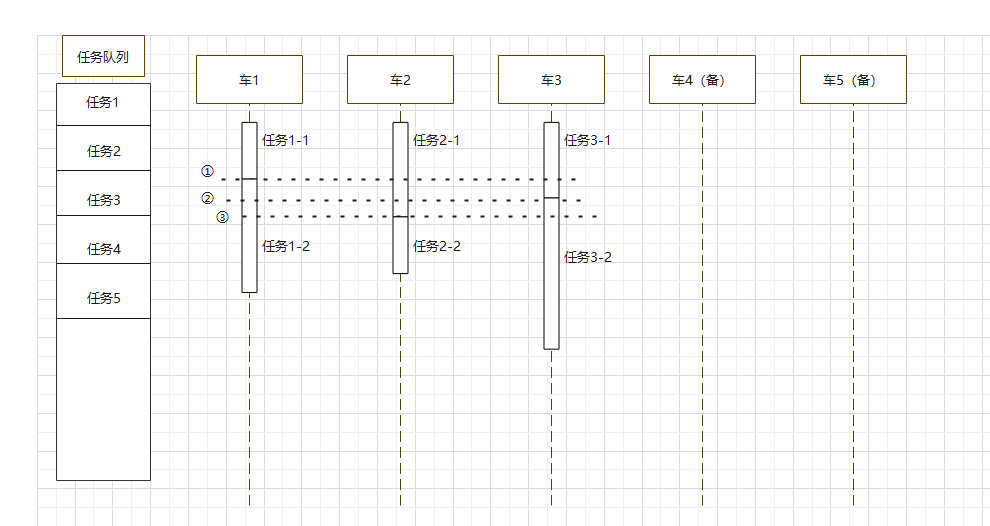
1. 现阶段的问题
2. 路径占用障碍未传，存在车辆相撞问题
3. 先进后出场景
4. 相同目的地坐标点问题（门口等待区折中方案）
5. 任务执行时调度不支持换车充电
6. 任务规划逻辑

