队伍编号	900342
题号	B

环形穿梭车系统的设计与调度

摘 要

本文主要研究的是环形穿梭车系统的最优调度问题。随着科技的不断发展，自动化技术的成熟，智能化仓库的搭建面临严峻的机遇与挑战，环形穿梭车系统作为智能化仓库的重要工具，其系统设计与调度直接关系着企业运输效率，是提升企业竞争力的内在要求[1]。所以，研究环形穿梭车最优调度系统具有重要意义。

首先对现有的穿梭车系统进行分析，从穿梭车的任务分配，延误原因分析建模，再通过单目标优化求出总任务最小完工时间。其次，再考虑穿梭车长度的因素，利用单目标优化求出总任务最小完工时间。再利用穿梭车延误时间、穿梭车数量变化两个指标，对环形穿梭车系统运行效率进行评价。最优通过进出货口的位置、轨道长度、最优穿梭车数量、最小延误时间得到最优的参数优化设计。

针对问题一，是研究总完工时间最小的问题。通过仔细分析，将问题一分成了三个模型进行建模，首先确定任务执行顺序，建立**了穿梭车任务规划模型**，得到了最优的任务执行顺序，见附录表一。再通过分析得出影响任务完成时间最主要的因素是穿梭车装卸货时造成延误，于是建立**穿梭车延误分析模型**，得到了计算穿梭车延误时间的模型。最后，建立**总完工时间最小单目标优化模型**，通过 MATLAB2013b 动态仿真，Lingo17.0 求解单目标优化，得到了穿梭车数量为 3,6,9 时，对应的最小完工时间分别是 10700 秒，5437 秒，3583 秒。

针对问题二，是在问题一的基础上，多考虑穿梭车长度下的最优最小时间，由于其他条件与第一问相同。通过分析，在问题一的模型上，对穿梭车之间的最短距离调整为穿梭车的长度，从而建立了考虑穿梭车长度的**最小时间单目标优化**模型，通过 Lingo17.0 求解单目标优化，得到了穿梭车数量为 3,6,9 时，对应的最小完工时间分别是 11204 秒，5786 秒，4053 秒。

针对问题三，是研究在表 1 的系统参数下，对环形穿梭车系统运行效率进行研究。首先，根据分析环形穿梭车系统效率的额主要因素有穿梭车的数量，以及穿梭车的行驶延误时间作为主要的评价指标，并建立了**环形穿梭系统评价模型**，分别对两种评价指标的影响作用进行分别建模，再通过评价模型得到了两种指标合力作用下影响的系统效率，并与穿梭车数量为 1 时的效率进行对比，得到了穿梭车数量为 3,6,9 时，对应的系统总体效率分别为对比参照效率的 2.98 倍，5.86 倍，6.82 倍。

针对问题四，是对环形穿梭车系统进行参数优化设计，故建立了**基于多目标优化的环形穿梭车系统参数优化模型**，目标函数分别从 A，B 侧**进出货口的位置，轨道长度，穿梭车数量，复合作业次数最多**进行优化求解，得到最优参数 A 侧轨道长度 60m，B 侧轨道长度：105m，弯道长度：30m，最优穿梭车数量为 8 辆以及进出货口的位置(见问题四求解)。

最后，对模型进行了优缺点分析，对模型做了一些改进和推广。

关键词： 穿梭车系统；多目标优化；任务优化；动态仿真；

目 录

一、 问题重述 1
1.1 问题背景 1
1.2 需解决的问题 1
二、 问题分析 1
2.1 问题的重要性分析 1
2.2 问题的整体分析 2
2.3 问题的关键分析 2
2.4 问题的具体分析 2
三、 符号说明 3
四、 模型假设 3
五、 数据预处理 4
六、 模型的建立与求解 5
6.1 总完工时间最小的单目标优化——问题一 5
6.1.1 模型准备 5
6.1.2 建模思想 5
6.1.3 模型建立 5
6.1.4 建模思想 7
6.1.5 建模准备 8
6.1.6 模型建立 8
6.1.7 建模思路 9
6.1.8 模型建立 9
6.1.8 算法设计 10
6.2 考虑穿梭车长度的最小时间优化模型——问题二 12
6.2.1 建模思路 12
6.2.2 建模准备 12
6.2.3 模型建立 12
6.2.4 建模思路 13
6.2.5 模型建立 14
6.2.6 算法设计 15
6.3 环形穿梭系统效率评价模型——问题三 16
6.3.1 模型思路 16
6.3.2 模型建立 17
6.3.3 算法设计 18
6.4 基于多目标优化的环形穿梭车系统参数优化设计——问题四 18
6.4.1 建模思路 18
6.4.2 模型建立 19
6.4.3 模型计算及结果 21
七、 模型评价 22
7.1 模型的优点 22
7.2 模型的缺点 22
八、 模型推广 22
九、 参考文献 22

一、问题重述

1.1 问题背景

本文研究的是环形穿梭车系统的最优设计与调度的问题。随着科技的不断发展，工厂机械化的逐渐成熟，物流行业面临着更多的考验，利用优化计算与信息技术，对穿梭车的调度进行优化管理，可以降低更多的生产成本，节约更多的运输时间[2-3]。

利用穿梭车系统将货物从指定进货口运送到指定的出货口，实现货物的出库。当货物较多时，有限的穿梭车数量就很难满足快速出货的要求，因此合理地对出入库运输系统的路线进行优化调度就显的非常的重要[4]。如果路线调度不合理，就可能会出现大规模的运输堵塞现象，更严重可能会造成系统崩溃[5]。

设计一类优秀的环形穿梭车系统，能够将仓库管理系统与电气控制系统更加有效地结合，在现有的研究里，环形穿梭车有准确度高，精度好，速度快的特点，相对于原有的机械运输方法来看，利用穿梭车系统可以在原有出库效率上增加 25%[6-7]。所以利用环形穿梭车系统是现代机械运作中较优的选择。但是，如果没有合理的穿梭车分配方案，会增加堵塞时间以及减少运行效率，这是当今环形穿梭车系统面临的最大难题，值得对该问题进行更加深入的研究[8-9]。

总而言之，研究环形穿梭车系统的最优设计与调度问题是具有重大意义，这对节约仓库管理成本与时间，使企业更具有国际竞争力的重要手段。

1.2 需解决的问题

问题一，在不计穿梭车实际长度的情况下，建立一种调度 N 辆穿梭车来完成各个进货口中待处理货物的数学模型及相应的求解算法，并且所得的总完工时间最小。此外，在表 1 的参数下，在 $N=3,6,9$ 时，计算处理货物所需的时间。

问题二，在考虑穿梭车实际长度的情况下，建立一种调度 N 辆穿梭车来完成各个进货口中待处理货物的数学模型及相应的求解算法，并且所得的总完工时间最小。此外，在表 1 的参数下，在 $N=3,6,9$ 时，计算处理货物所需的时间。

问题三，在表 1 系统参数条件下，建立数学模型，对环形穿梭车系统运行效率进行评价。

问题四，对此环形穿梭车系统进行参数优化设计，并提出实际可行的改进建议。

二、问题分析

2.1 问题的重要性分析

在对穿梭车系统进行优化的研究中，虽然已经取得了一定的进展，但是由于科技的高速发展，穿梭车系统的复杂性，高效率的要求也越来越高，以往的任务调度安排效率依旧比较低，对于不同企业的要求，我们需要根据具体问题进行具体分析，以寻求更佳的解决方案，主要对任务运输路径进行优化，以及对穿梭车执行对应任务进行建模，以减少其堵塞所造成的时间上的延误，从而提高整个仓库的调度效率，对于提高企业的管理水平，以及企业的整体竞争力，有着重要作用。

2.2 问题的整体分析

从整篇文章来看，是对环形穿梭车系统进行设计以及调度优化的问题。第一问，在不计穿梭车实际长度的情况下，建立一般化的调度 N 辆穿梭车的数学模型，并且求出总完工时间最小的时间，第二问，是在第一问的基础上，考虑穿梭车长度的情况，对第一问的模型进行扩充修改，得到考虑穿梭车长度的最小时间。第三问是在表一的参数下，对穿梭车效率进行评价，最后，进行参数优化设计，提出实际可行的改进建议，使得改进后的效率最优。

