的一些内容参见“附录 4-存储技术与物理设计”。

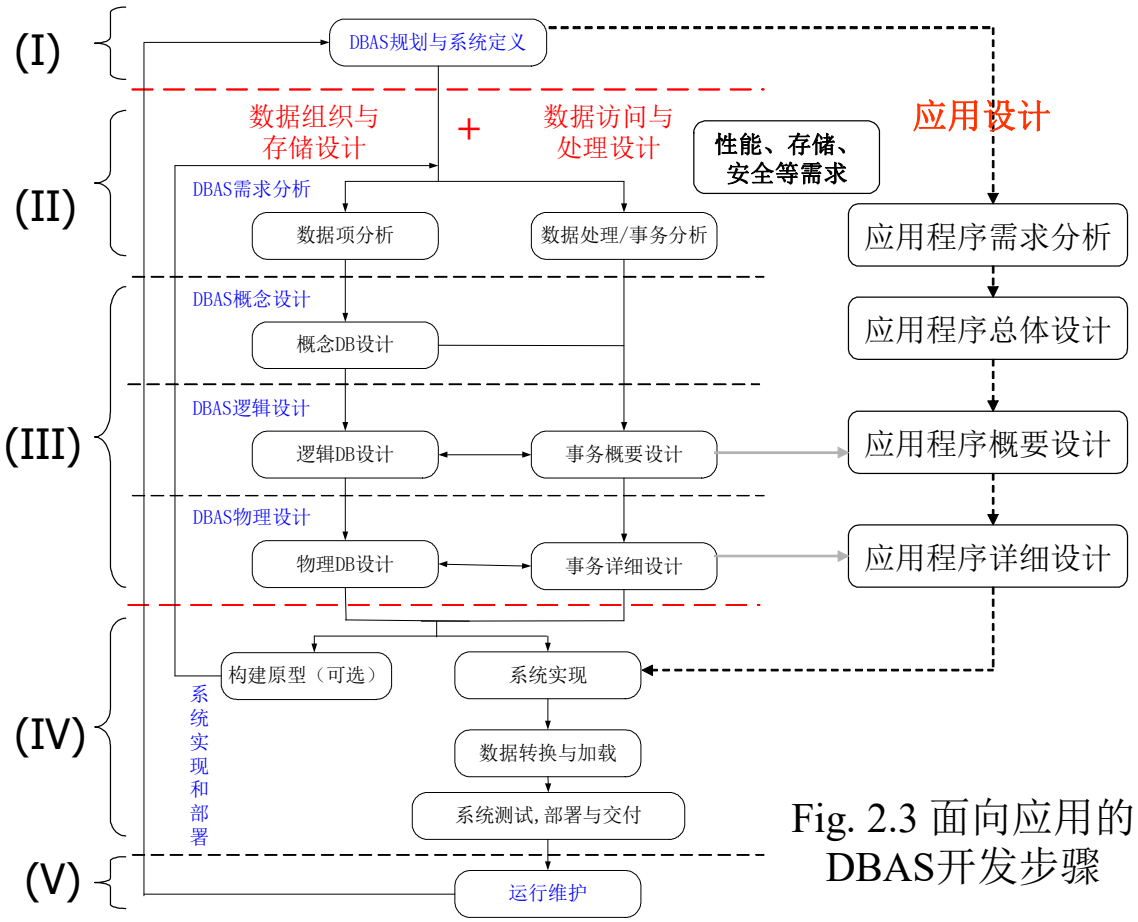


图 1 数据库应用系统 DBAS 的设计过程和生命周期

系统的层次结构如图 2 所示：

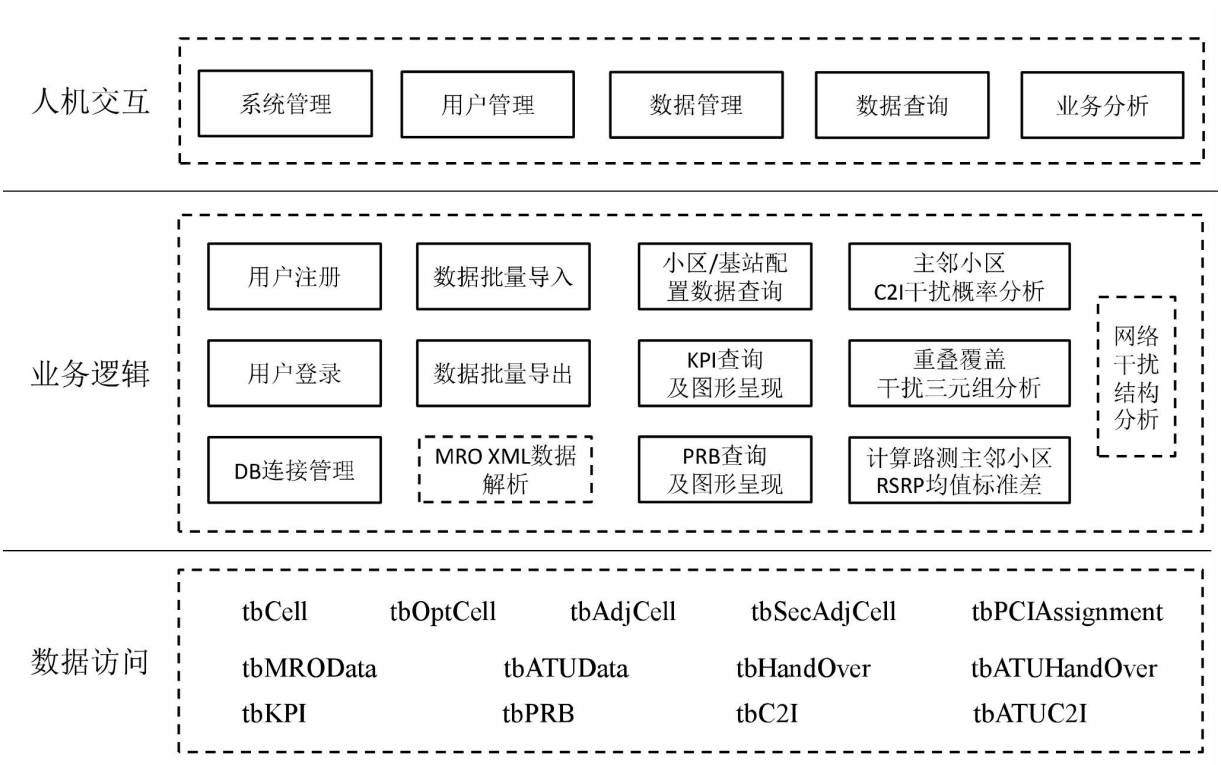


图 2　LTE 干扰分析系统层次结构

三．需求分析

所开发的 TE 网络干扰分析系统，实现如下功能。

3.1 系统界面

1) 登录界面

登录界面要求有：（1）用户登录，（2）用户注册/撤销等功能，用户分为系统管理员和普通用户两类。

用户通过口令、密码登录系统，注意区分登录系统的口令和连接数据库的口令，普通用户不允许知道连接数据库的口令。

2) 界面布置

要求界面布置合理，简洁美观。

注意设计界面时合理布置一级、二级菜单。一级菜单对应系统管理、用户管理、数据管理、

业务查询、业务分析五个主要功能。由一级菜单下的五个系统功能，分别点击进入二级菜单，二级菜单项对应上述五个功能下的具体细分业务功能，如“数据管理”下的“数据导入”。

选择数据库表时要求使用下拉菜单，不能将所有表名平铺在界面上。

3.2 用户管理

用户分为系统管理员、普通用户两大类，两类用户通过账号名、密码登录系统。

管理员可以查看用户注册信息，添加删除普通用户；管理员可以查看数据库连接、后台数据库服务器及数据库的配置信息，可以设置、修改数据库连接时长、数据库缓冲区大小等参数。

普通用户可以自己注册，但需要管理员审批通过。或者：普通用户自己不注册，只能使用管理员分配的帐号，登录系统，完成各项业务查询和业务分析功能。

3.3 系统管理

由系统管理员查看数据库连接、后台数据库服务器及数据库的配置信息，如数据库表所在的物理分区，修改数据库连接时长、数据库缓冲区大小等参数。

3.4 数据管理

3.4.1. 通过数据导入程序，批量导入以下 4 种数据

1) 网络配置信息导入

从外部导入 Excel 表形式的网络配置信息，即 LTE 网络配置数据库中的小区/基站工参表 tbCell，并存入数据库表 tbCell；

2) KPI 指标信息导入

从外部导入 Execl 表形式的天级的 KPI 指标统计数据（表“优化区 17 日-19 日 KPI 指标统计表”tbKPI），存入数据库表 tbKPI；

3) PRB 干扰信息导入

从外部导入 Execl 表形式分钟级的 PRB 干扰数据（表“优化区 17 日-19 日每 PRB 干扰 查询-15 分钟”tbPRB），存入数据库表 tbPRB；

4) MRO 数据导入

从外部导入 MRO 测量报告数据表 tbMROData，并存入数据库表 tbMROData；

批量数据导入要求:

— 从外部数据文件向数据库中某关系表导入一条主键值不为空的数据（对应于一个元组）时，根据

该条数据中的主属性/主键，判断数据库表中是否已有本条数据；

如果本条数据在数据库表中不在，则将该条数据作为新数据插入数据库。如果本条数据已经存在，则删除数据库中已有数据，将该条数据插入数据库，即用新数据覆盖数据库中的已有旧数据。

例如，当向数据库中的小区表 tbCell（CITY，SECTOR_ID，SECTOR_NAME，… ）中导入一条数据(‘三门峡’, ‘124672-0’, ‘三门峡渑池刘果-HLHF-1’, …)时，由于 SECTOR_ID 为主键，因此根据 124672-0 判断本条数据是否已在数据库 tbCell 表中，然后将本条数据插入数据库或覆盖已有数据。

