

队伍编号	MC2208157
题号	B

基于蚁群算法和遗传算法的小车（AGV）调度问题研究

摘 要

随着电商的兴起，无人仓逐渐作为自动化仓储物流系统的发展方向和目标。无人仓的目标就是指实现仓库作业流程的无人化操作包括入库、存储、分拣、出库等。但由于AGV在路径规划中不可避免地会出现碰撞，系统死锁等问题，严重影响路径优化，降低了系统柔性和使用效率，这是目前AGV研究领域的一大热点和难点。AGV的避碰问题主要分为两类，一类是对静态障碍物的躲避，另一类是将系统中其他AGV视为动态障碍物，避免碰撞。[4]因此，AGV避碰路径优化的任务内容即在具有静态、动态障碍物的环境中，能够按照用时最少、路程最短的要求，寻找一条从出发点到目的地的无碰路线。文章针对所给的三个问题，建立了多约束条件下的AGV调度模型，并结合题目所给的图和表制定最优的AGV行动路径和最短的运行时间。在该问题中，建立了AGV的行动模式模型、AGV的静态防碰撞模型、AGV的动态防碰撞模型

针对问题一，在某些地图节点无法到达以及需要将储位节点的货物运送到拣选节点等诸多基础限制条件的基础上，模型考虑基于蚁群算法和遗传算法的最优路径和最短时间的AGV调度数学模型。综合考虑任务中各台AGV的最短路径、储位等节点的坐标位置等因素。通过蚁群算法，规划每个AGV的最短行动路径，同时，避免出现多台AGV同时前往同一个储位节点的情况，使用遗传算法进行优化。通过MATLAB软件编写程序和利用数学不等式建立起调度模型，在不考虑碰撞，AGV只能走直线以及求得最短路径等限制下，计算得出AGV总的最短路程为3113个单位距离（一单位距离长度为一格）

车号	1	2	3	4	5	6	7	8	9	10
路程	126	122	149	235	128	218	69	254	295	130
车号	11	12	13	14	15	16	17	18	19	总长
路程	230	154	146	215	160	162	76	85	194	3113

由图可知问题一中最短AGV路程为76，最长为295，总路程为3113

针对问题二，由于考虑到需要平衡拣选工位的负载，同时预防小车的局部拥堵，将商品数量尽可能的均分到不同的拣选节点，在问题一模型的基础上，需要增加一个新的

限制条件：每个拣选点位对应一个由若干个规划而成的储位节点的集合。增加一个新的要求：尽可能的将商品数量平均分配到拣选工位上。根据这两个建立起通过遗传算法优化过的蚁群算法来进行编写 MATLAB 程序，建立起在尽可能平均分配商品数量的最短 AGV 运行路程，最后计算得出最短的 AGV 路径总长度为 3320 个单位距离

车号	1	2	3	4	5	6	7	8	9	10
路程	138	199	132	149	162	202	246	237	214	370
车号	11	12	13	14	15	16	17	18	19	总长
路程	139	158	147	133	132	107	110	202	216	3320

由图已知问题二中最短 AGV 路程为 107，最长的为 370，总路程为 3320

针对问题三，对于存在 AGV 的碰撞约束和死锁可能的模型优化问题，可以在问题一核问题二的基础上，通过建立 AGV 运动的动态规划模型来选择某时刻各个 AGV 的最优行动路径，通过建立优先级锁定和等待策略的小车防撞模型，来实现小车间动态避让。优先级别为 1,2,3, ...,19。即编号越靠前的 AGV，优先级别越高，相对优先级别第的小车在进行等待策略时需要对相对较高优先级的 AGV 等待，等较高优先级的 AGV 通过后再行动最后通过修改之前蚁群算法和遗传算法的限制条件便可以求解出不出现碰撞的最优解决策略，由于有通过算法加入一定的随机性，最后得出的运行最长路径的 AGV 的最短路程也是相对最优解为 363 个单位距离，若用时间表示，最短时间为 363（速度设定为单位速度 1）。

序号	1	2	3	4	5	6	7	8	9	10
路程	138	199	132	149	162	202	248	237	207	362
时间	138	199	139	149	162	205	248	237	207	363
序号	11	12	13	14	15	16	17	18	19	
路程	139	158	145	133	132	107	110	202	216	时长
时间	140	160	153	137	133	113	119	218	217	363

（表中为各个 AGV 运行路程数值上不等于运行时间，完成所有任务的时间取决于最晚完成任务的 AGV，即 10 号小车的运行时间）

由此得出问题三的最优解为 363 个单位时间

（后文中所有搬运机器人简称为小车）

关键词：多 AGV 路线规划；蚁群算法；遗传算法；动态规划

目录

1. 问题的重述	1
1.1 问题的背景.....	1
1.2 问题一.....	2
1.3 问题二.....	2
1.4 问题三.....	2
2. 基本假设	2
3. 符号说明	3
4. 问题分析	3
4.1 问题一.....	3
4.2 问题二.....	4
4.3 问题三.....	4
5. 模型的建立与求解	5
5.1.1 问题一模型建立:	6
5.1.2 算法分析:	8
5.1.2.1 蚁群算法[5]	9
5.1.2.2 遗传算法分析[5]	10
5.1.3 具体解法及答案	12
5.2.3.1 思路一	20
5.2.3.2 思路二	20
5.2.3.3 思路三	21
5.3.2.1 动态分析	27
5.3.2.2 优先级分析	27
5.3.2.3 等待策略分析	27
5.3.2.4 禁忌搜索算法[3]	28
6. 模型评价	34
7. 参考文献	34
8. 附录	35
8.1 问题相关数据.....	35
8.1.1 问题一相关数据参考: “问题一相关数据.xlsx”	35
8.1.2 问题二相关数据参考: “问题二相关数据.xlsx”	35
8.1.3 问题三相关数据参考: “问题三相关数据.xlsx”	35
8.2 主要软件	35
8.3 相关主要源代码及程序	36

1. 问题的重述

1.1 问题的背景

仓库管理(也叫仓储管理)是对仓库内货物的接收、存储、发货等一系列活动进行有效的控制管理,维护仓库货物并确保日常经营活动正常进行。传统的仓库管理模式大多是人工管理模式,而这种管理方式有着不少问题,特别是因为人力效率低下的因素导致的“爆仓”现象。随着电商的兴起,无人仓逐渐作为自动化仓储物流系统的发展方向和目标。无人仓的目标就是指实现仓库作业流程的无人化操作,包括入库、存储、分拣、出库等。在国内,无人仓技术起步早,且发展速度快,然而,现在无论国内还是国外,许多电商巨头都开始建立无人仓。因此,在未来发展中,通过加快技术升级,降低运行成本,精准服务消费者来提高自身竞争力才能使无人仓技术应用得以致胜。目前,在无人仓技术设施中,典型的搬运设备有输送线、自动导引车(Automated Guided Vehicle,简称AGV)、穿梭车、类Kiva机器人、无人叉车等等,智能搬运设备的应用大大提高了作业效率,但在运行中,如何快速找到最优路径并防止碰撞成为当前应用中值得研究的问题。

AGV 是一种智能搬运设备,装有电磁学或光学等自动引导装置,并且能够沿着规定的路径行驶,具有安全保护及移栽功能。作为无人仓库运营的关键设备,其工作效率直接影响无人仓库整体运营效率。合理的AGV行驶路径不仅能提升无人仓库内产品流通效率与工作效率,还能减少AGV派发数量并节约能耗,增加其有效工作时间。然而,随着搬运设备数量的不断扩张,AGV在路径规划中不可避免地会出现碰撞,系统死锁等问题,严重影响路径优化,降低了系统柔性和使用效率,这是目前AGV研究领域的一大热点和难点。AGV的避碰问题主要分为两类,一类是对静态障碍物的躲避,另一类是将系统中其他AGV视为动态障碍物,避免碰撞。因此,AGV避碰路径优化的任务内容即在具有静态、动态障碍物的环境中,能够按照用时最少、路程最短的要求,寻找一条从出发点到目的地的无碰路线。

1.2 问题一

对于问题一，假设不考虑小车的可能的碰撞问题，在无人仓的模型下设计调度，根据附件中的订单数据，仓库中的库存数据，使得每个小车尽可能忙的同时，最小化的行走总路径。

1.3 问题二

对于问题二为了更好地平衡拣选工位的负载，同时预防小车的局部拥堵，根据拣选工位和库存商品数量对仓库地图进行动态分区。也就是对仓库内储位上的每个托盘，都指定一个默认拣选工位。进一步，根据某段时间内所需出库商品的库存分布，再结合问题一的 AGV 调度算法，更合理地均衡每个拣选工位在某段时间内的工作量。

1.4 问题三

对于问题三在问题一和问题二的基础上，进一步考虑小车的碰撞和拥堵问题。当仓库内同时有多个 AGV 在执行任务时，不可避免有些 AGV 在某个路径节点上相遇。特别地，如果两个 AGV 在一条货架窄巷道上相遇，那么需要其中一个 AGV 避让。在合理的假设下，设计算法和防碰撞策略，使得小车能智能地避免碰撞。特别是在一些特殊节点处（如托盘回收处），避免出现多个 AGV 的拥堵，和可能的死锁场景。

2. 基本假设

假设一：刚开始可正常运行的小车不会因内部原因发生故障

假设二：所有小车任何情况下速度相同且为一

假设三：每个小车占据大小刚好为一个栅格

假设四：拣选工位，存储仓位可以让小车通过

假设五：所有小车相撞对其运行无影响（仅对问题一、问题二适用）

假设六：每个拣选工位的拣选商品速度相同（仅对问题一、问题二适用）

假设七：每次小车在拣货完成后都回到此次任务的托盘节点，简化了回库与回收问题。

假设八：对于 b 个停靠点的约束，我们在第一问中假设为无穷大，忽略 b 个停靠点对问题造成的影响。

3. 符号说明

符号	相关意义
I	AGV 节点集合
J	托盘节点集合
K	拣选工位节点集合
l	拣选工位处理商品数量下限
u	拣选工位处理商品数量上限
L_j	每个托盘默认节点
x_{ij}	从节点 i 到节点 j
n	商品数量
t_{ij}	从节点 i 到节点 j 的时间

4. 问题分析

4.1 问题一

问题一要求我们在不考虑 AGV 之间的碰撞体积和拥堵情况的前提下对 AGV 的路程进行最小化，首先，由于有一个 AGV 故障，排除在外，同时考虑使用蚁群算法进行路径优化和实现避障功能，又考虑到有可能出现多个 AGV 前往同一个储位节点而造成路程浪费和不必要的拥堵，为了避免此类问题发生，希望通过先到先得的思想，将 AGV 进行编号，给每个 AGV 分配不同的储位节点，同时使用遗传算法综合考虑所有可能产生的路程长度，求出所有可能性的最小值，即最小的路程长度，在全场 AGV 的速度保持不变的前提下，即为相对的全局最优解

4.2 问题二

在问题一所建立的模型的基础上，问题二需要增加一个新的约束条件同时还要增加一个新目标，即每个小车拣选的商品数量尽可能在一个范围 P 内（ P 的上限和下限需要由经验来确定，可以多次尝试，获得一个较为合适的上下限（ l 为下限， u 为上限））。由此，每个储位节点就会确定一个的拣选工位，根据拣选工位的不同，将同一个拣选工位的划分在同一个集合 A_1, A_2, A_3, \dots ，由同一个小车进行搬运。由此 19 辆可以使用的小车，有 18 辆可以进行分配（命名时，1 号小车搬运 A_1 ，以此类推），剩余一辆小车。由于小车搬运的距离与商品数量无关，而与托盘数量相关，所以有较多托盘的集合，运行时间会较多。即可以让剩余的那辆小车一起运送托盘数量最多的集合，由此使每辆小车的移动距离大致相同，达到让每个小车尽可能忙的同时最小化小车的行走总路径的目的。设定一系列的 A_1, A_2, A_3, \dots 为对应每一个机器人所需要前往的储位节点

