

# 喷淋布线思路文档

## 1. 数据准备

### 1.1 数据结构

#### 1.1.1 输入

##### 墙体信息

主要包含建筑墙和剪力墙两种，输入形式都为 polyline 形式框线的List

建筑墙，其中可能包含门窗防火卷帘等信息（处理方式和建筑墙相同），属于可穿障碍物（支管不可穿）

```
1 List<polyline> walls;
```

剪力墙，属于不可穿障碍物

```
1 List<polyline> shearWalls;
```

##### 喷头点位

point3d格式的List，包含喷头的坐标信息

```
1 List<point3d> sprayPoints;
```

##### 起点（含方向），终点

起点后第一段管线方向固定

```
1 Point3d startPoint;  
2 Point3d endPoint;  
3 Direction startDirection;
```

##### 车位排（含支管方向）

polyline 形式框线List，其中0-1点位对应车道相邻框线，顺序随机。

需要用户输入指定支管相对于车道的方向（平行/垂直）

```
1 List<polyline> parkingLots;
2 bool isParallel2Lane;
```

## 分组过后的UCS信息

按图的形式输入，点位关系使用邻接表表示，点位编号随机。

```
1 list<graph> ucsPoints;
```

### 1.1.2 喷淋点位

根据输入的点位坐标信息和UCS分组情况重新构建喷淋点位数据结构，以网状的有向图结构存储（相当于同一组内每一行和每一列都是双向链表结构），便于布线的时候访问相邻点位。

每个喷淋点位内需要存储的信息包括：点位坐标、UCS角度（保证在0-90之间）、所在场景类别、支管排布方向（-1：未分配支管，0：水平，1：垂直）

```
1 public class sprayPoint{
2     Dictionary<string, int> neighborMap = new Dictionary<string, int>{
3         {'up', 0}, {'right', 1}, {'down', 2}, {'left', 3}
4     };
5     public coordinatePos pos; //(x,y)
6     public double ucsAngle; //0-90
7     public Scenes scene;
8     public int branchDir; //-1:undefined, 0:0-90, 1:90-180
9     public bool assigned; //0:unwatered, 1:watered
10    public sprayPoint? upNeighbor;
11        public sprayPoint? rightNeighbor;
12        public sprayPoint? leftNeighbor;
13        public sprayPoint? downNeighbor;
14
15    //init
16    public sprayPoint(double posx, double posy){
17        this.pos.posx = posx;
18        this.scene = Scenes.0thers;
19        this.branchDir = -1;
20        this.neighbors = new sprayPoint[4];
21    }
22    public setScene(Scenes scene);
```

```
23     public setUcsAngle(double ucsAngle);
24     public setNeighbor(int idx, sprayPoint neighbor); //string dir
25 }
```

### 1.1.3 管道信息

主要是支干管，使用 polyline 存储，设置类变量记录是否从起点连通。

## 1.2 场景划分

首先，根据整个建筑的外围墙线和防火分区取交并，获得当前布线区域范围及其内部喷淋、墙线等数据，然后在当前区域执行场景划分。场景划分根据布线需求决定，并非完全按照人为指定分类。将场景分为车位排、塔楼（剪力墙密集区域）、小房间、较宽走廊（包含宽度大于8000的坡道和具有明显方向性的走廊区域）、狭窄走廊（包含宽度小于8000的坡道）和其他大空间几个部分，按如下顺序进行场景划分和标注。

```
1  enum Scenes{
2      Parking, SmallRoom, Tower, Corridor, NarrowCorridor, Others
3  }
```

### 1.2.1 车位排

根据 polyline 输入，获得车位排内部的点位并在点位上标注。可以在这一步根据用户输入的支管和车道方向关系，同时完成内部点位上支管方向的标注，以减少后续遍历步骤。

