会议记录

2019.4.28 周日 上午 晴 第三次会议记录

会议记录

会议地点：北京邮电大学图书馆研讨间517

实验名称：模拟外卖派单

与会者：李佳航、崔思颖、吴川宇

文档记录者：崔思颖

会议内容摘要：

问题说明：

1.遍历餐馆后经过食客的路线的选择

2.监控模块需要监控的变量有哪些；

3.完成每个食客派单的时间t[i]的纠正；

4.回溯时间点的判断

5.节点之间具体的走法

6.某个食客重复下单或某餐厅多次被点餐的情况

7.建图、建树的实现

8..路线的优化及多骑手问题

讨论意见：

1.食客与食客之间路线的选择：使用贪心算法，每次都去往最近的点

2.第i个订单派单完成的时刻t[i]=max{d,t0[i]}(t0[i]为第i单的下单时间，d为开始配送订单的时刻加上配送所需时间);当LLP（res[i]到cli[0]的距离+cli[0]到cli[i]的距离）>30时回溯到上一家餐馆

3.节点之间的走法：先判断两节点的相对方向，从而排除不可能走的位置，后建立树结构；判断并记录所有路径，length最大者即为可选路径

4.在分析路线走法时只建道路图

5.判断下一个配送的食客是否会超出期限，如若超出，则回溯到相应餐馆的前一个餐馆，并增加一名骑手。该骑手的派单序列则为上一个骑手无法按时派送的单，如若该骑手也有无法按时派送的单，那么则再雇一名骑手，将这些单转移给新的骑手，以此类推

6.如果有食客或餐馆重复下单，那么需要将相应位置节点移出数组visited[]

7.建图：

（1）储存整张区域地图，每个节点的结构设置变量tag，标识餐厅、食客、道路，并有大小为4的数组，标识该节点的四周位置

（2）关系图：仿邻接矩阵，初识化用0、1、MAX，如果为1，则需要求出最短路径并储存在节点的结构体内

8.建树：建的过程中要判断筛选，且各个子树需要独立

struct node

{int value;

struct node \*next)

tree[0]->value=0;tree[0]->next=p;p=malloc(sizeof…

注意参考《数据结构》一书

9.SLP（res[i]）为经过餐馆的最短路径之和

最后结果：

建图、建树、构思算法优化——崔思颖、吴川宇

算法优化、概要设计、核心模块的代码实现——李佳航

记录时间 2019.5.7 .