

主要程序清单和相应程序简要说明

毕设的程序采用:

- **Python 3.5**
- **Ubuntu 16.04**
- **PyCharm 2017.3**
- **Jupyter Notebook**

第一个层次为网络的整体的架构，按树形结构表示如下：

```
program
├── README.md
├── setup.py
├── examples
└── udntools
```

文件类型和文件说明如下表所示：

文件名	文件类型	文件说明
README.md	-	存放介绍毕设内容的一个超链接
setup.py	-	注册编写的包到Python的环境变量列表里
udntools	d	超密集组网的工具包，是程序的主要部分
examples	d	用于存放应用工具包得到的理论和仿真图

整个程序为两级架构：

- "udntools" 软件包为用于实现网络分析基本功能的库文件集合
- "examples" 为应用 "udntools" 工具包得到的理论与方针分析结果

"udntools" 层将在本章的剩余部分进行介绍， "examples" 层将在下一章中介绍。

"udntools" 工具包

"udntools" 工具包用于表达基站、用户、信道和网络的特性，为程序的主要部分，其中的内容的树形结构表示如下：

```
udntools
├── bs
├── channel
├── __init__.py
├── region
├── ue
└── utils
```

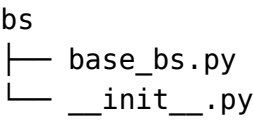
文件名	文件类型	文件说明
bs	d	表示微基站属性的包
ue	d	表示用户的属性的包
channel	d	表示信道的特性的包

文件名	文件类型	文件说明
region	d	网络性能仿真的接口
utils	d	辅助实现仿真的方法

下面对上述的 5 个子包分别进行说明：

"bs" 子包

"bs"子包中的内容的树形结构表示如下：



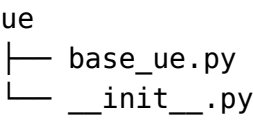
主要的类及类中的对象和方法

- BaseBS
 - Inherit from class **Object**

名称	类型	说明
bs_number_	对象	基站的个数
bs_power_	对象	基站的功率
bs_distribution_	对象	基站的分布
bs_position_	对象	基站的位置
set_bs_to_region	抽象方法	在区域内生成基站
select_ue	抽象方法	寻找基站服务的用户集合

"ue"子包

"ue"子包用于表征用户的特性，其中的内容的树形结构表示如下：



主要的类及类中的对象和方法

- BaseUE
 - Inherit from class **Object**

名称	类型	说明
ue_number_	对象	用户的个数
ue_distribution_	对象	用户的分布
bs_position_	对象	用户的位置
set_ue_to_region	抽象方法	在区域内生成用户

"Channel" 子包

"channel" 子包中的内容的树形结构表示如下：

```
channel
├── __init__.py
├── large_fade_channel.py
└── small_fade_channel.py
```

主要的类及类中的对象和方法

- LargeFadeChannel
 - Inherit from class **Object**

名称	类型	说明
path_loss_factor_	对象	大尺度衰落系数
large_fade_factor_matrix	方法	发射信号经过大尺度衰落后的功率

- SmallFadeChannel
 - Inherit from class **Object**

名称	类型	说明
small_fade_	对象	小尺度衰落的类型
h_matrix_	对象	信道系数
generate_h_matrix	方法	生成小尺度衰落系数矩阵

"Region" 子包

"region" 子包中的内容的树形结构表示如下：

```
region
├── base_region.py
├── comp_service_region.py
├── __init__.py
└── service_region.py
```

主要的类及类中的对象和方法

- BaseRegion
 - Inherit from class **Object**

名称	类型	说明
x_min	对象	区域的左边界
x_max	对象	区域的右边界
y_min	对象	区域的下边界
y_max	对象	区域的上边界
ground_position_	对象	用于存放生成的格点坐标
get_ground	方法	得到格点坐标的矩阵

- ServiceRegion
 - Inherit from class **BaseRegion, BaseBS, BaseUE**

名称	类型	说明
bs_ue_dict_	对象	存放基站对应其所服务的用户的字典
kill_ue	方法	删除区域内的用户
set_bs_to_region	方法	区域内基站部署
set_ue_to_region	方法	区域内基站部署
set_ue_sigma	方法	设定用户的分散程度
set_ue_distribution	方法	设定用户的分布
select_ue	方法	得到bs_ue_dict_

