

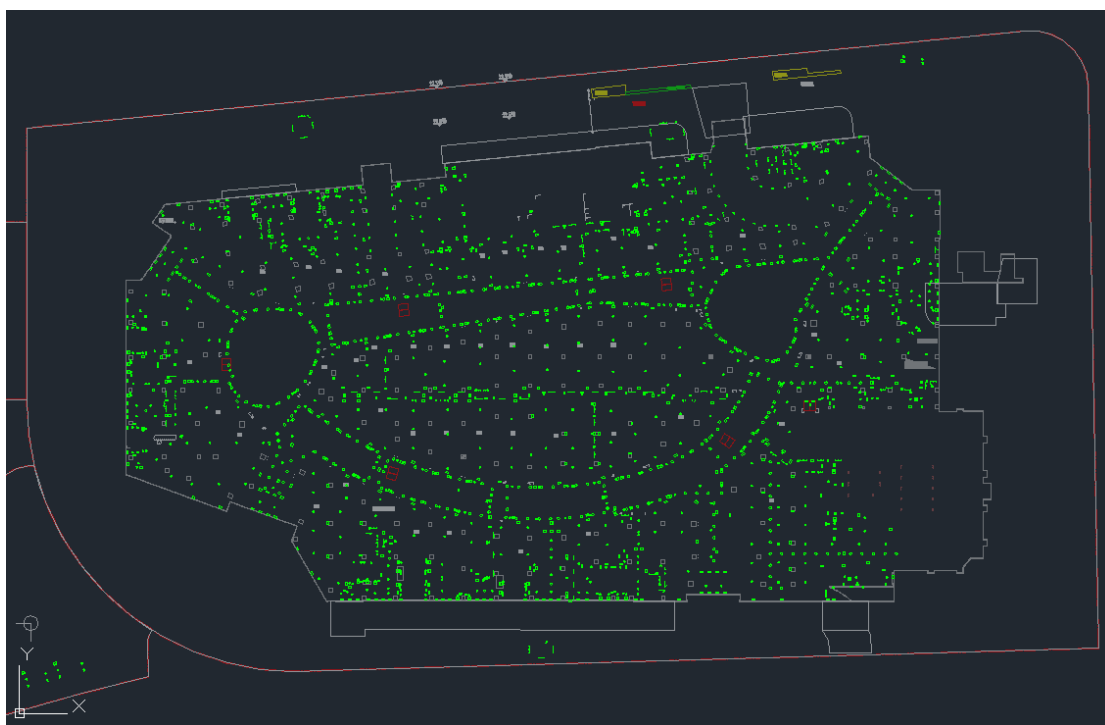
火灾报警项目管线连接部分开发文档

1 概述

2 需求

2.1 输入

- 1) 建筑+结构平面图（墙、门、窗、洞、防火卷帘、防火分区、楼梯、房间名称、柱、剪力墙）；
- 2) 设备布置点位；
- 3) 设备分组的最大点位数量和最小点位数量；
- 4) 电源位置（起点位置）。



2.2 输出

起点到设备管线路径以及设备间管线路径



2.3 处理要求

设备在连线前需先确定分组，然后按组连接。分组信息可以由用户输入，否则需要算法实现设备的自动分组。

2.3.1 分组要求

- 1) 任意一组设备的数量在给定的范围内；
- 2) 分组时不能仅考虑直线距离上的相近，需考虑设备之间是否存在不可逾越的障碍物；
- 3) 各分组的设备数量尽量相同；

2.3.2 连线要求

- 1) (约束) 当同时存在公共空间和私有空间时, 用于连接公共空间设备点位的管线不能穿越私有空间, 只有用于连接该私有空间内设备点位的管线可以进入该私有空间;
- 2) (优化) 线路中最长的路径长度 (起点至任意点位的最短线路长度的最大值) 应尽可能小, 且不应超过限定值;
- 3) (约束) 同一组设备应由给定起点开始相互连接;
- 4) (约束) 自第一个点位开始可以出现树状分支, 任意设备的分支数量不能超过 4 个;
- 5) (约束) 管线不能穿越洞口、窗、防火卷帘、防火分区;
- 6) (优化) 管线应尽可能沿逐渐远离电源的方向延伸;
- 7) (美观) 管线之间应尽可能减少相互交叉;
- 8) (美观) 管线应横平竖直, 沿水平或竖直方向铺设。