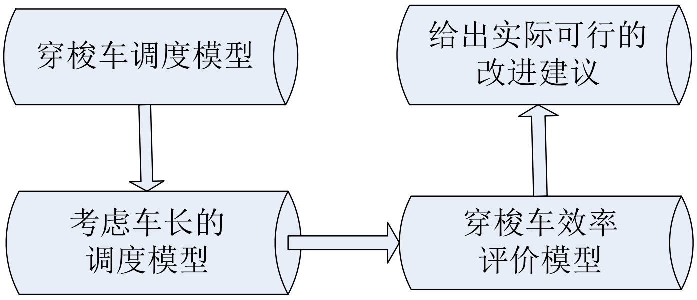

图 1 整体分析图

从图中可以看出，给出的问题是层层递进的，问题与问题之间有着紧密的相关性。

2.3 问题的关键分析

◆ **问题一中完工最小时间的转化**

通过分析，由于题中是要求求目标完工时间最小，再通过分析，若穿梭车均匀速运动，在不装卸货物时是不会产生延误，影响延误时间的最大因素为穿梭车的装卸货，则可以将目标为总完工时间转化为最小停车延误时间，若前车装卸货物时，不影响后面车辆的正常行驶时，则该环形穿梭车系统效率最大。

◆ **问题一中关键指标的确定**

对目标由穿梭车完工最小时间转化到最小停车延误时间后，为了计算穿梭车是否延误，通过分析，引入了穿梭车的路程与穿梭车之间的距离作为两个判别延误的重要工具，其中，后一辆穿梭车的路程不能大于前一个穿梭车的路上，按照出发顺序的穿梭车，最后一辆穿梭车加一圈的路程也要大于第一辆穿梭车。对于穿梭车之间的距离，可以通过路程判断是否延误后，对穿梭车延误时间进行计算。

◆ **问题三中运行效率评价指标的选择**

由于要对前两问计算的效率进行分析，得到了运行效率评价指标分别为**穿梭车延误的总时间**与**穿梭车的数量**，分别计算器影响效率的比例，再结合以上因素，与只有一辆穿梭车的情况作为对比，得到最后的评价方案。

2.4 问题的具体分析

针对问题一，是建立货物调度的数学模，来对装载 A 侧进货口货物的穿梭车完成当前任务后，应该去 B 侧哪一个出货口进行取货，利用目标优化安排 A 侧所有物品与 B 侧进货口的物品进行一一匹配，简单化模型，利用穿梭车的路程、穿梭车之间的距离，对车辆延误时间进行计算，最终得出总完工最小时间。

针对问题二，相对于问题一，增加了穿梭车长度，相当于在延误计算时，后车的行驶路程不能等于前车的行驶路程，并且必须差一个穿梭车的长度，增加条件后，使模型

更加的真实合理，得到的结果更符合实际。

针对问题三，是对环形穿梭车系统运行效率进行评价，为了合理的对环形穿梭车系统运行效率进行评价，可以采用影响运行效率最大的穿梭延误时间总合，以及穿梭车的数量进行研究。再结合以上因素，与只有一辆穿梭车的情况作为对比，得到一个比较准确的评价方案。

针对问题四，是对环形穿梭车进行参数优化设计，为了合理的对参数进行优化，根据影响效率的因素，设计更优的穿梭车运行方案，减少参数对此类穿梭车的效率影响，从而得到更优的参数设计。

三、符号说明

符号	说明	单位
$R_{A_iB_j}$	表示从 A 侧进货口 i 到 B 侧进货口 j 运输的货物数	—
R_{A_j}	表示 A 侧进货口 j 的货物总量	—
d_{Ajk}	表示 A 侧进货口 j 中的第 k 个货物在 B 侧的目标出货口	—
C	环形轨道的周长	s
R_{B_h}	表示 B 侧进货口 h 的货物总量	—
N	环形穿梭车的数量	—
m	进货口的总数	—
n	出货口的总数	—
m_A	A 侧的进货口总数	—
m_B	B 侧的进货口总数	—
n_A	A 侧的出货口总数	—
$t_{L/U}$	一次装卸货时间	s
n_B	B 侧的出货口总数	—
v_0	穿梭车的匀速行驶的速度	m
t_0	穿梭车匀速行驶所用时间	m/s

四、模型假设

全文假设	✧ 假设穿梭车到进出货口可以立即上货，立即卸货 ✧ 假设穿梭车按照逆时针方向运行 ✧ 假设穿梭车同一时间每次只能执行一个任务。 ✧ 假设穿梭车停车启动时，加减速时间不计
问题一补充假设	✧ 假设不计穿梭车的实际长度 ✧ 假设 A 侧进货口货物需要在 B 侧指定出货口卸货 ✧ 假设 B 侧进货口货物可以在 A 侧任意出货口卸货 ✧ 假设穿梭车行驶速度恒定为 1.5m/s

问题二补充假设	✧ 假设B侧进出货口的数量一样 ✧ 假设穿梭车之间的最小距离为穿梭车长 ✧ 假设以穿梭车车头位置作为穿梭车路程位置
问题三补充假设	✧ 假设以1辆穿梭车的情况对比效率

注：假设的准确性解释，从网络上收集的数据、资料应该真实可靠，只有在数据正确的情况下，求解模型得到的结果正确性才能更高。

假设的排他性解释，比如对于穿梭车车头位置作为路程的计算位置，假设使用穿梭车中部或者尾部作为路程计算的起始位置，对于我们的模型来说，影响是不大的，并且这种假设有利于模型的计算。

假设的合理性解释，比如假设穿梭车之间的最小距离为穿梭车车长，由于在问题二中要求考虑穿梭车的车长，所以可以将穿梭车之间的最短距离定为穿梭车车长。

假设的必要性解释，比如假设中不考虑穿梭车加减速的影响，如果在把这一因素加进去，会使该问题更加复杂，耗时更多，所以这样假设是有必要的。

五、数据预处理

◆ ***B*** **侧每个出货口出货数量**

由于要对穿梭车的运输路线进行规划，首先需要对表一的数据进行整理，得到每个*A*侧进货口的货物在*B*指定出货口处出货的数量，才能对每个任务完成后，下一个任务进行安排，从题中所给的附近1进行统计，得到下表：

表1 A侧货物在B侧出货口卸货的数量统计

进货口 / 出货口	A侧进货口1	A侧进货口2	总计
B侧出货口1	23	22	45
B侧出货口2	20	14	34
B侧出货口3	42	40	82
B侧出货口4	15	24	39
总计	100	100	200

◆ ***B*** **侧每个进货口待出货物数量**

由于需要对 B 侧进货口任务数进行统计，才能根据模型计算得到目标为总完工时间，故将附件2的数据进行统计，统计结果如下：

表2 B侧进货口货物数量

进货口	B侧进货口1 待出货物	B侧进货口2 待出货物	B侧进货口3 待出货物	B侧进货口4 待出货物
数量	100	51	71	100

六、模型的建立与求解

6.1 总完工时间最小的单目标优化——问题一

6.1.1 模型准备

由于要将题中给出的信息用数学语言进行描述，并且题中所给出的信息繁多，需要进一步整理，利用集合的形式，将穿梭车系统中的参数进行定义，模型中需要的定义如下：

设穿梭车集合 $\mathrm{ca}r=\left\{N_j \mid j=1,2,\cdots,n\right\}$

进货口集合为 $O=\left\{O_A,O_B\right\}$,其中 O_A， O_B 分别表示 A 侧进货口集合和 B 侧进货口集合，且

$$O_A=\left\{O_{Aj} \mid j=1,2,\cdots,m_A\right\}$$

$$O_B=\left\{O_{Bh} \mid h=1,2,\cdots,m_B\right\}$$

出货口集合为 $E=\left\{E_A,E_B\right\}$，其中 E_A， E_B 分别表示 A 侧出货口集合和 B 侧出货口集合，且

$$E_A=\left\{E_{Aa} \mid a=1,2,\cdots,n_A\right\}$$

$$E_B=\left\{E_{Bb} \mid b=1,2,\cdots,n_B\right\}$$

现需要处理的运输任务集合为 $task=\left\{task_j,task_h \mid j=1,2,\cdots,m_A,h=1,2,\cdots,m_B\right\}$，

即 A， B 两侧需运输的货物数集合为

$$R_A=\left\{R_{Aj} \mid j=1,2,\cdots,m_A\right\}$$

$$R_B=\left\{R_{Bh} \mid h=1,2,\cdots,m_B\right\}$$

6.1.2 建模思想

由于从 A 侧进货口的货物只能从 B 侧指定出货口卸货，为了使穿梭车运行的效率最大，则尽可能的满足穿梭车能够将在 A 侧进口货物运输到 B 侧指定出货口后，能从 B 侧进货口上货运送到 A 侧出货口。基于以上分析，通过尽力穿梭车任务规划模型，对穿梭车执行两侧任务的顺序进行规划，得到每一列穿梭车执行任务的顺序。

通过分析可得，A 侧进货口货物若在出货口 E_{B,n_b} 进行卸货，则该穿梭车在 B 侧只能从进货口 O_{B,m_b} 进行装货；若 A 侧进货口货物若在出货口 E_{B,n_b-1} 进行卸货，则该穿梭车在 B 侧可以从进货口 O_{B,m_b} 与 O_{B,m_b-1} 进行装货，换而言之，A 侧的货物在 B 侧出货的出货口越靠前，该穿梭车的下一次任务可以选择的 B 侧进货口就越多。从而从进货口最少的 O_{B,m_b} 入手，从而给出每个穿梭车执行运输任务的顺序。