- 所导入的数据文件的存放位置不固定，应允许用户在界面上选择导入文件的存放路径。
- 从外部数据文件导入数据时，选择表中典型字段，对导入的数据进行清洗，剔除不合格的数据项。例如，从文件 tbCell.xls 中导入小区配置信息时，对小区的经纬度信息进行清洗。如果发现文件中某行数据的经纬度数据不在合理范围内，该行数据不导入，并在导入日志文件（txt 文件）中记录改行数据编号。

说明：

tbMROData 为 csv 格式，不要将其强制改为 excel 格式，可以将导入功能实现两种格式，也可以采用 DBMS 提供的导入功能，将 tbMROData 和其它剩余表手工导入；或者：直接利用基本课程实验的数据库中已有数据。

3.4.2. 数据导出

在用户界面上用下拉列表显示数据库中的各个关系表，指定某些表，如 tbCell，将表中数据以 Excel 或 txt 格式的文件输出，要求输出文件中应当有表中各个属性名称，输出文件名为该关系表的名字。

至少从数据库中选择 3 张表，实现数据导出。

注意：导出的数据文件在系统中的存放路径必须是可选的，不允许在导出程序中采用固定的导出文件存放路径。

3.5 业务查询

至少完成以下几类业务信息的查询：

1. 小区配置信息查询

从 tbCell 表中查询小区信息，要求：

1）用户在查询界面的输入框中直接输入某小区的 Cell_ID/Sector_ID 或小区名称 CellName/SectorName，系统以列表方式输出该小区全部信息；

2）程序在查询界面的下拉列表中列出表 tbCell 中全部小区名称，用户通过下拉列表选择特定小区

名称，以列表方式输出该小区全部信息；

2. 基站 eNodeB 信息查询

从 tbCell 表中查询基站所属全部小区信息，要求：

用户在查询界面上通过输入框输入基站标识 eNodeBID 或基站名称 eNodeBName，或者通过下拉列表选择基站标识 eNodeBID 或基站名称 eNodeBName，以列表方式输出该基站全部小区信息；

3. 小区 KPI 指标信息查询

用户在查询界面上通过输入框输入小区名称，或者通过下拉列表选定表 tbKPI 中的某个特定小区，在右侧显示其属性列表，添加时间控件，查询网元某个时间段（天级）某个属性值的变化情况，并将结果用柱状图或折线图的方式呈现出来，例如查询网元从 2016 年 7 月 17 日到 7 月 19 日的 RRC 连接重建比率 (%)。

4. PRB 信息统计与查询

1） 根据表“优化区 17 日-19 日每 PRB 干扰 查询-15 分钟”，统计小时级的 PRB 干扰数据，生成一张新表 tbPRBnew，并导出到外部的 Excel 表中。

注：某一网元某一小时内的 PRB 干扰=该小时内 4 个 15 分钟平均值的和/4，要求计算所有的网元从 2016 年 7 月 17 日到 7 月 19 日的每个小时的 100 个 PRB 上检测到的干扰噪声的平均值。

2） 根据（1）中得到的小时级的 PRB 干扰数据表，在界面上根据网元名称以输入框或者下拉列表的方式选定某个特定网元，在右侧显示其属性列表，添加时间控件，查询网元某个时间段（小时级）某个属性值的变化情况，并将结果用柱状图或折线图的方式呈现出来，例如查询网元某一天从 01:00:00 到 13：00：00 每个小时内第 60 个 PRB 上检测到的干扰噪声的平均值。也可将时间段缩小到分钟级，此时用 tbPRB 进行查询，总之查询的时间粒度要和表对应。

注意事项：所有查询内容都要求可以导出，查询结果的图表要求可以导出，保存到可选文件存储路径下。

3.6 主邻小区 C2I 干扰分析

参照课程实验中的关系表 tbMROData 和 tbC2I。

tbMROData 记录了 LTE MRO 测量报告的内容。在 tbMROData 中，每一行表示位于服务/主小区 ServingSector 覆盖范围内特定位置的 1 个移动终端在一个特定时刻接收到的：1）来自服务/主小区 ServingSector 参考信号接收功率 LteScRSRP；2）来自周边某个邻小区 InterfereringSector 的参考信号接收功率 LteNcRSRP，该信号形成对主小区信号的干扰。

服务/主、干扰/邻小区间的 C2I 定义为：

C2I(ServingSector, InterfereringSector)= LteScRSRP - LteNcRSRP

C2I 值越小，说明邻小区对主小区的干扰越大。

对于一对主邻小区对<ServeringSector, InterfereringSector>，在主小区覆盖范围内，会有多个移动终端在不同地理位置、不同时刻收到许多条 MRO 测量报告。因此在 tbMROData 中，一对主邻小区会有许多条<LteScRSRP, LteNcRSRP>RSRP 测量值对。

例如<主小区 ID='5641-129'，邻小区 ID='15513-128'>有 6515 条测量值对。

将主邻小区对<ServeringSector, InterfereringSector>间的 C2I 看作符合正态分布的随机变量，根据 tbMROData 表中的多条<LteScRSRP, LteNcRSRP>RSRP 测量值对，可以计算 C2I 的均值 C2I_mean 和标准差 std；

说明：在 tbC2I 表中，主邻小区分别用 SCELL、NCELL 表示，有别于 tbMROData 表。

在此基础上，进一步计算：

1) 主小区信号比邻小区 RSRP 信号强度弱 9db 的概率，即主邻小区 RSRP 差值小于 9 的概率，

PrbC2I9(SCell, NCell)=Prb{C2I(SCell, NCell)<9}

PrbC2I9 越大，说明邻小区对主小区的干扰越大。

2) 主小区、邻小区的 RSRP 信号强度差值小于 6db 的概率 PrbABS6，即

PrbABS6(SCell, NCell)=Prb{ -6<C2I(SCell, NCell)<6}

PrbABS6 越大，说明主邻小区信号相差不大，主邻小区间出现重叠覆盖干扰。

具体要求和实现步骤如下：

1) 主邻小区 RSRP 差值（即 C2I）的均值 C2I_mean、标准差 std 计算

根据导入的 MRO 测量报告数据表 tbMROData 表中主小区和邻小区的参考信号接收功率 RSRP，计算主邻小区 RSRP 差值，该联系是多对多联系。

计算每对主邻小区<SCell, NCell>的 RSRP 差值 C2I，并计算出 C2I 的均值和标准差。

2) 主邻小区 RSRP 差值小于 9 的概率 PrbC2I9 计算

根据每对主邻小区 RSRP 差值的均值和标准差，可以得到正态分布，根据正态分布求出主邻小区 RSRP 差值小于 9 的概率 Prb9=Pr{C2I<9db}。

3) 主邻小区 RSRP 差值绝对值小于 6 的概率 PrbABS6 的计算

根据每对主邻小区 RSRP 差值的均值和标准差，可以得到正态分布，根据正态分布求出主邻小区 RSRP 差值绝对值小于 6 的概率（PrbABS6）。

4) 生成新表 tbC2Inew

新表包括六个属性：主小区 ID、邻小区 ID、主邻小区 RSRP 差值的均值、主邻小区 RSRP 差值的标准差、主邻小区 RSRP 差值小于 9 的概率、主邻小区 RSRP 差值绝对值小于 6 的概率。

注意：

1. 要求设置一个控制参数，将<LteScRSRP, LteNcRSRP>RSRP 测量值对的条数小于该参数的<LteScRSRP, LteNcRSRP>RSRP 测量值对先筛选掉，即当主邻小区间的干扰测量数据过少时，

不计算主邻小区间的 C2I 干扰值。

2. 根据 tbMROData 数据生成的 C2I 数据，存入新表 tbC2Inew 表中，tbC2Inew 中的数据应与数据库中原表 tbC2I 中的数据值不完全一样。

示例：

求一对主邻小区的信号强度差值的均值、标准差和相关概率：

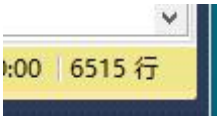
注：示例采用 SQL 语句只是表明一个计算过程（只计算了一个特定的主邻小区对），系统要求计算 tbMROData 全表的数据，需编写代码实现。

1) 选择一对特定小区的主邻小区的 RSRP 值（主小区 ID='5641-129'，邻小区 ID='15513-128'）

```
select LteScRSRP,LteNcRSRP from tbMROData
where ServingSector='5641-129' and InterferingSector='15513-128'
```

结果：

	LteScRSRP	LteNcRSRP
1	54	40
2	26	20
3	55	39
4	45	40
5	45	28
6	25	16
7	54	40
8	55	38
9	27	17
10	54	39
11	24	17
12	25	15
13	54	40
14	41	36
15	27	19
16	55	39
17	41	37
18	34	27
19	40	37
20	27	21



这一对小区间的<LteScRSRP, LteNcRSRP>RSRP测量值对共有6515个。

2) 计算主邻小区 RSRP 差值的均值和标准差（数据表中的 LteScRSRP 和 LteNcRSRP 类型为 int，而均值和标准差类型应该为 float，计算时要注意类型的转换）

```
select AVG(LteScRSRP-LteNcRSRP)mean,STDEV(LteScRSRP-LteNcRSRP) std
from tbMROData
where ServingSector='5641-129' and InterferingSector='15513-128'
```

结果 消息

	mean	std
1	8.96914811972371	6.41501441969715

3) 求 C2I 干扰概率

主邻小区RSRP差值的分布看作正态分布，已知均值和标准差，可以计算相关干扰概率。

- 差值小于9的概率PrbC2I9=0.501919;
- 差值的绝对值小于6的概率PrbABS6=0.311926;

3.7 重叠覆盖干扰小区三元组分析

根据表 tbC2Inew，找出所有的小区三元组<a, b, c>，其中 a、b、c 互为邻小区，并生成新表 tbC2I3，该表有三个属性，分别是三个小区的小区标识 ID，使得：

(1) PrbABS6(a, b)或 PrbABS6(b, a) ≥ x%;

(2) PrbABS6(a, c)或 PrbABS6(c, a) ≥ x%;

(3) PrbABS6(b, c)或 PrbABS6(c, b) ≥ x%;