- CompServiceRegion
 - Inherit from class **ServiceRegion, LargeFadeChannel, SmallFadeChannel**

名称	类型	说明
cluster_set_	对象	基站分簇得到的集合
cluster_bs_position_	对象	簇内基站的坐标
cluster_ue_set_	对象	根据基站分簇结果对用户分簇
cluster_ue_position_	对象	簇内用户的坐标
self.sir_array	对象	每个用户的信干比性能
cluster_by_kmeans	方法	基于 Kmeans 的基站分簇算法
cluster_by_dfs	方法	基于深度优先搜索的基站分簇算法
get_cluster_ue_position	方法	得到 cluster_ue_set_ 和 cluster_ue_position_
zbf_equal_allocation	方法	基于 ZFBF 的多用户联合传输算法
sir_array_sim	方法	得到用户的信干比的性能

"utils" 子包

"utils" 子包中的内容的树形结构表示如下：

```
utils
├── ase_theory.py
├── cdf.py
├── dfs_dict_by_distance.py
├── dim2_distance.py
├── __init__.py
└── pc_theory.py
```

文件名	文件类型	说明
ase_theory.py	-	用于得到单位面积谱效率的理论值
cdf.py	-	用于画概率累计分布函数图
dfs_dict_by_distance.py	-	实现深度优先搜索算法

文件名	文件类型	说明
dim2_distance.py	-	计算基站和用户之间的欧式距离
pc_theory.py	-	用于得到区域覆盖率的理论值

主要程序的流程图或实现框图

本章介绍主要程序的流程框图或实现框图，即软件的 "examples" 层的部分， examples 层的主要的模块为：

```
examples
├─ ase_sim
├─ capacity_map
├─ clustering_precoding
├─ dfs_clustering
├─ kmeans_clustering
└─ pc_sim
```

其主要的模块如下表所示：

模块名	说明
ase_sim	单位面积频谱效率的理论仿真
capacity_map	用于得到基站的拓扑结构和网络的遍历容量热力分布图
clustering_precoding	密集热点区域无线网络优化后的覆盖率性能分析
dfs_clustering	基于深度优先搜索的基站分簇算法的算法验证与分析
kmeans_clustering	基于 k - 均值的基站分簇算法的算法验证与分析
pc_sim	用于对密集热点区域无线网络的覆盖率的性能的分析