### 1.2.2 小房间

直观使用“面积较小”的特性选取防火区内的小房间，并在内部点位上标注。

划分完小房间后，将小房间对应的 Polyline 存储为一个 List，作为之后大空间布线时的“不可穿越墙体”使用。

### 1.2.3 塔楼

这里用“塔楼”来表示广义上剪力墙密集的所有区域。

先将“塔楼区域”概念离散化为“周围有很多剪力墙的喷淋点位”，根据单个点位周围固定半径圆内剪力墙的数量标注位于塔楼内部的喷淋点，然后构建包含所有塔楼点位的凸包，作为塔楼区域的划分。

### 1.2.4 走廊和坡道

依然是广义的“走廊”，包含坡道、大空间周围具有独立UCS的走廊、小房间门口的过道、需求文档中“大面积狭窄区域”中的非交叉部分等等。

对单个点位，取横纵两个方向的链表分别计数。设定走廊截面点位数量阈值（例如设为3），如果单个点位两个方向的链表中点位数量分别大于和小于设定的阈值，将该点位作为走廊处理，并取短边作为支管方向存储入点位信息。过当前点位沿短边方向做 line 并与近邻墙体取交线，计算交线之间的长度作为走廊的宽度，与8000比较。

做完这一步后，预期剩下的部分主要包括空旷且特征不明显的大空间、正交方向走廊的相交处和有弧度（UCS一直在变化）的坡道处，这些部分统一认为属于 Scenes.Others 。

TODO：该步默认除坡道以外的走廊都宽于8000，需要在实际实现中验证。

```
1  int xcnt = 1, ycnt = 1, flag = -1;
2  while(curPoint.rightNeighbor) xcnt++;
3  while(curPoint.leftNeighbor) xcnt++;
4  while(curPoint.upNeighbor) ycnt++;
5  while(curPoint.downNeighbor) ycnt++;
6  if(xcnt > threshold && ycnt <= threshold){
7      flag = 0;
8      curPoint.branchDir = 0;
9  }else if(xcnt <= threshold && ycnt > threshold){
10     flag = 1;
11     curPoint.branchDir = 1;
12 }
13 if(flag >= 0){
14     //延长取交线，计算宽度
15     curPoint.scene = Scenes.Corridor / NarrowCorridor;
16 }
```

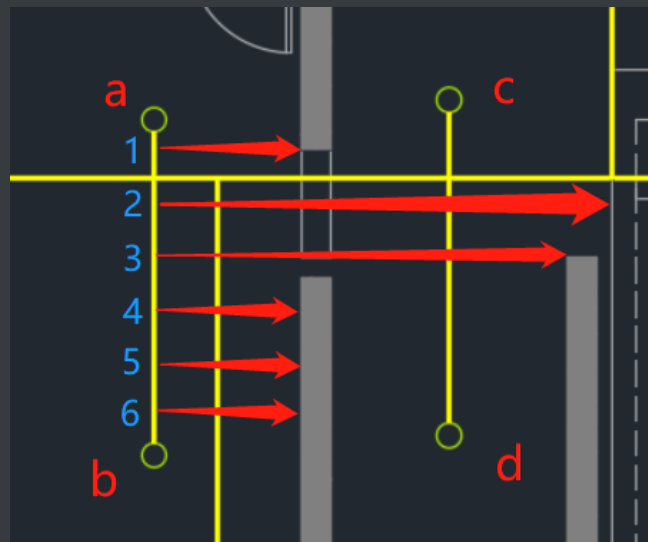
## 2. 布线思路

### 2.1 整体思路

整体的布线思路类似于蒙特卡洛树搜索（MCTS）算法，通过随机+推演逐渐建立搜索树，每个节点内存储访问次数和得分总和，每一次推演完成之后将得分back propagate至所有前向节点，在后续访问时根据节点的UCB值决定访问的节点。

本项目中需要决策的内容包含“管线交叉点在哪两个喷淋之间”以及“具体在什么位置引出管线”两个方面，为了表述清晰，把前者称为“点的位置”，后者称为“线的位置”。考虑到点的位置离散而线的位置连续，并且在点的位置确定之后，线的位置通常只有固定的若干小段可以选择（段内的具体位置通常影响不大），因此将点的位置作为MCTS的搜索节点，在每个节点内置线的位置作为最后的线路规划位置。

