建邺区MOOPs-SL 实验报告

需要打开的数据和服务包括：

理论空间地图MXD

启动MongoDB服务；

运行代码工程；

实验报告文档复现一遍；

# 1、定量评价指标

Objecteve count 2,3,4

Popsize=100,500,1000

Iteration=1000

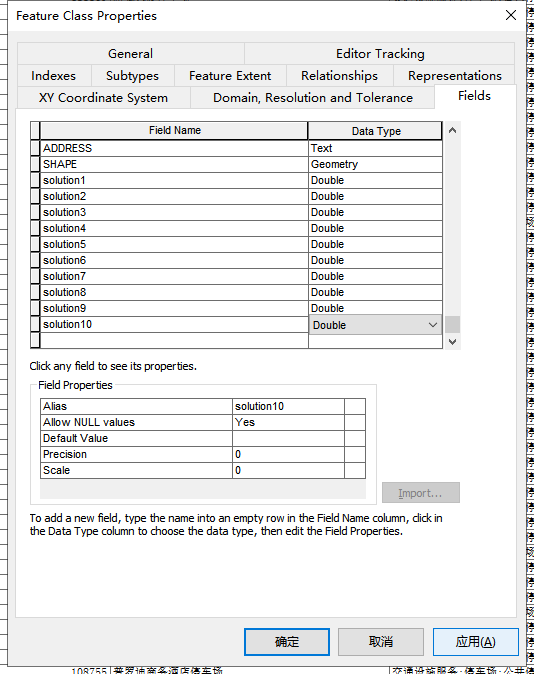
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试函数 | 求解问题 | 分类 | 种群规模 | 迭代次数 | SS | 耗时 |
|  |  | A1 | 200 | 10 | -0.922 | 15 |
|  |  | A1 | 200 | 10 | 0.227 | 16 |
|  |  | A1 | 500 | 20 | 0.975 | 102 |
|  |  | A1 | 500 | 20 | 0.928 | 104 |
|  |  | A1 | 500 | 40 | 0.865 | 207 |
|  |  | A1 | 500 | 40 | 0.849 | 209 |
|  |  | A1 | 1000 | 20 | 0.890 | 308 |
|  |  | A1 | 1000 | 20 | 0.968 | 303 |
|  |  | A1 | 200 | 20 | 0.140 | 89 |
|  |  | A1 | 200 | 20 | -0.183 | 88 |
|  |  | A1 | 500 | 100 | 0.095 | 900 |
|  |  | A1 | 500 | 100 | 0.0617 | 893 |
|  |  | A1 | 1000 | 100 | 0.0385 | 2389 |
|  |  | A1 | 1000 | 100 | 0.109 | 2374 |
|  |  | A1 | 5000 | 1000 |  |  |
|  |  | A1 | 5000 | 1000 |  |  |

# 2、建邺应用案例

用建邺的数据来做

复制park\_jy图层，然后得到park\_ky\_test图层

给park\_jy\_test图层增加10个字段，存放solution



# 昆山实验

## 001准备数据

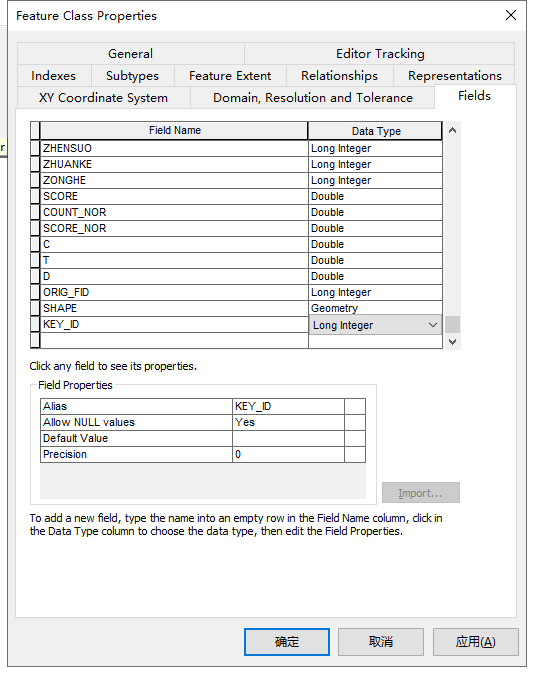
需要三个图层数据，demand，provider，潜在供给点，三个层都是点层数据，每个层必要的关键字段为：