4.3 问题三

对于问题三，由于考虑到不同的 AGV 之间会存在碰撞体积，即对于每一个 AGV 而言，其他正在运行的 AGV 为障碍物，同时障碍物的位置坐标随着时间也在发生改变，问题转化为动态的避障行为。同时也会出现死锁的情况导致 AGV 拥堵和碰撞，因此我们考虑在问题一和问题二的基础上，首先将货物数量如问题二中尽量平均到每一个拣货节点上，从而在整体上预先减少可能出现的死锁和拥堵问题，并且在问题一的模型上进行模型优化，考虑使用动态分析的方法和优先级排序的方法来实现避障和搬运任务，优先级排序即为每个 AGV 的编号，编号小的 AGV 优先级别相对较高，在等待过程中优先进行，所以避障主要通过等待策略来实现。AGV 间的动态避障最后通过运行时间最长的 AGV 运行时间（类似短板效应）即为 19 辆 AGV 的总运行时间。

5. 模型的建立与求解

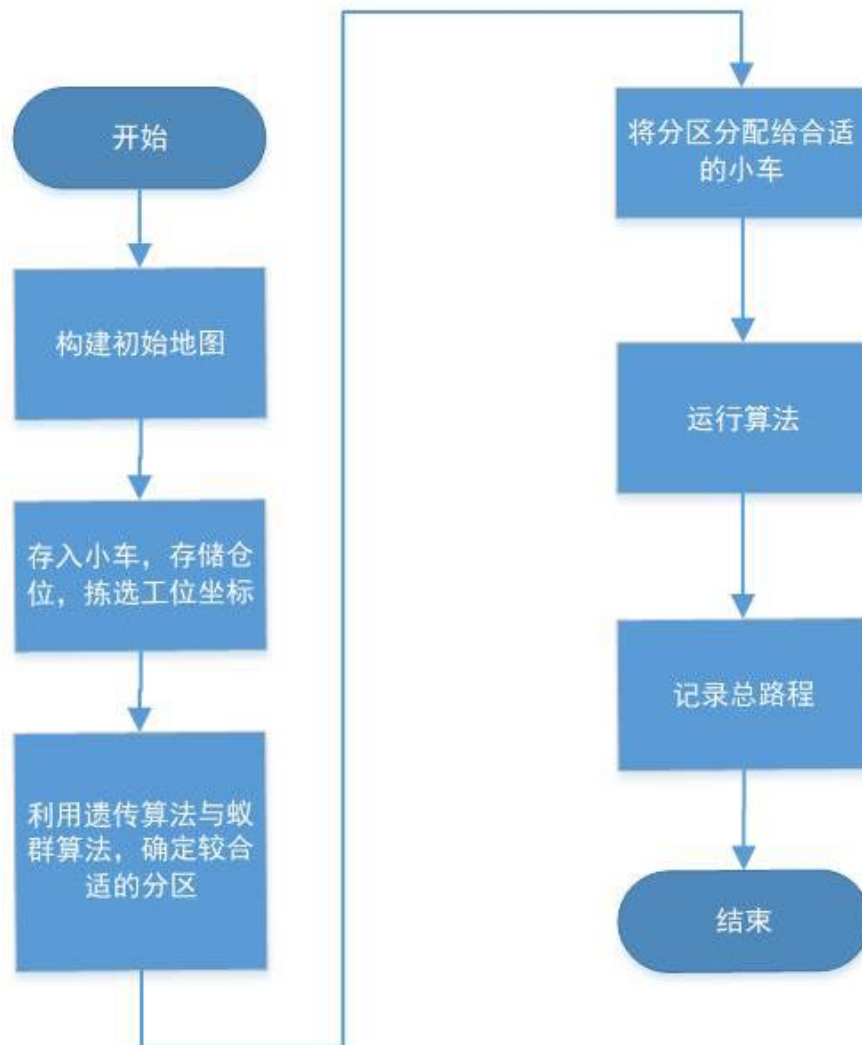


图 1，总体流程

5.1.1 问题一模型建立：

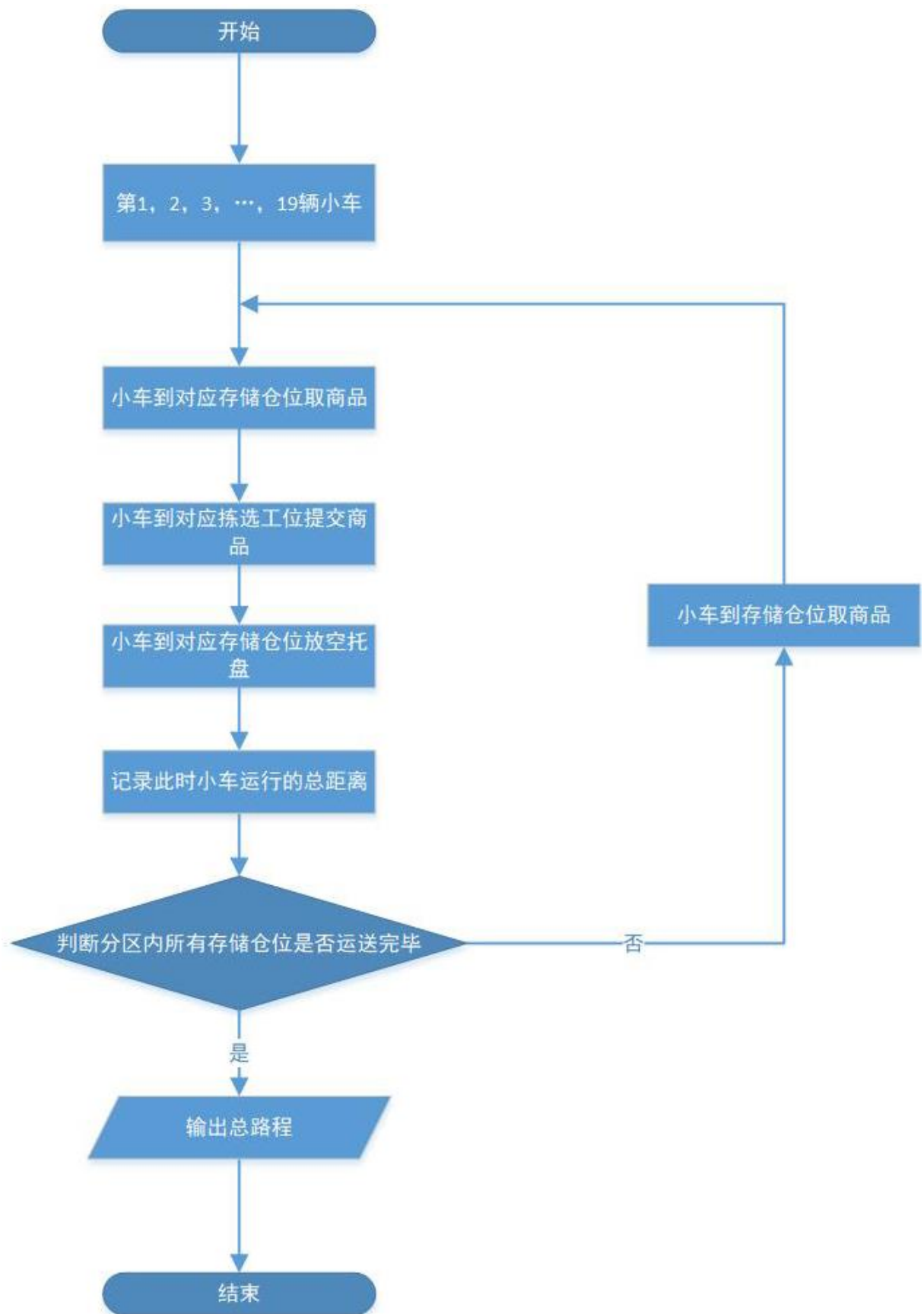


图 2，问题一求解流程

对于问题来说，我们可以用有向图直观分析[2]

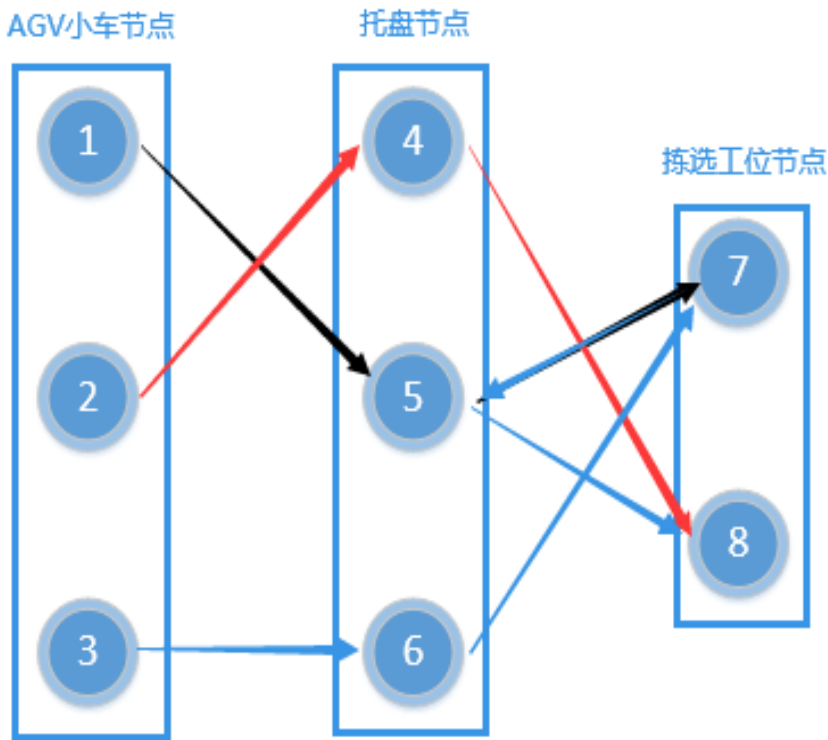


图 3 运行有向示意图

我们把 AGV 小车，托盘和拣选工位分为三类节点，如上图所示

例如在图中 1 号小车路线是 1-5-7

2 号小车路线是 2-4-8

3 号小车路线是 3-6-7-5-8

AGV 节点集合：I

托盘节点集合：J

拣选工位节点集合：K

所有节点集合 $M=I \cup J \cup K$

定义决策变量：

$$x_{ij} = \begin{cases} 1, & \text{小车经过边 } (i, j) \\ 0, & \text{otherwise} \end{cases}$$

约束 1：每个 AGV 小车节点有且仅有一个后继节点，并且后继节点为托盘节点。

$$\sum_{j \in J} x_{ij} = 1, \forall i \in I$$

$$x_{ij} = 0, \forall i \in I, \forall j \notin J$$

约束 2：每个托盘节点有且仅有一个前继节点和后继节点，并且所有后继节点为拣货工位节点，所有前继节点为 AGV 小车节点。

$$\sum_{i \in I \cup K} x_{ij} = \sum_{i \in K} x_{ji} = 1, \forall j \in J$$

约束 3：来到所有拣货工位上的车次必须等于离开拣货工位上的车次，等于托盘数。

$$\sum_{i \in I} \sum_{j \in J \cup \{o\}} x_{ij} = \sum_{i \in I} \sum_{j \in J} x_{ji} = n$$

此处 n 为订单数， o 为设立的虚拟节点，假设小车在完成任务后都会到达虚拟节点 o ，保证每辆小车在完成任务后不会停留在拣货工位上，也保证了来到所有拣货工位上的车次必须等于离开拣货工位上的车次，等于托盘数，此处简化为 o 点为每次任务的托盘节点。

$$\text{目标函数: } \min \sum_{i \in N} \sum_{j \in N} x_{ij} t_{ij}$$

其中 t_{ij} 代表节点 i 到节点 j 所花费的时间，目标函数是使求得时间最短， t_{ij} 在问题中可通过最短路径求得，所以在建模过程中 t_{ij} 已经是常数，不用做过多考虑。