例如下图，若算法当前迭代决策在点位ab之间向cd之间布支干管，在该节点访问次数为0（即执行第一次访问）时，考虑支干管和喷头的最小间距之后，在点位之间离散5-10个布线位置做局部的快速搜索。在图示例子中，1456号布线方案很快会遇到障碍，3号在较远处会遇到障碍，而2号相对畅通无阻。因此在ab的决策节点上将13的中点作为线的位置存储。若有多个位置可供选择（例如如果bd之间的墙体不连续），可以考虑将多个位置以list的方式存入节点供随机挑选。



## 2.2 特殊场景布线

在开始构建MCT时，首先根据划分的场景规划特殊位置的线路大致走向，包括车位排、塔楼、小房间、走廊和坡道。

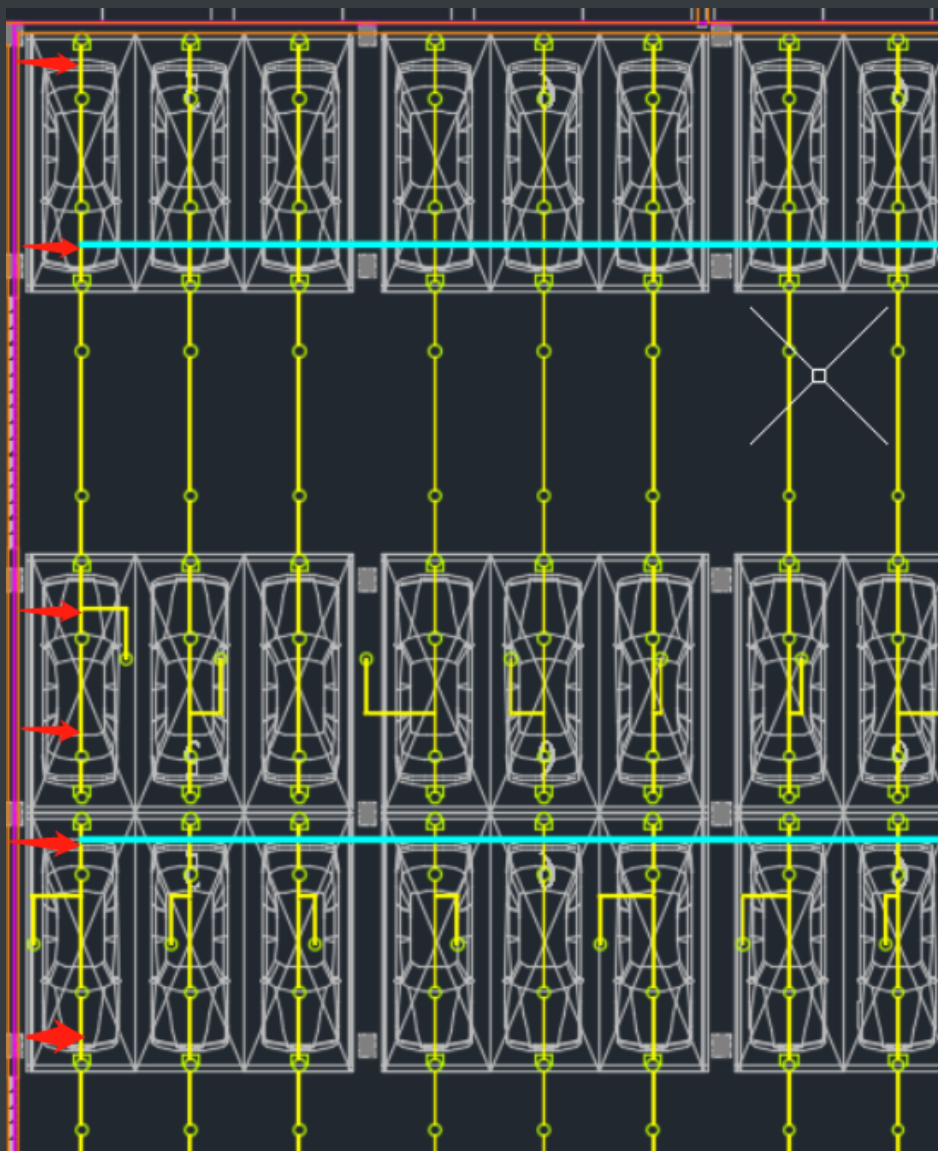
### 2.1.1 车位排布线

车位排布线模式比较单一，选择最先完成。根据用户输入的方向信息（水平或垂直，就目前的样图来看似乎垂直居多）选择布线方案。

#### 垂直

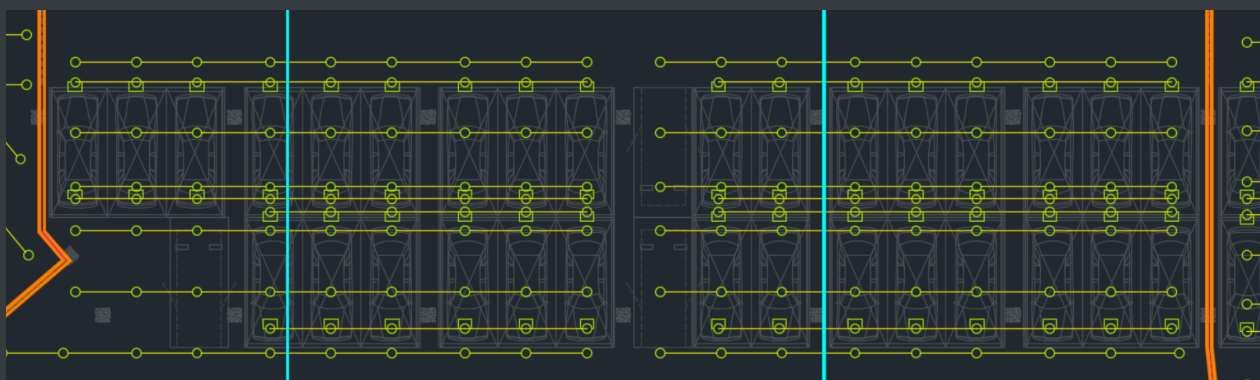
根据车位排输入0-3之间的距离判断属于双停车位或者单停车位。可以认为车位排上支干管的位置在单停车位的（0-1/4, 3/4-1）位置，在双停车位的（0-1/8, 3/8-5/8, 7/8-1）位置，与喷淋点位取交集获得离散的可布线位置。