Demand：KEY\_ID，population，name，X，Y

如果是多个时刻，则population可以有多个；

Provider：KEY\_ID，Scale，name，X，Y

潜在供给点与Provider的结果是一样的，但是为了反填结果，需要增加solution字段，数量根据自己的需求来加，可以是10个；



## 002图层数据入SDE库

为了便于后继的操作（读写）数据，为了可以直接从oracle中操作和统计，所以需要将图层数据存入SDE库中；

存入KUNSHAN SDE中，前提需要：

新建好KUNSHAN的表空间；

create tablespace EMS

logging

datafile 'E:\app\Administrator\product\11.2.0\dbhome\_1\database\EMS.dbf'

size 50m

autoextend on

next 50m maxsize unlimited

extent management local;

create user

EMS

identified by

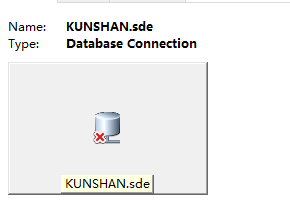
EMS

default tablespace EMS

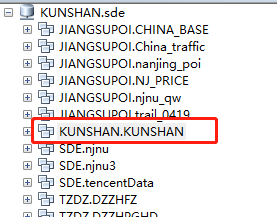
temporary tablespace TEMP;

grant connect,resource,dba to EMS;

初始化企业级数据库；

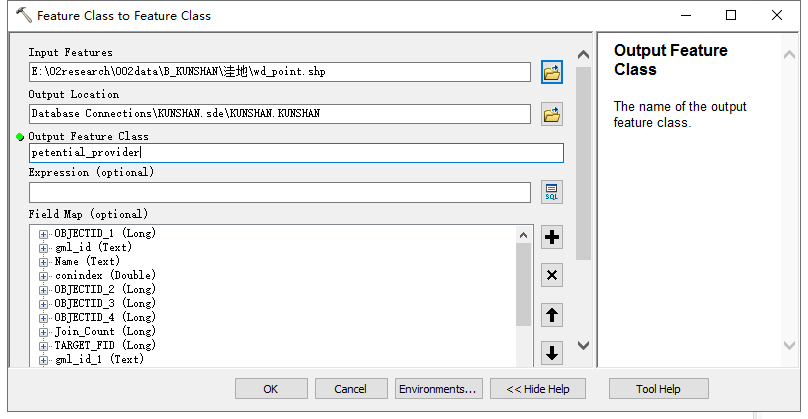


构建数据集；

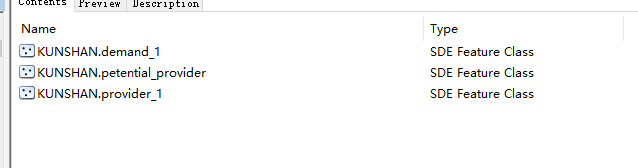


无需注意设置版本（设置了版本容易出现Oracle查到的数据为空的情况，）；

导入图层数据；



得到结果：



## 003初始化数据至MongoDB中

注意：provider的Key\_ID和petential\_provider的Key\_ID不可以有重复，所以需要加一个处理：

update petential\_provider t set t.Key\_Id=t.Key\_Id+1000;

select t.Key\_id from petential\_provider t;

