五一数学建模竞赛

承　诺　书

我们仔细阅读了五一数学建模竞赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与本队以外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其它公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们愿意承担由此引起的一切后果。

我们授权五一数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

参赛题号（从 A/B/C 中选择一项填写）：A

参赛队号：T1295124844

参赛组别（研究生、本科、专科、高中）：本科

所属学校（学校全称）：南京财经大学

参赛队员：　队员 1 姓名：侯孟敏

队员 2 姓名：梅宇

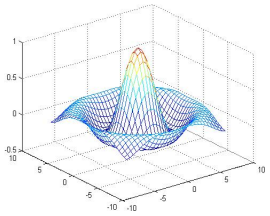
队员 3 姓名：侯孟慧

联系方式：　Email：1840686224@qq.com　联系电话：18851117990

日期：　2022　年 5　月　1　日

（除本页外不允许出现学校及个人信息）

五 一 数 学 建 模 竞 赛



题 目：血管机器人的订购与生物学习

关键词：多阶段决策、动态规划、模拟退火算法、多目标寻优

摘　要：

本文通过分析，确定决策变量和约束条件，建立了多决策的动态规划模型，用Python 代码的迭代算法和循环嵌套算法等对模型进行求解，对满足治疗所需的容器艇和操作手数量进行了分析和研究，得出题目要求的数据。

针对问题一，本文首先分析论文实现时的决策变量和约束条件，建立动态规划模型，再设计出循环嵌套算法，根据具体流程通过 Python 编程实现计算，代码见附录一，求解出题目中 1-8 周满足治疗每周所需增购的容器艇为（0,0,0,3,0,0,0,0），操作手为（14,0,0,28,0,0,0,0），最低运营成本为：7625 元。

针对问题二，本文先根据题目要求，增加额外的约束条件，定义一系列变量表示 1-104 周容器艇和操作手的损毁数目等。基于问题一，容器艇和操作手数量减去每周损毁的数量，确定目标函数和决策变量，设计迭代算法，并根据具体流程通过 Python 编程实现计算，详细代码见附录二，求解出 1-104 周满足治疗所需的最低运营成本：667560 元，其余相关结果见表 6-2，最后对比分析问题一和问题二 1-8 周的结果数据。

针对问题三，本文先在问题二的基础上，调整约束条件：降低损毁率至 10%，然后运用问题二建立的模型，同样使用迭代算法，根据流程通过 Python 编程实现计算，详细代码见附录四，求解出第 1-104 周总共需要购买的容器艇数量：498 个，操作手数量：2338 个，最低运营成本：439740 元。其余相关结果见表 7-2。

针对问题四，本文先对题目中所改变的约束条件，进行分析比对，确定条件改变后的数据变化，遵循问题三，根据新的约束条件和决策变量，参考模拟退火算法，使用 MATLAB 编程实现计算，求解出调整后的 1-104 周总共购买的容器艇和操作手的数量。

针对问题五，方案一：在问题四得到的 1-104 周最优结果的基础上，改变约束条件，考虑最新约束下满足第 105-112 周需求的最佳购买方案。方案二：在问题四的约束条件下，运用相同的模型，通盘考虑第 1-112 周的最佳购买方案。最后运用 MATLAB 编程得出两个方案的最低运营成本，比较两者的差额。

（前面两页必须使用模板格式，否则论文检测不通过）

此页为论文正文开始处

一、问题重述

随着微机电系统的发展，人类已经可以加工越来越小的机器。血管机器人的研究学习已成为国家关注的热点。这些血管机器人可以携带药物放入血管里定点治疗与血管相关的疾病，还可以充当血管清道夫，清除病毒，保持人体健康。因而，血管机器人的购买方案成为研究机构和医疗机构的重点关注问题。

请针对“血管机器人的订购与生物学习”问题，查阅资料并解决以下问题：

1. 根据所给数据，在一个熟练操作手最多“指导”10 个新购买的操作手进行生物学习的条件下，考虑使运营成本达到最低，并满足 1-8 周治疗所需的容器艇和操作手，给出最佳的购买方案。

2. 基于问题一，鉴于血管机器人在患者血管中工作时，碰到巨噬细胞有一定风险损毁，在血管机器人每周工作损毁率达到 20%的情况下，研究满足 1-104 周需求的最佳购买方案，将第 1-8 周的结果数据与问题一的第 1-8 周的结果数据进行对比分析，并将相关数据填入表格。

3. 基于问题二，修改以下部分条件：每名熟练操作手可以“指导”新操作手的数量调整为不超过 20 个；每周血管机器人的工作损毁率降低为 10%。研究满足 1-104 周需求的最佳购买方案，并将相关数据填入表格。