下面按以论文为线索依次介绍上述的子模块：

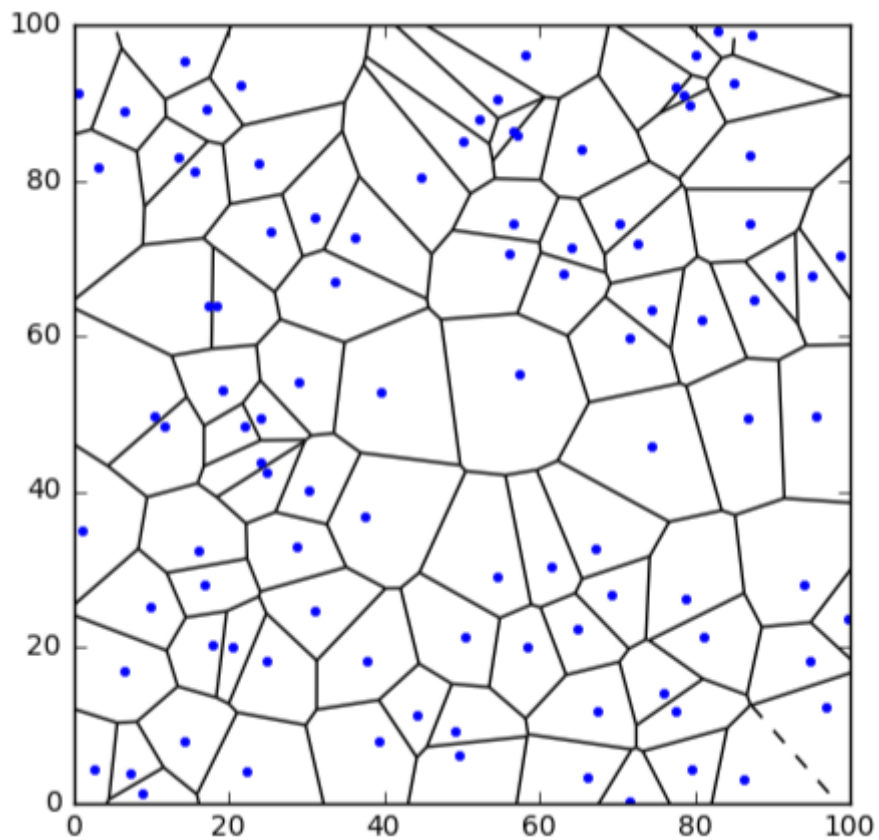
对基站的网络拓扑结构的仿真：

- 所属子模块：
 - examples/capacity_map
- 对应论文中的 图 3-3

仿真的步骤为：

- **Step 0**：设置网络的服务区域的边界，设置基站的数量
- **Step 1**：生成服从泊松点过程的坐标
- **Step 2**：根据生成的坐标画出基站的网络拓扑图，并画出反应每个微基站服务范围的 Voronoi 图

得到的仿真结果图如下图所示：



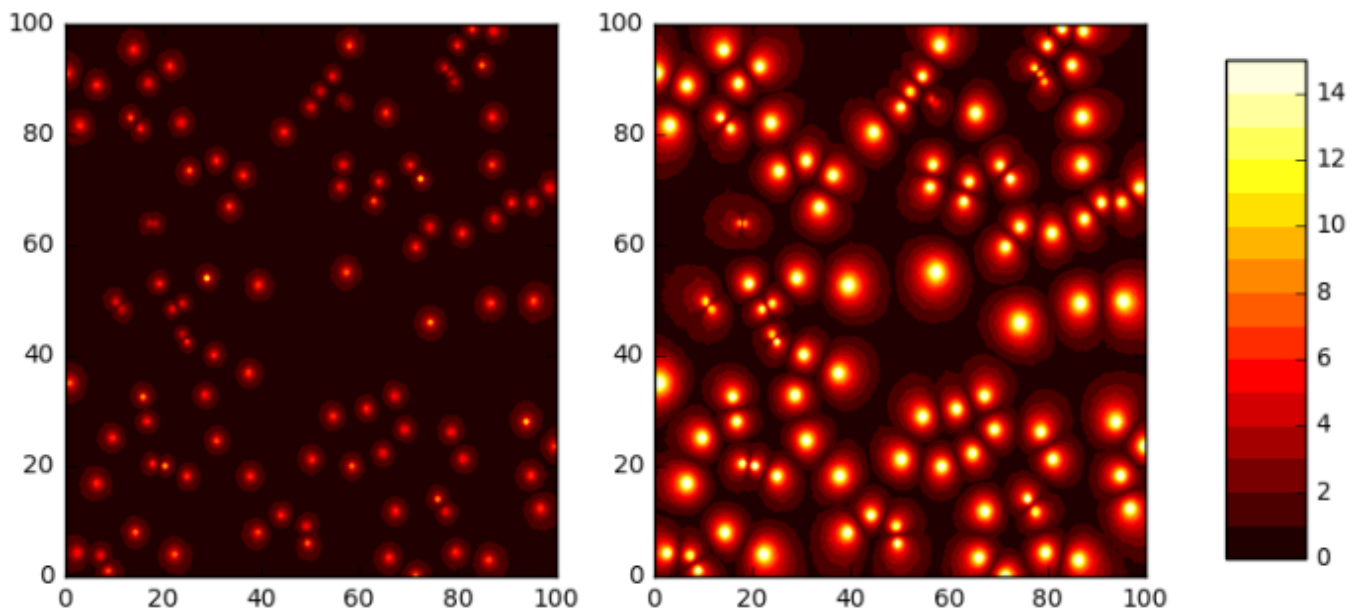
对网络中各个位置的遍历容量的热力分布图的仿真图：

- 所属子模块：
 - examples/capacity_map
- 对应论文中的 图 3-4

仿真的步骤为：

- **Step 0**：设置基站拓扑结构，大尺度衰落系数分别设置为2.0和4.0
- **Step 1**：对 ground_position_ 数据结构中的所有坐标求遍历容量，信道是平稳的可以用时间遍历的方法求均值
- **Step 2**：画出基站的热力分布图