对于问题一来说我们做出以下假设：

- 1) 每次小车在拣货完成后都回到此次任务的托盘节点，简化了回库与回收问题。
- 2) 对于 b 个停靠点的约束，我们在第一问中假设为无穷大，忽略 b 个停靠点对问题造成的影响。

5.1.2 算法分析：

5.1.2.1 蚁群算法[5]

蚁群算法的基本原理来自于自然界蚂蚁觅食的最短路径原则。蚂蚁虽然没有敏锐的视力，但是能在其走过的路径上释放一种蚂蚁特有的分泌物——信息素，信息素能够使得一定范围内的其他蚂蚁能够察觉并由此影响他们以后的行为。当一些路径上通过的蚂蚁越来越多时，其留下的信息素也越来越多，以致信息素强度增大（随时间的推移信息素会逐渐减弱），所以蚂蚁选择该路径的概率也就越高，从而增加了该路径的信息素强度，这种选择过程被称为蚂蚁的自催化行为。由于其原理是一种正反馈机制，因此，也将蚂蚁王国理解为所谓的增强型学习系统。

$$P_{IJ}^k(T) = \begin{cases} \frac{[\tau_{IJ}(T)]^\alpha * [\eta_{IJ}(T)]^\beta}{\sum_{S \in ALLOW_J} [\tau_{IS}(T)]^\alpha * [\eta_{IS}(T)]^\beta}, & J \in ALLOW_K \\ 0, & J \notin ALLOW_K \end{cases}$$

其中:

k 表示第 k 只蚂蚁: k= 1,2,3,4, ..., M

I j 为栅格地图的具体坐标;

$\tau_{IJ}(T)$ 表示 T 时刻 I 栅格到 j 栅格的信息素, 信息素初始值一般设为较小常数;

$\eta_{IJ}(T)$ 是栅格 I 到栅格 j 的启发信息, $\eta_{IJ}(T) = 1/d_{IJ}$, d_{IJ} 表示栅格 i 到栅格 j 的距离;

$j_k(i)$ 为蚂蚁 k 在栅格 I 的可行域;

α, β 分别表示信息素和启发信息的重要程度;

禁忌表 $Tabu_k$ 存入蚂蚁 k 已经路过的栅格, 若蚂蚁陷入死胡同后无后续栅格可选, 则默认该蚂蚁已经死亡, 算法删除该蚂蚁及其所寻路径。

每次迭代中当所有蚂蚁寻径完成后, 对所有路径上的信息素进行更新

$$\Delta\tau_{ij}(T+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(T)$$

$$\Delta\tau_{ij}(T) = \sum_{k \in K_{ij}} \Delta\tau_{ij}^k(T)$$

$$\Delta\tau_{ij}^k(T) = Q/L_k$$

其中, ρ 是信息素挥发系数, $\Delta\tau_{ij}(T)$ 是本次循环节点 I j 间信息素增量, K_{ij} 为经过节点 I j 的所有蚂蚁, $\Delta\tau_{ij}^k(T)$ 为第 k 只蚂蚁所留下的信息素量, L_k 是蚂蚁 k 所寻路径的长度, Q 为一正常数。在全局信息素更新完成后, 把所有蚂蚁放回原起点重新寻径, 经过 G 次迭代完成后, 计算出最短路径。

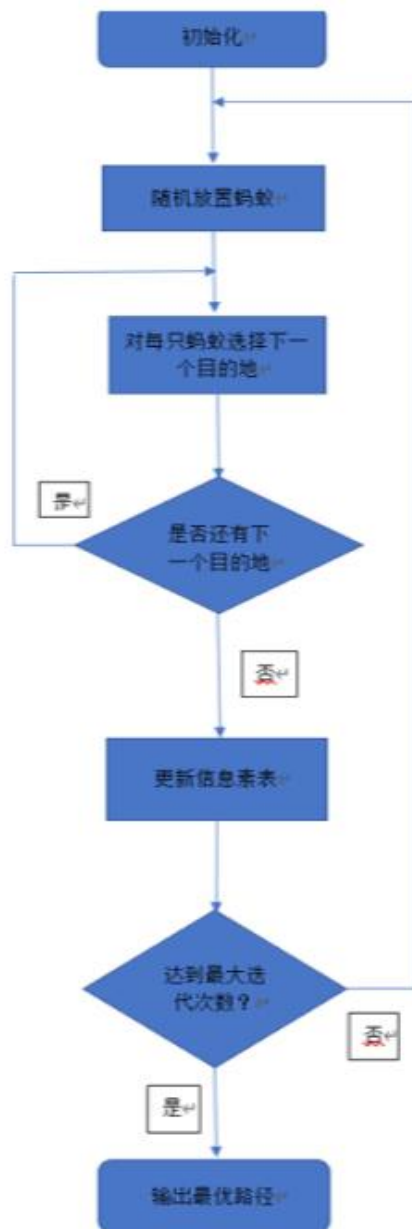


图 4，蚁群算法流程图

5.1.2.2 遗传算法分析[5]

遗传算法是一种基于自然选择和基因遗传学原理，借鉴了生物进化优胜劣汰的自然选择机理和生物界繁衍进化大基因重组，突变的遗传机制的全局自适应概率搜索算法。在这里使用遗传算法主要是为了避免多个 AGV 同时前往同一个储位节点的情况，同时相当于对蚁群算法进行算法优化，跳出局部最优解，走向全局最优解，具体公式：

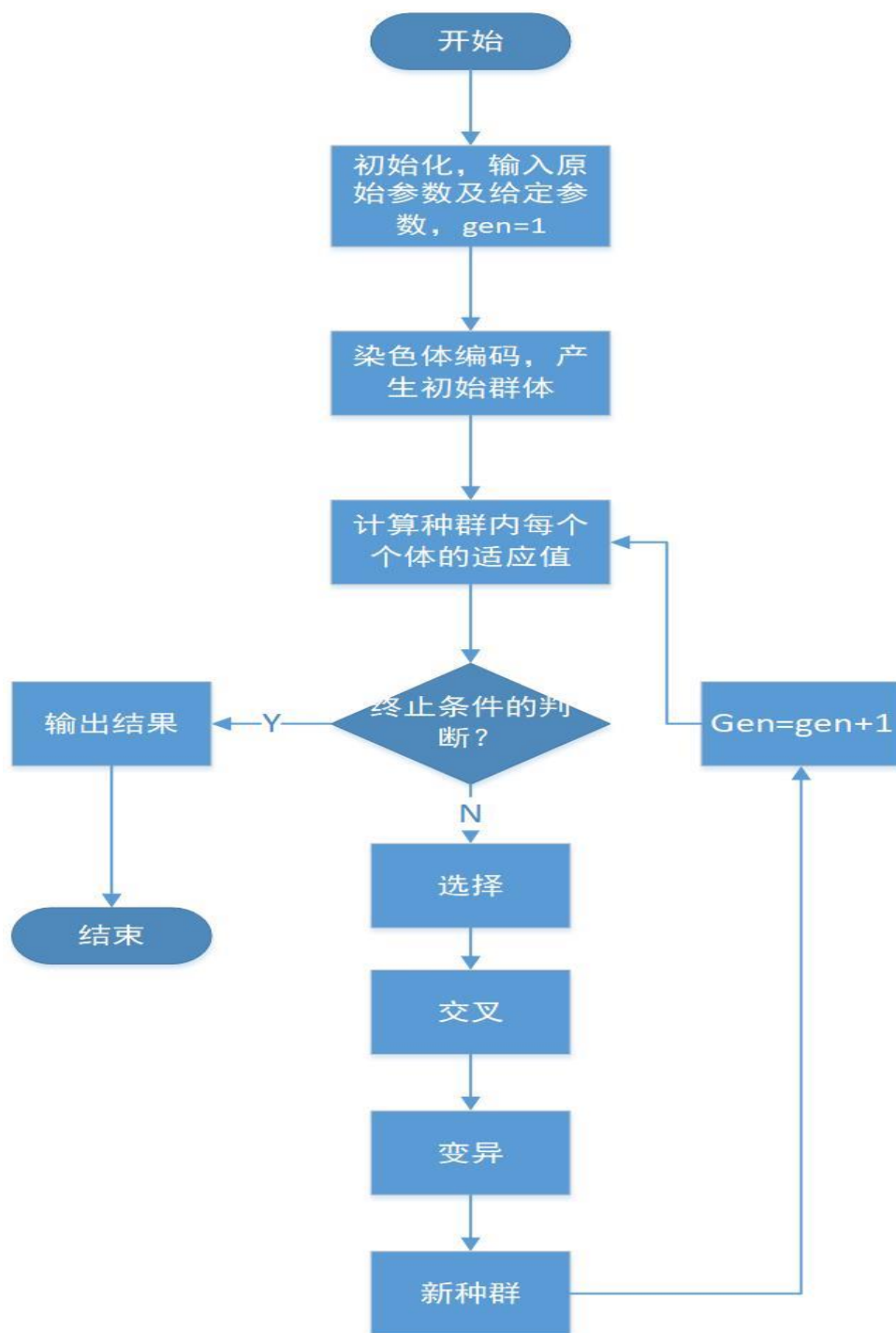


图 5, 遗传算法流程图

- a) 初始化: 设置进化代数计数器 $t=0$, 设置最大进化代数 T , 随机生成 M 个个体作为初始群体 $P(0)$ 。
- b) 个体评价: 计算群体 $P(t)$ 中各个个体的适应度。
- c) 选择运算: 将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体

中个体的适应度评估基础上的。

- d) 交叉运算:将交叉算子作用于群体。遗传算法中起核心作用的就是交叉算子。
- e) 变异运算:将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因值作变动。群体 $P(t)$ 经过选择、交叉、变异运算之后得到下一代群体 $P(t+1)$ 。
- f) 终止条件判断:若 $t=T$, 则以进化过程中所得到的具有最大适应度个体作为最优解输出, 终止计算。

5.1.3 具体解法及答案

第一步, 根据算法的实现, 小车无法到达存储仓位和拣选工位所在的节点, 所以我们规定算法中小车到达的地点为存储仓位和拣选工位的前一个节点。

第二步, 有些小车初始位置在存储仓位或拣选工位, 由于算法的问题, 小车在存储仓位和拣选工位的节点是无法移动的, 为了避免这种情况发生, 我们可以将这些特殊的起点所在的节点删去 (即将地图上相应障碍节点即数据中用 1 表示的点, 换成空白节点即数据中用 0 表示的点), 因为我们规定的目的地是存储仓位或拣选工位的前一个节点, 所以删去对最终的结果并没有影响。

第三步, 首先建立一个地图模型, 地图模型中包含了算法的实现需要得到实际的所有节点的坐标。根据已有数据, 我们可以创造出为 01 矩阵的地形图, 如果为 1 表示障碍物, 则 0 表示小车可移动的区域。其次, 根据分析和已有数据, 采用矩阵存储着关于小车的起始点、存储仓、拣选工位的信息, 即记录着程序运行的起始点、终止点。最后, 使用遗传算法和蚁群算法调用这些数据。

第四步, 经过分析可以发现, 如果让小车直接去离自己最近的一个存储仓位进行运送, 则可能出现不同的小车同时运送同一个存储仓位的货物的情况出现 (即出现抢货现象的问题)。这里我们可以通过将存储仓位进行分区来避免这种情况的发生, 即利用遗传算法将所有的存储仓位按照某种方式划分成 19 个区域 (此问题中不考虑小车相撞, 因此直接到达最近的拣选工位即可), 再将这 19 个分区通过蚁群算法分别分给相对较近的小车, 通过这种方法确定好目标, 即可避免上述可能出现的问题。

第五步, 关于是将空的托盘放到空存储仓位还是托盘回收处。可以这样假设,

根据上述分区的方式，小车运送的存储仓位都是相邻的，即大多数空存储仓位和下一个目标存储仓位的距离为 1（非相邻的因其影响很小，所以不予考虑），而小车先去托盘回收处，再到下一个目标存储仓位，其运动距离一定会大于 1。这种思路，显然是将空托盘放到存储仓位所算出的总路程更短。