在进行任务路径规划时，已规划的路径作为路径障碍物是避免相撞的主要因素，因此在路径规划时需计算对应时间基准的路径障碍。

规划示例如下：

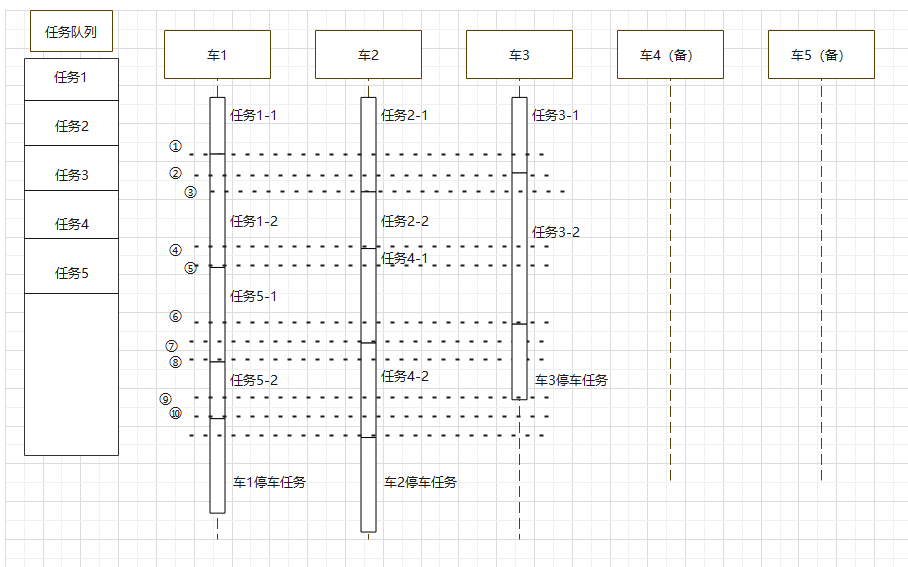
例如：规划5个任务，选3辆车并行方案，2辆空闲车作为备选。

Ps：任务1-1表示任务1的第1阶段路径



如上图所示，第一批次规划流程如下：

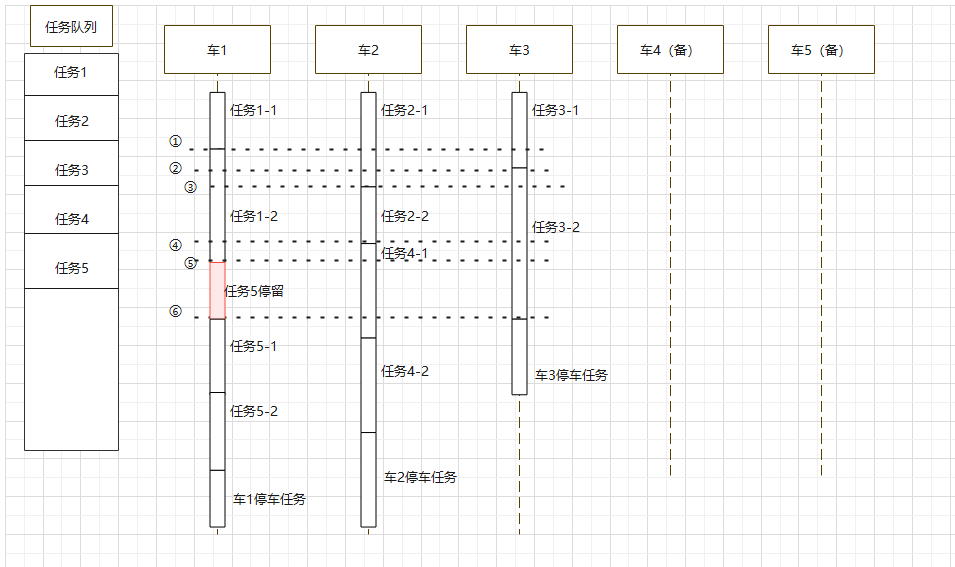
1. 任务1分配车1，任务2分配车2，任务3分配车3，无路径障碍，进行三个任务的第1阶段规划，输出路径任务1-1，任务2-1，任务3-1。其中任务1-1路径最短。
2. 任务1-1路径先执行完（虚线①），规划任务1-2路径，基于虚线①的基准，截取任务2-1，任务3-1的剩余路径作为路径障碍物传入，规划出任务1-2路径。
3. 然后任务3-1路径执行完（虚线②），规划任务3-2路径，基于虚线②的基准，截取任务1-2，任务3-1的剩余路径作为路径障碍物传入，规划出任务3-2路径。
4. 最后任务2-1路径执行完（虚线③），规划任务2-2路径，基于虚线③的基准，截取任务1-2，任务3-2的剩余路径作为路径障碍物传入，规划出任务2-2路径。至此，第一批次任务规划完成。如图所示，任务2总路径最短，车2先释放出来。



如上图所示，规划后续批次任务

1. 车2先释放（虚线④），任务4分配车2进行路径规划，基于虚线④的基准截取1-2，3-2剩余路径作为路径障碍物传入，规划出任务4-1路径。
2. 然后任务1路径完成，车1释放（虚线⑤）。任务5分配车1进行路径规划，基于虚线⑤基准截取任务4-1，任务3-2的剩余路径作为路径障碍物传入，规划出任务5-1路径。
3. 然后任务3路径完成，车3释放，无其他任务，规划停车阶段。
4. 依此类推，基于图中的虚线处基准进行路径截取作为后续任务障碍物传入进行路径规划。