得到的仿真结果图如下图所示：



对网络的覆盖率的性能的理论分析和仿真：

- 所属子模块：
 - examples/pc_sim
- 对应论文中的 图 3-5，图 3-6，图 3-7

根据论文中的分析可以得到，在密集热点区域无线网络的场景中，网络的覆盖率上界的表达式，该表达式也为覆盖率的近似式，如下式所示：

$$P_c(T, \lambda, \alpha, \sigma) = \frac{1}{1 + \rho(T, \alpha)} + \frac{\rho(T, \alpha)}{1 + \rho(T, \alpha)} \cdot \frac{1}{2\pi\sigma^2\lambda(1 + \rho(T, \alpha)) + 1} \quad (1)$$

其中：

$$\rho(T, \alpha) = T^{2/\alpha} \int_{T^{-2/\alpha}}^{\infty} \frac{1}{1 + u^{\alpha/2}} du \quad (2)$$

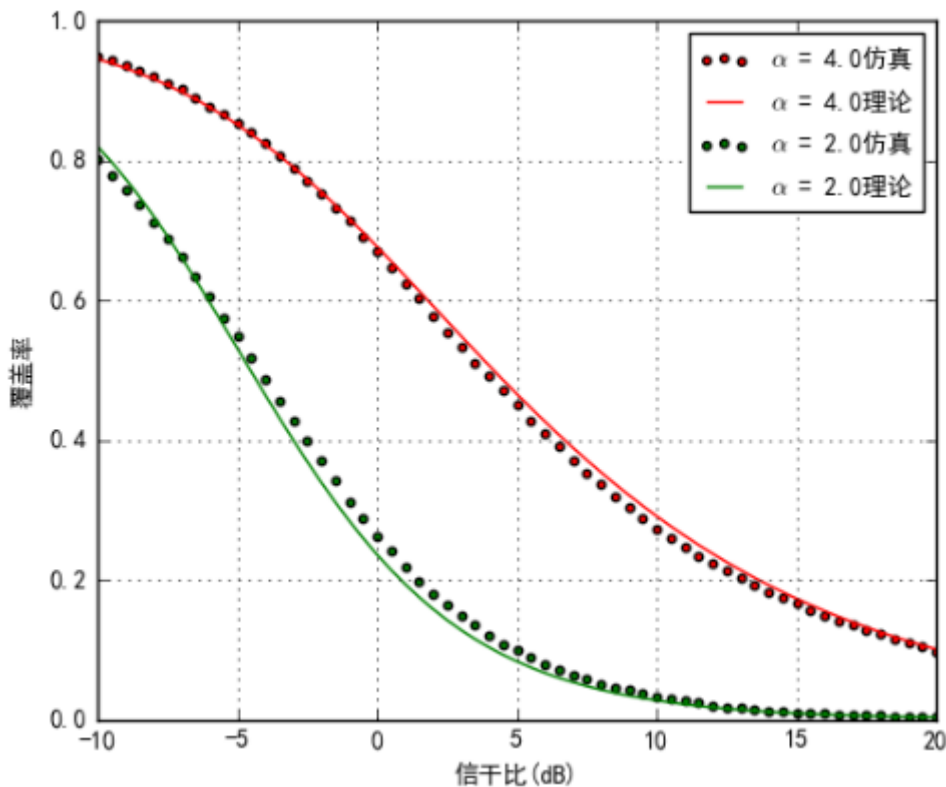
P_c 为覆盖率， T 为信干比门限， λ 为微基站的密度， α 为信道的衰落系数， σ 为用户的分散程度。

