# 6.1-7.31总结

# 死锁解除（预防与解除）

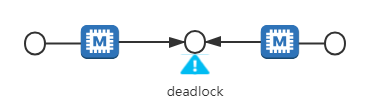
## 功能

防止拥堵，在死锁发生后解除死锁

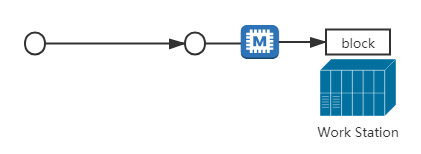
## 死锁场景

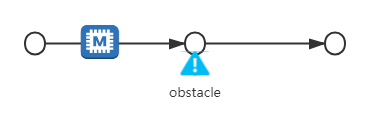
### 可能出现的死锁场景

1. 两辆AGV节点相向冲突，在无向图场景下发生概率较大，有向图几乎不可能发生



1. 互斥区占用时间过长，例如工作站出现问题



1. 故障AGV堵塞路径或节点
2. AGV在同一条路径上相向冲突，在AGV速度较快时会出现

### 死锁情况判定

1. A AGV的下一条路径被B占用，且B的下条路径被A占用
2. AGV等待时间过长（需要设定阈值，作为最终的判定条件）
3. 出现未知障碍物（超时的一种算作异常状态，可以使用deadlock处理）
4. AGV在同一条路径上，且方向相反

## 依赖库

1. Jsoncpp：用来解析json文件和通信
2. Thread：用来实现多线程

## 封装类

### deadlock类

用来处理单一节点的死锁解锁问题，线程安全

#### unlock(int status);处理死锁点，清理死锁AGV，

按优先级与AGV是否可以接受路径确定避让AGV（mir需要起终点，可以自主规划路径；Seer需要给定路径），

status分三种情况

status=1：两车相向冲突；

status=2：障碍物冲突；

status=3：等待时间过长

#### void resetmap(string lockpint);重设地图，

将避让AGV周围的路径改写为双向路径（要解除死锁必须要倒车，因此单向路径无法使用）

#### void newpath();退避路径的选择，

将死锁点设为不可用，规避死锁点，规划新的路径以解锁，调用路径规划类，传入修改后的地图，增加不可通行情况下的路径规划。解锁路径规划的算法拥堵系数的权重应尽量提高，以避免对正常运行的AGV产生影响

#### string predict(AGVinfo AGV1info);预测节点

通过AGV当前位置与角度，遍历地图，查找其运动方向上的节点来预测AGV要经过的节点，以判断是否死锁及查找死锁点

## simplejson

封装了jsoncpp，可以保存，打开json文件，处理json数据及字符串的转换

## simplelog

记录运行日志，包括AGVstatus和指令

# 数据文件

## Mapspec

拓扑地图，json格式，包括节点及其连通点，物理坐标和路径长度

## AGVpath

AGV的路径，json格式，路径规划模块维护

## Rastermap

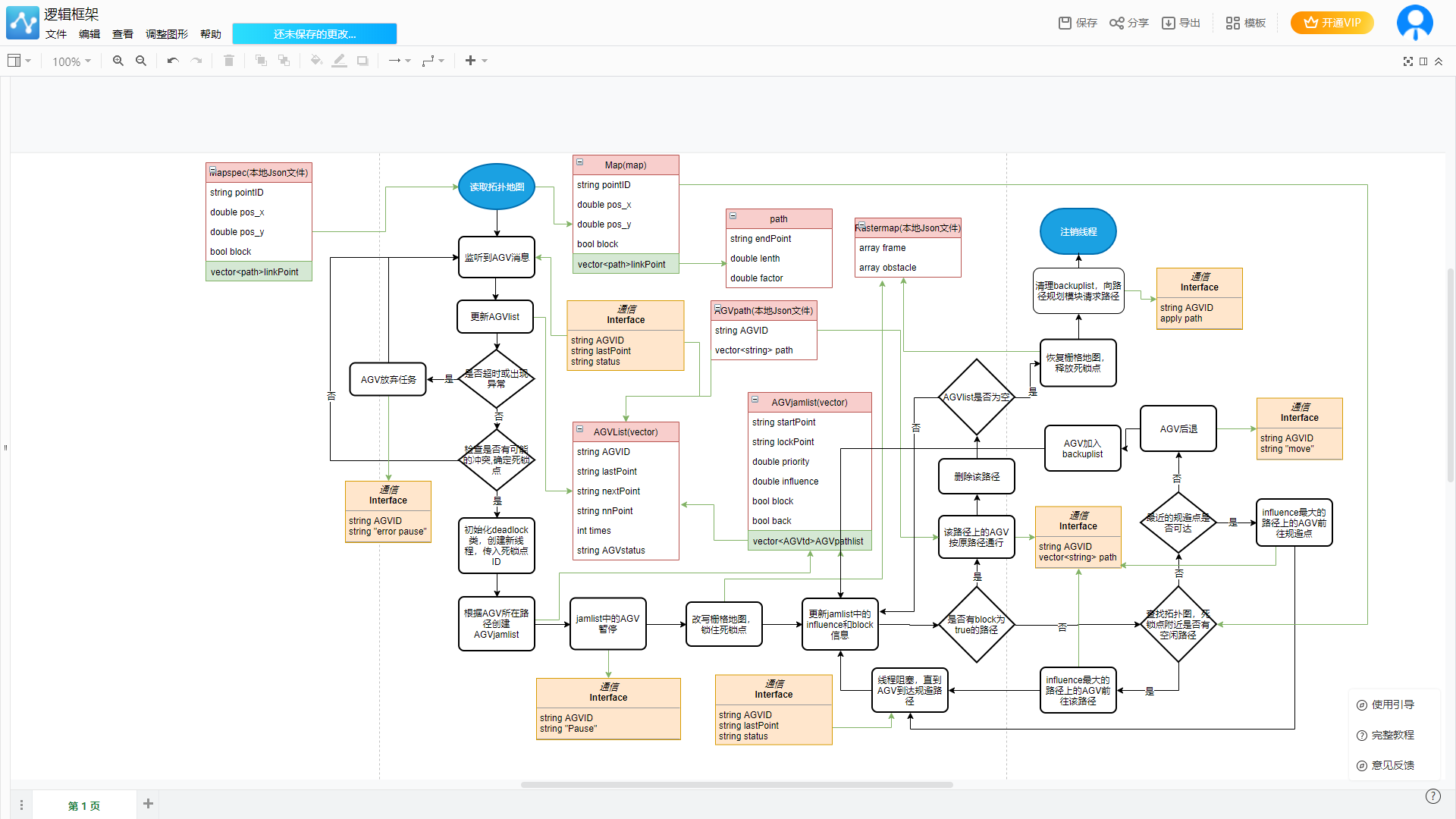
栅格地图，json格式，路径规划模块用来计算time\_table，在发生死锁时要把死锁点锁住，避免新的AGV前往死锁点