3 模型分析

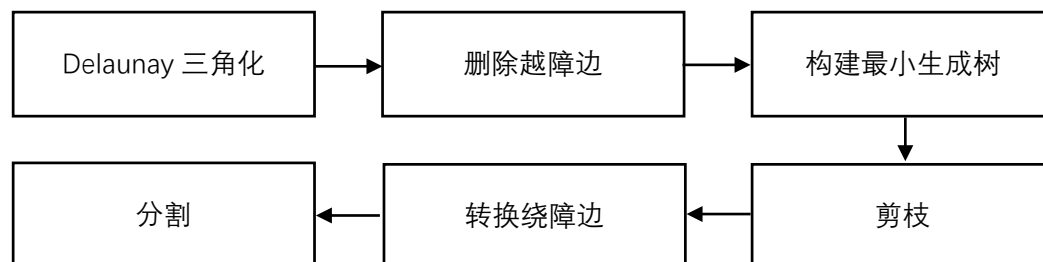
- 1) V_d : 设备布置点位的二维坐标的集合
- 2) O : 不可穿过的结构, 即障碍物的集合, 每个障碍物为一个多边形
- 3) V_o : O 中所有多边形的顶点, 即障碍顶点的集合
- 4) 不可穿区域: 洞口、窗、防火卷帘、防火分区的内部, 以及区域框线的外部
- 5) 每组的最大设备数量 $DMAX$, 最小设备数量 $DMIN$

4 算法思路

4.1 分组

主要思路为: 构建一个考虑绕障距离的最小生成树, 利用贪心策略, 从树中选择一条边, 将树一分为二, 并对分割后的子树递归地继续分割, 直到森林中所有子树均满足分组要求。

流程图如下:



1) Delaunay 三角化

设点集 $V = V_d \cup V_o$, 对 V 进行 Delaunay 三角化, 形成三角网 $G_t = G(V, E_t)$, 其中 E_t 为三角网中的边集

2) 删除越障边

越障边指的是有部分线段落在障碍物内部的边。这一步中删去 E_t 中的越障边。

3) 构建最小生成树

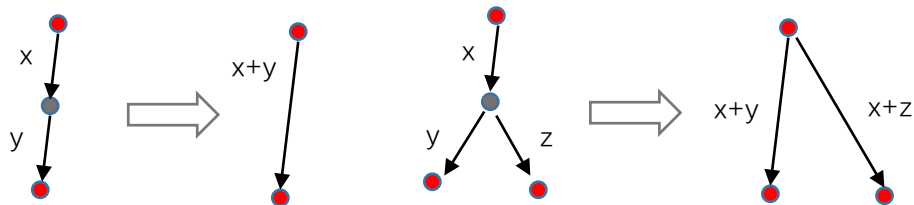
在 G_t 中得到最小生成树 MST

4) 剪枝

不断删去 MST 中为障碍顶点的叶子节点，直到所有叶子节点属于 V_d

5) 转换绕障边

将 MST 中剩余的障碍顶点表达的长度信息转换到新边中，如下图示例：



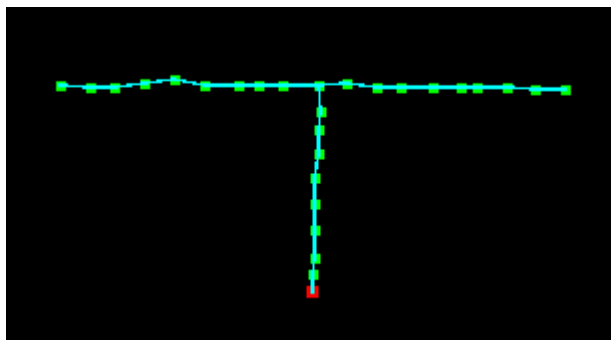
6) 分割

对 OAMST 中的每条边计算评价值，选择评价值最小的边进行分割，递归地对子树继续分割，直到树中节点数符合要求，目前贪心策略采用的评价函数主要考虑：

- 分割后两子树的位置关系，对角关系最佳，上下关系和左右关系次之
- 分割后两子树的“凝聚力”，用方差表示
- 分割后两子树的节点数，尽量均匀，有一边低于 DMIN 直接判负
- 分割后两子树的节点数除以 DMAX 的余数尽量大。这是为了减少小分组的出现