给出上述表达式的数值解，并和经过蒙特卡洛仿真的到的结果进行对比。

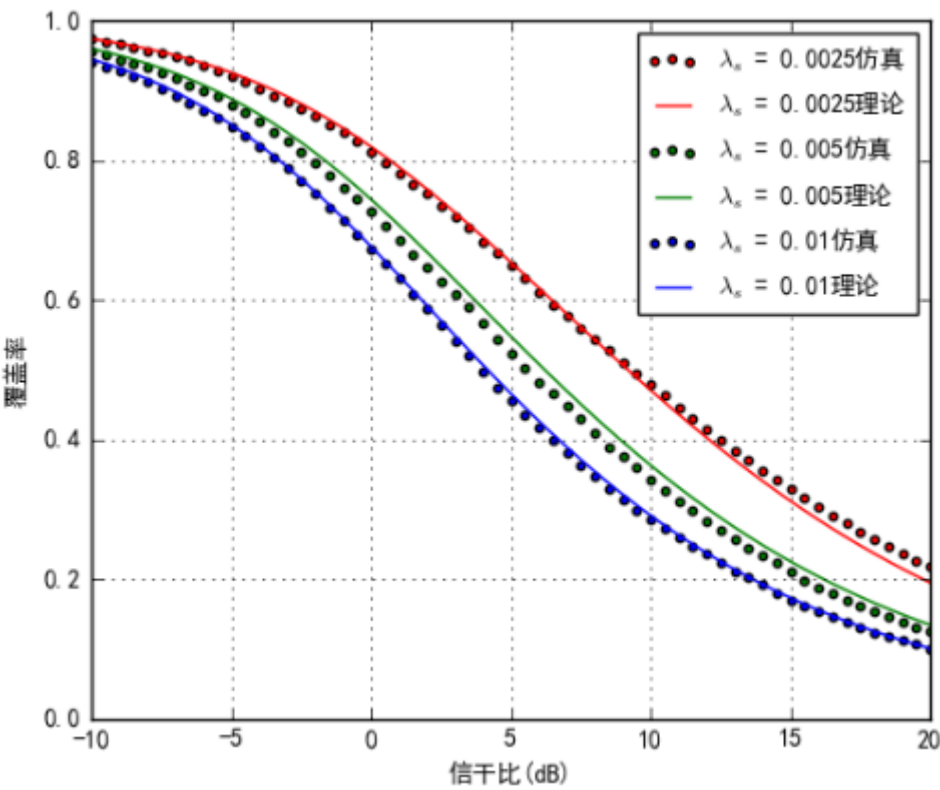
仿真的步骤为：

- **Step 0**：采用控制变量法，讨论信道衰落系数（图 3-5），微基站的密度（图 3-6），用户分散程度（图 3-7）对覆盖率的影响，基站的网络拓扑为泊松点过程，用户的分布为二维高斯分布。
- **Step 1**：根据给定的参数对覆盖率性能进行仿真
- **Step 2**：根据给定的参数求覆盖率的数值解
- **Step 3**：画出曲线对比分析