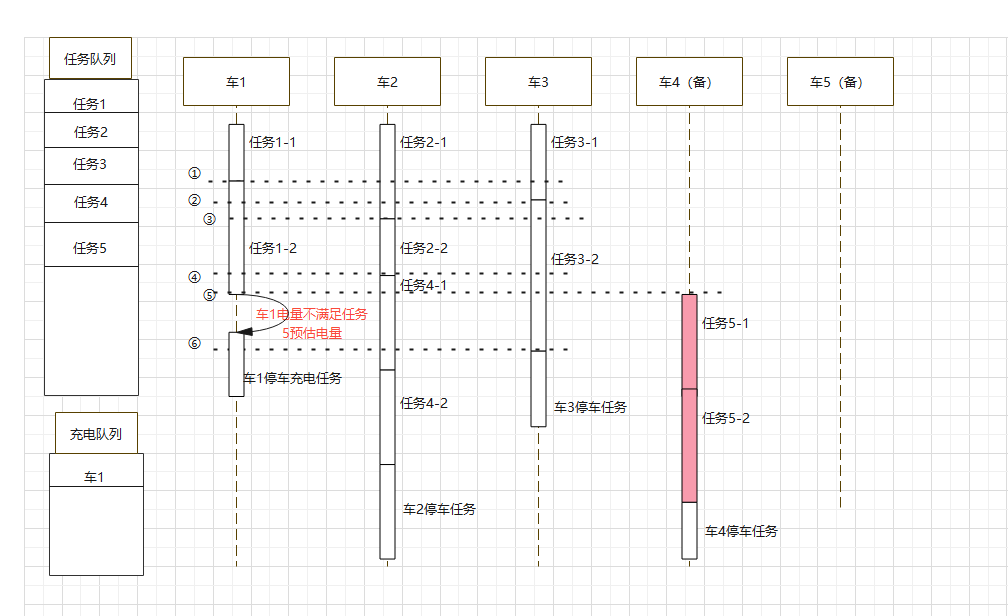
情况1：路径规划不出时的情况



如上图所示：

车1释放后，任务5分配车1进行路径规划（虚线⑤），假设任务3-2剩余路径作为障碍规划不出任务5-1的路径。任务5则需要停留，等待该路径释放后，在虚线⑥的时间基准上再进行路径规划。

情况2：车剩余电量不满足任务预估电量时，需换车规划

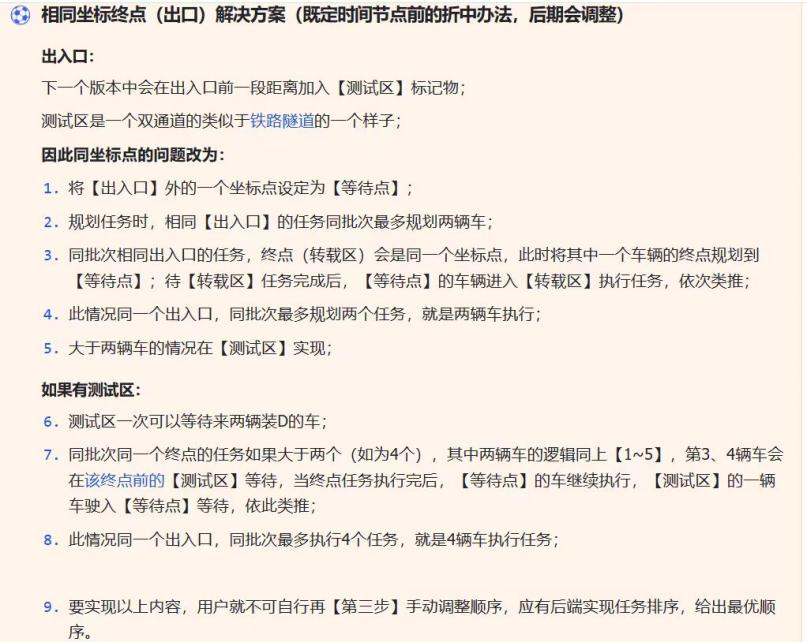


如上图所示：

车1释放后，接受任务5进行规划，假设预估电量大于车剩余电量，车1不满足执行。任务5则需换车执行，选择车4，在同样的时间基准点上进行路径规划（虚线⑤），路径障碍物也相同。

特殊情况：无备选车辆？

情况3：相同目的地（货架/出入口）的情况



目前出了一版折中方案，以设置等待点的方式进行路径冲突规避。

在路径规划时，同一批次中存在相同目的地（货架/出入口）时，将后续任务改变规划结束点为等待点，然后在任务调度时动态根据货架/出入口内任务执行情况调度任务等待或进入目的地。

1. 指令任务规划结果的执行顺序存储

采用链表的方式存储任务间的执行关联顺序。

如：

任务1（车1）-> 任务5（车1）-> 车1停车充电任务

任务2（车2）-> 任务4（车2）-> 车2停车充电任务

任务3（车3）-> 车3停车任务

数据库表存储：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 任务 | 停留步长 | 车辆 | 下一个任务 | 其他字段··· |
| 任务1 | 0 | 车1 | 任务5 |  |
| 任务2 | 0 | 车2 | 任务4 |  |
| 任务3 | 0 | 车3 | 停车任务 |  |
| 任务5 | 20 | 车1 | 停车任务 |  |
| 任务4 | 0 | 车2 | 停车任务 |  |

1. 任务调度

不考虑实际运行中动态障碍物的情况，按照以上的任务规划逻辑及任务的关联顺序，有序的进行任务调度。