考虑同方向的车位排及中间车道布线统一规划，为了均匀起见，在初次布线时优先考虑在每一个车位排的同一对应位置分布支干管，例如单停车位的3/4-1位置对应双停车位的7/8-1位置。若有车位排靠近墙体，则优先考虑在远端布线。初次布线完成后根据是否满足在两停车位之间车道的供水、相邻支干管之间喷淋点位数的极差等，进行简单的遍历调整，选择得分较高的方案（可能设定分数阈值）作为MCT的节点。



## 水平

计算一条车道上喷淋点的数量，优先八个一组进行划分，若结束处极差过大则考虑减少单根支管所带喷头数（从8开始递减）直至极差低于设定的阈值，按分组结果垂直车道布支干管。



## 2.1.2 小房间布线

对小房间区域取矩形外包框。若外包框狭长且内部点位较多，或者小房间都直接和外部相连，考虑沿外包矩形靠近大空间的长边布支干管（若有两边则可以作为MCT节点）。若外包框没有明显狭长形态，则跳过该步，在完成大空间布线之后再进行考虑（TODO）。

### 2.1.3 塔楼布线

塔楼部分的布线相对外部独立，目的是在塔楼区域内找到一根穿墙和弯折都比较少，且覆盖所有点位的折线。可以设计成一棵独立且状态更少的MCT，对于狭长塔楼，折线方向优先沿长边；对于凸包面积较小的塔楼，布线方案类似小房间处理，存储凸包外围作为后续布线时不可穿越墙体，在完成大空间布线之后进行考虑。