，x 为用户设定的阈值参数。

在根据上述方法计算出的三元组<a, b, c>中，小区 a、b、c 相互间信号强度差别不大，没有主导小区，出现严重重叠覆盖，会严重影响通话质量、下载速率等 KPI 性能指标。

注意：

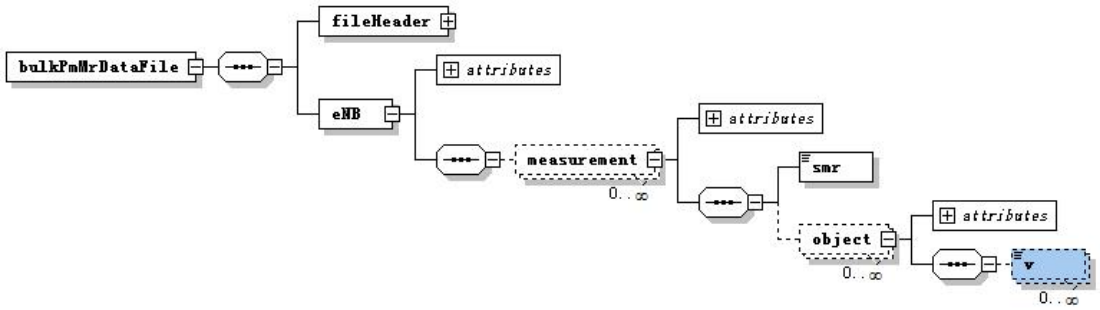
- 要求在设计系统时，阈值参数 x 是可配置的，在界面上输入不同的参数值显示不同的结果，例如 x=70；
- PrbABS6 即主邻小区 RSRP 差值的绝对值小于 6 的概率；
- 求小区三元组时要求去重，例如三元组<a, b, c>等同于三元组<a, c, b>和<b, c, a>，三者只需保留一个即可；
- 计算三元组涉及的小区数目、小区间干扰数据条数非常大，需要从性能角度考虑计算速度。不要只是简单地使用 SQL 语句、多表连接操作进行分析。

3.8 MRO/MRE 测量数据解析【选做】

3.8.1. MRO 数据简介

MRO 是终端用户每 480ms 发送一次的测量报告样本数据，一般以小时为单位存放在 XML 文件

中，因此数据量很大。利用 MR 数据我们能够计算得到当前网络干扰情况，分析得出干扰严重的小区后，可以选定这些小区作为优化小区展开优化工作。XML 格式的测量报告文件的结构图如图所示：



上图中<smr></smr>标签存储相应属性名，<object></object>标签存储属性名相对应的记录。下表给出主要字段含义。

字段名称	数据含义
eNBid	服务小区基站编号
MR.LteScRSRP	服务小区参考信号接收功率
MR.LteNcRSRP	邻区参考信号接收功率
MR.LteScRSRQ	服务小区参考信号接收质量
MR.LteNcRSRQ	邻区参考信号接收质量
MR.LteScEarfcn	服务小区频点
MR.LteScPci	服务小区 PCI
MR.LteNcEarfcn	邻区频点
MR.LteNcPci	邻区 PCI
MR.Tadv	时间提前量
MR.AOA	eNodeB 天线到达角

3.8.2. MRO 解析目标、步骤及预期结果

目标：编写程序，自动完成 MRO 原始数据的解析，具体包括压缩包解压、XML 文件解析、关键信息提取、主邻小区计算、RSRP 差值对统计等多个步骤，并将提取的结果保存到数据库中。

MRO 解析思路：

Step1. 一般情况下，从运营商拿到的原始 MR 数据不对 MRO、MRS、MRE 进行区分。因此，我们首先要遍历 $FilePath$，同时使用正则表达式对该文件夹下的文件进行过滤，得到 MRO 文件列表 $MroFile=\{f_1,f_2,...,f_n\}$。

注：利用正则表达式获取文件夹下所有 MRO 文件（MRO 文件的文件名中都包含有 MRO 字样，如 TD-LTE_MRO_ALCATEL_nanyang2_130897_20160913000000_1.xml.gz）。

Step2. 顺序获取列表中下一个文件 f_i，对 f_i 进行解压操作。

Step3. 对 xml 文件进行解析，从中获取服务小区基站 ID、主邻小区 Pci、频点 Earfcn 等信息。

Step4. 利用 Step3 中获得的主邻小区参数信息从公参信息表中查询获得主邻小区 ID。

Step5. 利用剔除规则，判断该条记录是否有效。若有效，则保存数据，等待批量插入数据库。剔除操作主要考虑主邻小区 ID、接收信号强度 RSRP、邻小区频点 NcEarfcn 以及物理识别码 NcPci 几个属性的取值。小区频点通常比较固定，包含 38400、37900、38908 等，但是考虑到各个地区可能存在一些出入，我们在进行解析前，先从公参信息表中获取到当前地区包含的所有频点信息集合 $Earfcn=\{e_1,e_2,...,e_n\}$。根据《中国移动测量报告技术要求规范》，我们可以得到小区物理标识的取值范围是 0~503，RSRP 的上报值范围是 0~97。由此，我们可以得到剔除规则。

$$NcEarfcn \notin Earfcn \,||\, NcPci < 0 \,||\, NcPci > 503$$
$$||\, Rsrp < 0 \,||\, Rsrp > 97 \,||\, ID_{邻} = NULL || ID_{主} = NULL$$

如果解析得到的属性信息满足上述表达式，则该条数据存在异常，我们要将其剔除。

Step6. 如果 $i = n$，所有文件解析完毕，将内存中数据批量插入数据库；否则，令 $i = i + 1$，返回 Step2 继续执行。

通过上述解压解析过程，可以得到完整的 MR 数据并快速解析出每条 MR 记录中小区标识。设计数据库表为 tbMROData 表：存储拆分后的 MRO 采样点主邻小区对 ID 及 RSRP 等信息，各字段如下表所示：

字段名称	字段类型	长度（B）	说明
MroID	int	4	采样点 ID
ServingSector	varchar	15	主小区 ID
InterferingSector	varchar	15	邻小区 ID
LteScRSRP	int	4	主小区 RSRP
LteNcRSRP	int	4	邻小区 RSRP
LteNcEarfcn	int	4	邻小区频点
LteNcPci	smallint	2	邻小区 PCI

解析结果示例如下：

MroID	ServingSector	InterferingSector	LteScRSRP	LteNcRSRP	LteNcEarfcn	LteNcPci
20	127634_0	243398_32	48	38	39148	201
20	127634_0	243730_32	48	34	39148	382
21	127634_0	243398_32	48	38	39148	201
21	127634_0	243730_32	48	34	39148	382

21	127634_0	272579_144	48	34	39148	203
81	127634_0	127634_2	55	48	37900	201
81	127634_0	233883_0	55	45	37900	189
81	127634_0	127634_1	55	44	37900	203

3.8.3. 多线程+数据库表分区

当数据量大时，上述过程可以使用多线程+数据库表分区技术来并行解析和分布式存储。

多线程并行解析：原始 MRO 数据中一个基站的所有数据文件存放在同一文件夹下，因此可以将集合 $MroFile$ 按照基站 ID 划分为文件簇 $MroCluster = \{c_1, c_2, ..., c_m\}$。多线程模式下，由于同一个文件簇中的多个文件的解析数据存在同一主键的 RSRP 差值对，如果以单个文件 $f_i, 1 \le i \le n$ 为执行单位，必须要解决多个线程之间的线程通信问题，需要申请一个共享数据块，以供多个线程互斥的访问并写入数据。为了避免这种互斥访问临界变量的时间开销，可以以文件簇 $c_j, 1 \le j \le m$ 作为执行单位，一个线程负责一个文件簇中所有数据文件的处理。

数据库表分区分布式存储：由于原始 MR 数据过于庞大，解析后所得有效数据也是海量的，如果仅仅存放在一张数据库表中，会使数据库表的索引文件变大，数据的查询、插入操作都会比小表更加耗时。表分区将一个逻辑大表分解为多个物理小表，因此也包含有多个索引文件，每次对大表操作都会转换到对应的小表上进行实际操作，使数据的查询、插入等操作速度得到提升。分区表可以存放在不同物理磁盘上，并行读写大大提升 IO 速度。

3.9 网络干扰结构分析【选做】

3.9.1. 网络干扰结构图及其分析

关系表 tbCell 描述了 LTE 网络中小区配置信息，实际网络中小区数目高达成千上万。tbC2I 表刻画了小区间的干扰关系，表 tbC2I 的内容如下：

	A	B	C	D	E	F	G	H	I
1	CITY	SCELL	NCELL	PrC2I9	C2I_Mean	std	SampleCo	WeightedC2I	
2	sanxia	124673-0	259772-0	99	2.19	2.71	62	6138	
3	sanxia	124673-0	259595-1	94	3.72	3.36	72	6768	
4	sanxia	124673-0	274161-12	93	4.44	3.12	54	5022	
5	sanxia	124673-0	259778-0	92	2.92	4.37	1806	166152	
6	sanxia	124673-0	253771-0	91	4.06	3.67	70	6370	
7	sanxia	124673-0	124813-0	88	3.36	4.73	100	8800	
8	sanxia	124673-0	47444-0	88	4.38	3.94	656	57728	
9	sanxia	124673-0	15472-129	88	1.86	6.01	86	7568	
10	sanxia	124673-0	124711-2	87	4.08	4.33	18626	1620462	
11	sanxia	124673-0	253917-2	87	3.77	4.69	5156	448572	
12	sanxia	124673-0	253901-0	85	5.2	3.6	530	45050	
13	sanxia	124673-0	15513-128	85	3.6	5.2	10042	853570	
14	sanxia	124673-0	253819-1	81	4.9	4.73	4536	367416	
15	sanxia	124673-0	124673-1	79	4.47	5.73	48231	3810249	
16	sanxia	124673-0	253929-1	79	4.86	5.2	56	4424	
17	sanxia	124673-0	15476-128	78	4.74	5.48	2116	165048	