第六步，要使用遗传算法和蚁群算法，首先建立一个地图模型，然后阐述该算法的执行情况（如图 2 与图 3 所示），具体如下：

G: 地形图为 01 矩阵，如果为 1 表示障碍物

SP: 起始点, 例: [1, 20]

EP: 终止点, 例: [20, 1]

K: 迭代次数（指蚂蚁出动多少波）

M: 蚂蚁个数（每一波蚂蚁有多少个）

Alpha: 表征信息素重要程度的参数

Beta: 表征启发式因子重要程度的参数

Rho: 信息素蒸发系数

Q: 信息素增加强度系数

ROUTES: 每一代的每一只蚂蚁的爬行路线

PL: 每一代的每一只蚂蚁的爬行路线

Tau: 输出动态修正过的信息素

要建立蚁群算法模型，先要构造启发式信息矩阵，用胞元结构存储每一代的每一只蚂蚁的爬行路线，用矩阵存储每一代的每一只蚂蚁的爬行路线长度，之后启动 K 轮蚂蚁觅食活动，每轮派出 M 只蚂蚁。

前提建立好后，下面进行具体的算法实施：

第一步，状态初始化。将爬行路线长度，禁忌表，爬行路线，邻接矩阵均初始化，再将当前节点初始化为起始点，并判断该起点的周围节点序号。

第二步，判断可以前往的节点。在这里我们设置一个觅食停止条件：蚂蚁未遇到食物或者陷入死胡同或者觅食停止

第三步，转轮赌法选择下一步怎么走。规定小车的运动方式，只能上下左右四个方向进行运动，并计算需要用到的距离（用来判断是否较优的解）。然后通过建立概率分布来确定下一步的目的坐标。

第四步，状态更新和记录。当蚂蚁移到下一个节点时，记录此时坐标，累加已走过的路径。然后将已访问过的节点从禁忌表中删除周围的点，然后统计再次可以

前往的节点。

第五步,记下每一代每一只蚂蚁的觅食路线和路线长度。

第六步,更新信息素。将更新量初始化,信息素挥发一部分,新增加一部分。

第七步,获得最优结果并记录。

第八步,计算坐标。

最后模型构建完毕之后,执行程序得到最终数据后,并对结果进行优化。

利用地图数据和小车的初始位置数据画出整体的一张初始地图,并将不同类的坐标用不同颜色表示。只需将 1 换成对应数字,再规定不同的颜色即可。现规定栅格中 0 为可移动区域,1 为不可移动区域,2 为存储仓位,3 为拣选工位,4 为回收空托盘处,如图 4 所示。

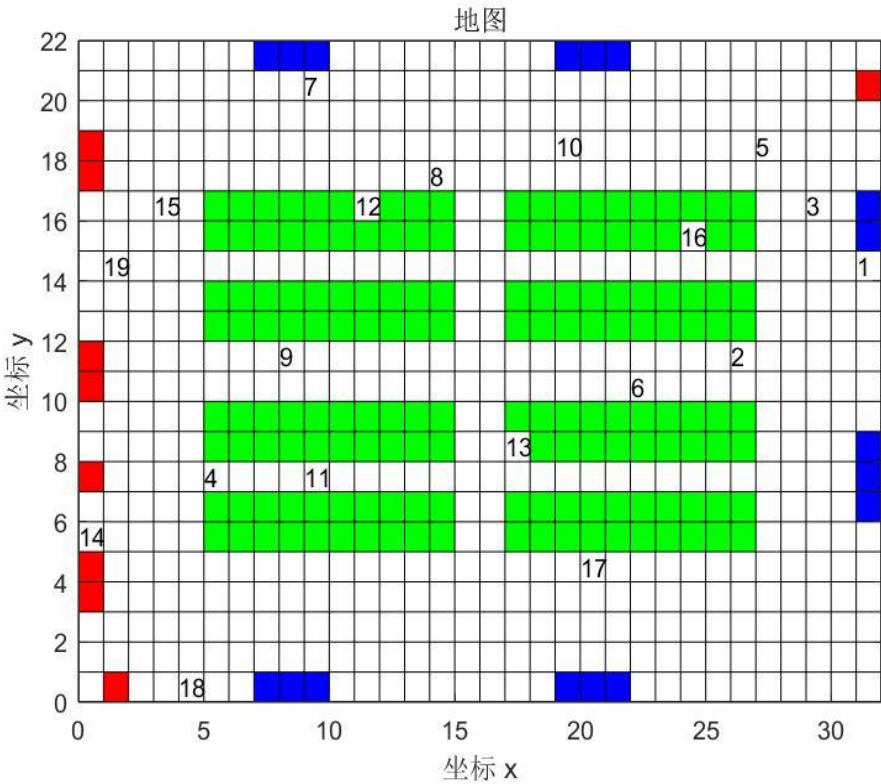


图 6, 地图及 AGV 坐标图

利用小车运行的路径数据和地图数据,画出实际的运行轨迹地图(如图),有三个步骤:第一步,画出彩色的地图;第二步,利用算出来的数据,画出最短爬行的路径;第三步,标出起点和终点。然后得出最后结果

5.1.4 最后结果

车号	1 号	2 号	3 号	4 号	5 号小 车	6 号小 车	7 号小 车	8 号小 车	9 号小 车	10 号 小车
单个 小车 单 次 路 程	11	11	13	5	7	5	7	8	1	3
	10	9	13	14	10	15	5	16	17	5
	10	8	12	15	9	14	10	15	18	4
	9	8	12	15	9	14	10	15	18	4
	9	7	11	16	8	13	4	14	18	3
	8	7	11	16	8	13	4	14	18	3
	8	6	10	17	7	12	3	13	19	4
	7	6	10	17	7	12	3	13	19	4
	7	9	9	11	11	16	4	19	20	10
	6	9	9	11	11	16	4	19	20	10
	6	8	8	12	10	15	5	18	21	11
	6	8	8	12	10	15	5	18	21	11
	6	7	6	11	9	14		17	20	12
	5	7	6	11	9	14		17	20	12
	5	6	5	12		13		16	19	13
		6	5	12		13		16	19	13
				13						
				13						
单个 小车 总 路 程	113	122	148	233	125	214	64	248	288	122
	11 号 小车	12 号 小车	13 号 小车	14 号 小车	15 号 小车	16 号 小车	17 号 小车	18 号 小车	19 号 小车	
	3	12	1	9	5	8	3	5	7	
	17	16	13	12	11	14	7	6	13	
	16	6	12	13	12	13	6	5	14	
	16	6	12	13	12	13	6	5	14	
	15	15	11	14	13	12	5	4	15	
	15	15	11	14	13	12	5	4	15	
	16	7	10	15	14	11	4	3	13	
	16	7	10	15	14	11	4	3	13	
	17	14	12	16	10	10	3	4	14	
	17	14	12	16	10	10	3	4	14	
	16	8	11	12	11	9	4	6	15	
	16	8	11	12	11	9	4	6	15	
	15	13	10	13	12	8	5	7	16	
	15	13	10	13	12	8	5	7	16	
	10			14		7	6	8		
	10			14		7	6	8		
单个 小车 总 路	230	154	146	215	160	162	76	85	194	总 路 程 : 3099

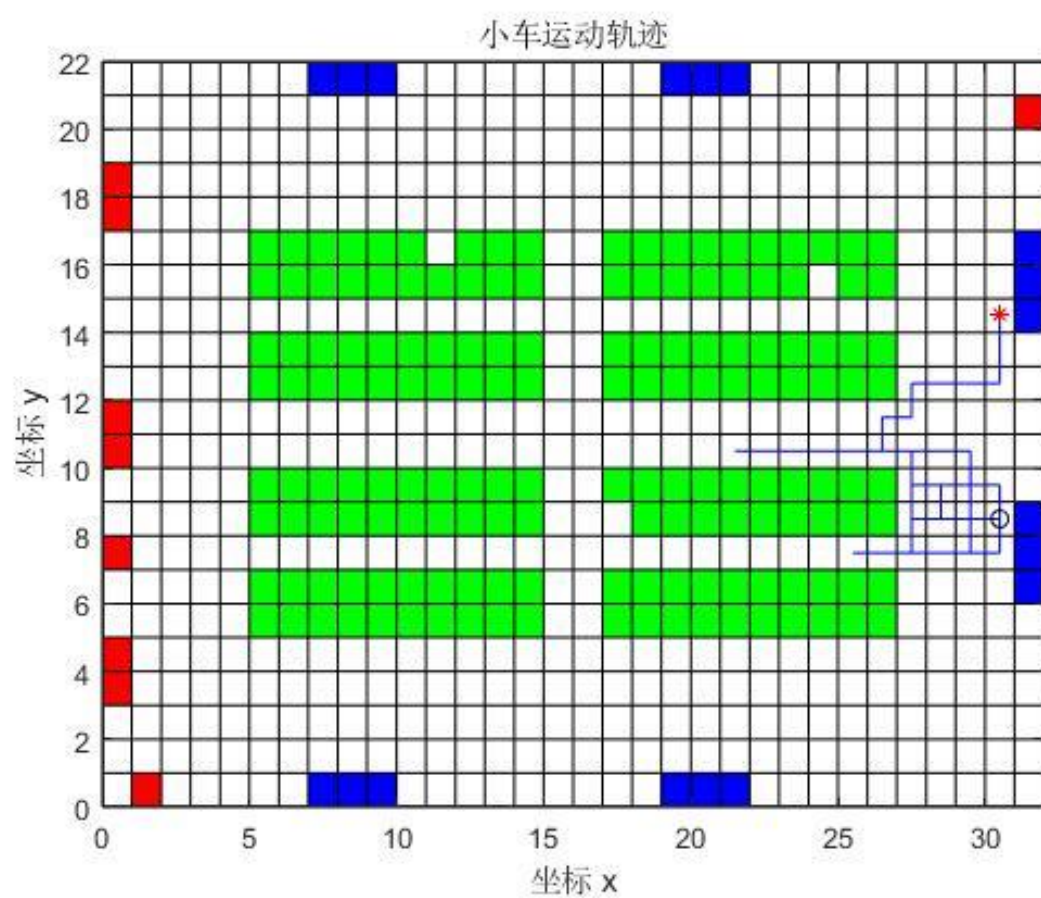
[illegible]