### 2.1.5 走廊和坡道布线

狭窄走廊随机沿一边（考虑与墙柱的距离）垂直支管布线，较宽走廊优先沿中线。

## 2.3 其余部分布线

### 2.3.1 连接处布线

首先处理坡道UCS变化处和走廊连接处的布线。直观的做法是取近邻的支干管在当前UCS内做交点，根据交点情况（如十字交叉处）需要对近邻支干管“线的位置”进行微调。

### 2.3.2 零散点位覆盖

在前序布线过程中，每布完一根支干管之后，将确定由其供水的喷淋点位的 `assigned` 值设为1，表示完成供水。特殊场景布线结束之后，对于未完成供水的点位进行小范围的遍历，如果周边有支干管则可以认为能被供水，可以将 `assigned` 设为1。该步骤用于去除一些没有被供水的零散点位。

至此，可以认为剩下的未供水点位几乎都位于空旷大空间。

### 2.3.3 起点连线

考虑从起点向多段平行管线做垂线（干管）。垂足位置不一定在起点（可以是起点方向延长线上的离散位置，同样可以作为MCT节点）

在初始连线时，为了使出入口连线尽可能最长，假设入口位置是 $(x1, y1)$ ，出口是 $(x2, y2)$ ， $|x1 - x2| > |y1 - y2|$ ，则初始优先沿 $y1 - y2$ 方向寻找管线并作垂直。可以认为MCTS正式开始工作。从起点开始沿给定的方向，根据上述做法依次将可以进行转折的点作为MCT的节点，在访问过程中，记录转折的数量、连通管道的数量、穿墙数量等，MCTS的终点是当前所有支干管都被连通。

在这里需要加入后处理，后处理的任务包括：

1. 小范围遍历未供水的点位，同零散点位覆盖步骤。如果剩余未供水点位较多，认为是存在空旷大空间，i.e. MCTS需要继续执行。在空旷大空间内选择支管方向（可以根据矩形外包框形态选择窄边，或者作为MCT节点），同车位排水平的方式进行分组和支干管排布，从起点或邻近支干管连接供水。
2. 环路检测，发现环路之后进行打断



### 3. 连接出口，保证出入口连线最长

完成后处理之后，根据弯折、穿墙数量等指标对相应布线方案进行打分，将打分结果back propagate至所有前序节点，然后执行下一次搜索。

## 3. TODO

### 3.1 未解决问题

1. 如何保证出入口连线是全图最长管线，是放在最后调整还是每一次有新的管线连接都相应执行一次调整，具体调整算法
2. 算法不能保证连线是树的形态，MCTS结束后可能会存在环路，后处理部分在发现环路之后如何决定一个最佳的断开环路的位置，还是说干脆把环路这个约束也用来打分
3. 具体的打分机制如何设计（主要是弯折、穿墙数量和支干管数量）
4. 是否需要调整MCTS在分叉处的选择方案，例如不采用UCB的比较，而是设置随机权重，可能可以照顾到一些被不小心淘汰掉的优秀布线方案，并且可能可以在权重处减少弯头，即给拐点更远的位置分配更高的权重。如果是这样，要怎么定义初始的权重，怎么设置后续的更新方案。
5. 起点处的弯折和通路（因为起点的管线方向固定）
6. 小房间和塔楼的具体布线方案
7. 整个算法的时间复杂度不确定，复杂图可能会因为搜索状态过多出现复杂结果

### 3.2 待验证的约束

1. 干管和支干管清晰区分 i.e. 干管不能连出支管，干管与支干管少共用
2. 转角的角度，尤其是不同UCS相接处
3. 主线就近连接，虽然尽量让起点连接各个方向的管道，但仍然有非常大的可能性出现绕路情况
4. 小于8000的坡道，支管不能跨梁
5. 支干管分布均匀（包括车位排复杂处的切分方案等）