4. 基于问题三，增加约束条件：购买优惠政策，研究满足问题三条件及购买优惠政策条件下，第 1-104 周的容器艇和操作手的调整方案，并将相关数据填入表格。

5. 基于问题四，考虑两种方案，预测 105-112 周的血管机器人的使用需求，并比较两种方案的第 1-112 周最低运营成本的差额。

二、问题分析

问题一：本小题主要是根据 1-8 周具体数据，利用线性规划中的基本最优解概念使目标函数达到最优值的基本可行解，运营成本达到最低为目标函数，每周购买的操作手和容器艇数目为决策变量，每周用于训练的操作手数目为操作手购买量/10，据此写出对应算法。在 Python 的编译环境下，根据约束条件，得出满足 1-8 周治疗所需的容器艇和操作手的最佳购买方案。

问题二：本小题主要是根据 1-104 周的具体数据，在问题一的模型下，添加新的约束条件：血管机器人工作损毁率 20%，在 Python 的编译环境下，利用 for 循环体运行代码，统计每周购买和需要保养的容器艇和操作手数量，以及每周参与训练的操作手数量和总成本。

问题三：本小题主要是基于问题二的变形求解，对问题二的模型进行局部约束条件的修改，研究第 1-104 周既能满足治疗，又能使运营成本达到最低所需要购买的容器艇和操作手的数目。

问题四：本小题主要是基于问题三的条件，目标函数为运营成本，决策变量与上述相同，每周购买的容器艇和操作手是多少，因此，当我们在进行问题四的建模时，只需要更改问题三的约束条件。解决方案方法与问题三相同，同时参考鲸鱼算

法，以此来调整第 1-104 周里总共购买的容器艇和操作手的购买方案，使运营成本达到最低。

问题五：本小题主要是基于问题四的得出的结果，进行预测。在方案一的情况下：调整 104 周以后的容器艇和操作手价格，按照问题四的思路预测满足第 105-112 周的血管机器人的需求；在方案二下：按照问题四的思路，增加预测周数至 112 周，考虑第 1-112 周的血管机器人需求。最后比较两种方案的第 1-112 周最低运营成本的差额。

三、模型假设

1、所有数据均为题目所提供，具有可靠性。
2、在所有问题中，假设一个熟练工训练后仍需要保养。

四、符号说明

变换符号	符号说明
arm	机械臂
container	容器艇
Initial_arm	初始机械臂数量
Initial_container	初始容器艇数量
weeks_arm	存储需要购买机械臂的周数
weeks_container	存储需要购买容器艇的周数
maintenance_costs_arm	机械臂的保养费用
maintenance_costs_container	容器艇的保养费用
add_arm	新增的机械臂数量
add_container	新增的容器艇数量
total_cost	总成本
expenses_arm	机械臂的购买费用
expenses_container	容器艇的购买费用
training_expenses_arm	训练机械臂的费用
weekly_damage_arm	每周损毁的机械臂数量
weekly_damage_container	每周损毁的容器艇数量
weekly_maintenance_arm	每周需要保养的机械臂数量
weekly_maintenance_container	每周需要保养的容器艇数量
len(container)	周数
Initial_arm[i]	第 i 周所拥有机械臂之和
Initial_container[i]	第 i 周所拥有容器艇之和
v	第 i 周参加训练的操作手数量

五、问题一模型的建立和求解

5.1 基于多阶段决策中的动态规划的最优购买方案评定模型

根据题目条件所知，血管机器人的容器艇和机械手的医疗需求会随着周数的变化发生改变，基于该条件下，可知最佳购买方案的拟定为典型的多阶段决策问题。

考虑到该多阶段决策问题所求为最优解，可利用 Python 建立数学模型，通过变量 i(周数)将多阶段决策问题巧妙化为 i 个非线性规划问题。

数学模型应考虑如下约束条件：

目前所处第 i 周小于总周数

$$i<Len(container)$$

当前第 i 周和下一周医疗所需操作手小于现有操作手

$$arm[i]+arm[i+1]>initial\_arm$$

当前第 i 周医疗所需容器艇小于现有容器艇

$$container[i]>initial\_container$$

当所处第 i 周不满足以上约束条件时，需根据每小题的具体条件进行 add_arm ,add_container 操作，即增加购买操作手和容器艇。最后得到最优的容器艇和操作手的购买数量和购买时间，综合统计运营成本。

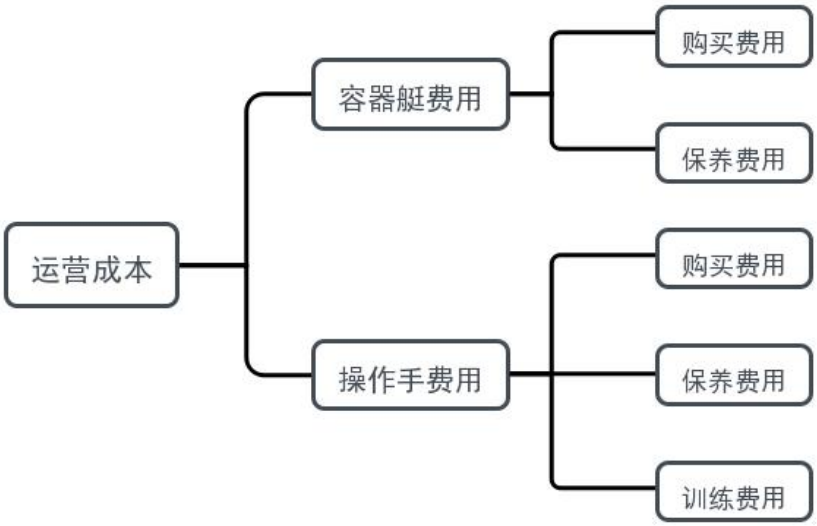


图 1 运营成本的影响因素

表 5-1 血管机器人的需求（1-8 周）

周	血管机器人的使用数量	容器艇所需数量	操作手所需数量
第一周	11	11	44
第二周	5	5	20
第三周	4	4	16
第四周	7	7	28
第五周	16	16	64
第六周	6	6	24
第七周	5	5	20
第八周	7	7	28