图 7, 小车运动轨迹例子

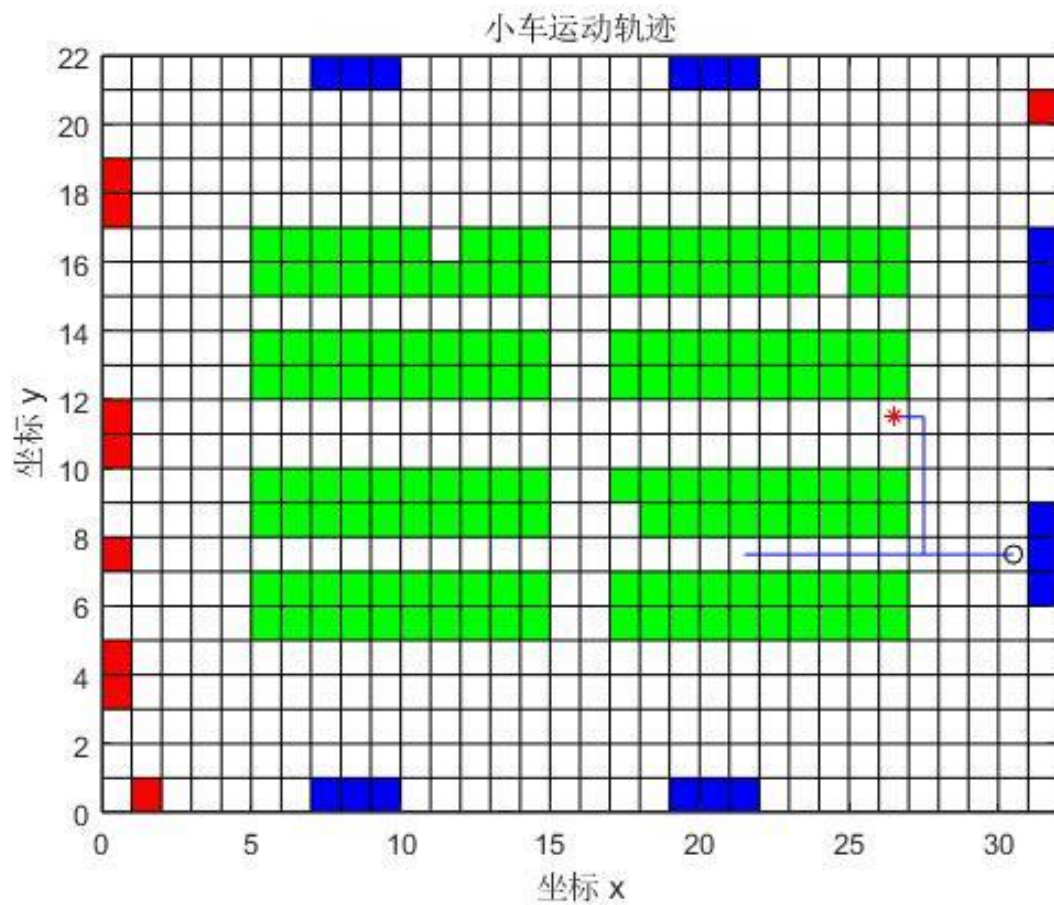


图 8，小车运动轨迹例子

5.2.1 问题二模型

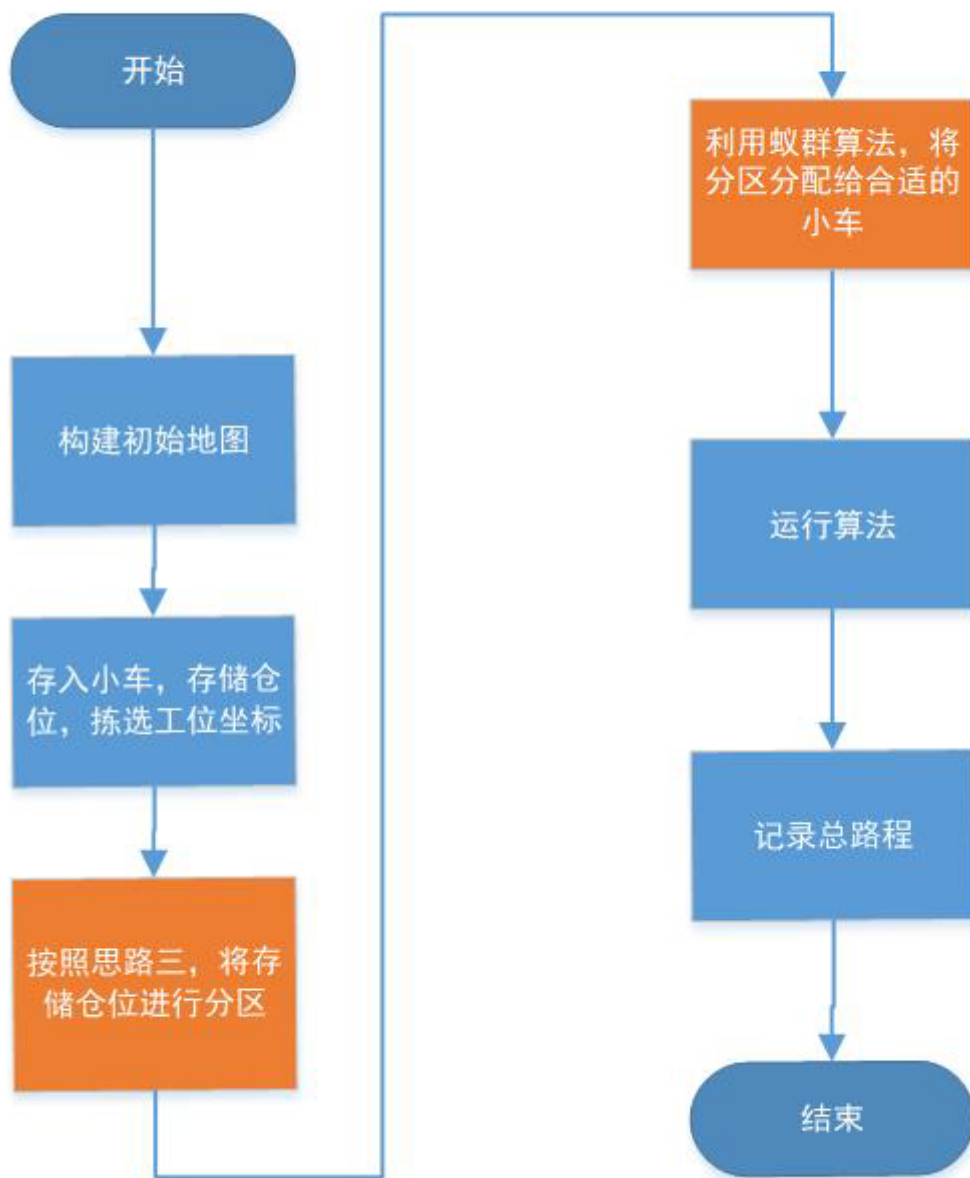


图 9，问题二解题流程

[2]问题 2 是在问题 1 的基础上增加 1 个目标和 1 个约束，目标是指使得任务区域均衡规划，约束是指储位上的每个托盘都制定了一个默认拣选工位，指每个托盘的后继节点约束 2 将 K 变为 $L_j \forall j \in J$ ，指每个托盘都有自己的默认节点，显然有

$L_j \subseteq K$ ，得到的新约束为：

AGV 节点集合：I

托盘节点集合：J

拣选工位节点集合：K

所有节点集合 $M=I \cup J \cup L_j$

定义决策变量：

$$x_{ij} = \begin{cases} 1, & \text{小车经过边 } (i, j) \\ 0, & \text{otherwise} \end{cases}$$

约束 1：每个 AGV 小车节点有且仅有一个后继节点，并且后继节点为托盘节点。

$$\sum_{j \in J} x_{ij} = 1, \forall i \in I$$

$$x_{ij} = 0, \forall i \in I, \forall j \notin J$$

约束 2：每个托盘节点有且仅有一个前继节点和后继节点，并且所有后继节点为拣货工位节点，所有前继节点为 AGV 小车节点。并且储位上的每个托盘都对应指定的拣选工位。

$$\sum_{i \in I \cup K} x_{ij} = \sum_{i \in L_j} x_{ji} = 1, \forall j \in J$$

约束 3：来到所有拣货工位上的车次必须等于离开拣货工位上的车次，等于托盘数。

$$\sum_{i \in I} \sum_{j \in J \cup \{o\}} x_{ij} = \sum_{i \in I} \sum_{j \in J} x_{ji} = n$$

目标是指每个拣选工位对应的托盘商品总量尽可能平均，将其转化为约束处理，即规定每个拣选工位处理商品总量受到上下限限制。

$$l \leq \sum_{j \in J} x_{ji} \leq u, \forall i \in K$$

参数 l 和 u 是根据平均商品总量所决定，计算每个区域内的商品总量与平均商品总量作比较，从而得出参数 l 和 u 。

5.2.2 算法分析

第二问与第一问的算法种类一样，不再赘述

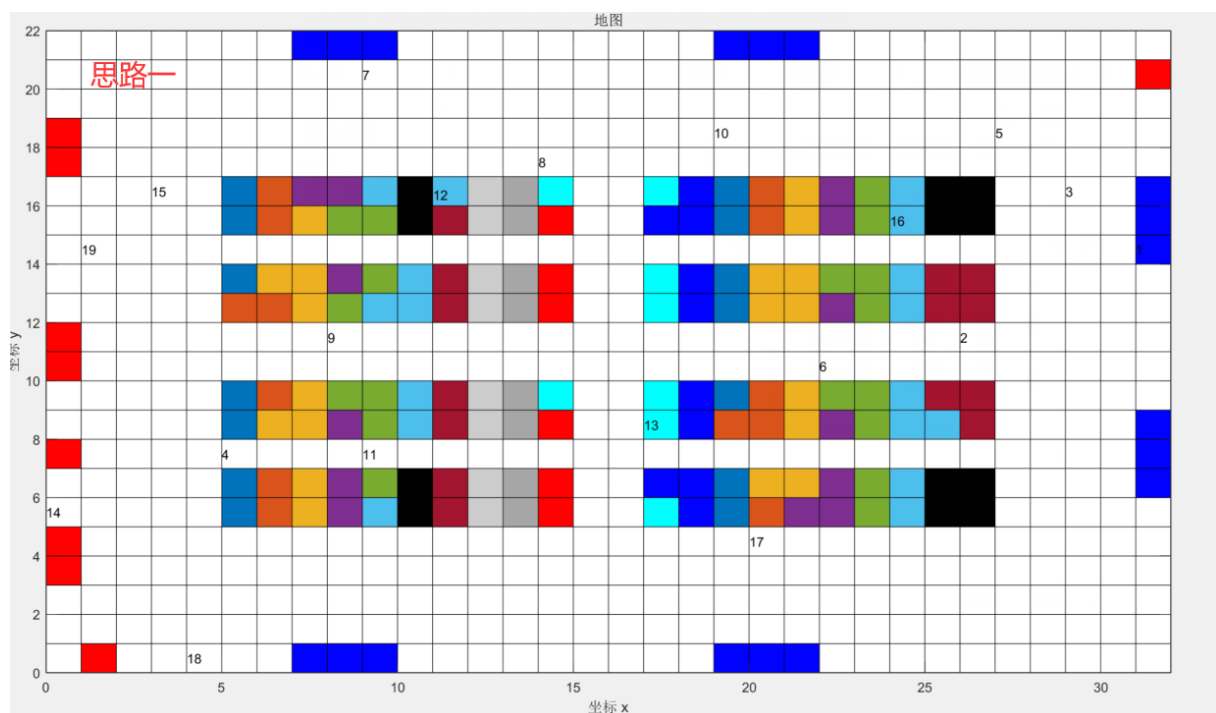
5.2.3 具体解法及答案

想要实现每个拣选工位的工作量相同，只需让小车送到每个拣选工位的商品数量相同。但在实际问题中，让到达每个拣选工位的商品数量相同几乎不可能，且按照这样算法也几乎无法实现。但如果我们把商品数量的值限定在一定范围内，让完全平均变成尽可能平均，那么实现起来就相对简单。