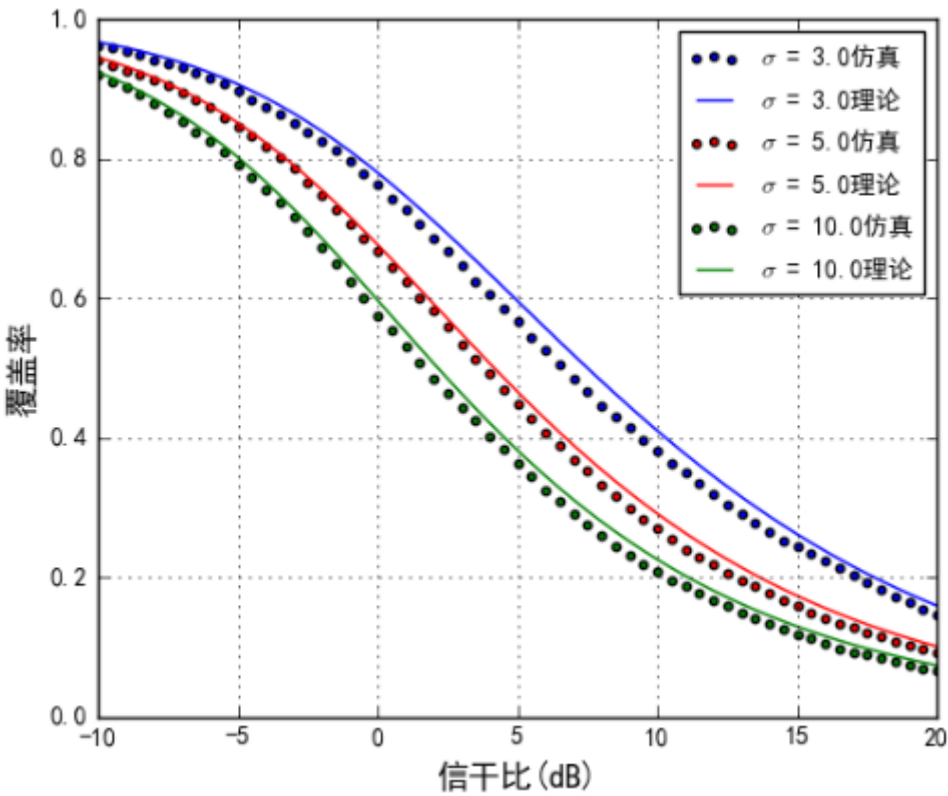
覆盖率关于信道衰落系数的曲线图如下：



覆盖率关于微基站密度的曲线图如下：



覆盖率关于用户分散程度的曲线图如下：



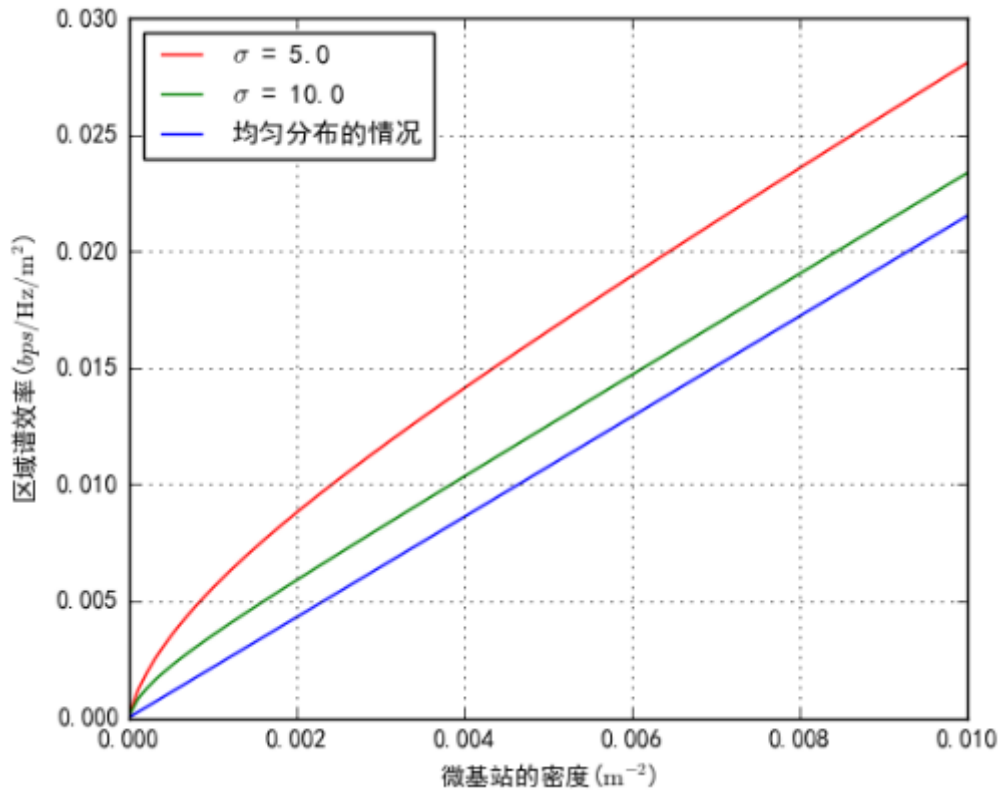
对网络的单位面积谱效率的性能的理论分析：

- 所属子模块：
 - examples/ase_sim
- 对应论文中的 图 3-8

根据论文中的分析可以得到区域内单位面积谱效率的表达式如下式所示：

$$\eta_{ASE} = \lambda_s \int_{t>0} \frac{1}{1 + \rho(2^t - 1, \alpha)} + \frac{\rho(2^t - 1, \alpha)}{1 + \rho(2^t - 1, \alpha)} \cdot \frac{1}{2\pi\sigma^2\lambda(1 + \rho(2^t - 1, \alpha)) + 1} dt \quad (3)$$