其中 SCELL 和 NCELL 分别是主小区和邻小区的 id，C2I_Mean 为它们的同频干扰均值。

目标：

- 以关系表 tbCell、tbC2I 中的小区作为图中的节点，将 tbC2I 表中的一行数据作为连接本行主小区/服务小区 SCELL 与邻小区/被服务小区/干扰小区所对应的图节点间的一条边，C2I_Mean 作为边的权值，权值越大，干扰越大，这样就构建了一个网络干扰拓扑结构图。
- 针对网络干扰拓扑结构图，使用社区检测工具和算法进行社区划分，将网络干扰拓扑结构图划分为多个簇，使得同属于同一社区/簇的小区间干扰尽可能强，而分属于不同社区/簇的小区间的干扰尽可能小。
- 在此基础上，结合小区位置、天线高度、发射信号功率、小区覆盖等情况，分析网络干扰结构，并对划分和分析结果进行可视化，以便分析网络中各个小区的干扰特性。

3.7.2. 社区检测简介

一个社区由一组连接紧密的节点组成，同时这些节点与社区外部的节点连接稀疏。社区检测又称社区发现，就是在复杂网络中发现这些连接紧密的社区结构。与节点聚类不同的是，聚类更多注重节点属性的差异，而社区检测更多注重节点之间关系（边）的紧密程度。

Louvain[1]算法是一种基于多层次优化 Modularity[2]的算法，它的优点是快速、准确，被 Andrea[3] 认为是性能最好的社区发现算法之一。Modularity 函数最初被用于衡量社区发现算法结果的质量，它能够刻画发现的社区的紧密程度。那么既然能刻画社区的紧密程度，也就能够被用来当作一个优化

[1] Blondel V D , Guillaume J L , Lambiotte R , et al. Fast unfolding of communities in large networks[J]. Journal of Statistical Mechanics Theory & Experiment, 2008.

[2] Newman M E J , Girvan M . Finding and Evaluating Community Structure in Networks[J]. Physical Review E, 2004, 69(2 Pt 2):026113.

[3] Andrea, Lancichinetti, Santo, et al. Community detection algorithms: A comparative analysis[J]. Physical Review E, 2009.

函数，即将结点加入它的某个邻居所在的社区中，如果能够提升当前社区结构的 modularity。

Modularity 的定义如下：

$$Q = \frac{1}{2m}\sum_{i,j}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j)$$

$$\delta(u,v) = \begin{cases} 1 when\ u==v \\ 0\ else \end{cases}$$

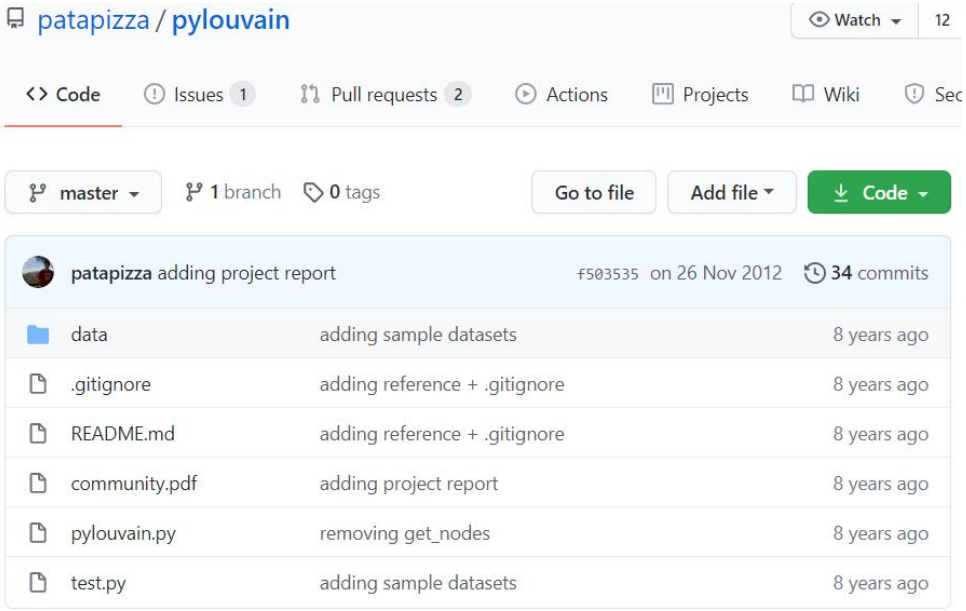
其中，A_{ij}节点i和节点j之间边的权重，网络不是带权图时，所有边的权重可以看做是1；$k_i = \sum_j A_{ij}$表示所有与节点i相连的边的权重之和（度数）；c_i表示节点i所属的社区；$m = \frac{1}{2}\sum_{ij} A_{ij}$表示所有边的权重之和（边的数目）。

Louvain 算法包括两个阶段，在步骤一它不断地遍历网络中的结点，尝试将单个结点加入能够使 modularity 提升最大的社区中，直到所有结点都不再变化。在步骤二，它处理第一阶段的结果，将一个个小的社区归并为一个超结点来重新构造网络，这时边的权重为两个结点内所有原始结点的边权重之和。迭代这两个步骤直至算法稳定。

3.7.3. 代码示例

1) 社区检测源码下载

此处我们使用的社区检测算法是 louvain 算法，它的一个 python 实现的下载地址：https://github.com/patapizza/pylouvain。下载 zip 解压到本地。



其中 pylouvain.py 是社区检测算法，data 文件夹存放要划分的网络数据，test.py 是测试程序。另外 community.pdf 是社区检测算法原论文。

该代码所需的环境较为简单，运行只需要 python3 的基本环境，可以在 python 3 环境下直接运行 test.py

即可。

2) 干扰数据预处理

写在前面：此处数据是直接使用 python 的 pandas 库直接读取 excel 表格作为示例进行干扰分析，同学们可以结合数据库进行改进，包括 ODBC/JDBC/PDBC 连接、单表查询等。

在 pylouvain.py 中提供了.gml 和.txt 两种格式的数据源网络的读取方法，其中.gml 文件存储网络数据结构类似字典格式，比较繁琐，而.txt 的比较简单，每一行有两列代表一条边的两个节点，，每一行也可以放三列，第三列存储边的值。

由于我们的数据是 excel 表，且表中含有其他无关列，我们只需要其中的 SCELL、NCELL、C2I_Mean 这三列的内容，所以我们需要写一个预处理代码，将 excel 表有用内容提取出来存储到.txt 中。

1. Pandas 库是 excel 文件读取很常用的库，我们需要在 python 环境中安装它。安装命令 pip install pandas

2. 用 excel 打开 10. tbC2I.xlsx，然后点击另存为.csv，这样就将文件格式改为了.csv，方便代码的读取。

3. 新建一个 python 文件，写入代码如下，注意其中文件地址要换成自己的文件地址：

```
# -*- coding: utf-8 -*-
import pandas as pd
csv_data = pd.read_csv(r"C:/Users/22117/Desktop/xxx 地区 LTE 网络数据-2020-09-03/xxx 地区 LTE 网络数据-2020-09-03/10. tbC2I.csv", encoding = 'utf-8',usecols=['SCELL','NCELL','C2I_Mean' ])
save_path = r"C:/Users/22117/Desktop/xxx 地区 LTE 网络数据-2020-09-03/xxx 地区 LTE 网络数据-2020-09-03/tbC2I.txt"
f = open(save_path,"a+", encoding='utf-8')
for index,row in csv_data.iterrows():
        f.write(row['SCELL'] +" "+ row['NCELL'] +" "+ str(row['C2I_Mean']) + "\n")
f.close()
```

4. 此段代码运行结束会生成一个.txt 文件，其中的内容如下图：

每一行有三列，前两列代表边的两个节点，第三列代表边的值。

5. 将 pylouvain.py 中的如下位置的 Int 改为 float，因为我们的边的权重不是整型，而是浮点型。

```
@classmethod
def from_file(cls, path):
    f = open(path, 'r')
    lines = f.readlines()
    f.close()
    nodes = {}
    edges = []
    for line in lines:
        n = line.split()
        if not n:
            break
        nodes[n[0]] = 1
        nodes[n[1]] = 1
        w = 1
        if len(n) == 3:
            w = int(n[2])
        edges.append(((n[0], n[1]), w))
    # rebuild graph with successive identifiers
    nodes_, edges_ = in_order(nodes, edges)
    print("%d nodes, %d edges" % (len(nodes_), len(edges_)))
    return cls(nodes_, edges_)
```

float

3) 网络可视化

由于原代码的测试部分没有任何输出展示，完成的功能仅仅是社区检测算法划分网络，但是没有画出网络效果图，而进行下一步的干扰分析需要将网络可视化，所以需要加上网络画图功能。

1. Matplotlib+networkx 环境搭建

   此处的画图使用 matplotlib 和 network 库，这两个库的安装都十分简单快速，跟上面 pandas 的安装相似，安装命令为 pip install matplotlib 和 pip install networkx。