先只考虑商品数量尽可能平均，不考虑小车运行距离。

5.2.3.1 思路一

按照从上到下进行 s 形计算，累加得到商品总数 L，若 L 属于 P，则所选的存储仓位为集合 A，由此依次确定 A1, A2, A3……



图片 10，思路一示意图

5.2.3.2 思路二

按照从左到右进行 S 型计算 累加得到商品总数 L，若 L 属于 P，则所选的存储仓位为

集合 A，由此依次确定 A1，A2，A3……

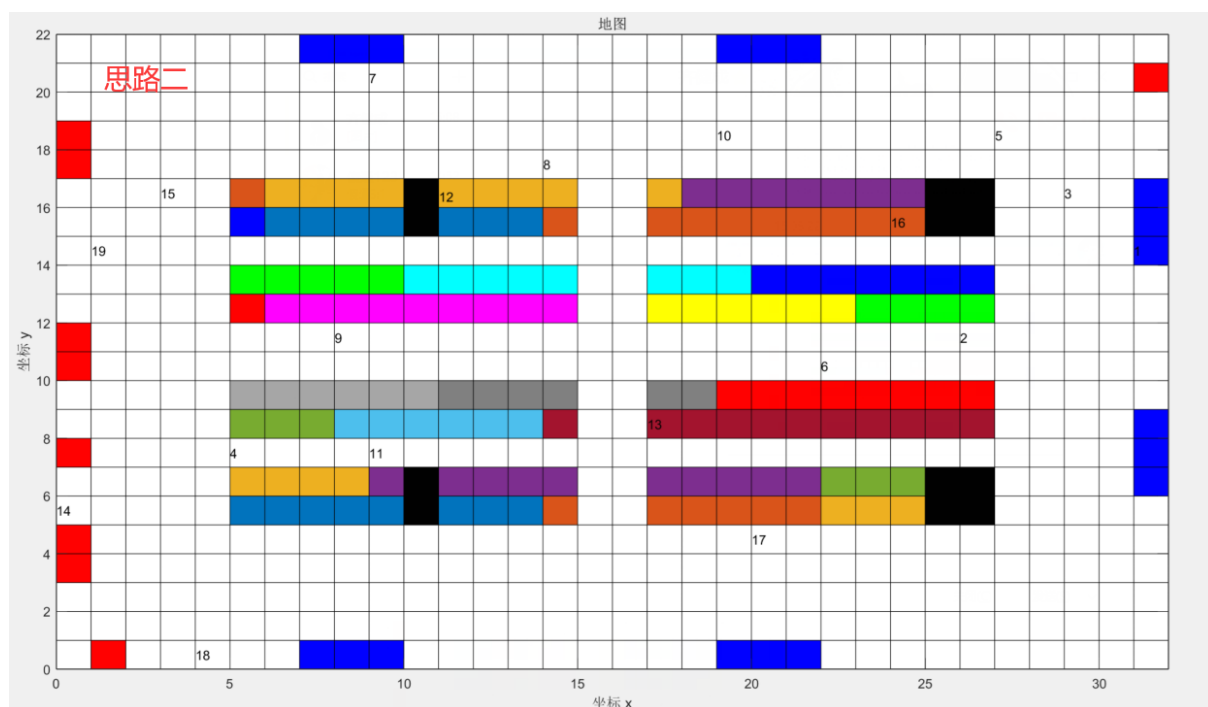
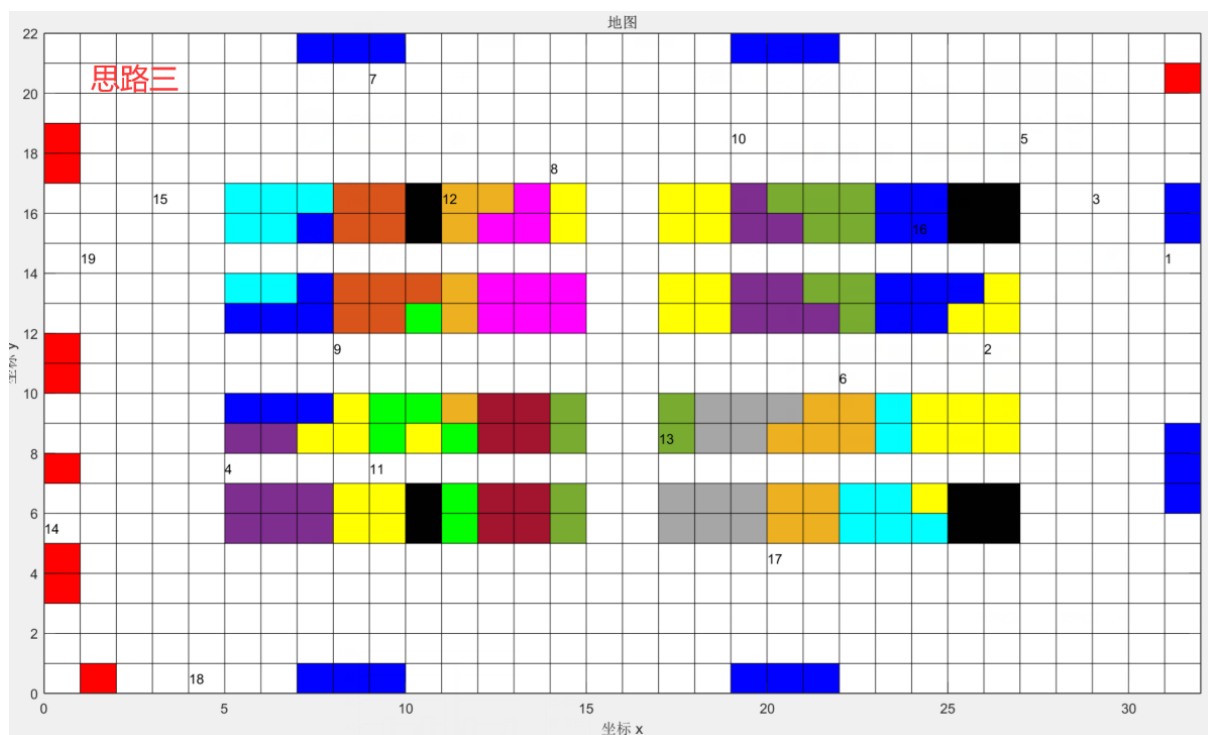


图 11， 思路二示意图

接下来考虑相对最近的小车距离

5.2.3.3 思路三

结合上述两种思路与第一题的结果，可以发现以块状分区得到的集合 A 较优。分块中每块的商品总数为 L，若 L 属于 P，则 L 为可行区域，由此确定了集合 A。



图片 12， 思路三示意图

以下是计算后的具体数据：

1 号小 车	2 号小 车	3 号小 车	4 号小 车	5 号小 车	6 号小 车	7 号小 车	8 号小 车	9 号小 车	10 号小 车
4	10	17	2	6	2	3	1	2	6
10	9	9	13	10	14	3	10	21	22
7	12	10	17	9	15	4	13	20	19
7	12	10	17	9	15	4	13	20	19
6	13	7	14	9	12	15	14	19	15
6	13	7	14	9	12	15	14	19	15
5	10	7	14	12	12	16	12	16	18
5	10	7	14	12	12	16	12	16	18
6	10	10	3	10	12	15	14	5	18
6	10	10	3	10	12	15	14	5	18
8	11	8	4	10	13	16	12	16	21
8	11	8	4	10	13	16	12	16	21
8	12	11	15	11	13	17	17	16	22
8	12	11	15	11	13	17	17	16	22
7	11			12	16	18	16		19
7	11			12	16	18	16		19
7	11					19	15		19
7	11					19	15		19
8									16
8									16
位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	位移总 和
138	199	132	149	162	202	246	237	207	362
11 号小 车	12 号小 车	13 号小 车	14 号小 车	15 号小 车	16 号小 车	17 号小 车	18 号小 车	19 号小 车	
5	1	2	6	3	1	1	12	7	
12	11	13	5	7	6	5	10	13	
12	10	14	4	6	7	6	9	14	
12	10	14	4	6	7	6	9	14	
15	14	16	3	5	5	7	13	14	
15	14	16	3	5	5	7	13	14	
9	14	17	10	12	10	14	12	15	
9	14	19	10	12	10	14	12	15	
11	17	18	10	12	9	13	13	15	
11	17	18	10	12	9	13	13	15	
14	18		11	13	10	12	12	16	
14	18		11	13	10	12	12	16	
			11	13	9		16	12	
			11	13	9		16	12	

			12				15	12	
			12				15	12	
位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	位移总 和	总位移
139	158	147	133	132	107	110	202	216	3378