6.1.3 模型建立

为了使问题得到简化，现给出如下假设：

假设 1： B 侧货物在 A 侧卸货时，出货口的优先级从 1 到 n 依次增加。

假设 2： 假设不计穿梭车的实际长度。

➢ **穿梭车任务规划模型**

A 侧进货口货物运输后，对 A 侧进货口的货物到 B 侧目的出货口的数量进行统计，统计表如下：

表 3 A 侧进货口的货物到 B 侧目的出货站的数量

进货口 / 出货口	A 侧进货口 1	A 侧进货口 2	……	A 侧进货口 m_a	总计
B 侧出货口 1	$R_{A_1B_1}$	$R_{A_2B_1}$	……	$R_{A_{m_a}B_1}$	R_{AB_1}
B 侧出货口 2	$R_{A_1B_2}$	$R_{A_2B_2}$	……	$R_{A_{m_a}B_2}$	R_{AB_2}
……	……	……	……	……	……
B 侧出货口 n_b	$R_{A_1B_{n_b}}$	$R_{A_2B_{n_b}}$	……	$R_{A_{m_a}B_{n_b}}$	$R_{AB_{n_b}}$
总计	R_{A_1}	R_{A_b}	……	$R_{A_{m_a}}$	

其中 $R_{A_iB_j}$ 表示需要从 A 侧进货口 i 到 B 侧进货口 j 运输的货物数；R_{A_i} 表示 A 侧进货口 i 的货物总量，R_{AB_j} 表示 B 侧出货口 j 出口的总量。

◆ **A 侧进货口货物运输到 *B* 侧出货口 n_b 方案**

由于 A 侧进货口货物运输到 B 侧出货口 n_b，仅仅可以从 B 侧进货口 m_b 近货，若 A 侧进货口货物多于 B 侧进货口 m_b，则会出现穿梭车空跑的次数，若 B 侧进货口 m_b 多于 A 侧进货口货物，则可以较为有效的进行分配，其定义式如下：

$$f(R_{B_{m_b}}-R_{AB_{n_b}})=\begin{cases}R_{B_{m_b}}-R_{AB_{n_b}} & (R_{B_{m_b}}-R_{AB_{n_b}}>0)\\ 0 & (R_{B_{m_b}}-R_{AB_{n_b}}<0)\end{cases}$$

其中 $f(R_{B_{m_b}}-R_{AB_{n_b}})$ 表示将 A 侧进货口运输到 B 侧出货口 n_b 后，下次任务选择 B 侧进货口 m_b 进货后，进货口 m_b 的进货量。$R_{B_{m_b}}$ 表示 B 侧进货口 m_b 的数量，$R_{AB_{na}}$ 表示 A 侧进货口物品运输到出货口 n_a 的数量。

◆ ***A* 侧进货口货物运输到 *B* 侧出货口 n_b-1 方案**

安排完 A 侧进货口货物运输到 B 侧出货口 n_b 后的调度顺序，需要对 A 侧进货口货物运输到 B 侧出货口 n_b-1 后的进货口安排进行调度。

若 $f(R_{B_{m_a}}-R_{AB_{n_b}})=0$，则说明 B 侧进货口 m_b 的货物已全部安排调度到 A 侧出货口出

货，则进货口 m_b 已无可以调度的货物，并且 A 侧进货口的货物运输到 B 侧出货口 n_b-1 后，只能从 B 侧进货口 m_b-1 进行装载货物。

若 $f(R_{B_{m_a}}-R_{AB_{n_b}})>0$，则说明 B 侧进货口 m_b 的货物已没有全部安排调度到 A 侧出货口出货，则进货口 m_b 还有可以调度的货物，则 A 侧进货口的货物运输到 B 侧出货口 n_b-1 后，先从 B 侧进货口 m_b 处进行装载货物，若进货口 m_b 处货物装载完，再从进货口 m_b-1 装载货物。

$$f(R_{B_{m_a-1}}-R_{AB_{na}-1})=\begin{cases} f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}} & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na-1}}>0) \\ 0 & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b}-1}-R_{AB_{na}-1}<0) \end{cases}$$

其中 $f(R_{B_{m_a-1}}-R_{AB_{na}-1})$ 表示表示将 A 侧进货口运输到 B 侧出货口 n_b-1 后下次任务选择 B 侧进货口 m_b 进货后，进货口 m_b 的进货量。$R_{B_{m_b-1}}$ 表示 B 侧进货口 m_b-1 的数量，$R_{AB_{na-1}}$ 表示 A 侧进货口物品运输到出货口 n_a-1 的数量。

◆ **A 侧进货口货物运输到 B 侧出货口 n_b-n 方案**

安排完 A 侧进货口货物运输到 B 侧出货口 n_b-n 后的调度顺序，需要对 A 侧进货口货物运输到 B 侧出货口 n_b-1 后的进货口安排进行调度。

若 $f^{n-1}(R_{B_{m_a}}-R_{AB_{n_b}})=0$，则说明 B 侧进货口 m_b 到进货口 n_b-n+1 的货物已全部安排调度到 A 侧出货口出货，则进货口进货口 m_b 到进货口 n_b-n+1 已无可以调度的货物，并且 A 侧进货口的货物运输到 B 侧出货口 n_b-n 后，只能从 B 侧进货口 m_b-n 进行装载货物。

若 $f^{n-1}(R_{B_{m_a}}-R_{AB_{n_b}})>0$，则说明 B 侧进货口 m_b 的货物已没有全部安排调度到 A 侧出货口出货，则进货口 m_b 还有可以调度的货物，则 A 侧进货口的货物运输到 B 侧出货口 n_b-n 后，先从 B 侧进货口 m_b 处进行装载货物，若进货口 m_b 处货物装载完，再从进货口 m_b-n 装载货物。

$$f(R_{B_{m_a-1}}-R_{AB_{na}-1})=\begin{cases} f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}} & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na-1}}>0) \\ 0 & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}-1}<0) \end{cases}$$

其中 $f(R_{B_{m_a-1}}-R_{AB_{na}-1})$ 表示表示将 A 侧进货口运输到 B 侧出货口 n_b-1 后下次任务选择 B 侧进货口 m_b 进货后，进货口 m_b 的进货量。$R_{B_{m_b-1}}$ 表示 B 侧进货口 m_b-1 的数量，$R_{AB_{na-1}}$ 表示 A 侧进货口物品运输到出货口 n_a-1 的数量。

➢ **穿梭车延误分析模型**

6.1.4 建模思想

通过以上对穿梭车任务进行规划，得到了最优的规划方案。在对穿梭车延误原因进行分析，穿梭车若在轨道上正常行驶，不受前车的阻挡，则穿梭车运行不会产生延误，所以产生延误的原因是由于前车装卸货时，后车与前车之间的距离减小到 0，则后车只能选择停车等待，所以若要使延误最小，则需要前后车之间的距离最大。

并且在参数的选择上，通过分析，选择了穿梭车的位移$s_i(i=1,2,\cdots,n)$与穿梭车之间的距离$d_{i,i-1}(i=2,3,\cdots,n)$作为判断是否产生延误，以及产生延误的时间。从而建立了穿梭车延误分析模型，对延误的时间进行计算，得到最小的延误时间。

6.1.5 建模准备

首先，对穿梭车开始之间的距离进行初始化,由于需要得到最大的穿梭车之间的距离，则$d_{i,i+1}=\dfrac{C}{N}(i=2,3,\cdots,n)$，其中$C$环形轨道周长，$N$表示穿梭车的数量。

将 A 侧出货口 1 处定为穿梭车行驶的起始点，并按照顺时针运行，则级穿梭车 1 的位移为$s_1=0$，穿梭车i的初始位移为$s_i=-(i-1)\times\dfrac{C}{N}$，

6.1.6 模型建立

◆ **全部穿梭车没有进行装卸货物的情况**

当所有穿梭车没有进行装卸货物时，穿梭车i的位移增加量为v_0t_0，则可以得到匀速直线行驶后，穿梭车i的位移以及穿梭车之间的距离为：

穿梭车i的位移 $s_i'=s_i+v_0t_0$

穿梭车i与$i+1$之间的距离$d_{i,i+1}'=s_i'-s_{i+1}'=s_i-s_{i+1}=d_{i,i+1}$

其中s_i'为匀速运动t_0之后得到的穿梭车i的路程，s_i为匀速运动前穿梭车i的路程，v_0为穿梭车的行驶速度，t_0为全部穿梭车匀速直线运行的时间。

◆ **存在穿梭车准备进行装卸货物的情况**

当存在穿梭车i进行装卸货物时，则有可能会产生延误，装卸货物的时间为t_1，当穿梭车i准备进行货物装卸时候，对其前面的穿梭车不产生影响，只会对其后面的穿梭车产生影响，装卸货物完成后，判断是否会对后面的穿梭车产生影响，以及产生多大的延误如下公式：