注意：统计一下三张表的点的数量：

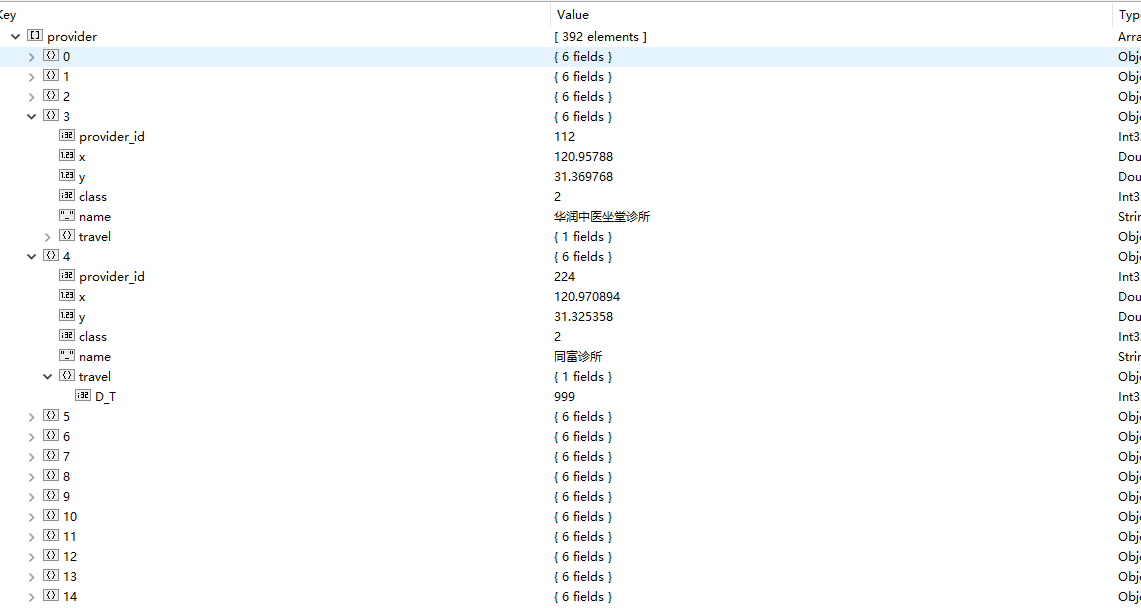
select count(0) from demand\_1; ---386

select count(0) from provider\_1; ---286

select count(0) from petential\_provider; ---106

注意，运行initial\_provider.py文件的时候需要打断点条件；

查看MongoDB中的collection结果

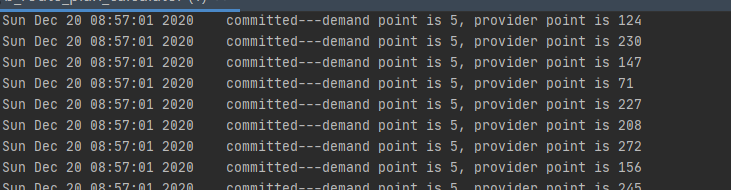


## 004运行RPD计算

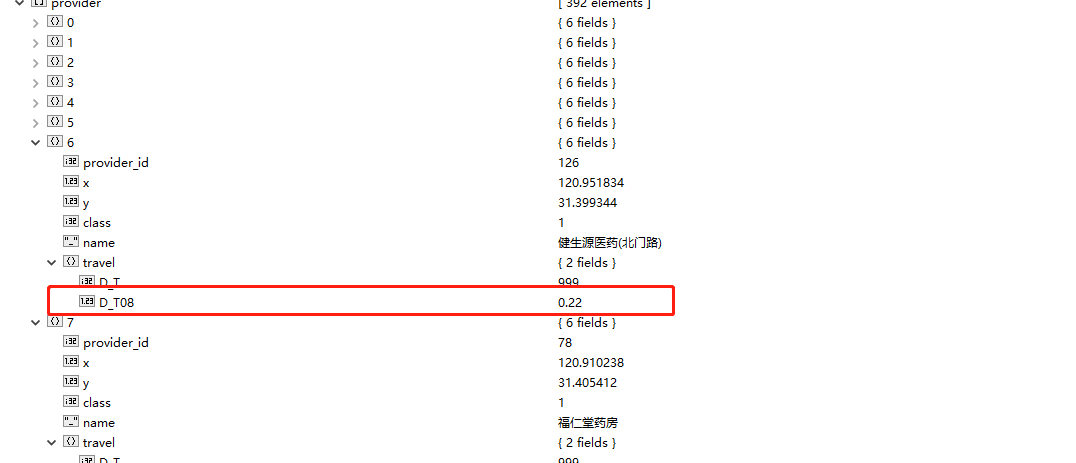
修改：b\_route\_plan\_calculator.py文件中的一些参数