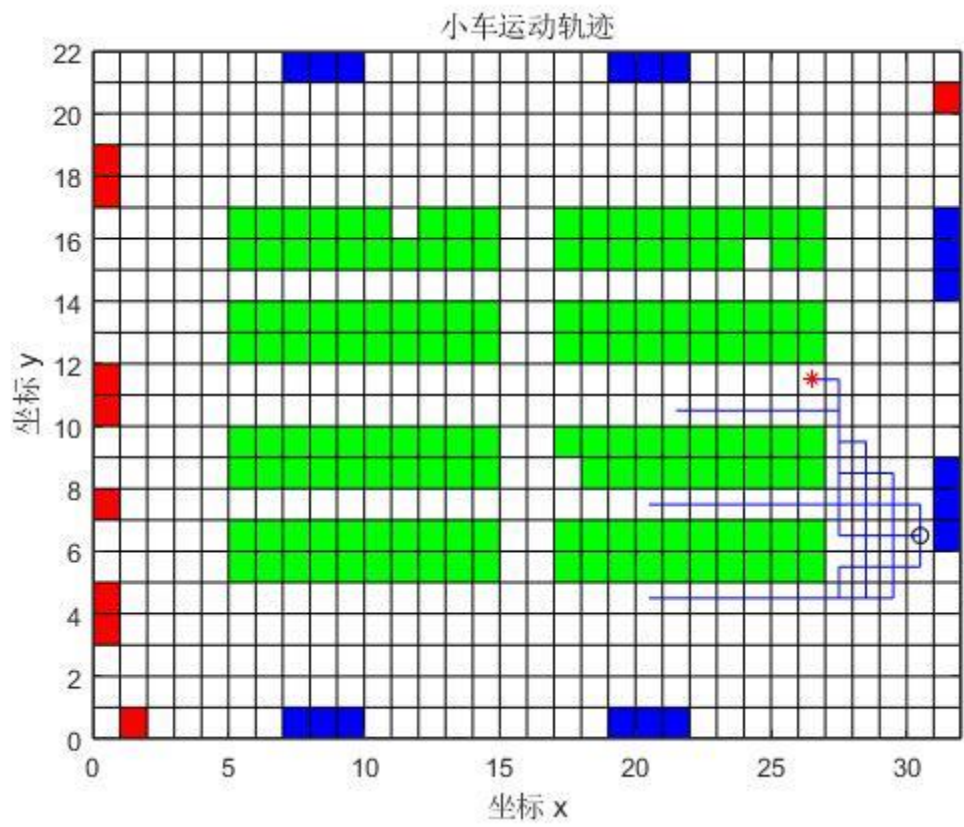


图 13, 小车运行轨迹例子

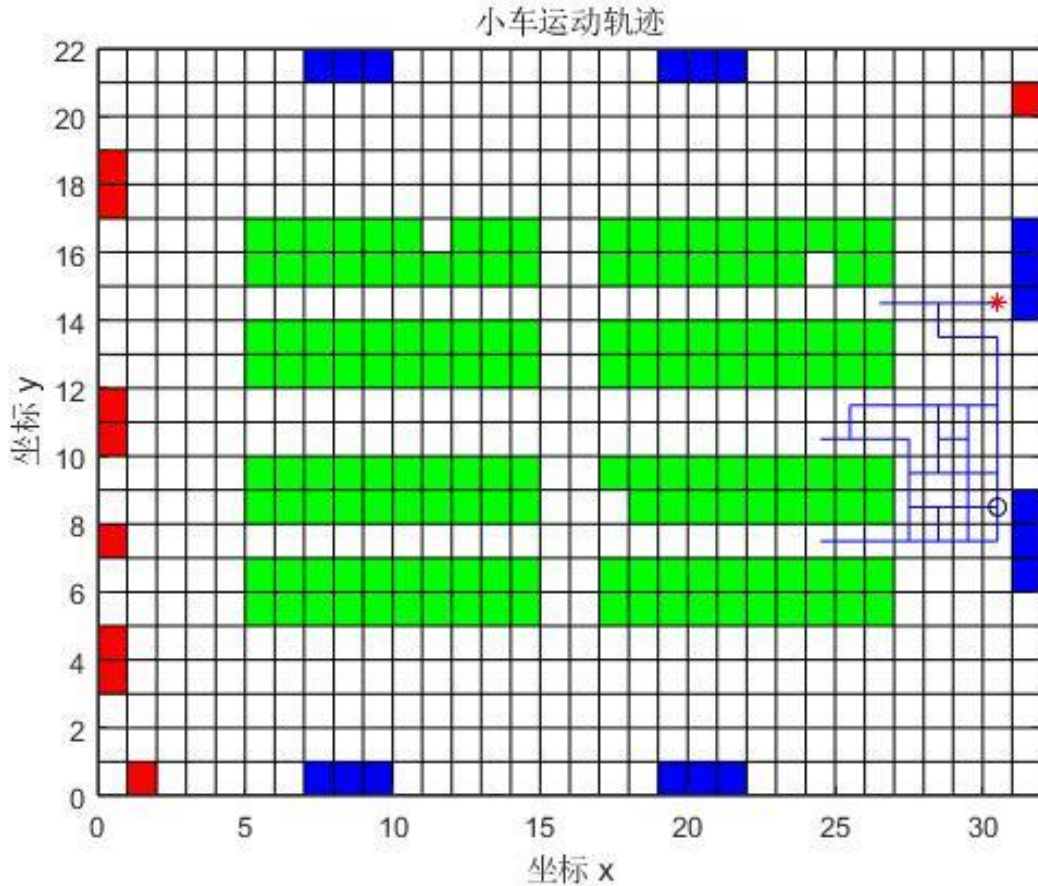


图 14, 小车运行轨迹例子

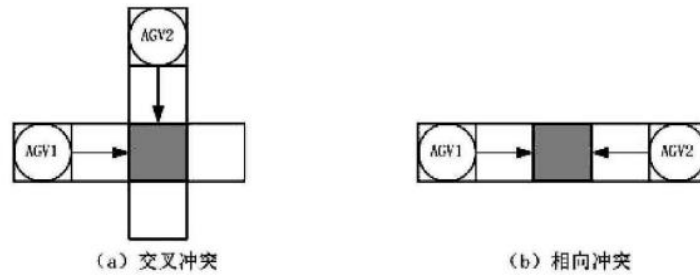
5.3.1 问题三模型[1]

- 1) 每台 AGV 占据一个栅格，单位移动速度等于栅格边长；
- 2) 每台 AGV 速度相同，且空载与载物速度一致，转弯时耗为常数；
- 3) 单位可行栅格的大小要确保 AGV 能够顺利通行，避免因栅格大小的因素对 AGV 的路径规划造成不必要的影响；
- 4) 每台 AGV 同一时刻只能接受一个任务，完成后方能执行下一任务；
- 5) 两台 AGV 之间规定最小安全距离，设置为一个车身长度。

多 AGV 避障路径问题描述

路径冲突是指在系统中，设某一 AGV 路径规划已完成时，其走过节点会有相应的时间窗，离另一 AGV 的规划路径有在某一时段的不同节点，称此节点为路径冲突节点。路径冲突情况复杂，包括静止冲突、交叉冲突、相向冲突、追击冲突，因为本文假设 AGV 速度俱定，且两车之间保持最小安全距离，然后主要研究如何避免动态障碍物，所以主要强调交叉冲突和相向冲突两种。对于该两种冲突（如图 12）的具体描述如下：

图 15，两种冲突示意图



1) 交叉冲突

如图 2(a)所示，由于每台 AGV 运行速度一致且平稳，每个栅格大小相等，因此 AGV 1 与 AGV 2 将要同时到达图中的阴影栅格中，即 AGV 将要在阴影栅格中发生路径冲突。这种现象称为多 AGV 系统路径规划中经常遇到的交叉冲突问题。

2) 相向冲突

如图 2 (b) 所示，AGV1 与 AGV2 相向行驶，由于一台 AGV 占据一个栅格，此路径只允许一台 AGV 通过，即不能同时供两台 AGV 通过，因此，两 AGV 若要继续前行到达阴影部分，则会发生路径冲突。这种现象称为多 AGV 系统路径规划中经常遇到的相向冲突问题。

基于弹性时间窗的改进模型

在蚁群算法的基本模型中，蚂蚁在搜索路径时是以所处栅格与相邻栅格直线距离的倒数为启发式信息，直线距离作为依据，不能好的反映 AGV 在实际路径中的运行时间；两点之间路径搜索盲目性较大，未知因素较多；信息素在更新时是依据第 k 只蚂蚁所留下的信息素量，信息素量相对较大，搜索速度慢。因此，为了缩短 AGV 收到任务后进行路径规划到完成任务的时间，针对传统蚁群算法中的这些不足，本文对算法部分作出的改进如下：

①为了提高算法搜索的精确性，提高蚂蚁对终点的可见度，本文以蚂蚁所处节点与

目标点的曼哈顿距离的倒数为启发式信息. 如式 (5) 所示

$$\eta(P) = \frac{1}{|(E(x) - P(x))| + |(E(y) - P(y))|} \quad (5)$$

注: E 代表终点栅格, P 代表蚂蚁所处栅格.

②为了减少不良路径信息素的干扰, 同时为提高蚂蚁搜索的速度, 本参考陈国良等专家通过设置蚁群信息素更新机制来避免传统蚁群算法收敛性差、局部最优缺点的思路

^[17], $\Delta\tau_{ij}^k(t)$ 改进为 选取第 k 只蚂蚁所留下的信息素中的最短路径作为信息素更新的依据. 如式 (6) 所示

$$\begin{aligned} \Delta\tau_{ij}^k(t) &= Q/L_k \\ L_k &= \min L \end{aligned} \quad (6)$$

在传统系统中, 在 AGV 即将发生碰撞时, 系统没有依据来决定 AGV 的先后通过顺序, 那么只能面临碰撞, 出现系统错误, AGV 将无法再继续运行. 因此, 为了 AGV 在路径规划过程中解决避碰问题, 本文对多 AGV 系统作出的改进: 对系统中的每台 AGV 进行任务优先级排序. 可以根据命令下达的时间顺序, 即 AGV 所承担订单任务的时间先后进行排序, 时间在前的任务优先级高, 时间在后的任务优先级低, 即将发生碰撞时, 任务优先级高的优先通过, 任务优先级低的相应采取等待或绕行.

基于 AGV 任务优先等级排序后, 可有效解决路径冲突问题, 面对不同类型的冲突, AGV 可采取的策略如下:

①当在某一节点发生路径交叉冲突的时候, 解决的方法可以通过等待策略解决, 任务优先级低的 AGV 停车等待, 优先级高的 AGV 先行通过.

② 当在某一节点发生路径相向冲突的时候, 等待策略无法解决冲突, 则采取的策略是对任务优先级低的 AGV 重新进行路径搜索, 寻找避碰前提下的最优路径.

实施步骤

1) 环境建模. 建立仓储环境栅格模型, 根据任务优先级顺序依次给 AGV 分配任务, 确定已分配任务 AGV 的目标节点, 对承载任务优先级最高的 AGV 首先进行路径搜索, 初始化每个节点的时间窗表和其它信息参数.

2) 路径初步规划. 设置迭代计数器 $G = 0$, 将 M 只蚂蚁放在起始位置上, 并把初始点加入到禁忌表中, 根据转移概率公式计算出蚂蚁 k 所有可转移节点的概率, 根据轮盘赌规则选择下一节点, 更新禁忌表, 存储所有蚂蚁的所寻路径, 并保存路径长度, 删除死亡蚂蚁, 默认其路径长度为无穷大; 根据信息素更新公式进行全局信息素更新, 并清空禁

忌表;判断G是否等于最大迭代次数,若是,流程结束,搜索出该AGV的最优路径;若否,重复进行上述步骤.

3) 计算时间窗. 计算该AGV经过节点的时间窗,更新每个节点的时间窗表. 按顺序搜索下一辆AGV的最优路径,并计算出该AGV所经过节点的时间窗,更新所有节点的时间窗表,比较每个节点是否发生时间窗冲突,如果没有冲突,则进行下一辆AGV的路径规划,如果有冲突,则将冲突节点加入禁忌表,则解决时间窗冲突,并进行再次比较.

4) 重新搜索. 如果在进行路径规划的过程中发生与相邻节点在时间窗上均发生冲突而导致任务无法执行的情况,就通过扩大时间节点的搜索范围,重复进行新一轮的搜索.

5.3.2 分析方法

5.3.2.1 动态分析

对于动态分析,由于AGV之间会有相互的作用,所以对于每一个单一的AGV而言,其行动路线会受到其他AGV的影响,所以考虑使用动态分析,在问题二AGV分布的基础上,其中一个AGV1每移动一个栅格便进行重新的路径最优规划,其他AGV在相同的时间内也移动一个栅格距离,而此时其他的AGV相对于AGV1而言是无法通过的栅格,即地图G所代表的矩阵中的1。每次每个AGV都通过运算获取每一步之后的最优解。从而达到全局最优解

5.3.2.2 优先级分析

考虑准备使用等待策略来进行避让以及防撞功能的实现,因此,哪一个AGV进行避让变通的尤为重要,通过对每个AGV赋予优先级并且固定,即不管AGV上的货物为什么,AGV之间的避让只考虑AGV自身所携带的优先级,优先级别高的有限进行路径行进,优先级别低的AGV先等待较高优先级的AGV通过后再通过

5.3.2.3 等待策略分析

考虑到不同的AGV之间可能会有冲突,如果一起前进而不进行避让的话,势必会造成AGV的无法使用。因此我们考虑通过等待策略来进行避障处理。

5.3.2.4 禁忌搜索算法[3]

接下来，我们建立禁忌算法模型

第一步，在搜索中，构造一个短期循环记忆表（禁忌表），禁忌表中存放刚刚进行过的 $|T|$ （ T 称为禁忌表）个邻居的移动，这种移动即解的简单变化。

第二步，我们规定禁忌表中的移动称为禁忌移动。对于进入禁忌表中的移动，在以后的 $|T|$ 次循环内是禁止的，以避免回到原来的解，从而避免陷入循环。 $|T|$ 次循环后禁忌解除。

第三步，判断小车进行 n 次移动后状态，如 $n < |T|$ ，那么小车不会陷入循环，如 $n \geq |T|$ ，那么小车将进入循环状态，始终保持 $|T|$ 个移动。

5.3.3 具体解法及答案

这里建立一个前提，小车所处坐标也是障碍物。

首先，为了避免相撞，我们需要知道哪辆小车先停止运动，这里我们引入优先级判断。为了方便算法的实施，我们规定按照小车的编号来排序，即 1 号小车优先级最高，19 号小车优先级最低。

其次，需要提前判断是否会相撞。这里我们以小车的中心为参照，当两辆小车间的距离小于或等于 2 时，即接下来会相撞，这时，优先级低的小车等待，优先级高的小车进行路径判断。

以下是 Car3.m 的具体实施步骤：

第一步，加入所有初始数据数据，如小车初始位置，目标位置。

第二步，运行蚁群算法，进行路径规划。

第三步，每运行一个栅格，利用禁忌算法进行防撞判定，优先级高的先运行，优先级低的进行等待。

第四步，记录小车运行过程并计算总时间，总路程。

第五步，存储数据。

对于最终的结果，由于蚁群算法的随机性，获得的结果也有一定随机性。其中有较优的解，也有较差的解，甚至由于优先级判定导致被其他小车阻碍行动而产生死循环。以下为第一种解：