对其进行数值分析可得到单位面积频谱效率与密度的关系如下图所示：



对微基站分簇的仿真示意图：

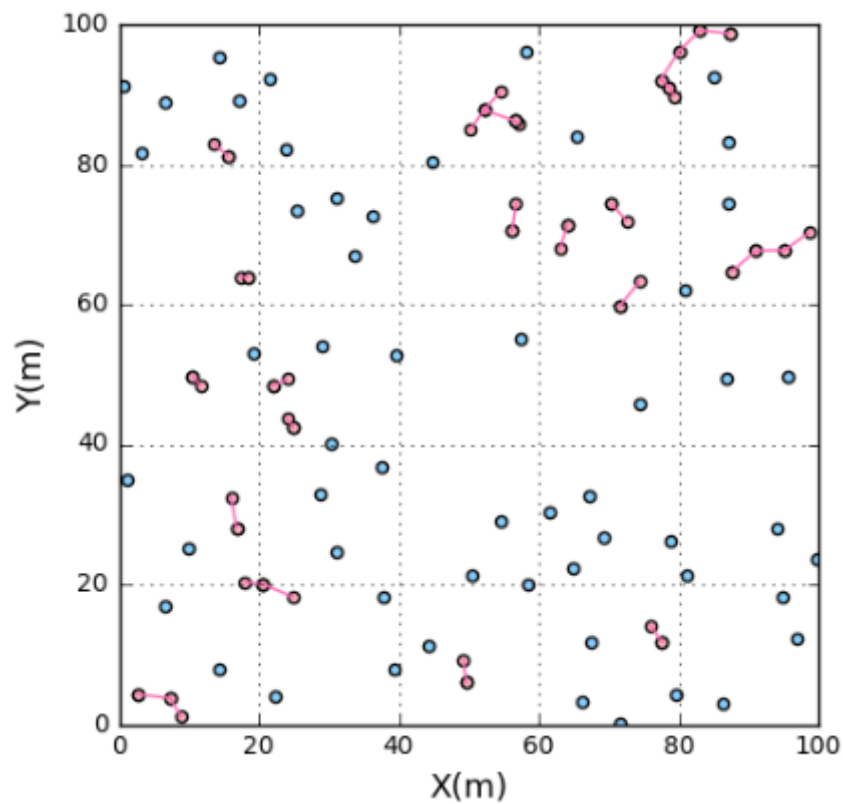
基于深度优先搜索的基站分簇算法结果仿真图：

- 所属子模块：
 - examples/dfs_clustering
- 对应论文中的 图 4-3

仿真的步骤为：

- **Step 0**：输入基站拓扑结构，输入距离门限
- **Step 1**：使用深度优先搜索算法对基站进行分簇
- **Step 2**：画出分簇的仿真结果图