穿梭车i完成装载货物时的位移：$s_i'=s_i+v_1t_0=s_i$

穿梭车$i+n$在穿梭车i完成装载货物时的位移：

$$s_{i+n}'=s_{i+n}+v_0t_2+v_1t_3 \quad (i+n<N)$$

$$t_2+t_3=t_0$$

其中v_0为穿梭车匀速行驶的速度，t_2为穿梭车匀速行驶的时间，v_1为穿梭车$i+n$装卸货物的时间，

◆ **穿梭车卸货延误情况分析**

之前的模型计算了穿梭车i进行装卸货物时，每辆穿梭车在穿梭车i进行装卸货物的时间里的位移状况，由于题中要求不能超车，则判断是否延误的依据为穿梭车i后面的穿梭车的位移不能超过穿梭车i的位移，若超过前车的位移，则会产生延误，则通过该计算方法计算是否产生延误，具体公式如下：

$$g(i,i+n)=\begin{cases}0 & (s_i \ge s_{i+n})\\ \dfrac{-d_{i,i+n}+v_0t_2+v_1t_3}{v_0} & (s_i < s_{i+n})\end{cases}$$

其中 $g(i,i+n)$ 表示穿梭车 $i+n$ 的延误时间，$d_{i,i+n}$ 表示穿梭车 i 到穿梭车 $i+n$ 的距离，v_0 为穿梭车匀速行驶的速度，t_2 为穿梭车匀速行驶的时间，v_1 为穿梭车 $i+n$ 装卸货物的时间。

从公式中可以看出当穿梭车 i 的路程大于穿梭车 $i+n$ 的路程时，则穿梭车装卸货物时，不会对穿梭车 $i+n$ 产生影响，当穿梭车 $i+n$ 的路程大于穿梭车 i 的路程时，穿梭车不会产生延误。

➢ 总完工时间最小单目标优化模型

6.1.7 建模思路

由于需要计算总完工时间，并且需要总完工时间最小，并且根据以上模型的分析，延误的最主要原因为穿梭车上下货时，造成后面穿梭车的延误，从而影响完工时间。若要使任务的总完工时间最短，则需要尽可能的减少穿梭车的延误时间之和，则将完工时间最小转化为延误时间之和最短。

6.1.8 模型建立

◆ 决策变量的确定

由于从 A 侧进货的穿梭车运输到 B 侧出货口后，需要从 B 侧进货口再次进货运输到 A 侧出口，则从 B 侧哪一个进货口进货可以最大限度减少后面穿梭车的延误时间，从而使目标函数得到最小的完工时间，所以决策变量为 x_{ij} 为 A 侧进货口任务 i 完成后，是否从 B 侧进货口 j 进行进货。

$$x_{ij}=\begin{cases}0, \text{执行完任务}i\text{后, 不选择}B\text{侧进货口}j\text{进货}\\ 1, \text{执行完任务}i\text{后, 选择从}B\text{侧进货口}j\text{进货}\end{cases}$$

◆ 目标函数的确定

由于题中要求建立总完工时间最短的数学模型，总完工时间最小可以转化为穿梭车延误时间之和最小，故确定目标函数为最小的穿梭车延误时间之和，记为：

$$\min\sum_{i=1}^{N} g(i,i+n) \quad (i+n<N)$$

其中，$g(i,i+n)$ 表示穿梭车 $i+n$ 在穿梭车 i 影响下的延误时间。N 表示穿梭车的数目

◆ 约束条件的确定

约束条件 1：B 侧出货口货物总数限制条件

对于选择从 B 侧进货口 j 进行进货的数量有限，则选择进货口 j 进行进货的次数应该等于进货口 j 的货物数量，得到：

$$\sum_{i=1}^{M} x_{ij} = R_{B_j}\,(j=1,2,\cdots,m_B)$$

约束条件 2：穿梭车装卸货造成的延误限制

$$g(i,i+n)=\begin{cases}0 & (s_i \ge s_{i+n})\\ \dfrac{-d_{i,i+n}+v_0t_2+v_1t_3}{v_0} & (s_i < s_{i+n})\end{cases}$$

其中 $g(i,i+n)$ 表示穿梭车 $i+n$ 的延误时间， $d_{i,i+n}$ 表示穿梭车 i 到穿梭车 $i+n$ 的距离，v_0 为穿梭车匀速行驶的速度， t_2 为穿梭车匀速行驶的时间， v_1 为穿梭车 $i+n$ 装卸货物的时间。

约束条件 3：B 侧进货口安排符合规划

$$f(R_{B_{m_a-1}}-R_{AB_{na}-1})=\begin{cases}f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}} & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na-1}}>0)\\ 0 & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}-1}<0)\end{cases}$$

其中 $f(R_{B_{m_a-1}}-R_{AB_{na}-1})$ 表示表示将 A 侧进货口运输到 B 侧出货口 n_b-1 后下次任务选择 B 侧进货口 m_b 进货后，进货口 m_b 的进货量。$R_{B_{m_b-1}}$ 表示 B 侧进货口 m_b-1 的数量，$R_{AB_{na-1}}$ 表示 A 侧进货口物品运输到出货口 n_a-1 的数量。

约束条件 4：B 侧进货口与 B 侧出货口匹配限制

$$\sum_{j=1}^{n_a} f(R_{B_i}-R_{AB_j}) = R_{B_i}(1,2,\cdots,m_a)$$

其中 R_{B_i} 表示 B 侧进货口的货物数量。

综上所述，建立的**总完工时间最小单目标优化模型：**

$$\min \sum_{i=1}^{N} g(i,i+n)\ \ (i+n<N)$$

$$s.t.\begin{cases}\displaystyle\sum_{i=1}^{M} x_{ij} = R_{B_j}\,(j=1,2,\cdots,m_B)\\ g(i,i+n)=\begin{cases}0 & (s_i \ge s_{i+n})\\ \dfrac{-d_{i,i+n}+v_0t_2+v_1t_3}{v_0} & (s_i < s_{i+n})\end{cases}\\ f(R_{B_{m_a-1}}-R_{AB_{na}-1})=\begin{cases}f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}} & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na-1}}>0)\\ 0 & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}-1}<0)\end{cases}\\ \displaystyle\sum_{j=1}^{n_a} f(R_{B_i}-R_{AB_j}) = R_{B_i}(1,2,\cdots,m_a)\end{cases}$$

6.1.8 算法设计

◆ **算法思想**

由于是建立了三个模型，第一个是穿梭车任务规划模型，对穿梭车执行任务的顺序进行规划，在建立了穿梭车延误分析模型，对时间延误的问题进行分析，并且得到了判断延误的方法，以及判断出穿梭车装卸货时是否延误，是一种动态规划，可以使用MATLAB 仿真对情况进行模拟，得到延误的时间。最后建立了总完工时间最小的单目标优化模型，计算得出完工的最小时间。

◆ **算法步骤**

Step1：初始化，得到每辆穿梭车的初始路程$s_i(i=1,2,\cdots,n)$，以及穿梭车之间的距离$d_{i,i-1}(i=2,3,\cdots,n)$；

Step2：利用 *MATLAB* 对动态模型进行仿真模拟，通过每辆穿梭车的初始路程$s_i(i=1,2,\cdots,n)$，以及穿梭车之间的距离$d_{i,i-1}(i=2,3,\cdots,n)$的变化，得到每辆穿梭车所造成的延误时间；

Step3：利用 *Lingo* 软件对单目标优化模型进行求解，从而得到最小的总完工时间；

◆ **求解结果**

当$N=3,6,9$时，初始化穿梭车的初始路程见下表：

表 4 穿梭车的初始路程

参数 / 穿梭车数量 N	初始路程(m)	穿梭车间隔$d_{i,i+1}=\frac{C}{N}$
3	$s_i=-33\frac{1}{3}i+33\frac{1}{3}$	$d_{i,i+1}=33\frac{1}{3}$
6	$s_i=-16\frac{2}{3}i+16\frac{2}{3}$	$d_{i,i+1}=16\frac{2}{3}$
9	$s_i=-11\frac{1}{9}i+11\frac{1}{9}$	$d_{i,i+1}=11\frac{1}{9}$

利用 *MATLAB* 编程，得到穿梭车任务规划的结果，得到的结果如下表

表 5 穿梭车任务规划

序号	出货口 4 卸货穿梭车	出货口 4 卸货穿梭车	出货口 4 卸货穿梭车	出货口 4 卸货穿梭车	A 侧空载穿梭车
B 侧进货口 4	39	61	0	0	0
B 侧进货口 3	0	21	30	0	0
B 侧进货口 2	0	0	4	45	2
B 侧进货口 1	0	0	0	0	100