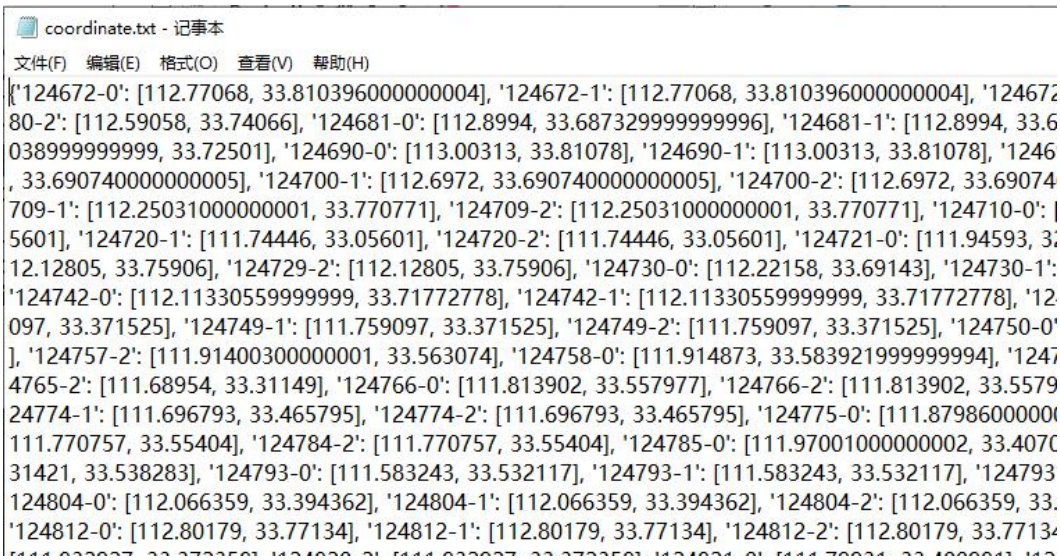
2. 构建节点 id 和坐标对应的字典

一般来说 networkx 画社交网络图时，由于社交网络节点没有特定的坐标，所以为了让节点不随机摆放，而是呈现一定的规律，networkx 中内置了一些布局函数如 spring_layout。但是我们的网络中节点是有确定的位置坐标的，也就是它的经纬度。利用经纬度作为坐标来展现网络更加符合问题场景。

每个小区的经纬度信息在表格 1.tbCell.xlsx 中。由于 networkx 接受的节点坐标输入是一个字典，字典的 key 就是节点的 id，value 就是坐标，所以我们同样需要利用 pandas 读取表格。首先将 1.tbCell.xlsx 另存为.csv 文件，然后新建 python 文件，写入如下代码（注意将文件地址换成自己的文件地址）：

```
# -*- coding: utf-8 -*-
import pandas as pd
csv_data = pd.read_csv(r"C:/Users/22117/Desktop/xxx 地区 LTE 网络数据-2020-09-03/xxx 地区 LTE 网络数据-2020-09-03/1.tbCell.csv", usecols=['SECTOR_ID','LONGITUDE','LATITUDE' ])
save_path = r"C:/Users/22117/Desktop/xxx 地区 LTE 网络数据-2020-09-03/xxx 地区 LTE 网络数据-2020-09-03/coordinate.txt"
coordinate_dict = {}
for index,row in csv_data.iterrows():
    coordinate_dict[str(row['SECTOR_ID'])] = [row['LONGITUDE'], row['LATITUDE']]
with open(save_path,"a+", encoding='utf-8') as f:
    f.write(str(coordinate_dict))
```

得到的 coordinate.txt 文件就保存了节点和坐标对应的字典，内容如下图：



3. 主函数

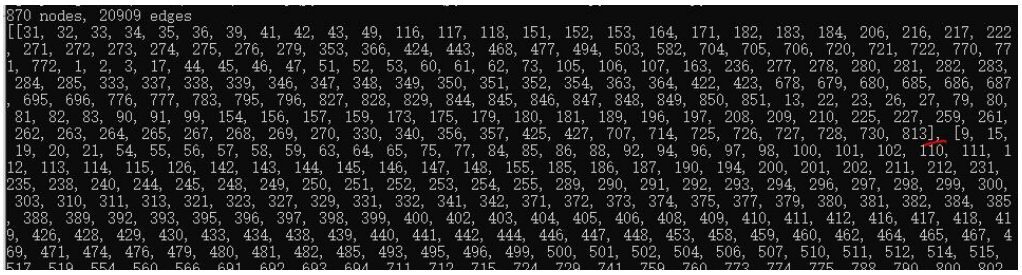
在与 pylouvain.py 同目录下新建一个 python 文件，写入如下代码（注意将文件地址换成自己的文件地址）：

```
# -*- coding: utf-8 -*-
from pylouvain import PyLouvain
import math
from matplotlib import pyplot as plt
import networkx as nx

filepath = 'data/tbC2I.txt'

# 获取社区划分
pyl = PyLouvain.from_file(filepath)
node_dict = pyl.node_dict # key 是 253916-2 的形式，value 是编号的形式
reverse_node_dict = dict(zip(node_dict.values(), node_dict.keys()))# key 是编号的形式，value 是 253916-2 的形式
partition, q = pyl.apply_method()
print(partition)
print("模块度：",q)

# 给各个社区节点分配颜色
community_num = len(partition)
print('community_num:',community_num)
color_board = ['red','green','blue','pink','orange','purple','scarlet']
color = {}
for index in range(community_num):
    print("社区"+str(index+1)+":"+str(len(partition[index])))
    for node_id in partition[index]:
        color[node_id] = color_board[index] # color 为一个字典，key 为编号形式的节点，value 为所属社区的颜色
new_color_dict = sorted(color.items(), key=lambda d:d[0], reverse = False)# 将 color 字典按照 key 的大小排序，并返回一个 list
```

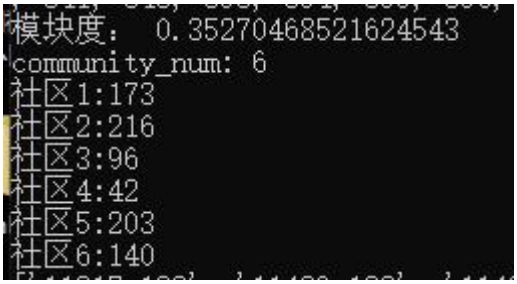
```
node_list = [reverse_node_dict[item[0]] for item in new_color_dict] #存储编号从小到大顺序对应的 253916-2 的形式的节点
color_list = [item[1] for item in new_color_dict]#存储 node_list 中对应的节点颜色
print(node_list)
print(color_list)

#构建 networkx 无向图
G = nx.Graph()
f = open(filepath, 'r')
lines = f.readlines()
f.close()
edge_list = [] #存储边列表
edge_width = [] #存储边列表对应的边粗细
edge_color = [] #存储边列表对应的边颜色
for line in lines:
    n = line.split()
    if not n:
        break
    G.add_edge(n[0],n[1],weight=float(n[2]))
    edge_list.append([n[0],n[1]])
    if color[node_dict[n[0]]] == color[node_dict[n[1]]]:#如果边的两端颜色相同
        edge_color.append(color[node_dict[n[0]]]) #则使用点的颜色作为边的颜色
    else:
        edge_color.append('c') #否则使用其他颜色
    if float(n[2]) > 40.0: #阈值
        edge_width.append(float(n[2])/100.0)
    else:
        edge_width.append(0.0)

# 可视化
plt.figure(figsize=(200,200))
f = open(r"C:/Users/22117/Desktop/老师给的优化/xxx 地区 LTE 网络数据-2020-09-03/xxx 地
```

```
区 LTE 网络数据-2020-09-03/coordinate.txt", encoding='utf-8')
pos_dict = eval(f.read())
f.close()
_node = [int(item.split("-")[-1])%4 for item in node_list] #提取后缀模 4 取余
node_0_index_list,node_1_index_list,node_2_index_list,node_3_index_list = [], [], [], []
for index,item in enumerate(_node): #划分不同后缀余数的群，以便给每个群分配一个节点的形状 node_shape 防止都用圆形，导致同一经纬度的节点重叠在一起
    if item == 0:
        node_0_index_list.append(index)
    if item == 1:
        node_1_index_list.append(index)
    if item == 2:
        node_2_index_list.append(index)
    if item == 3:
        node_3_index_list.append(index)
print("node_list:",_node)
nx.draw_networkx_nodes(G, pos_dict, nodelist=[node_list[i] for i in node_0_index_list],node_shape=7, node_color=[color_list[i] for i in node_0_index_list], node_size=50)
nx.draw_networkx_nodes(G, pos_dict, nodelist=[node_list[i] for i in node_1_index_list],node_shape=4, node_color=[color_list[i] for i in node_1_index_list], node_size=50)
nx.draw_networkx_nodes(G, pos_dict, nodelist=[node_list[i] for i in node_2_index_list],node_shape=5, node_color=[color_list[i] for i in node_2_index_list], node_size=50)
nx.draw_networkx_nodes(G, pos_dict, nodelist=[node_list[i] for i in node_3_index_list],node_shape=6, node_color=[color_list[i] for i in node_3_index_list], node_size=50)
nx.draw_networkx_edges(G, pos_dict, edgelist = edge_list, width = edge_width, alpha=1, edge_color=edge_color)
plt.show()
```