包括：collection\_name和n\_rows

执行文件



查看运行时间后的结果：

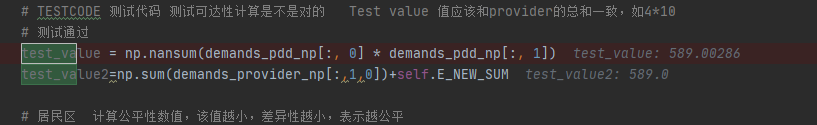


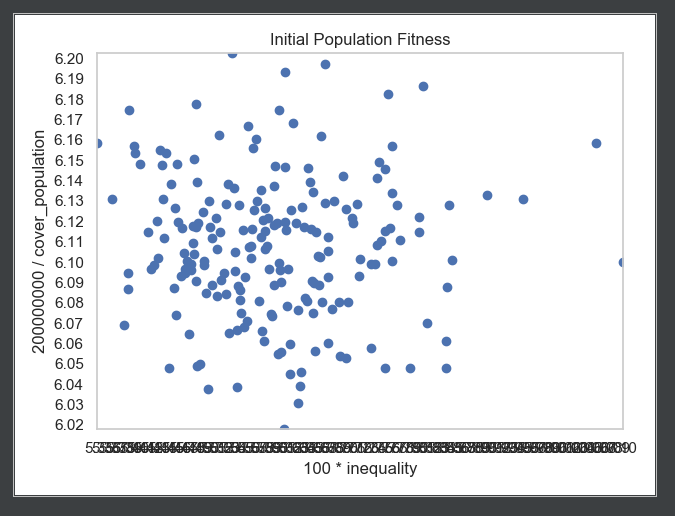
## 007运行MoPs代码

需要修改的地方主要包括2个地方：

一个是初始化的参量，一个是过程的参量（主要包括：fitness函数，可视化函数，正确性验证函数）；

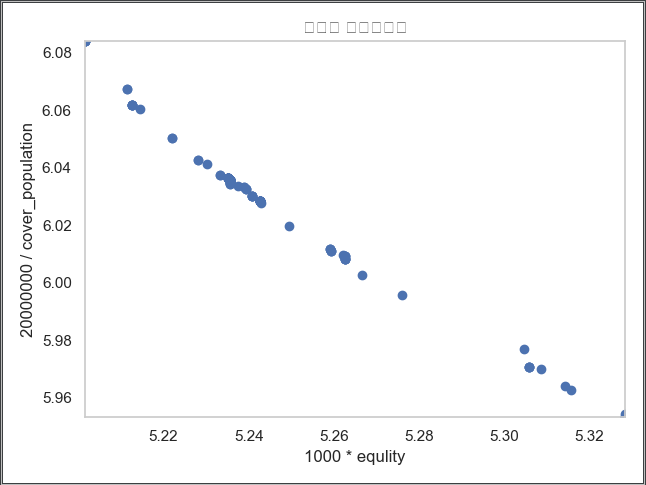
正确性验证函数：





可以设置多组运行的结果，得到一个大的列表；

主要影响的参数包括：初始化参量；



## 008将非支配解存入到petential\_provider图层上

首先需要重新填入provider\_id list，注意需要是petential provider的；

'update petential\_provider set solution1=0.0,solution2=0.0,solution3=0.0,solution4=0.0,solution5=0.0,solution6=0.0,solution7=0.0,solution8=3.0,solution9=0.0,solution10=0.0 where KEY\_ID=1096'