最后，利用 *Lingo* 软件对单目标优化模型进行求解，从而得到最小的总完工时间，计算得到最后的时间结果为

表 6 不同穿梭车数量下的最小的总完工时间

穿梭车数量 N	3	6	9
最小完工时间 T(s)	10700	5437	3583

6.2 考虑穿梭车长度的最小时间优化模型——问题二

➢ **考虑穿梭车长度情况延误分析模型**

6.2.1 建模思路

由于第二问是要在考虑穿梭车长度的情况下，计算完成任务的最小完成时间，由于其他的条件与第一问的条件相同，通过分析，可以利用问题一中的模型，通过改变约束条件，从而建立考虑穿梭车长度的最小时间优化模型，在之前模型的基础上，增加穿梭车车长作为约束条件。

通过以上对穿梭车任务进行规划，得到了最优的规划方案。在对穿梭车延误原因进行分析，穿梭车若在轨道上正常行驶，不受前车的阻挡，则穿梭车运行不会产生延误，所以产生延误的原因是由于前车装卸货时，后车与前车之间的距离减小到 0，则后车只能选择停车等待，所以若要使延误最小，则需要前后车之间的距离最大。并且我们以穿梭车的车头作为穿梭车的位置。

并且在参数的选择上，通过分析，选择了穿梭车的位移 $s_i(i=1,2,\cdots,n)$ 与穿梭车之间的距离 $d_{i,i-1}(i=2,3,\cdots,n)$ 作为判断是否产生延误，以及产生延误的时间。从而建立了穿梭车延误分析模型，对延误的时间进行计算，得到最小的延误时间。

6.2.2 建模准备

首先，对穿梭车开始之间的距离进行初始化,由于需要得到最大的穿梭车之间的距离，则 $d_{i,i+1}=\dfrac{C}{N}(i=2,3,\cdots,n)$，其中 C 环形轨道周长， N 表示穿梭车的数量。

将 A 侧出货口 1 处定为穿梭车行驶的起始点，并按照顺时针运行，则级穿梭车 1 的位移为 $s_1=0$，穿梭车 i 的初始位移为 $s_i=-(i-1)\times\dfrac{C}{N}$，

6.2.3 模型建立

◆ **全部穿梭车没有进行装卸货物的情况**

当所有穿梭车没有进行装卸货物时，穿梭车 i 的位移增加量为 v_0t_0，则可以得到匀速直线行驶后，穿梭车 i 的位移以及穿梭车之间的距离为：

穿梭车 i 的位移 $$s_i'=s_i+v_0t_0$$

穿梭车i与$i+1$之间的距离$d_{i,i+1}'=s_i'-s_{i+1}'=s_i-s_{i+1}=d_{i,i+1}$

其中s_i'为匀速运动t_0之后得到的穿梭车i的路程，s_i为匀速运动前穿梭车i的路程，v_0为穿梭车的行驶速度，t_0为全部穿梭车匀速直线运行的时间。

◆ **存在穿梭车准备进行装卸货物的情况**

当存在穿梭车i进行装卸货物时，则有可能会产生延误，装卸货物的时间为t_1，当穿梭车i准备进行货物装卸时候，对其前面的穿梭车不产生影响，只会对其后面的穿梭车产生影响，装卸货物完成后，判断是否会对后面的穿梭车产生影响，以及产生多大的延误如下公式：

穿梭车i完成装载货物时的位移：$s_i'=s_i+v_1t_0=s_i$

穿梭车$i+n$在穿梭车i完成装载货物时的位移：

$$s_{i+n}'=s_{i+n}+v_0t_2+v_1t_3 \quad (i+n<N)$$

$$t_2+t_3=t_0$$

其中v_0为穿梭车匀速行驶的速度，t_2为穿梭车匀速行驶的时间，v_1为穿梭车$i+n$装卸货物的时间，

◆ **穿梭车卸货延误情况分析**

之前的模型计算了穿梭车i进行装卸货物时，每辆穿梭车在穿梭车i进行装卸货物的时间里的位移状况，由于题中要求不能超车，则判断是否延误的依据为穿梭车i后面的穿梭车的位移不能超过穿梭车i的位移，若超过前车的位移，则会产生延误，则通过该计算方法计算是否产生延误，具体公式如下：

$$g(i,i+n)=\begin{cases} l_{RGV} & (s_i-l_{RGV}\ge s_{i+n}) \\ \dfrac{-d_{i,i+n}+v_0t_2+v_1t_3}{v_0} & (s_i-l_{RGV}<s_{i+n}) \end{cases}$$

其中$g(i,i+n)$表示穿梭车$i+n$的延误时间，l_{RGV}表示穿梭车车长，$d_{i,i+n}$表示穿梭车i到穿梭车$i+n$的距离，v_0为穿梭车匀速行驶的速度，t_2为穿梭车匀速行驶的时间，v_1为穿梭车$i+n$装卸货物的时间。

从公式中可以看出当穿梭车i的路程大于穿梭车$i+n$的路程时，则穿梭车装卸货物时，不会对穿梭车$i+n$产生影响，当穿梭车$i+n$的路程大于穿梭车i的路程时，穿梭车不会产生延误。

➢ **总完工时间最小单目标优化模型**

6.2.4 建模思路

由于需要计算总完工时间，并且需要总完工时间最小，并且根据以上模型的分析，延误的最主要原因为穿梭车上下货时，造成后面穿梭车的延误，从而影响完工时间。若

要使任务的总完工时间最短，则需要尽可能的减少穿梭车的延误时间之和，则将完工时间最小转化为延误时间之和最短。

6.2.5 模型建立

◆ 决策变量的确定

由于从 A 侧进货的穿梭车运输到 B 侧出货口后，需要从 B 侧进货口再次进货运输到 A 侧出口，则从 B 侧哪一个进货口进货可以最大限度减少后面穿梭车的延误时间，从而使目标函数得到最小的完工时间，所以决策变量为 x_{ij} 为 A 侧进货口任务 i 完成后，是否从 B 侧进货口 j 进行进货。

$$x_{ij}=\begin{cases} 0,\text{执行完任务}i\text{后, 不选择}B\text{侧进货口}j\text{进货} \\ 1,\text{执行完任务}i\text{后, 选择从}B\text{侧进货口}j\text{进货} \end{cases}$$

◆ 目标函数的确定

由于题中要求建立总完工时间最短的数学模型，总完工时间最小可以转化为穿梭车延误时间之和最小，故确定目标函数为最小的穿梭车延误时间之和，记为：

$$\min\sum_{i=1}^{N} g(i,i+n)\ \ (i+n<N)$$

其中，$g(i,i+n)$ 表示穿梭车 $i+n$ 在穿梭车 i 影响下的延误时间。N 表示穿梭车的数目

◆ 约束条件的确定

约束条件 1：B 侧出货口货物总数限制条件

对于选择从 B 侧进货口 j 进行进货的数量有限，则选择进货口 j 进行进货的次数应该等于进货口 j 的货物数量，得到：

$$\sum_{i=1}^{M} x_{ij} = R_{B_j}\ (j=1,2,\cdots,m_B)$$

约束条件 2：穿梭车装卸货造成的延误限制

$$g(i,i+n)=\begin{cases} l_{RGV} & (s_i-l_{RGV}\ge s_{i+n}) \\ \dfrac{-d_{i,i+n}+v_0t_2+v_1t_3}{v_0} & (s_i-l_{RGV}<s_{i+n}) \end{cases}$$

其中 $g(i,i+n)$ 表示穿梭车 $i+n$ 的延误时间，$d_{i,i+n}$ 表示穿梭车 i 到穿梭车 $i+n$ 的距离，v_0 为穿梭车匀速行驶的速度，t_2 为穿梭车匀速行驶的时间，v_1 为穿梭车 $i+n$ 装卸货物的时间。

约束条件 3：B 侧进货口安排符合规划

$$f(R_{B_{m_a-1}}-R_{AB_{na}-1})=\begin{cases} f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}} & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na-1}}>0) \\ 0 & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}-1}<0) \end{cases}$$

其中 $f(R_{B_{m_a-1}}-R_{AB_{na}-1})$ 表示表示将 A 侧进货口运输到 B 侧出货口 n_b-1 后下次任务选择 B 侧进货口 m_b 进货后，进货口 m_b 的进货量。$R_{B_{m_b-1}}$ 表示 B 侧进货口 m_b-1 的数量，$R_{AB_{na-1}}$ 表示 A 侧进货口物品运输到出货口 n_a-1 的数量。