3.7.4. 结果展示

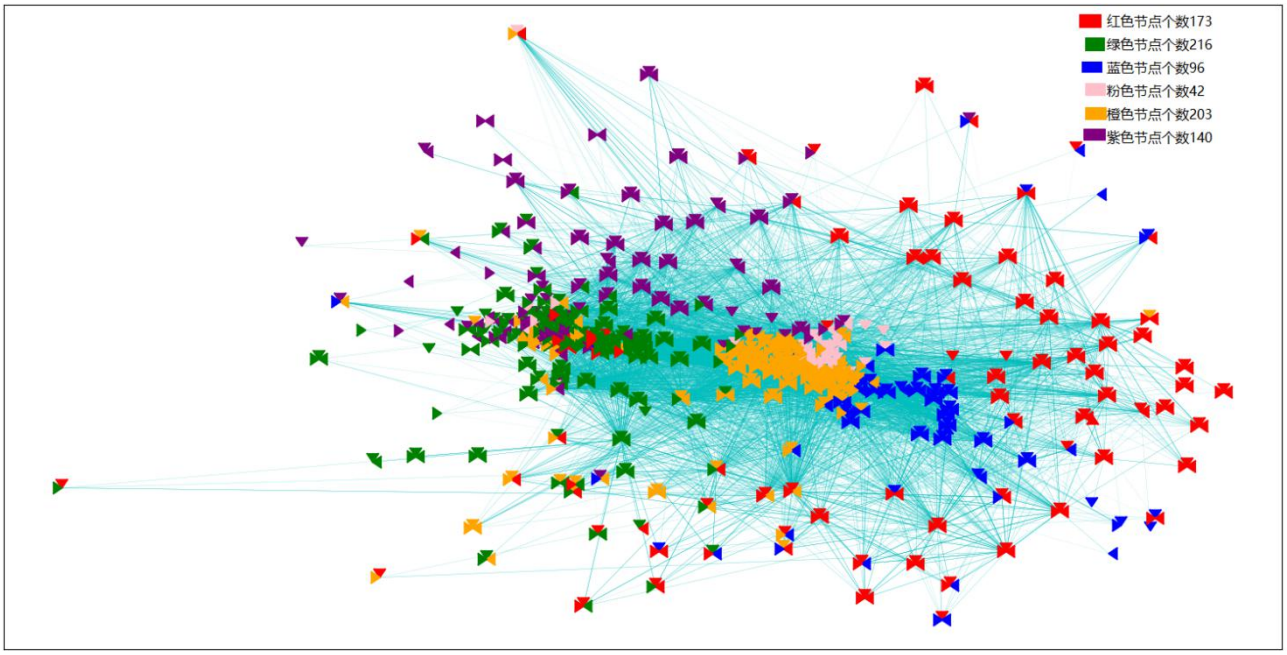


上图为控制台输出的结果，干扰网络有 870 个节点，20909 条边，louvain 算法将此网络划分为六个部分，如上图输出的 list。



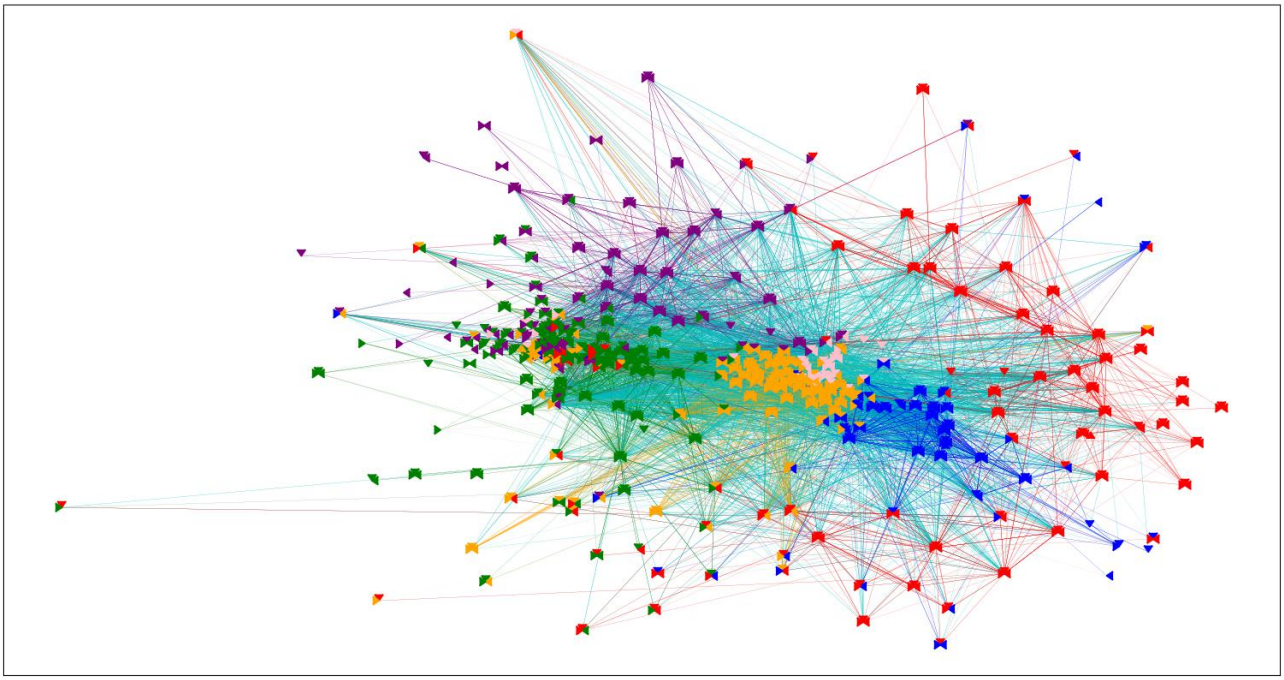
算法将网络划分成 6 个社区，每个社区的节点数分别为：173、216、96、42、203、140。最终的模块度为 0.3527。

将这六个社区的节点着以不同的颜色，边的粗细取决于边值大小，边的颜色统一用蓝青色，且每个节点以其经纬度作为坐标，考虑到一个基站可能有两个或者三个小区，他们的经纬度相同，为了避免重叠，将同一基站的小区按一定的规则画成不同方向的三角形。得到如下的网络图：



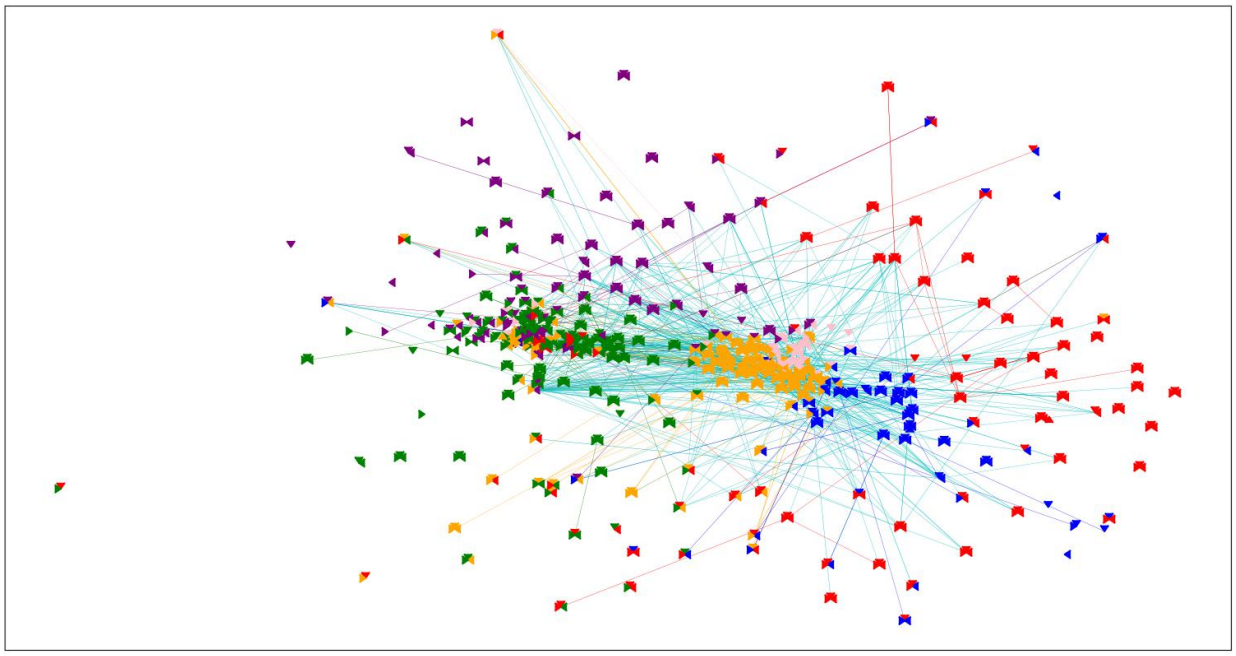
可以发现优化全局模块度的 Lovain 算法划分的社区内节点在坐标空间上也比较相近，我们在社区划分中没有使用到坐标信息，使用的是小区之间的 C2I 干扰强度作为边权，而干扰是与距离相关的，所以能在坐标空间体现出聚集性。从上图可以发现，仍有一些社区节点离得比较远，所以 C2I 干扰不能完全表现空间。

为了看出社区划分的效果，将边着以不同颜色：若边两端节点属于同一社区就着以该社区的颜色，否则蓝青色。如果蓝青色线条比较细、少，则说明社区划分效果较好，因为蓝青色线表示不同社区的连线。



从上图可以发现，整体上看，中间部分的蓝青色线还是比较多的，说明划分的社区之间的干扰还是比较严重。考虑到线条太多，线条遮挡严重，可以设置线条的边权大于某个阈值才显示，下图我设置了阈值为 20，边权大于 20 的线才会被显示。

【问题：工具软件的分析结果能否在本次课程设计开发的系统的界面上呈现出来】



可以发现，蓝青色的线条仍然很多，说明划分的社区之间干扰比较严重。造成这种现象的原因可能是网络的干扰较为复杂，存在很远距离的严重干扰，也有可能是划分的社区不够好。由于该可视化比较简单，不能对某一颜色的线的起点和终点进行统计和显示，故而不能确定地分析出确切的原因，有兴趣的同学可以在这方面进行改进可视化效果。

四． 系统架构和应用程序设计

4.1 系统架构

参照图 2，系统采用三层 B/S 架构。其中数据访问层采用 GaussDB 数据库系统（如 openGauss, GaussDB(MySQL)，或者 PostgreSQL、MySQL，存储小区/基站配置数据、C2I 干扰数据、MRO 测量报告、路测数据、小区性能指标数据等；业务逻辑层实现系统管理、数据导入/导出、业务查询模块、业务分析（PRB 小时级计算、干扰分析、MRO 数据解析等）等功能。用户通过浏览器访问系统，使用系统提供的各项功能。

开发语言可采用 Java、Python、C、C++、C#等。

为提高系统开发效率，建议（但不强求）采用一些简单易学的 Web 开发框架和工具，如：

（1）Springboot + VUE、Springboot + freemarker 模板，并借鉴 CSDN、知乎等技术网站/论坛上提供的开发案例和系统源码，示例：

一 springboot+vue.js 搭建图书管理系统开源项目，

https://blog.csdn.net/weixin_45132238/article/details/112390505

- **现在开发网站 web 应用一定要前后端分离吗**-知乎- Zhihu
  https://www.zhihu.com/question/442535354
- **基于 Vue+Nodejs+MongoDB 小区社区综合治理管理系统毕业生源码案例设计【知乎】**
  基于 Vue+Element+nodejs+Express+mongoDB 社区综合治理管理系统
  https://blog.csdn.net/saber04/article/details/107086249