## 通信接口

通信格式使用json

1. MR定时定点发送消息TrafficStatus
2. 向MR发送控制指令TrafficControl，包括move、pause、error\_pause
3. 向路径规划模块请求路径

## 避让逻辑



* 1. 双AGV避让

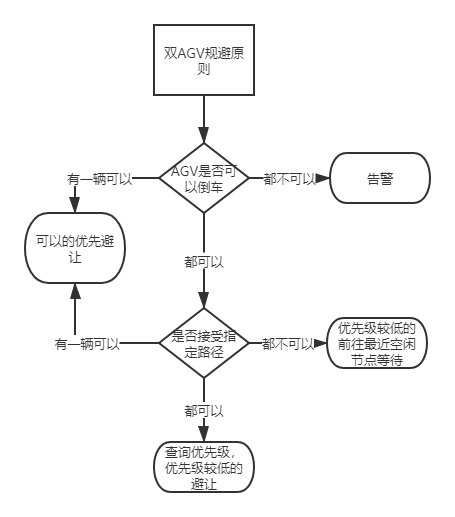
１）锁定该节点，在当前AGV离开该节点前后续AGV不可通过；

２）在锁定该节点的前提下，基于现有时间窗或拥堵为后续ＡＧＶ重新规划执行任务的路线；

３）计算当前AGV通过后一节点的新时刻，记作后一AGV的等待时间，基于此计算该AGV后续的各节点、路径的时间窗，判断与后续路径己知的时路径是否存在冲突，如存在，舍弃该路线；如不存在，暂时保留；

４）判断新生成的路线的时间成本与等待后的路线的时间成本；时间成本小的作为最终的最优路径；

５）生成任务路径命令，将重规划的路径派发给后一AGV。



* 1. 多AGV避让

1)确定冲突点;

2)遍历与冲突点连通的节点;

3)寻找所有在这些path上的AGV，加入冲突AGV表；

4)遍历冲突AGV表，计算每条path上的AGV数量之和

5)循环进行如下操作：若AGV数量大于二，循环;若等于1，结束;

(1)寻找优先级最小的path；

(2)将该path上的AGV按到冲突点的距离（根据AGV当前位置计算）push避让AGV表；

(3)pop避让AGV表，避让AGV将冲突点设置为不可用，依次重新规划路径（这样可以做到整条路径上的AGV全部避让，减少循环次数）

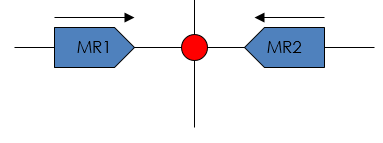
## 实际场景测试

场景：

AGV数量：2台seer

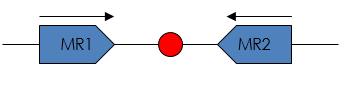
AGV速度：0.2m/s

Case1：节点相向冲突，已通过



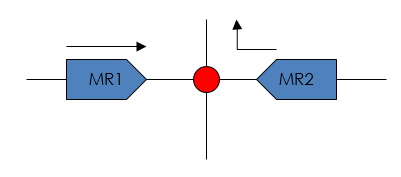
一台车会去另一路径避让，待另一台车通过后，继续运行

Case2：路径上相向冲突，测试通过



一台车会后退，待另一台车通过后，继续运行

Case3：转弯直行冲突，测试通过



MR1会暂停，等MR2通过后，继续运行

Case4：超时，部分测试通过

在前方没有障碍物时，会按原路运行

## 存在的问题

1. 目前测试，MR速度只有0.2m/s，在速度较快时的情况还没有测试
2. 目前只测试了两台车，三台车以上的情况还没有测试
3. MR在两个节点相距较近时，可能会部分阻挡其他节点，死锁解除模块无法发现这一情况，需要进行处理

# 工作计划安排

## 八月：

1. 测试AGV在快速运行的情况下的表现，尽可能减少响应时间，优化算法
2. 死锁处理与路径规划的融合
3. 确定地图的规划要求，尽量预防死锁和拥堵、冲突发生的可能
4. 软件可视化
5. 总结工作，整理文档