5.2 容器艇和操作手购买数量的计算公式

当第 i 周不满足其对应的约束条件时，通过以下计算：

$$add\_arm[i]=arm[i]+arm[i+1]-initial\_arm \quad (5\text{-}2\text{-}1)$$

$$add\_container[i-1]=container[i]-initial\_container \quad (5\text{-}2\text{-}3)$$

将每周购买的容器艇和操作手数量存储到相应的矩阵，得到如下结果：

容器艇购买=(0 0 0 3 0 0 0 0)

机械手购买=(14 0 0 28 0 0 0 0)

表 5-2 每一周的购买方案（1-8 周）

周	container[i]（个）	arm[i]（个）	add_container[i]（个）	add_arm[i]（个）
第一周	11	44	0	14
第二周	5	20	0	0
第三周	4	16	0	0
第四周	7	28	3	28
第五周	16	64	0	0
第六周	6	24	0	0
第七周	5	20	0	0
第八周	7	28	0	0

5.3 运营成本的计算公式

增购容器艇费用 expenses_container：

$$\sum_{0}^{7} add\_container[i]*200 \quad (5\text{-}3\text{-}1)$$

增购机械臂费用 expenses_arm:

$$\sum_{0}^{7} \text{add}\_arm[i]*100 \tag{5-3-2}$$

容器艇保养费用 maintenance_costs_container:

$$\sum_{0}^{7} (initial\_container[i] - container[i])*10 \tag{5-3-3}$$

操作手保养费用 maintenance_costs_arm：

$$\sum_{0}^{7} (initial\_arm[i] - arm[i])*5 - add\_arm[i]*5 \tag{5-3-4}$$

操作手训练费用 training_expenses_arm：

$$\sum_{0}^{7} (initial\_arm[i] - 50)*10 + [arm[i]/10 + 1]*5 \tag{5-3-5}$$

表 4-3 运营成本详细（1-8 周）

周	增购容器艇费用（元）	增购操作手 用费（元）	容器艇保养费（元）	操作手保养费（元）	操作手训练费（元）（含熟练手）	运营成本（元）
第一周	0	140	20	20	160	340
第二周	0	0	80	220	0	300
第三周	0	0	90	240	0	330
第四周	600	280	60	165	310	1415
第五周	0	0	0	140	0	140
第六周	0	0	100	340	0	440
第七周	0	0	110	360	0	470
第八周	0	0	90	320	0	410
合计	600	4200	550	1805	470	7625

六、问题二模型的建立和求解

6.1 基于问题一的最优购买方案评定模型

根据问题二的条件，增加约束，运用多阶段决策中的动态规划进行分析，分为以下几个步骤：第一，分析每周能够满足治疗的容器艇和操作手的数目，判断各需

要增购的数目以及最合适的购买时间；第二，统计运营总成本；第三，将问题一和问题二第 1-8 周的结果数据进行比较分析，并完成表格。

6.2 血管机器人损毁数目的计算公式

$$weekly\_damage\_container[i]=[container[i]*0.2+0.5] \quad (6\text{-}3\text{-}1)$$

$$weekly\_damage\_arm[i]=weekly\_damage\_container[i]*4 \quad (6\text{-}3\text{-}2)$$

表 6-1 血管机器人损毁数量

周	所需的机器人数目（个）	损毁的容器艇数目（个）	损毁的操作手数目（个）
第一周	11	2	8
第二周	5	1	4
第三周	4	1	4
第四周	7	1	4
第五周	16	3	12
第六周	6	1	4
第七周	5	1	4
第八周	7	1	4
…	…	…	…

6.3 容器艇和操作手购买数量的计算公式

基于问题一，已知量和决策变量不发生改变，除此外，因为多增加一个损毁数量的约束条件，initial_arm[i],initial_container[i]在第 i 周结束时会发生改变

$$initial\_arm[i]=initial\_arm[i]-weekly\_damage\_arm[i] \quad (6\text{-}3\text{-}1)$$

$$initial\_container[i]=initial\_container[i]-weekly\_damage\_container[i] \quad (6\text{-}3\text{-}2)$$