（2）SSH 框架，前后端分离，前端采用 JSP+HLML+JS，后端采用传统 JDBC 与 HQL 相结合。

示例：2017 级课程设计案例

用户管理中的权限管理也可以采用现成框架，如 OAuth2，Shiro 等，样例参见：

- **如何设计一个通用的权限管理系统，https://juejin.cn/post/6850037267554287629**

也可以或两层 C/S 架构和对应的开发环境/工具，如 Qt。

4.2 数据批量导入程序

导入数据时，要求数据库中已有对应的表结构。

如前所述，向数据库表导入数据时，必须根据导入的数据在数据库中是否已经存在，采取 insert 或 update 动作。如果完全由导入程序逐条判断并导入，导入过程耗时费力，将非常慢。为此，采用数据库系统提供的批量导入技术，实现对导入数据的清洗和快速导入。

需要利用 bulkinsert、bulkcopy 等批量导入语句，开发批量导入程序和定义关系表上的导入触发器，2 者协同配合完成数据清洗和数据批量导入工作。

以从数据文件 Cell.xls 向关系表 tbCell 中批量导入小区配置数据为例，其原理如下图所示。

第 1 步. 分批读取输入文件数据

导入程序分批读取导入数据文件，每次从数据文件 Cell.xls 文件中批量读取多行（如 K=50 行）数据，存放在内存数据结构 InputBuffer 中，例如在 C#语言中的 datatable 数据结构。

在 C#语言中，采用 datatable 结构的 InputBuffer 具有数据库表 tbCell 完全一致的逻辑结构。

第 2 步. 数据清洗

在导入程序将 K 行数据逐条读入 InputBuffer 的过程中，对导入数据进行清洗，判断每条数据的各个属性值是否符合数据类型要求、取值是否在合法范围，剔除掉来自于数据文件的不符合要求的输入数据。并在导入日志（txt 文件）中记录被剔除的数据在输入文件的位置、编号。

第 3 步. 批量插入数据

对经过清洗后、存放在 InputBuffer 中的多条数据，采用数据库提供的批量插入语句，例如 BulkInsert、

bulkcopy，配合定义在关系表 tbCell 上的导入触发器，将这些数据全部插入数据库。

DBMS 收到由批量插入语句传来的需要导入 tbCell 表的数据后，自动启动定义在 tbCell 表上的导入触发器，由导入触发器对这批数据进行 insert 或 update 操作。具体如下（以 SQL Server 为例）：

（1）当用 bulkinsert 向 tbCell 中插入数据时，在系统内部会自动生成一个临时关系表 tbInserted，该表存放被插入的数据，并且定义在该表上的导入触发器是可以直接访问 tbInserted 的。

（2）导入触发器依次扫描 tbInserted 中的每个元组，如果某个元组中的 CellID 在数据库表 tbCell 中已经出现，则将该元组放到临时关系表#UpdateTemp。按照此种方式，找出导入数据中需要 Update 操作的全部元组，并暂时存放在#UpdateTemp。

（3）tbInserted 中存放的是对应于 Insert 或 Update 操作的 2 类元组，因此将 tbInserted 与#UpdateTemp 做集合差操作，得到的就是以前没有在 tbCell 中出现过、需要 Insert 的元组，这些元组缓存在#InsertTemp 中。

（4）触发器对临时关系表#InsertTemp、#UpdateTemp 中的元组分别执行 Insert、Update（也可以先 delete、后 insert）操作，将数据导入 tbCell 中。

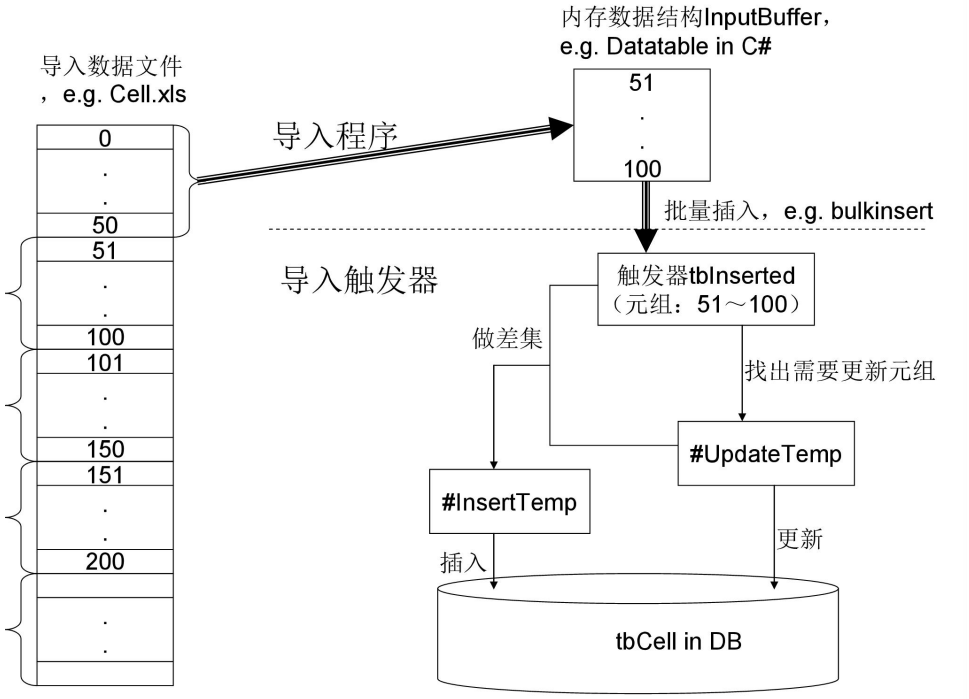


图 4-1 数据批量导入过程示意

注意事项：

一 不允许将数据导入文件的全部内容一次性地都读到内存中

原因：实际应用中，有可能导入数据文件非常大，如 10G 大小的路测数据文件。要求先对文件内容进行清洗，然后再导入数据库，为了清洗数据，必须将文件中数据先读入内存。数据文件过大，一次性全部读入内存会导致死机。

正确做法：根据内存大小、需要导入的数据文件中一行数据的大小，计算允许一次性导入内存的数据量，如 400M。如果数据文件没有超过 400M，则一次性导入文件；如果文件大于 400M，分批导入。

简单做法：分批导入，每次导入固定行的数据，如 50 行。

- **数据文件中多行数据读入内存、进行清洗后，不要再写回到磁盘文件中，应直接批量写入数据库，避免对 1 个大的数据文件（如 10G）多次读写磁盘。**

C#/Java 中用 bulk insert、bulk copy 等操作可以将内存中的数据批量地插入数据；如果采用 Python 语言编写数据导入程序，可以采用 ExcuteMany 方法实现数据批量插入数据库。

4.3 存储过程与触发器

本系统的多个功能需要使用一系列涉及数据库访问的计算过程，如根据 15 分钟级 PRB 数据计算小时级 PRB 指标、根据主邻小区 RSRP 差值计算均值和标准差等。

可以将上述这些涉及到数据库访问的计算过程开发为存储过程，并编译、存储在数据库服务器上，应用程序的各功能模块直接调用这些存储过程，从而实现查询执行计划重用，提高系统开发和程序运行效率。

此外，数据导入功能需要使用触发器来实现。

4.4 数据库物理文件/文件组设计

数据库在物理层面由多种数据库文件组成，如存放应用数据的数据文件、存放日志的日志文件。

数据库物理设计时需要结合数据库平台提供的机制，创建数据库，将数据库文件根据实际需要安排在不同的磁盘分区中。

例如，SQL Server 数据库的数据库文件分为主、辅文件、日志文件、文件组，数据库物理设计时，可以利用 create database 语句部署在不同物理分区。

查阅资料，根据课程设计采用的 openGauss、PostgreSQL 等数据库的数据文件类型、日志文件结构，以及数据库文件部署方式，将数据库的数据文件安放在不同的分区中，不要求只放在默认分区。

4.5 索引设计

为提高系统的数据查询效率，需要根据用户访问情况，在 tbCell 等关系表的相关属性上建立合适的索引，并利用 DBMS 平台提供的机制，观察分析索引执行效果，如查看 SQL 语句查询执行计划，分析索引有无起到预期作用。设计结果记录在数据库表设计文档中。

索引建立的原则参见课程讲义、“实验 8 数据查询分析与优化”。

五． 课程设计内容

在上学期的课程实验基础上，按照前述原理和方法，遵循“图 1 数据库应用系统 DBAS 的设计过程和生命周期”，设计开发 LTE 网络干扰分析系统。

系统设计包括数据存储的设计和数据处理设计两方面，即数据库表设计、软件模块结构设计（如系统模块化分、各模块输入/输出和功能描述）、索引设计等。

根据系统设计开发过程和结果，完成课程设计报告中，课程设计报告主要内容如下：

- （报告第 1 页）封面：北京邮电大学课程设计报告

  按照封面规定的格式，填写小区成员信息和人员分工、课程设计内容（简介）。
- 课程设计的目的、内容
- 系统需求分析

  从系统界面、数据管理、业务查询、业务处理等几方面，归纳系统应实现的功能，以及应到达的性能。
- 系统设计

  包括：系统架构/层次结构设计，软件模块设计，

  数据库设计（E-R 图设计、数据库逻辑设计/关系表设计、函数/触发器设计、索引及数据库文件设计、批量导入设计等），

  接口访问接口设计等。
- 系统实现

  软硬件平台，开发环境，编程语言，采用的框架
- 系统运行示例

  系统提供的 5 大类业务功能的运行示例截图
- 总结

注意：

- 课程设计报告文档命名规则：**班级_姓名 1_姓名 2 数据库课程设计报告**
- 系统代码作为单独附件提交

六． 注意事项

1. 选择合理的开发语言和环境，保证各项实验内容均能按照要求完成；
2. 先使用 create database 语句创建数据库，合理设计数据库主文件、辅文件，将数据文件、日志文件、索引文件安排在不同磁盘分区中；