得到的结果图如下图所示：



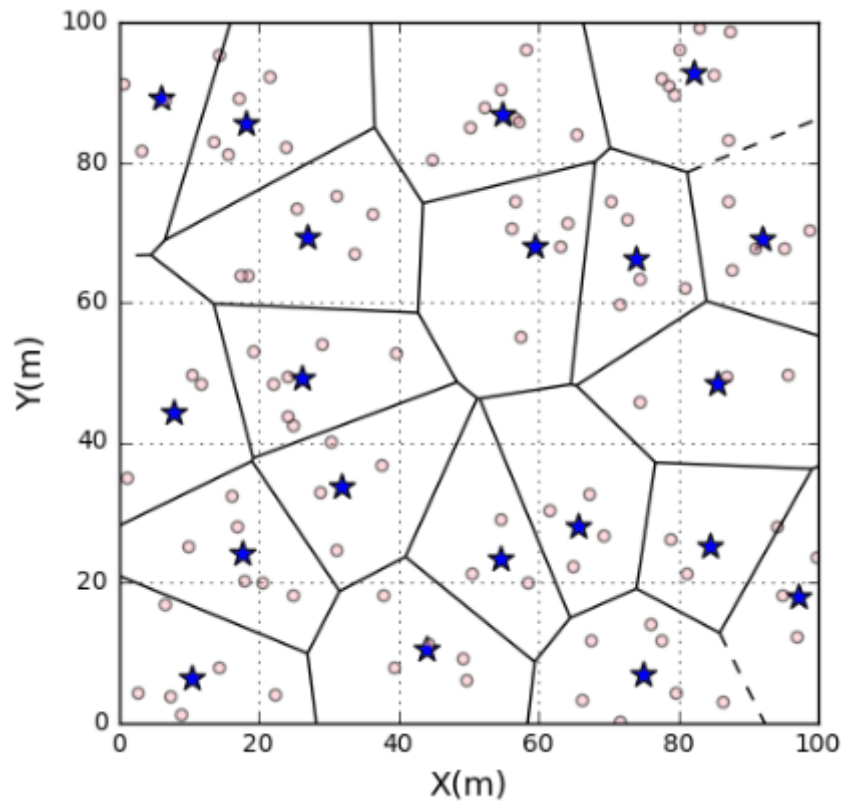
基于 k - 均值的基站分簇算法结果仿真图：

- 所属子模块：
 - examples/dfs_clustering
- 对应论文中的 图 4-4

仿真的步骤为：

- **Step 0**：输入基站拓扑结构，均值个数
- **Step 1**：使用 k 均值算法对基站进行分簇
- **Step 2**：画出分簇的仿真结果图

得到的结果图如下图所示：



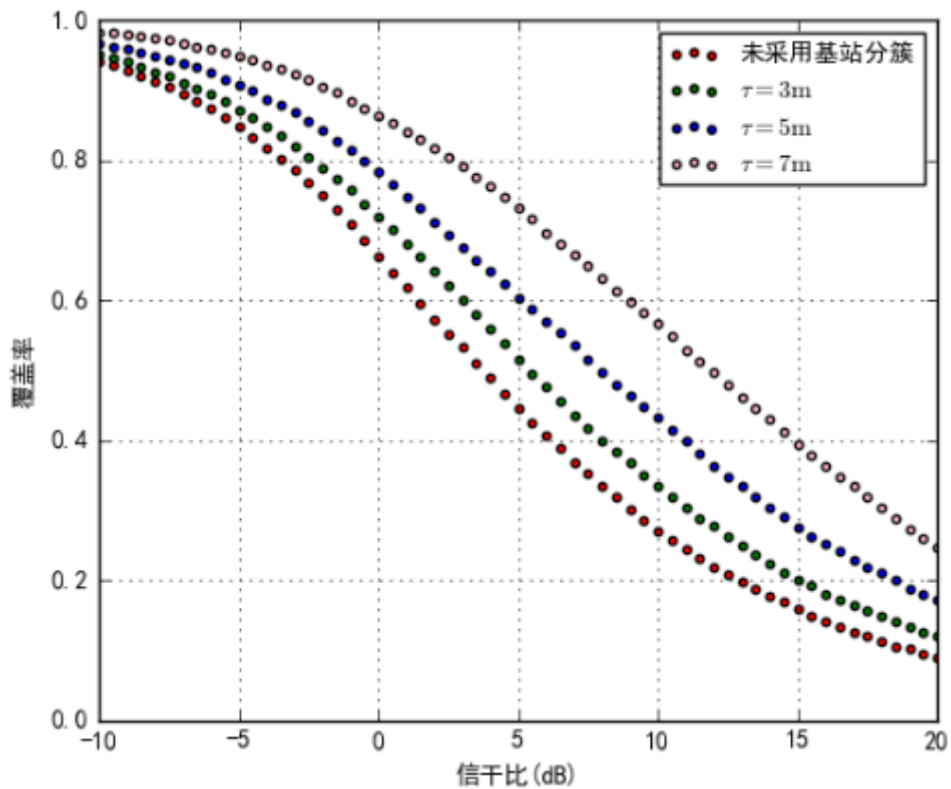
对区域内网络性能优化的仿真曲线图：

- 所属子模块：
 - examples/clustering_precoding
- 对应论文中的 图 4-5，图 4-6

仿真步骤为：

- **Step 0**：输入基站拓扑结构，输入距离门限
- **Step 1**：对基站使用深度优先搜索算法（图 4-5），k - 均值算法（图 4-6）进行分簇
- **Step 2**：采用 ZFBF 技术对网络的性能进行优化
- **Step 3**：得到优化后的结果

使用深度优先搜索算法对基站分簇，再进行性能优化的结果为：



使用 k 均值算法对基站分簇，再进行性能优化得到的结果为：