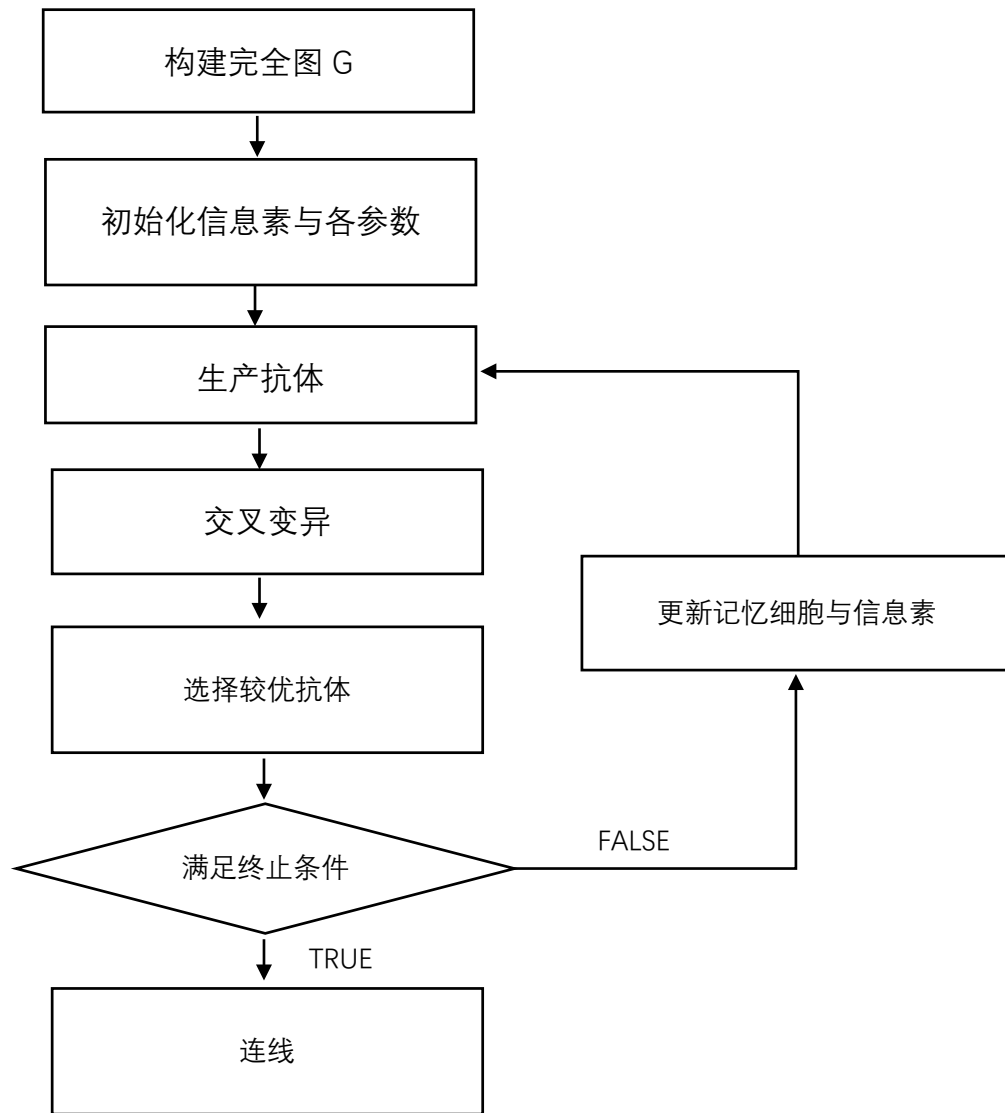
7) 特殊情况

由于使用生成树的策略，每个点只由一条边连到图中，会有特殊情况找不到分割边，例如：



$DMAX=25$ ， $DMIN=12$ ，图中有 28 个点，割完总有一子树节点数小于等于 9

特殊处理策略：按密度聚类

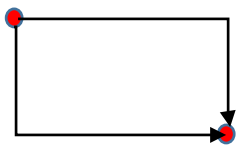


4.2 连线

主要思路为将两个点之间是否连接和如何连接的问题分开来，先以 V_d 构建完全图 G ，在 G 中通过蚁群-免疫算法得出一个认为最优的生成树 T_{best} ，表达各点之间的连接关系，最后根据 T_{best} ，遵循横平竖直原则将点逐一连接。

4.2.1 建图

建立完全图 $G = K(V_d)$ ，任意两点 v_i, v_j 之间的权重 $W(i, j)$ 为考虑障碍开销的曼哈顿距离。定义两点之间的曼哈顿路径有两条，如下：



权重计算遵循以下规则：

- 1) 若两点直接连线不穿过障碍物， $W(i, j) = |x_i - x_j| + |y_i - y_j|$
- 2) 若两点的两条曼哈顿路径有一条不穿过障碍物，同 1)

3) 否则, 根据穿过的障碍物加权, 选择加权后权重小的那一条曼哈顿路径

4.2.2 生成树

采用蚁群-免疫算法。使用蚁群算法从 G 中得到解, 使用免疫算法筛选和保存解。

免疫部分:

- 采用 Prüfer 编码表示一条抗体, 由于一个 Prüfer 编码可以对应完全图中的一棵树, 所以一个抗体即一个解。
- 采用局部记忆细胞保存局部最优解, 全局记忆细胞保存全局最优解, 与抗体表示相同。

蚁群部分:

- 采用信息素表示一条边的选择价值, 每次循环后会挥发一部分, 然后由最佳路径进行更新。
