目前能想到的需要实现的逻辑：

1. 由小车的当前cross选择下一条next\_road，下一个next\_cross

思路：

小车的出发点即是第一个current\_cross，根据current\_cross\_id可以从cross\_data中找到next\_road\_id，根据next\_road\_id和current\_cross\_id可以从road\_data里面找到next\_cross。

这里的next\_road是一个集合（最多4条最少2条），如果有的路走过（在car\_path里面有），就不会选这条路。

这里找到的next\_cross是一个集合，如果里面有终点，就直接选这一个点。

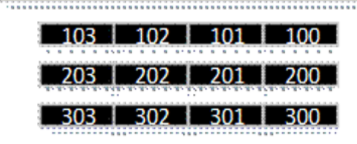
next\_cross是否双向也需要考虑，暂时不考虑length，speed等影响蚁群选择路径的因素。只考虑帮助蚂蚁找打可以选择的路。

参考： AG0.1.py

1. 小车出发时，如何选车道，如何不超车，出发是否会堵住？

解释：

如下图所示，这是赛题里面给出的车道选择规则，该如何实现赛题中的车道选择规则，以及如何防止超车（优先级？根据前车速度改后车速度？前车出路口之后，根据什么规则出发后车速度的恢复？），出发时怎么判断被堵住。需要有切实可行的思路。



1. 路口处能否通行的逻辑，如何通行的逻辑
2. 车辆是直行还是左转还是右转的判断

3，4点其实是相互关联的，4的转向判断那天已经给出（根据currnent\_road,next\_road），但是怎么选择next\_road和第3点关系其实很大，如果道路不通或者堵住，我们是可以选择换路的，而换路之后车的转向也变了，那么路口的调度顺序也会发生变化。所以我觉得还是得先选路，然后确定转向，然后进行路口调度。

这里的调度包括赛题中的v1,v1，SV1，SV2之类的判断，也包括车辆调度顺序的判断，还包括车辆退回（前方挡住，调度延后，死锁的根源）。这里我觉得是调度最困难的地方。需要多讨论讨论。

1. 路径选择

以蚁群算法为基础，我觉得可以加入道路平均速度、道路末尾速度、道路车辆数等影响蚂蚁对道路的选择。这个可以放后面一些讨论。

1. 车辆移动，到达终点

这个逻辑比较简单，根据时间片来移动即可，这个和第2点也是相互关联的，不能超车变道的情况下如何实现车辆移动。

车辆移动到终点之后，理论上是把车辆“移除”，并输出car\_path的，这里“移除”该以怎样的方式表现和实现，也需要考虑清楚。

1. 信息素的更新

这里由于每辆车的起始点和终点都不一样，信息素会相互干扰，而且和标准蚁群算法的矩阵记录信息素浓度方法不同，我们没法放在矩阵里记录，暂时不考虑信息素更新。

1. 限制搜索策略

保证蚂蚁不是乱走，而是有“方向”的走。

1. 整体的大循环框架

每循环一次，时间片就加1，当所有车辆到达目的地时，循环结束，这里需要一个伪代码作为整体指导。