当第 i 周不满足其对应的约束条件时，通过以下计算得到第 i 周的购买方案：

$$add\_arm[i]=arm[i]+arm[i+1]-initial\_arm[i] \quad (6\text{-}3\text{-}3)$$

$$add\_container[i]=container[i]-initial\_container[i] \quad (6\text{-}3\text{-}4)$$

$$initial\_arm[i]=initial\_arm[i]-weekly\_damage\_arm[i] \quad (6\text{-}3\text{-}5)$$

$$initial\_container[i]=initial\_container[i]-weekly\_damage\_container[i] \quad (6\text{-}3\text{-}6)$$

6.4 运营成本相关数据的计算公式

保养的操作手数量 weekly_maintenance_arm：

$$\sum_{0}^{103} initial\_arm[i]-arm[i] \quad (6\text{-}4\text{-}1)$$

保养的容器艇数量 weekly_maintenance_container：

$$\sum_{0}^{103} initial\_container[i]-container[i] \quad (6\text{-}4\text{-}2)$$

参加训练的操作手数量 v：

$$\sum_{0}^{103} ([add\_arm[i]/10+1]+add\_arm[i]) \quad (6\text{-}4\text{-}5)$$

增购容器艇费用 expenses_container：

$$\sum_{0}^{103} add\_container[i]*200 \tag{6-4-6}$$

增购机械臂费用 expenses_arm:

$$\sum_{0}^{103} add\_arm[i]*100 \tag{6-4-7}$$

容器艇保养费用 maintenance_costs_container:

$$\sum_{0}^{103} (initial\_container[i]-container[i])*10 \tag{6-4-8}$$

操作手保养费用 maintenance_costs_arm：

$$\sum_{0}^{103} (initial\_arm[i]-arm[i])*5-add\_arm[i]*5 \tag{6-4-9}$$

操作手训练费用 training_expenses_arm：

$$\sum_{0}^{103} (initial\_arm[i]-50)*10+[arm[i]/10+1]*5 \tag{6-4-10}$$

表 6-2 问题 2 相关结果数据

周次	购买的容器艇数量	购买的操作手数量	保养的操作手数量	保养的容器艇数量	参与训练的操作手数量（含“熟练工”和“新手”）	总成本（单位：元）
第 12 周	5	20	26	1	22	3360
第 26 周	0	0	100	9	0	590
第 52 周	21	128	59	0	141	18705
第 78 周	16	40	172	0	44	8500
第 101 周	12	80	308	0	88	12820
第 102 周	17	36	344	0	40	9120
第 103 周	25	92	306	0	102	16750
第 104 周	0	0	308	0	0	1540
1-104 周（总计）	879	3826	13102	131	4234	667560

表 6-3 运营成本详细（1-8 周）

周	增购容器艇费用（元）	增购操作费用（元）	容器艇保养费（元）	操作手保养费（元）	操作手训练费用（元）（含熟练手）	运营成本（元）
第一周	0	1400	20	20	160	1600
第二周	0	0	60	180	0	240
第三周	0	0	60	180	0	240
第四周	1600	4400	20	75	490	6585
第五周	0	0	0	120	0	120
第六周	0	0	70	260	0	330
第七周	0	0	70	260	0	330
第八周	600	1200	40	190	140	2170
合计	2200	7000	340	1285	790	11615

6.5 问题一和问题二中第 1-8 周的结果数据对比分析

对比表 5-3 和表 6-3，由于问题二增加了血管机器人损毁率达到 20%的约束条件，所以满足问题二中治疗所需的操作手和容器艇的购买量应适当增加，因此对应的运营成本也有所增长。