- 采用浓度抑制跳出局部收敛。浓度用{平均节点分支数/节点总数}表示, 节点分支数指的是一个节点关联的边中信息素大于某常数的边数。由定义知, 浓度范围在 $2/n$ 到 1 之间。浓度大: 说明可选路径多, 倾向按信息素大小来选择路径; 浓度小: 说明可选边变少, 趋于收敛, 此时倾向于按信息素大小设置概率, 轮盘随机选择路径来跳出停滞。

1) 初始化信息素与各参数

各顶点之间的信息素初始化为 $A/\text{权重}$, A 为常数。设置蚂蚁数量 m 。

设置全局记忆细胞数量, 局部记忆细胞数量等参数。

2) 产生抗体

每个蚂蚁都寻找一个生成树, 即产生一个抗体。

蚂蚁寻路遵循 prim 算法, 每次根据浓度来决定选取下一个点的策略 (蚁群浓度抑制部分)。

3) 交叉变异

以两种方式产生更多新抗体:

- 局部记忆细胞分化出抗体与新产生的抗体随机配对交叉
- 新抗体中的每个抗体按一定概率进行变异

4) 选择较优抗体

使这批新抗体与抗原结合, 计算亲和度, 即用适应性函数评价抗体, 考虑:

- 总路径长度
- 各点到起点的最短路径的最大值
- 逆回路数量
- 每个点的度不超过 4, 超过直接判负
- ...

选择最好的抗体尝试加入全局记忆库

5) 终止条件

达到指定迭代次数, 或某限定次数内最好全局记忆细胞未更新, 则结束循环, 取全局最好的抗体为最终解, 否则进行 6)

6) 更新记忆细胞和信息素

选择本次循环产生抗体中最好的一批更新局部记忆库, 根据这批抗体更新信息素, 回到 2)

4.2.3 连线

根据生成树中的连接关系，从起点开始连接

- 1) 曼哈顿路径上无障碍，则直接连接曼哈顿路径
- 2) 曼哈顿路径均有障碍，但直接连线无障碍，则直接连线
- 3) 否则将两点形成的矩形（或直线）以及所涉及的障碍物一起进行网格划分，使用 A*算法寻路

所有线连接后，对弯头数进行优化