约束条件 4：B 侧进货口与 B 侧出货口匹配限制

$$\sum_{j=1}^{n_a} f(R_{B_i}-R_{AB_j})=R_{B_i}(1,2,\cdots,m_a)$$

其中 R_{B_i} 表示 B 侧进货口的货物数量。

综上所述，建立的**总完工时间最小单目标优化模型：**

$$\min\sum_{i=1}^{N} g(i,i+n)\ \ (i+n<N)$$

$$s.t.\begin{cases}\sum_{i=1}^{M} x_{ij}=R_{B_j}\,(j=1,2,\cdots,m_B)\\ g(i,i+n)=\begin{cases}l_{RGV} & (s_i-l_{RGV}\ge s_{i+n})\\ \dfrac{-d_{i,i+n}+v_0t_2+v_1t_3}{v_0} & (s_i-l_{RGV}<s_{i+n})\end{cases}\\ f(R_{B_{m_a-1}}-R_{AB_{n_a}-1})=\begin{cases}f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}} & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na-1}}>0)\\ 0 & (f(R_{B_{m_b}}-R_{AB_{n_b}})+R_{B_{m_b-1}}-R_{AB_{na}-1}<0)\end{cases}\\ \sum_{j=1}^{n_a} f(R_{B_i}-R_{AB_j})=R_{B_i}(1,2,\cdots,m_a)\end{cases}$$

6.2.6 算法设计

◆ **算法思想**

由于是建立了三个模型，第一个是穿梭车任务规划模型，对穿梭车执行任务的顺序进行规划，在建立了穿梭车延误分析模型，对时间延误的问题进行分析，并且得到了判断延误的方法，以及判断出穿梭车装卸货时是否延误，是一种动态规划，可以使用 *MATLAB* 仿真对情况进行模拟，得到延误的时间。最后建立了总完工时间最小的单目标优化模型，计算得出完工的最小时间。

◆ **算法步骤**

Step1：初始化，得到每辆穿梭车的初始路程 $s_i(i=1,2,\cdots,n)$，以及穿梭车之间的距离 $d_{i,i-1}(i=2,3,\cdots,n)$；

Step2：利用 *MATLAB* 对动态模型进行仿真模拟，通过每辆穿梭车的初始路程 $s_i(i=1,2,\cdots,n)$，以及穿梭车之间的距离 $d_{i,i-1}(i=2,3,\cdots,n)$ 的变化，得到每辆穿梭车所造成的延误时间；

Step3：利用 *Lingo* 软件对单目标优化模型进行求解，从而得到最小的总完工时间；

◆ **求解结果**

当 $N=3,6,9$ 时，初始化穿梭车的初始路程见下表：

表 7 穿梭车的初始路程

参数 / 穿梭车数量 N	初始路程(m)	穿梭车间隔 $d_{i,i+1}=\dfrac{C}{N}$
3	$s_i=-33\frac{1}{3}i+33\frac{1}{3}$	$d_{i,i+1}=33\frac{1}{3}$
6	$s_i=-16\frac{2}{3}i+16\frac{2}{3}$	$d_{i,i+1}=16\frac{2}{3}$
9	$s_i=-11\frac{1}{9}i+11\frac{1}{9}$	$d_{i,i+1}=11\frac{1}{9}$

利用 MATLAB 编程，得到穿梭车任务规划的结果，得到的结果如下表

表 8 穿梭车任务规划

序号	出货口 4 卸货穿梭车	出货口 4 卸货穿梭车	出货口 4 卸货穿梭车	出货口 4 卸货穿梭车	A 侧空载穿梭车
B 侧进货口 4	39	61	0	0	0
B 侧进货口 3	0	21	30	0	0
B 侧进货口 2	0	0	4	45	2
B 侧进货口 1	0	0	0	0	100

最后，利用 Lingo 软件对单目标优化模型进行求解，从而得到最小的总完工时间，计算得到最后的时间结果为

表 9 最小的总完工时间

穿梭车数量 N	3	6	9
最小完工时间 T(s)	11204	5786	4053

6.3 环形穿梭系统效率评价模型——问题三

6.3.1 模型思路

通过问题分析，影响环形穿梭车系统效率的主要因素有穿梭车的数量，若穿梭车较少，则穿梭车之间不发生延误。若穿梭车较多，则由于穿梭车之间的距离过近，从而产生延误，但是与此同时，由于穿梭车较多，能够更好的加快运输的频率，从而加快货物的运输，故建立指标来对系统效率进行评价。

选择的指标为穿梭车的拥堵时间，以及穿梭车数量增加导致穿梭车效率的提高。若穿梭车数量增加，则必定会增加成本，但是可以减少完成任务的总时间，则需要将增加的效率，以及穿梭车的成本作为条件，对穿梭车效率进行评价。

6.3.2 模型建立

首先对影响穿梭车效率的指标进行分析，建立以穿梭车的数目，以及完成任务的总时间作为指标，再通过穿梭车的效益来对环形穿梭车的效益进行评价。

◆ **指标 1：穿梭车的延误时间**

穿梭车延误时间的改变使效率发生了改变，并且延误时间导致了总完工时间的增加，基于以上分析，得到穿梭车的延误时间为：

若 $s_i - l_{RGV} \ge s_{i+n}$，则穿梭车 i 对后穿梭车车 $i+n$ 不产生延误，故

$$g(i,i+n)=0;$$

若 $s_i - l_{RGV} < s_{i+n}$，则穿梭车 i 对后穿梭车车 $i+n$ 不产生延误，故

$$g(i,i+n)=\frac{-d_{i,i+n}+v_0t_2+v_1t_3}{v_0}$$

其中 $g(i,i+n)$ 表示穿梭车 $i+n$ 的延误时间，l_{RGV} 表示穿梭车车长，$d_{i,i+n}$ 表示穿梭车 i 到穿梭车 $i+n$ 的距离，v_0 为穿梭车匀速行驶的速度，t_2 为穿梭车匀速行驶的时间，v_1 为穿梭车 $i+n$ 装卸货物的时间。

穿梭车延误的效率为 $$\eta_1=\frac{T_{总}}{\sum_{i=1}^{N-1}\sum_{n=1}^{i+n\le N} g(i,i+n)+T_{总}}$$

其中 $T_{总}$ 为完成该任务的总效率。

◆ **指标 2：穿梭车数量变化**

穿梭车数量的变化使得效率发生了变化，一方面穿梭车数的增加加大了最大货物的吞吐量，基于以上分析，得到了穿梭车的效率变化公式：

穿梭车数量变化引起的穿梭车间距的变化：

$$d_{i,i+1}=\frac{C}{N}(i=2,3,\cdots,n)$$

其中 C 环形轨道周长，N 表示穿梭车的数量。

◆ **两车间距与效率之间的关系**

若两侧间距 $d_{i,i+1}<v_0t_0$，则穿梭车 i 进行装载货物时，若穿梭车 $i+1$ 进行匀速直线行驶，则会产生延误，若 $d_{i,i+1}\ge v_0t_0$，则穿梭车 i 进行装载货物时，无论穿梭车 $i+1$ 也进行装载货物，或匀速直线行驶，都不会产生延误，故穿梭车效率为：

$$\eta_2=\frac{T_{总}}{\frac{v_0t_0-d_{i,i+1}}{v}+T_{总}}$$

◆ **基于穿梭车延误时间及穿梭车数量变化的效率评价**

由于穿梭车数量的提高对效率的变化有高有低，再基于作为仓库管理来讲，需要使经济效益获得最大，每提高穿梭车数量便会对效率有所改变，这里我们假设穿梭车的使用价值为val，以经济价值来计算整体的收益情况，我们将$N_0=1$时的效率作为基本效率，来计算$N=6$，$N=9$时，所带来的效率提升。则总体效率模型为：

$$\eta=\frac{N}{N_0}\eta_1\eta_2=\frac{N}{N_0}\times\frac{T_{总}}{\sum_{i=1}^{N-1}\sum_{n=1}^{i+n\le N}g(i,i+n)+T_{总}}\times\frac{T_{总}}{\frac{v_0t_0-d_{i,i+1}}{v}+T_{总}}$$

6.3.3 算法设计

◆ **算法步骤**

Step1：初始化，得到每辆穿梭车的初始路程$s_i(i=1,2,\cdots,n)$，以及穿梭车之间的距离$d_{i,i-1}(i=2,3,\cdots,n)$；