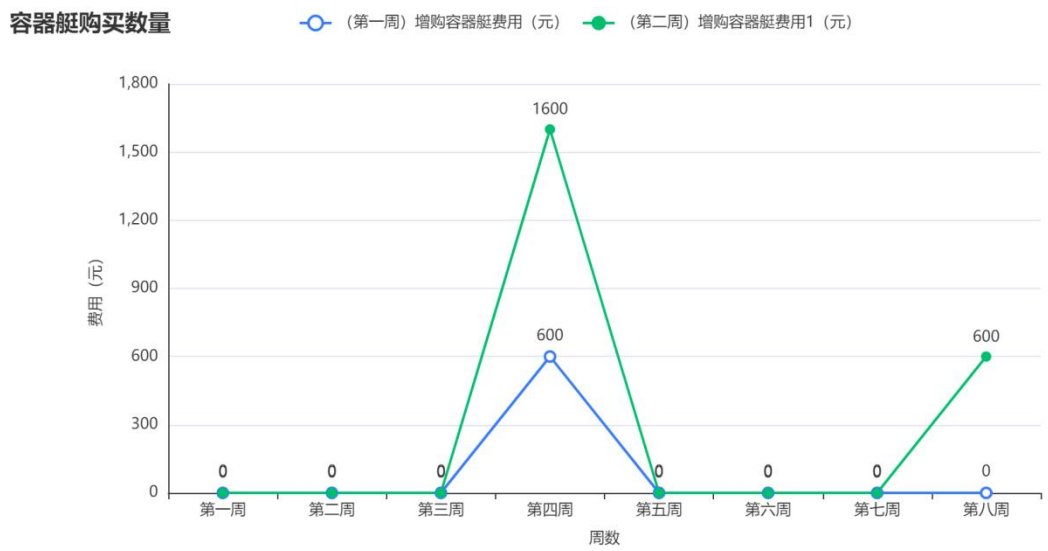


图 6-1 问题一和问题二第 1-8 周容器艇购买费用对比折线图

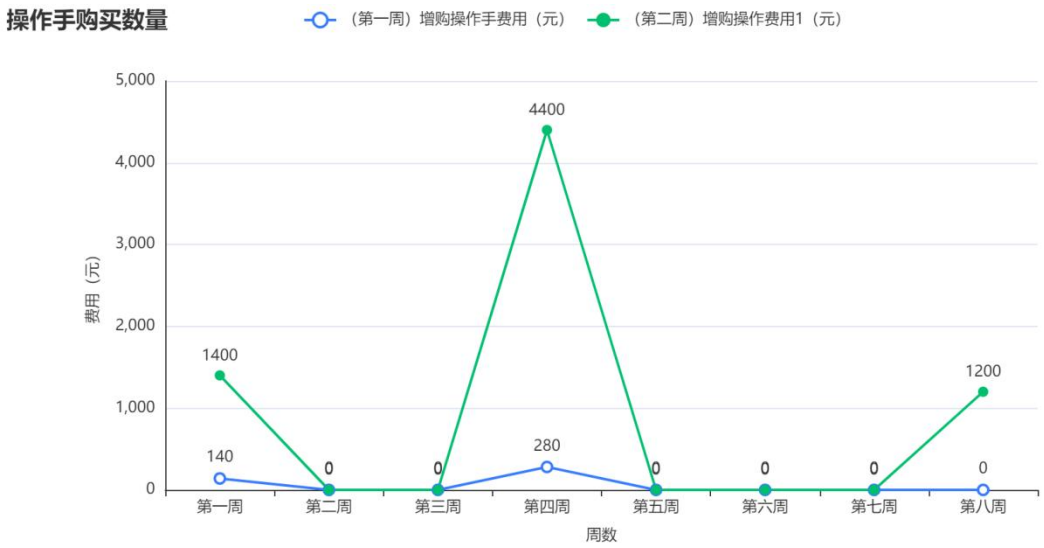


图 6-2 问题一和问题二第 1-8 周操作手购买费用对比--折线图

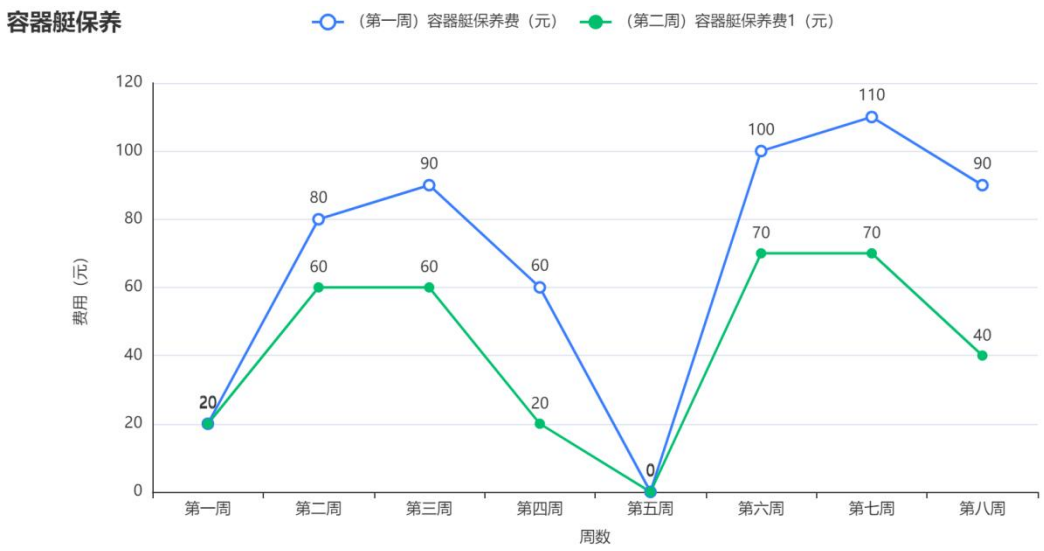


图 6-3 问题一和问题二第 1-8 周容器艇保养费用对比--折线图

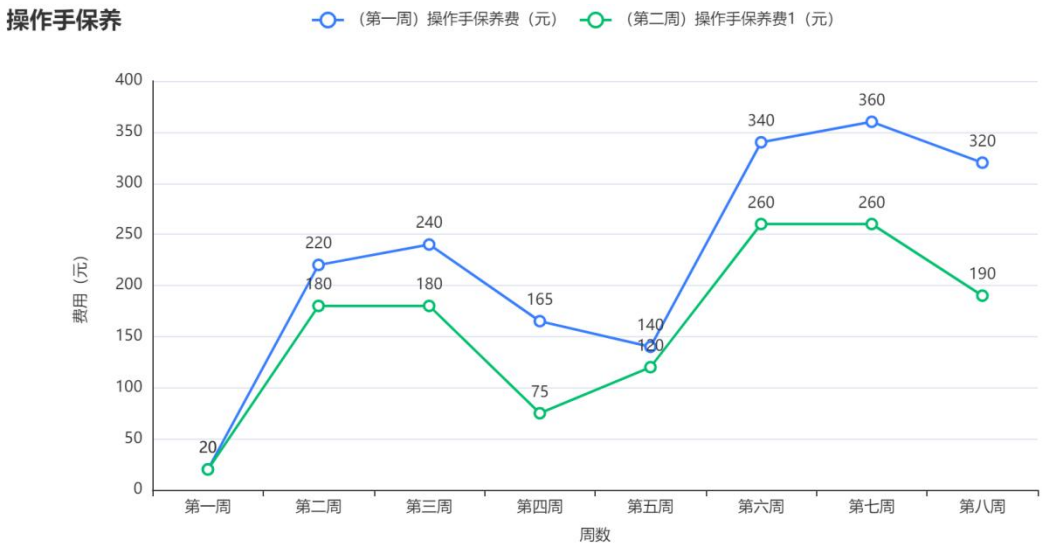


图 6-4 问题一和问题二第 1-8 周操作手保养费用对比--折线图

