

修改版文档

目的：将权重最大的连通边集合起来，并且计算处理道路网络需要的机器台数

输入参数：图结构（需要读取文件输入），各节点还对应一个节点权重

前期准备：图结构（邻接表，节点权重表）

1、初始化：

输入邻接表、节点权重表，负荷阈值（load_threshold）

对于节点权重表，生成两个列表（对应节点名字和节点权重）

维护一个记录节点是否已经访问的列表 if_visited_list，维护一个变量：需要的机器台数（param_a）

初始化过程中，对于 if_visited_list,将其中的全部元素设为 false，param_a 初始值设为 1

2、当 if_visited_list 不全为 true 时：

2.1 计算 if_visited_list 中为 false 的所有节点权重之和，如果大于 param_a，那么 param_machines += 1；

2.2 维护一个空列表 path_tmp=[]；

2.2 查找 if_visited_list 为 false 的所有节点中权重最大的那个节点，标记为已访问，然后这个节点就是路径开始节点，加入到 path_tmp 中；

2.3 计算 path_tmp 中节点的权重（设为变量 weight_loaded），当 weight_loaded < load_threshold 时，执行如下步骤：

2.3.1 维护一个列表 candidate_visit [{adj_names,adj_values}]初始化为空；

2.3.2 查找 path_tmp 中最后一个节点的邻接表，按边权重，将相邻的没有访问过的这些节点逐一加入候选访问列表；

2.3.3 如果 candidate_visit 为空，则退出循环；

2.3.3 将 candidate_visit 中的元素按各节点对应的权重从大到小排序；

2.3.4 如果 candidate_visit 中的最后一个元素（也就是权重最小的节点）的权重加上 weight_loaded >= load_threshold,退出循环，否则转到第 5 步；

2.3.5 从 candidate_visit 中从左往右遍历，找到第一个元素 使得对应节点权重加 weight_loaded < load_threshold,然后将该节点标记为已经访问，加入 path_tmp 中，返回到步骤 1；