Step2：计算出穿梭车在不同辆数的情况下，完成任务的总体延误时间；

Step3：计算出穿梭车在数量的变化下，两车间距对效率的影响比例；

Step4：利用总体效率模型的公式，计算得出最终的效率。

◆ **求解结果**

表 10 穿梭车数量对效率的影响

穿梭车数量 N	3	6	9
总体效率(相对于 N=1)	2.98	5.86	6.82

从求解结果可以看出，穿梭车数量的上升可以提高总体的效率，但是在穿梭车数量越来越多时，整体效率的增长较慢。

6.4 基于多目标优化的环形穿梭车系统参数优化设计——问题四

6.4.1 建模思路

环形穿梭车作为立体仓库的辅助设备被广泛应用与物流系统中，极大地促进了物流产业的发展，给人们的生活带来了便利。由问题一、二可知，环形穿梭车系统能够实现大量货物的调度分配，穿梭车在一个调度周期内（绕环形轨道一周）完成单一任务或复合任务。单一任务是指完成一次出货搬运和一次进货搬运，复合任务是指完成两次（或多次）出货搬运和两次（或多次）进货搬运，没有完成任务被称为空跑任务。在实际运行过程中，为使穿梭车尽可能多的完成复合作业，减少单一作业和空跑作业的产生，从而在短时间内搬运更多的货物，进一步提高环形穿梭车系统的运行效率。本文将从以下几个方面对环形穿梭车系统进行参数优化设计。

- 确保穿梭车搬运距离和最短

- 环形穿梭车系统采用的是封闭式的环形轨道，在运行中，易发生堵塞，当进出货口需进出的货物数量庞大时，堵塞不可避免。应确保轨道堵塞造成的停车次数最少。
- 减少穿梭车装好货物后出发的加速时间和准备卸物时的减速时间。在搬运任务时，避免穿梭车一直做匀速运动。
- 保证穿梭车行走一个调度周期完成的搬运任务最多。

6.4.2 模型建立

穿梭车搬运货物包含三个阶段，分别是：进货装载，运行搬运，卸货装载。那么完成一次货物搬运作业，即完成一次单一任务的作业时间为：

$$T = t_w + 2t_d + t_0$$

$$t_w = \sum_{x=1}^{a} t_x + \sum_{y=1}^{b} t_y + \sum_{z=1}^{c} t_z$$

其中，T：穿梭车完成单一搬运任务所需时间；t_w：穿梭车在一个调度周期运行搬运所需时间；t_d：穿梭车进货转载和卸货转载所需时间（本文规定进货装载和卸货转载所需时间相同）；　t_0：环形穿梭车系统对穿梭车位置的检测时间和通讯的时间（一般为定值）；　t_x：穿梭车加减速行驶所用时间，a 为在一个调度周期内，穿梭车加减速行驶的的次数；　t_y：穿梭车匀速行驶所用时间，b 为在一个调度周期内，穿梭车匀速行驶的次数；t_z：穿梭车停止等待的时间。

设穿梭车集合 $\mathrm{car} = \{\mathrm{c}_j \mid j{=}1,2,\cdots,\mathrm{N}\}$，进货口集合为 $O = \{O_A, O_B\}$,其中 O_A, O_B 分别表示 A 侧进货口集合和 B 侧进货口集合，且 $O_A = \{O_{Aj} \mid j{=}1,2,\cdots,m_A\}$，$O_B = \{O_{Bh} \mid h{=}1,2,\cdots,m_B\}$，出货口集合为 $E = \{E_A, E_B\}$，其中 E_A, E_B 分别表示 A 侧出货口集合和 B 侧出货口集合，且 $E_A = \{E_{Aa} \mid a{=}1,2,\cdots,n_A\}$， $E_B = \{E_{Bb} \mid b{=}1,2,\cdots,n_B\}$，现需要处理的运输任务集合为

$$task = \{task_j, task_h \mid j{=}1,2,\cdots,m_A, h{=}1,2,\cdots,m_B\}$$

现要求这些任务需在规定时间 $T_A^1,...,T_A^j,...,T_A^{m_A};T_B^1,...,T_B^h,...,T_B^{m_B}$ 内执行搬运任务，为实现环形穿梭车调度系统高效率运行，使复合任务最多，空跑任务最少，同时减少穿梭车起停、加减速的时间，建立数学模型如下：

◆ 目标函数的确定

目标函数 1：确保环形穿梭车绕环形轨道一圈做复合作业所用时间最少。

环形穿梭车绕轨道一圈，在一个调度周期内完成两次搬运任务，即将 A 侧进货口进的货物搬运至 B 侧出货口出货，再从 B 侧进货口进货运送至 A 侧出货口出货。这一过程需满足所用时间最少。

$$\min T = \sum_{j=1}^{m_A} T_j + \sum_{h=1}^{m_B} T_h$$

目标函数 2：确保穿梭车执行复合作业的次数最多，空跑作业次数最少。

$$\max \sum_{j=1}^{N} \sum_{i=1}^{e} x_{C_j}^{i}$$

目标函数 3：减少各个环形穿梭车的轨道长度，尽可能减少无用的路程

$$\min \sum_{j=1}^{N} \sum_{i=1}^{c} t_i$$

目标函数 4：调度最佳的出入口排列顺序，使得出入口位置更加合理

$$\min \sum_{j=1}^{N} \sum_{i=1}^{a} t_i$$

其中，N:穿梭车的数量；e:一辆穿梭车完成一个调度周期内的任务所搬运的货物数量；$x_{C_j}^{i}$:第j辆穿梭车执行分配给它的第i个搬运任务的状态，即$x_{C_j}^{i}=\begin{cases}1\\0\end{cases}$，其中1表示穿梭车正在执行搬运任务，0则表示穿梭车为空跑状态。

◆ 约束条件的确定

约束条件 1：确保各穿梭车能够准确到达进货口进货。

为保证穿梭车C_j能够准确到达A侧进货口O_{Aj}、O_{Aj-1}进货，保证穿梭车C_j能够准确到达B侧进货口O_{Bj}、O_{Bj-1}进货，定义如下约束条件：

$$L_j^A - L_{\min(j)}^A \geq 0, L_h^B - L_{\min(h)}^B \geq 0$$

$$L_{j-1}^A - L_{\min(j-1)}^A \geq 0, L_{h-1}^B - L_{\min(h-1)}^B \geq 0$$

其中，L_j^A：穿梭车C_j到A侧进货口O_{Aj}进货的距离；

$L_{\min(j)}^A$：穿梭车C_j到A侧进货口O_{Aj}进货的最短距离；

L_h^B：穿梭车C_j到B侧进货口O_{Bh}进货的距离；

$L_{\min(h)}^B$：穿梭车C_j到B侧进货口O_{Bh}进货的最短距离；

约束条件 2： 确保环形穿梭车能够连续执行搬运任务，避免空跑搬运

为使穿梭车C_j在执行搬运任务j后能继续执行搬运任务$j-1$，需保证环形穿梭车C_j执行搬运任务$j-1$的时间大于C_j执行搬运任务j的时间。

$$T_{sC_j}^{j+1} > T_{oC_j}^{j}, T_{sC_j}^{h+1} > T_{oC_j}^{h}$$

其中：$T_{sC_j}^{j+1}$：穿梭车C_j执行A侧第$j+1$个搬运任务的开始时间；

$T_{oC_j}^{j}$：穿梭车C_j执行A侧第j个搬运任务的结束时间；

$T_{sC_j}^{h+1}$：穿梭车C_j执行B侧第$h+1$个搬运任务的开始时间；

$T_{oC_j}^{h}$：穿梭车 C_j 执行 B 侧第 h 个搬运任务的结束时间；

约束条件 3：确保穿梭车执行搬运任务 j 的时间满足系统要求的最短时间。

$$T_A^j < T_{A\min}^j, T_B^h < T_{B\min}^h$$

其中，$T_{A\min}^j$：A 侧第 j 个搬运任务从开始下达到开始执行系统要求所需的最少时间；

T_B^h：B 侧第 h 个搬运任务从开始下达到开始执行所用时间；

$T_{B\min}^h$：B 侧第 h 个搬运任务从开始下达到开始执行系统要求所需的最少时间。

综上所述，建立的**环形穿梭车系统参数优化设计模型**为：

$$\min Z = \sum_{j=1}^{m_A} T_j + \sum_{h=1}^{m_B} T_h$$

$$\max \sum_{j=1}^{N} \sum_{i=1}^{e} x_{C_j}^i$$

$$\min \sum_{j=1}^{N} \sum_{i=1}^{c} t_i$$

$$\min \sum_{j=1}^{N} \sum_{i=1}^{a} t_i$$

$$s.t. \begin{cases} L_j^A - L_{\min(j)}^A \ge 0, L_h^B - L_{\min(h)}^B \ge 0 \\ L_{j-1}^A - L_{\min(j-1)}^A \ge 0, L_{h-1}^B - L_{\min(h-1)}^B \ge 0 \\ T_{sC_j}^{j+1} > T_{oC_j}^j, T_{sC_j}^{h+1} > T_{oC_j}^h \\ T_A^j < T_{A\min}^j, T_B^h < T_{B\min}^h \end{cases}$$

6.4.3 模型计算及结果

首先将多目标优化通过设置权重，将多目标优化转化为单目标优化，再利用前两问建立的模型，带入多目标进行优化，最后利用 Lingo 求解，得到了最优的出入口位置，并且得到了出入口之间的最优距离，以及轨道的最优长度长度，以及最优的穿梭车数量。

得到的结果如下：

- **进出货口位置：**

A 侧从左向右：出货口 3，出货口 2，出货口 1，进货口 2，进货口 1.