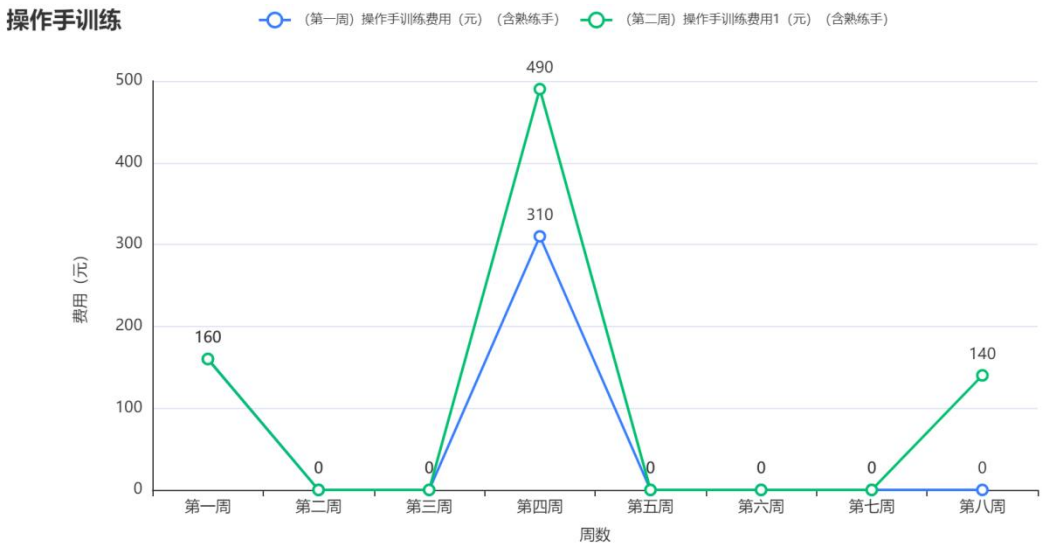


图 6-5 问题一和问题二第 1-8 周操作手训练费用对比--折线图

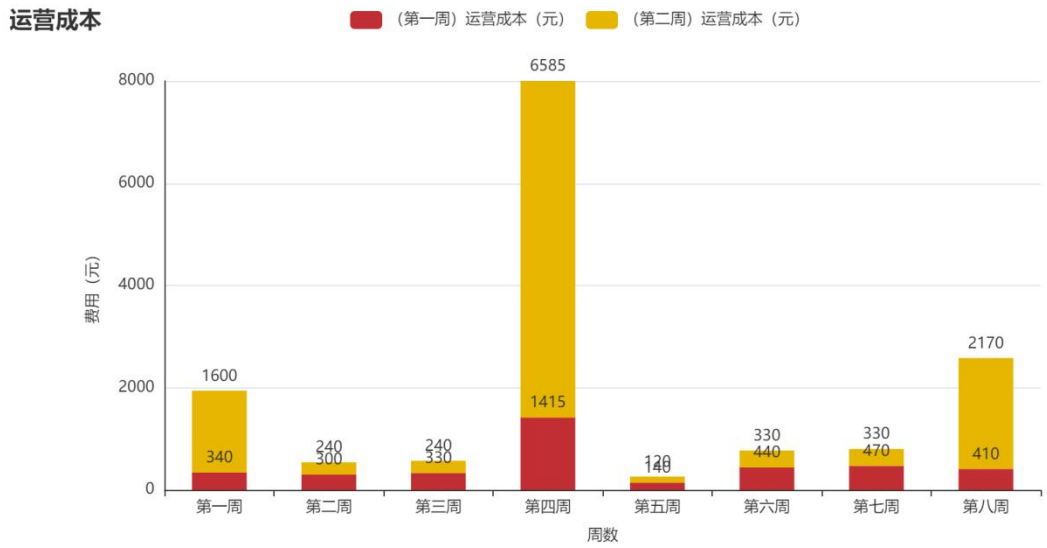


图 6-6 问题一和问题二第 1-8 周运营成本对比--堆积柱状图

七、问题三模型的建立和求解

7.1 基于问题二的最优购买方案评定模型

鉴于问题三本质上同问题二相同，仅仅修改了部分约束条件（损毁率降为 10%，熟练手可指导的新操作手调整为不超过 20），由问题二所得到的最优购买方案评定模型可不做修改，对部分参数进行修改即可得到问题三的答案。