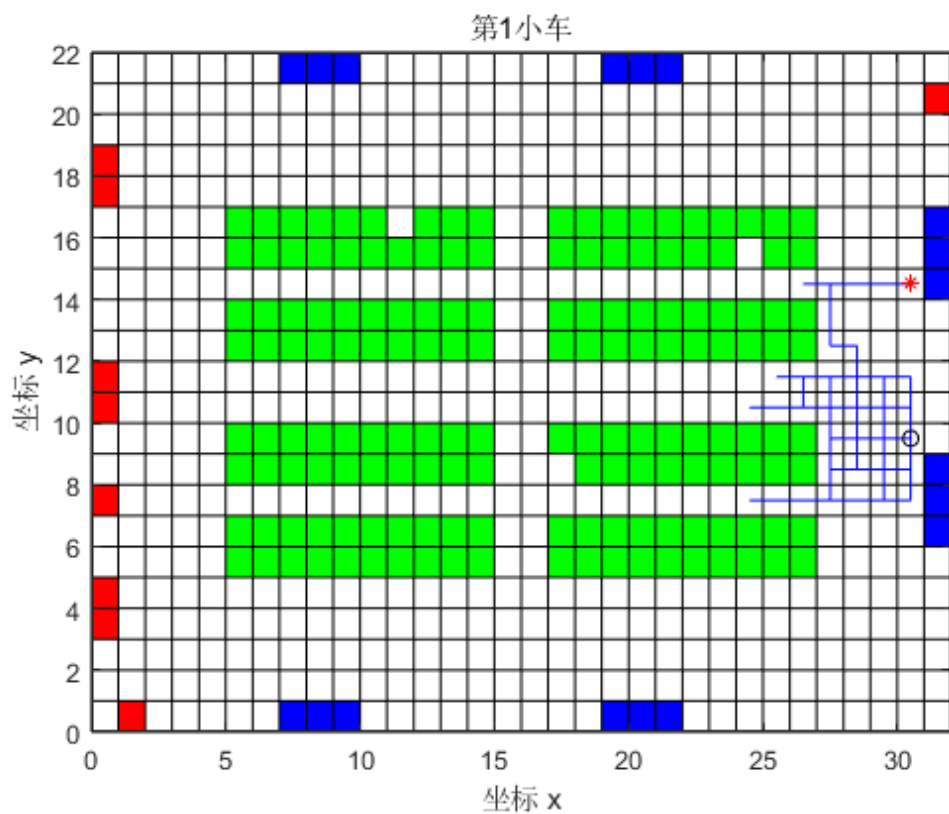


图 16， 第一种解法轨迹

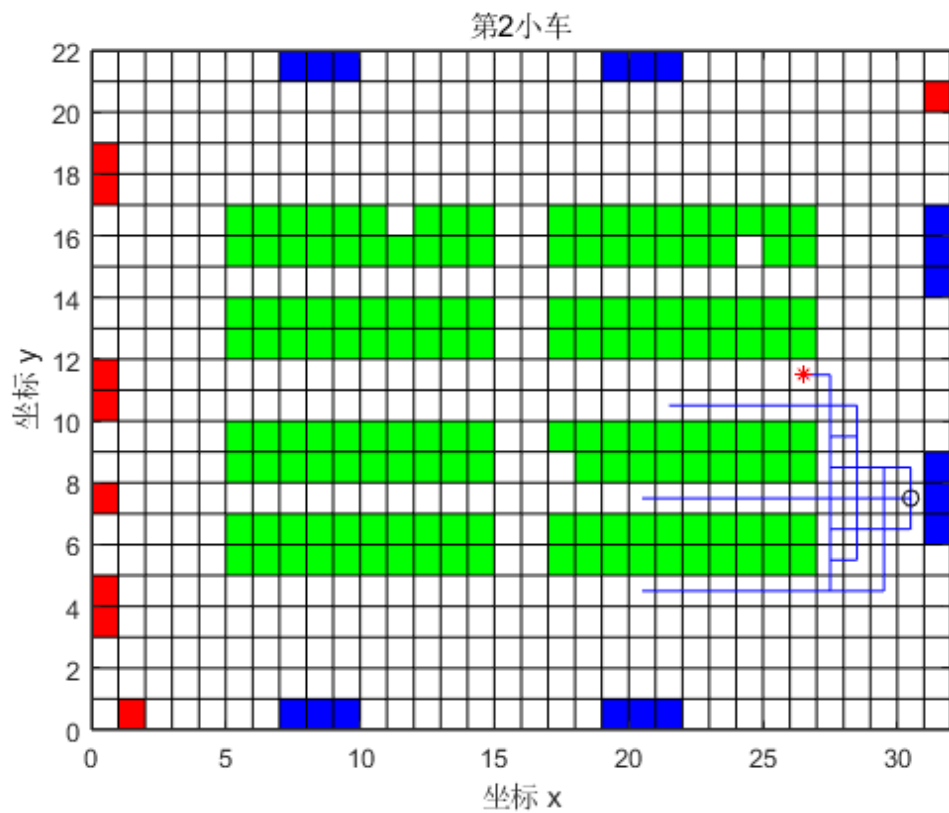


图 17， 第一种解法轨迹

以下是运行结果

序号	1	2	3	4	5	6	7	8	9	10
路程	138	199	132	149	162	202	248	237	207	362
时间	138	199	139	149	162	205	248	237	207	363
目的 总数	21	19	15	15	17	17	19	19	15	21
等待 时间	0	0	7	0	0	3	0	0	0	1
序号	11	12	13	14	15	16	17	18	19	
路程	139	158	145	133	132	107	110	202	216	总时
时间	140	160	153	137	133	113	119	218	217	363
目的 总数	13	13	11	17	15	15	13	17	17	
等待 时间	1	2	8	4	1	6	9	16	1	

由于小车运行的时间不同，根据木桶原理取最后一个结束的时间，为 363。

以下为第二种解

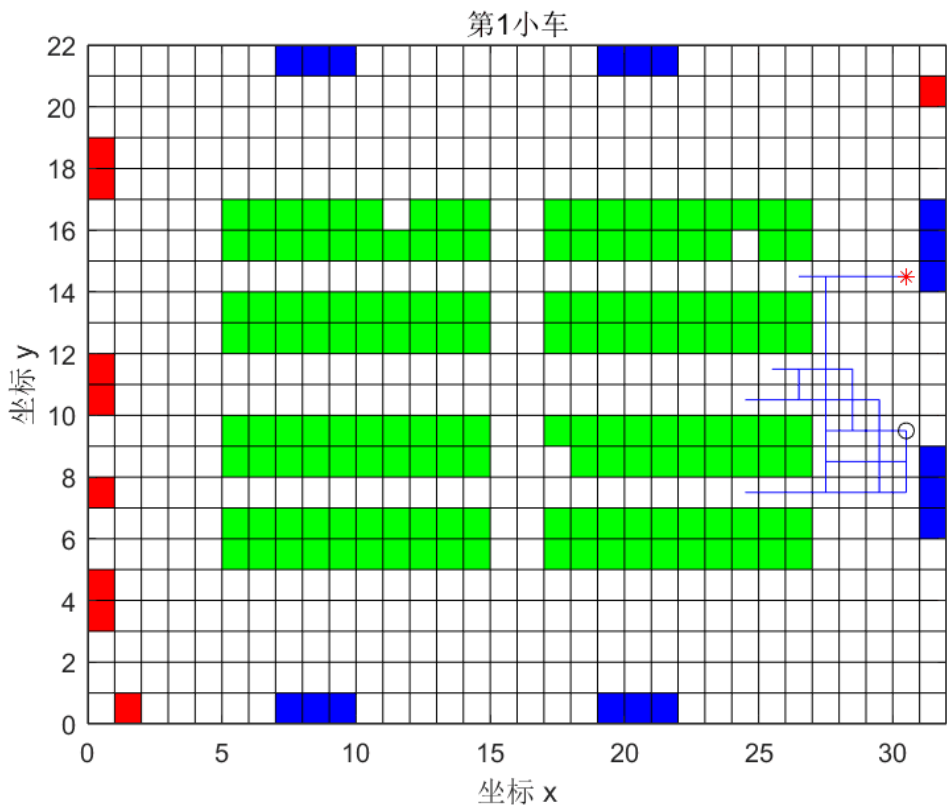


图 18， 第二种解法轨迹

以下是运行结果

序号	1	2	3	4	5	6	7	8	9	10
路程	138	199	132	149	162	202	246	237	207	362

图 19，第二种解法轨迹

小车编号	运行时间	已经运行了几个目的地	目的地的总数目
1	138	21	21
2	199	19	19
3	139	15	15
4	149	15	15
5	162	17	17
6	208	17	17
7	246	19	19
8	237	19	19
9	207	15	15
10	363	21	21
11	140	13	13
12	159	13	13
13	154	11	11
14	141	17	17
15	133	15	15
16	118	15	15
17	118	13	13
18	140731	14	17
	221	17	17

运行时间	运行状态（即 0 表示完毕, 1 表示未完毕）	已经运行了几个目的地	目的地的总数目
138	0	21	21
199	0	19	19
139	0	15	15
149	0	15	15

162	0	17	17
208	0	17	17
156	0	19	19
237	0	19	19
207	0	15	15
363	0	21	21
140	0	13	13
162	0	13	13
155	0	11	11
143	0	17	17
133	0	15	15
110	0	15	15
117	0	13	13
78105	1	14	17

运行时间	运行状态（即 0 表示完毕, 1 表示未完毕）	已经运行了几个目的地	目的地的总数目
138	0	21	21
199	0	19	19
139	0	15	15
149	0	15	15
162	0	17	17
207	0	17	17
248	0	19	19
238	0	19	19
207	0	15	15
364	0	21	21
140	0	13	13
163	0	13	13

155	0	11	11
139	0	17	17
133	0	15	15
109	0	15	15
119	0	13	13
2835	1	14	17
217	0	17	17

结果可以发现，所有死循环出现的地方都是第十八号小车，且都是运送了 14 个以后出现问题，说明 18 号小车的路径仍有可优化的地方，但相对于第一问、第二问已经有了巨大的提升。

6. 模型评价

本题为多 AGV 路径规划问题，在问题求解中需要考虑多种因素，本文仅对其中某些方面进行了研究和探讨，仍有诸多问题有待进一步解决，模型分析内容如下：

(1) 对于问题的求解建立了最佳模型，做出多种假设，使得求解更加清晰、简单，减少了网络复杂度。

(2) 本题求解使用蚁群算法和遗传算法，对于问题的适应性较强，对于问题的约束条件有更好的求解办法，但其缺点也十分明显，使用此算法运行时间较长，收敛速度较慢。

(3) 所建模型可拓展性较强，适应多种约束条件求解。

(4) 问题三设计了基于弹性时间窗的改进模型，提高了算法的精准性，避免了不良路径信息素的干扰，能够根据优先级前后判断解决碰撞与死解问题。

(5) 本文的问题求解主要以运筹学的思路来解答，提出了运筹优化模型，建立了较优模型。

7. 参考文献

[1] 杨洋, 张建敏, 刘艺林, 宋馨 “基于改进蚁群算法的无人仓的多 AGV 避碰路径优化

策略” [基于改进蚁群算法的无人仓的多 AGV 避碰路径优化策略 - 中国知网 \(cnki.net\)](#)

2022.04.14

[2] 郭超, 陈香玲, 郭鹏, 王强, 汪世杰. “基于时空 A*算法的多 AGV 无冲突路径规划”.

[基于时空 A*算法的多 AGV 无冲突路径规划 - 中国知网 \(cnki.net\)](#) 2022.04.15

[3] 杨洪涛, 郑之恒, 李英德, 陈青丰, 江伟光 “采用 AGV 分拣的型材下料车间成组调度问题研究” [采用 AGV 分拣的型材下料车间成组调度问题研究 - 中国知网 \(cnki.net\)](#)

2022.04.15

[4] 雷斌, 金彦彤, 王致诚, 赵蕊, 胡方鑫” 仓储物流机器人技术现状与发展” [仓储物流机器人技术现状与发展 - 中国知网 \(cnki.net\)](#) 2022.04.16

[5] 温正 《精通 MATLAB 智能算法》清华大学出版社 2015 年 4 月

[6] 王源 “无人仓的小车调度问题具体分析”

https://www.zhihu.com/question/527666985/answer/2439768816?utm_source=qq&utm_medium=social&utm_oi=1162684881497952256 2022.04.17

8. 附录

8.1 问题相关数据

8.1.1 问题一相关数据参考: “问题一相关数据.xlsx”

8.1.2 问题二相关数据参考: “问题二相关数据.xlsx”

8.1.3 问题三相关数据参考: “问题三相关数据.xlsx”

8.2 主要软件

MATLAB、Excel、Word