B 侧从左向右：进货口 1，进货口 4，进货口 2，进货口 3，出货口 4，出货口 3，出货口 2，出货口 1.

- **轨道长度：**

A 侧轨道长度：60m

B 侧轨道长度：105m

弯道长度：30m

➢ **最优穿梭车数量：**8 辆

七、模型评价

7.1 模型的优点

(1) 模型一中将问题进行了拆分，首先对穿梭车货物调度路线进行了建模安排，再考虑延误的模型，最后再计算最小时间，计算建模层次分明。

(2) 模型一建立的单目标优化模型，选择用最小延误时间来代替计算最小总任务时间，进行了问题的转化，更好地对该模型进行了计算。

(3) 模型二中在模型一的基础上进行了改进，较好的实现了问题之间的连接性，也为用 *lingo* 软件计算目标函数值，提高了精确度。

(4) 建立的模型能与实际紧密联系，结合情况对问题进行求解，使得模型具有很好的通用性和推广性。

7.2 模型的缺点

(1) 模型一中并没有计算定量的计算所有指标，只是通过语言合理的分析指标对结果的影响。

(2) 问题三中，对于实际情况，应该还有更多的影响因素，如果时间充足，可以给出更加全面的效率分析。

八、模型推广

(1) 在本文中，都没有考虑穿梭车停车启动的时间，如果考虑车辆的加速度，得到的结果将更加完整。

(2) 在本文中，主要是研究在单层情况下的穿梭车规划，在未来的研究中，可以在立体仓库里，进行更多的改进规划。

(3) 在第三问中，建立了多目标优化模型，该模型在计算过程中，通过加权平均的方法，将多目标优化转化为单目标优化，这种计算方法可以在其他很多多目标优化模型中得到应用。

九、参考文献

[1] 禹一童.基于自适应遗传算法的智能 RGV 动态调度策略的研究[J].中国新通信,2019,21(03):143-144.

[2] 向旺,吴双,张可义,徐健,王弘扬.基于排队论的环形穿梭车系统运行参数分析[J].制造业自动化,2018,40(06):151-153.

[3] 向旺. 环形穿梭车系统车数确定及其调度算法的研究[D].机械科学研究总院,2018.

[4] 江唯. 智能立体仓储系统运行效率优化技术研发[D].南京理工大学,2017.

[5] 江唯,何非,童一飞,李东波.基于混合算法的环形轨道 RGV 系统调度优化研究[J].计算机工程与

应用,2016,52(22):242-247.

[6] 胡建伟. 环形穿梭车运行调度系统研究与开发[D].南京理工大学,2016.

[7] 龙锋. 基于自适应遗传算法的 W 公司仓库货位分配与优化研究[D].华南理工大学,2015.

[8] 姜华.环形穿梭车在烟草工业企业物流系统中的应用[J].物流技术与应用,2015,20(02):120-123.

[9] 顾红,邹平,徐伟华.环行穿梭车优化调度问题的自学习算法[J].系统工程理论与实践,2013,33(12):3223-3230.

[10] [10]杨少华,张家毅,赵立.基于排队论的环轨多车数量与能力分析[J].制造业自动化,2011,33(16):102-104.

[11] 司守奎，孙玺菁，数学建模算法与应用[M]，北京：国防工业出版社，2011.

十一、附录

附件 1：问题一的程序

文件名：***fujian1.doc***

功能：计算穿梭车数 $N=3,6,9$ 的时间

```
public void sched(RoadSys sys, double time_pass) {
                ArrayList<Car>       cars_list = sys.cars_list;
                ArrayList<PipeInput>  PPI_list  = sys.PPI_list;
                ArrayList<PipeOutput> PPO_list  = sys.PPO_list;

                for(int i=0; i< PPI_list.size(); i++) {
                    PipeInput input = PPI_list.get(i);
                    if (input.getItem() == null) {
                        continue;
                    }
                    for(int j=0; j< cars_list.size(); j++) {
                        Car car = cars_list.get(j);
                        if (car.getAimPipe() != null || car.getItem() !=
null) {
                            continue;
                        }
                        car.setAimPipe(input);
                        //System.out.println("Action:set
input:"+car+"aim:"+car.getAimPipe().getPos());
                        break;
                    }
                }
                for(int j=0; j< cars_list.size(); j++) {
                    Car car = cars_list.get(j);
                    if (car.getAimPipe() != null || car.getItem() == null)
{
                        continue;
                    }
                    Item item = car.getItem();
                    if (item.ouput_idx != Pipe.OUTPUT_ANY) {
                        car.setAimPipe(new PipeOutput(Pipe.SIDE_B,
item.ouput_idx));
                        //System.out.println("Action:set output:"+car);
                    }else {
                        for(int i=0; i< PPO_list.size(); i++) {
                            PipeOutput output = PPO_list.get(i);
                            if (output.getSide() == Pipe.SIDE_A) {
                                car.setAimPipe(output);
                                //System.out.println("Action:set
output:"+car);
```

```
                                //System.out.println("Action:set 
output:"+output);
                                break;
                            }
                        }
                    }
                    break;
                }
                for(int i=0; i < cars_list.size(); i++) {
                    
    //System.out.println(""+cars_list.get(i).getItem()+"----"+((PipeI)));
                }
                total_time += time_pass;
                //System.out.println("Action:time pass "+ time_pass +" s / 
" + total_time + " s");
            }
        };
    }
    public static void main(String args[]) {
        Task1 task = new Task1();
        task.doSelf(3);
        task.doSelf(6);
        task.doSelf(9);
    }
}
```

表 1 问题一任务规划表

各进货口货物的处理顺序	A 侧进货口 1 的货物在 B 侧的目标出货口编号	A 侧进货口 2 的货物在 B 侧的目标出货口编号	A 侧进货口 1 的货物在 B 侧的目标进货口编号	A 侧进货口 1 的货物在 B 侧的目标进货口编号
1	2	4	3	4
2	1	4	3	4
3	4	2	4	3
4	3	1	4	2
5	1	3	3	4
6	4	2	4	3
7	3	1	4	2
8	3	2	4	3
9	1	3	3	4
10	4	2	4	3
11	3	4	4	4
12	2	1	3	2
13	2	3	3	4
14	2	1	3	2
15	1	4	3	4
16	3	4	4	4
17	4	1	4	2
18	1	3	3	4
19	3	3	4	4
20	4	3	4	4
21	2	3	3	4
22	3	2	4	3
23	2	3	3	4
24	4	3	4	4
25	2	3	3	4
26	1	2	3	3
27	3	4	4	4
28	3	1	4	2
29	3	2	4	3
30	1	4	3	4
31	1	4	3	4
32	1	3	3	4
33	1	3	3	4
34	3	4	4	4
35	4	3	4	4
36	4	3	4	4
37	1	3	3	4
38	1	3	3	4
39	3	2	4	3
40	3	1	4	2
41	3	3	4	4
42	3	1	4	2
43	3	1	4	2
44	2	3	3	4
45	3	4	4	4
46	1	3	3	4
47	2	3	3	3

48	1	4	3	4
49	3	3	4	3
50	3	3	4	3
51	3	1	4	2
52	3	2	4	3
53	2	3	3	3
54	3	2	4	3
55	4	1	4	2
56	3	3	4	3
57	3	4	4	4
58	3	1	4	2
59	1	2	3	3
60	3	4	4	4
61	3	1	4	2
62	1	4	3	4
63	4	3	4	3
64	2	3	3	3
65	2	1	3	2
66	4	3	4	3
67	4	1	4	2
68	2	3	3	3
69	3	4	4	4
70	2	1	3	2
71	1	2	2	3
72	3	1	4	2
73	3	3	4	3
74	4	3	4	3
75	1	3	2	3
76	2	4	3	4
77	3	4	4	4
78	2	4	3	4
79	1	4	2	4
80	3	4	4	4
81	3	1	4	2
82	3	4	4	4
83	2	1	3	2
84	2	3	3	3
85	3	3	4	3
86	1	2	2	3
87	1	3	2	3
88	3	3	4	3
89	2	3	3	3
90	2	3	3	3
91	3	1	4	2
92	3	1	4	2
93	3	4	4	4
94	3	2	4	3
95	3	3	4	3
96	4	1	4	2
97	1	4	2	4
98	4	3	4	3

99	3	4	4	4
100	1	3	2	3