7.2 血管机器人损毁数目的计算公式

$$weekly\_damage\_container[i]=[container[i]*0.1+0.5] \quad (7\text{-}2\text{-}2)$$

$$weekly\_damage\_arm[i]=weekly\_damage\_container[i]*4 \quad (7\text{-}2\text{-}1)$$

7.3 容器艇和操作手购买数量的计算公式

基于问题二，没有新的约束条件，仅仅修改损毁率和熟练手指导的新操作手数量，因而可以套用问题二的计算公式。

$$initial\_arm[i],initial\_container[i] \quad (7\text{-}3\text{-}1)$$

在第 i 周结束时会发生改变

$$initial\_arm[i]=initial\_arm[i]-weekly\_damage\_arm[i] \quad (7\text{-}3\text{-}2)$$

$$initial\_container[i]=initial\_container[i]-weekly\_damage\_container[i] \quad (7\text{-}3\text{-}3)$$

当第 i 周不满足其对应的约束条件时，通过以下计算得到第 i 周的购买方案：

$$add\_arm[i]=arm[i]+arm[i+1]-initial\_arm[i] \quad (7\text{-}3\text{-}4)$$

$$add\_container[i]=container[i]-initial\_container[i] \quad (7\text{-}3\text{-}5)$$

$$initial\_arm[i]=initial\_arm[i]-weekly\_damage\_arm[i] \quad (7\text{-}3\text{-}6)$$

$$initial\_container[i]=initial\_container[i]-weekly\_damage\_container[i] \quad (7\text{-}3\text{-}7)$$

7.4 运营成本相关数据的计算公式

保养的操作手数量 weekly_maintenance_arm：

$$\sum_{0}^{103} initial\_arm[i]-arm[i] \quad (7\text{-}4\text{-}1)$$

保养的容器艇数量 weekly_maintenance_container：

$$\sum_{0}^{103} initial\_container[i] - container[i] \tag{7-4-2}$$

参加训练的操作手数量 v：

$$\sum_{0}^{103} ([add\_arm[i]/20+1]+add\_arm[i]) \tag{7-4-3}$$

增购容器艇费用 expenses_container：

$$\sum_{0}^{103} add\_container[i]*200 \tag{7-4-4}$$

增购机械臂费用 expenses_arm:

$$\sum_{0}^{103} add\_arm[i]*100 \tag{7-4-5}$$

容器艇保养费用 maintenance_costs_container:

$$\sum_{0}^{103} (initial\_container[i] - container[i])*10 \tag{7-4-6}$$

操作手保养费用 maintenance_costs_arm：

$$\sum_{0}^{103} (initial\_arm[i] - arm[i])*5 - add\_arm[i]*5 \tag{7-4-7}$$

操作手训练费用 training_expenses_arm：

$$\sum_{0}^{103} (initial\_arm[i] - 50)*10 + [arm[i]/20+1]*5 \tag{7-4-8}$$

表 7-2 问题 3 相关结果数据

周次	购买的容器艇数量	购买的操作手数量	保养的操作手数量	保养的容器艇数量	参与训练的操作手数量（含“熟练工”和“新手”）	总成本（单位：元）
第 12 周	3	12	35	3	3	2135
第 26 周	0	0	116	11	0	690
第 52 周	18	116	78	0	122	16810
第 78 周	10	20	195	1	21	5195
第 101 周	1	40	354	0	42	6390
第 102 周	7	0	392	0	0	3360
第 103 周	16	44	361	0	47	9875
第 104 周	0	0	344	0	0	1720
1-104 周（总计）	498	2338	15758	281	2474	439740