   数据库创建完成后，再建立数据库表。可以采用基本课程实验的建表脚本；
3. 数据库表创建后，在后续的导入/导出、查询、分析过程中，不允许修改数据库中结构，如不允许 drop、create 关系表；但允许动态调整索引；

   注意区分：DB 设计者、DB 使用者的角色，权限不同；
4. 界面设计，详见 3.1，可以有自己个性化的设计，但必须涵盖所有功能，注意页面布局，尽量简洁、美观、直观，建议采用一级菜单-二级菜单项的界面排列方式；不要将全部功能安排在一个页面内；
5. 用户登录界面应区分开用户级口令、密码和数据库连接级口令密码，对用户级口令密码要有用户注册功能；
6. 导入时必须完成批量导入，不要一次性导入全表，而且数据库已存在表结构（为空表或旧数据）；

   导入表时要求通过路径选择任意的表；导出数据库中表和查询结果的图表都要求导出路径可选，不要直接在代码里固定表名和路径名；

   采用触发器，判断导入的数据是否主键与表中现有数据重复，采取 update 或 insert/delete 方式更新数据，建议：采用 update，但需要实验比较，哪个好一些？？

   数据分批时，要根据输入内容和可用内存估计，合理选择每次读入内存清洗的数据量。不能简单地输入读入 1 个货多个文件块。

   导入时：选择要导入的数据库表名/数据库中数据类型(如小区级工参)、要导入的输入文件名和路径；

   数据导出时，可以自定义导出文件名，选择文件存放路径

   防止数据导入时，分批读取导入文件中数据——不能全读到内存中，进行数据清洗（包括类型检查、范围检查）；对清洗后的数据直接由内存，通过数据库批量导入操作，写入数据库。不允许：将清洗后的数据写入磁盘文件，再有文件导入数据库。
7. 导入时，增加导入进度条；
8. 查询时，要求输入框和下拉列表的方式都实现，2 种方式放在同一个框中；

输入查询条件时，对数值型属性，如 eid，需要进行类型检查，防止输入非数值型数据导致死机；

9. 与时间相关的 KPI 和 PRB 查询时要注意：

   （1） 查询输入条件包括：

网元对象，KPI 指标/PRB，查询时段的起点、终点；

通过日历、时间控件选择起点、终点。

（2） 查询时段的起点、终点的时间粒度需与表中数据的时间粒度保持一致。例如，KPI 如果是天级数据，则 KPI 查询时需要选择年、月、日；PRB 数据的时间粒度为小时级，则需要输入年、月、日；

（3） 用图表呈现结果时，可以选用柱状图、折线图；注意纵坐标范围和单位坐标的选取，使得图形能有一个好的呈现效果；

10. 不能遗漏数据库存储过程和函数相关的实验内容；
11. 干扰分析计算 C2I 时，设置可配置的样本下限阈值。当主邻小区对的 C2I 样本数小于该阈值时，可以不计算该主邻小区对的 C2I 干扰的均值、标准差、Prb6 概率等。
12. 计算干扰三元组时，筛选主邻小区对的阈值参数，不要固定为 0.7，应该在界面上可配置；注意对三元组去重；
13. 编写系统代码时要注意代码的灵活性和通用性，在完成系统功能的基础上，可根据实际情况添加一些必要的功能；
14. 设计和实现系统时要多思考各种合理有效的实现方法，不要拘泥于指导书给出的实现方式。

七． 课程设计实验环境与工具

- DBMS:

  GaussDB 数据库系统平台，如 openGauss, GaussDB(MySQL)

- Programming languages:

  Python、C、C++、C#、Java

- DB interface

  ODBC, JDBC， ADO, Python/connector 等

- 开发框架

  Springboot + VUE、Springboot + freemarker；

  SSH [JSP+HLML+JS，后端采用传统 JDBC 与 HQL]

- GaussDB 课程资源

  openGauss、Gauss(MySQL)软件及文档资料

八． 课程设计要求

1. 分组完成课程设计，每组不超过 3-4 人。

   如果采用 B/S 架构和 Web 开发框架，且系统实现的功能比较多，设计和开发工作量较大，每组成员 4 人；如果没有采用开发框架，或采用 C/S 架构，每组三人。
2. 开发流程遵循图 1 所示的 DBAS 的设计过程和生命周期模型规定的步骤。每一步中应遵循数据库系统开发的基本原理和方法。
3. 课程设计文档应格式规范，内容完整。包括封面、目录、正文内容、案例图片、附录等
4. 课程设计完成后，各组提交设计文档、程序，并通过程序演示进行课程验收。
5. **重点注意事项：**

   1） 提交文档的名称统一采取如下形式：

   班级_姓名 1_姓名 2 数据库课程设计报告.doc

   2） 文档必修包括封面，封面格式参见：

   课程设计报告封面(分组版)

   在封面栏目“课程设计内容”中，填写课程设计内容、组员分工。

   3） 不要照抄指导书中内容

   4） 将所有设计文档合并在一个 Word 文件中，图片插入文档中合适位置；所有程序代码以附录形式单独放在一个文件中。
6. 实验必须包括用户管理、数据批量导入、存储过程与触发器、索引设计、查询指标图形化显示这 4 方面内容，如有缺失，扣分。
7. 第 8 周左右中期检查
   - 线上或线下验收
   - 各组应完成的实验内容不少于全部内容的 50%，保证课程设计进度
8. 最终课程设计验收在第 15-16 周，提前完成的组可提前预约验收
9. 按时提交文档，并进行验收。验收前，文档发至：yewen@bupt.edu.cn

验收时间、地点：待定

如果提前做完，可以提前预约验收。

北京邮电大学课程设计报告

课程设计名称	数据库系统原理课程设计	学院	计算机学院	指导教师	
班级	班内序号	学号	学生姓名	成绩	
2018211xxx			xxx		
2017211xxx			xxx		
课程设计内容	简要介绍课程设计的主要内容，包括课程设计教学目的、基本内容、实验方法和团队分工等 教学目的： 内容方法： 分工：				
学生课程设计报告(附页)					

课程设计成绩评定	遵照实践教学大纲并根据以下四方面综合评定成绩： 1、课程设计目的任务明确，选题符合教学要求，份量及难易程度 2、团队分工是否恰当与合理 3、综合运用所学知识，提高分析问题、解决问题及实践动手能力的效果 4、是否认真、独立完成属于自己的课程设计内容，课程设计报告是否思路清晰、文字通顺、书写规范 **评语：** **成绩：** 指导教师签名： 年 月 日

注：评语要体现每个学生的工作情况，可以加页

实验验收表格

班级	姓名，学号，分工
DB 平台，开发语言	

完成情况及评分点【待重新修订】		
实验内容及功能点		完成情况
文档内容（20 分）	1．实验目的、内容（2 分）	
	2.设计： 2.1 层次结构（2 分）； 2.2 业务层软件模块设计（2 分）；	（名称、功能、输入/输出、流程）
	2.3 数据库层设计： E-R 图设计（2 分）； 数据库关系表设计（2 分）：	（关系表定义、属于第几范式）
	2.4 存储过程和/或触发器/函数设计 （2 分）	
	2.5 索引设计（2 分）； 聚集、非聚集索引	
	2.6 数据库文件/文件组设计：数据库主文件、辅文件、日志文件、文件组，安排在不同的磁盘分区中（1 分）	
	2.7 数据库访问接口（1 分）：	软件模块使用的 API 接口
	3. 系统实现：（1 分）	软硬件平台、开发环境/语言、数据库平台
	4. 系统运行示例（系统登录、主要业务功能运行截图）（2 分） 5. 总结（1 分）	
登录界面（3 分）	口令、密码、注册	

批量导入（6 分）	1. 输入文件分组，如每 50 行一组；（2 分） 2. 对输入数据进行范围检查，如 SECTOR_ID 定义在合理区间内（至少 2 个属性）（2 分）	
	3. 批量插入命令 *bulkcopy*, *bulkinsert*（2 分）	
	4. 触发器：insert/delete/update，处理新插入数据、已有数据（4 分）	
数据导出（3+4 =7 分）	数据库表导出（已有表和新生成的表）	
	查询结果导出（包括图表）	
索引设计与分析（8 分）	主键聚集索引设计（2 分）	
	非聚集索引设计（至少 2 处）（3 分）	
	观察对比查询执行计划，比较索引效果（3 分）	
数据库文件 / 文件组物理设计（6 分）	观察：数据库主文件、辅文件、日志文件、数据库文件组，列出文件、文件组名称；（3 分）	
	Create Database 定义语句、所在磁盘分区；（3 分）	
采用日历 / 时间控件，输入查询条件 (10 分)	PRB 小时级呈现	
	对性能指标（KPI，PRB）按时间段查询，如 xx 年 xx 月 xx 日——xx 年 xx 月 xx，注意时间粒度和表一致!!!（至少 2 个指标查询）	
查询指标图形化显示（10 分）	至少 2 个指标显示；（3 分）	
	采用的图形显示控件；（2 分）	

存储过程/触发器/函数（4+4+4=12 分）	存储过程:	
	触发器:	
	函数：e.g. 计算主邻小区 RSRP 差值的均值、标准差。	
干扰分析三元组（5+5）=10 分）	计算 C2I，生成新表。	
	三元组查询结果。	
MRO 数据解析（4 分）	MRO 数据入库，XML 数据解析，表结构和索引设计	
网络结构干扰分析（4 分）	网络分析工具的集成， 分析结果的可视化 分析结果的解释	

九．参考文献

教材：

Abraham Silberschatz, Henry F.Korth, S. Sudarshan, **Database System Concepts** (Fifth Edition), Higher Education Press and McGraw-Hill Companies, Beijing, May, 2007

参考书目：

1. 蒋崇礼，培养计算机类专业学生解决复杂工程问题解决的能力，清华大学出版社，北京，2018 年 7 月