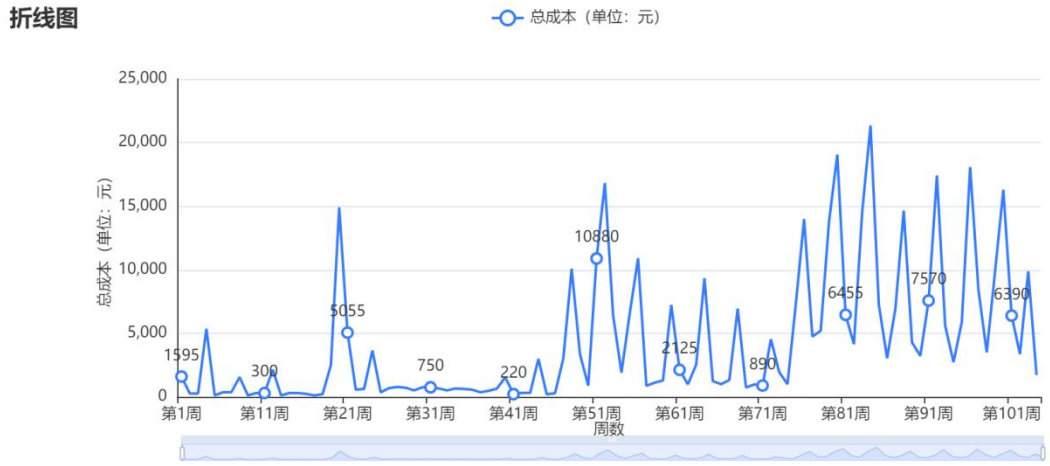


图 7-1 问题三周数和总成本的关系折线图

八、问题四模型的建立和求解

8.1 基于问题三的最优购买方案评定模型

问题四在问题三的基础上，改变了优惠政策的约束条件，目标函数和已知量不变，参考模拟退火算法[1][2]，对 1-104 周总共购买的容器艇和操作手数量进行调整，并完成相关表格。

8.2 优惠价格相关数据的计算公式

1. 随机生成一个初始值，并产生一个初始变化数量（容器艇），求出运营成本因此省下的钱ΔP_r

2. 若$\Delta P_k \leq 0$ 则变化数量生效并继续产生位移再判断；若$\Delta P_r > 0$ 则以值为 P_k 的概率来判断是否生效

3. Pk 公式，式中 c 为购买容器艇时，每个容器艇节省的钱

$$P_k = \frac{1}{1+e^{-\Delta P_r / c}} \tag{8-2-1}$$

4. 转第 1 步继续执行，直到达到平衡状态为止

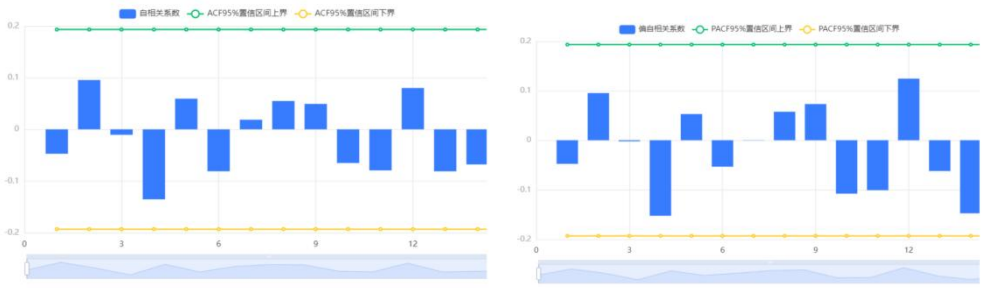
九、问题五模型的建立和求解

9.1 基于问题四的最优购买方案评定模型

本题预比较方案一，方案二最低运营成本差额，应先进行时间序列分析，得到 105-112 周血管机器人的使用需求。自回归移动平均模型是目前最常用的拟合平稳序列的模型，本题具体采用 ARMA 模型进行时间序列分析预测。

通过对自相关系数（ACF）和偏自相关系数（PACF）进行分析:

模型残差自相关图（ACF）　　模型残差偏自相关图（PACF）



得到下列数学模型

y(t)=0.898+0.125*y(t-1)-0.209*y(t-2)-0.27*y(t-3)+0.515*y(t-4)-0.272*y(t-5)　(9-1-1)

从而得到 105-112 周血管机器人使用需求

时间（周）	使用需求
105	111
106	105
107	103
108	106
109	112
110	109
111	107
112	110

十、模型改进

在问题二和问题三中，因为题目要求每周的血管机器人损毁数目按四舍五入取整，在 Python 的编译环境下，四舍五入取整函数为 round()，但是 round(0.5)=0 而不是 1，这就导致我们问题三的容器艇总共多买了 14 个，操作手多买了 56 个，总成本多了 7715 元。所以我们使用了 int(损毁的数量+0.5)来进行四舍五入。

十一、参考文献

[1]老居搞机，模拟退火算法（附代码），https://www.jianshu.com/p/3586a6e0e246，2022 年 5 月 3 日

[2]Steven，你也能看懂的：退火算法，https://blog.csdn.net/weixin_44613063/article/details/104151489，2022 年 5 月 3 日

附录：

附录一：问题一代码

```
import math

arm = [44, 20, 16, 28, 64, 24, 20, 28]
container = [11, 5, 4, 7, 16, 6, 5, 7]
initial_arm = 50
initial_container = 13
weeks_arm = []
weeks_container = []
maintenance_costs_arm, maintenance_costs_container = 0, 0
num1 = 0
num2 = 0
add_arm = []
add_container = []
total_cost = 0

for i in range(len(arm)):
    if (i + 1 < len(arm)):
        s = arm[i] + arm[i + 1]
        if (s > initial_arm):
            num1 = s - initial_arm
            add_arm.append(num1)
            initial_arm = s
            weeks_arm.append(i)

        if (container[i] > initial_container):
            num2 = container[i] - initial_container
            add_container.append(num2)
            initial_container = container[i]
            weeks_container.append(i - 1)
    maintenance_costs_arm = maintenance_costs_arm + (initial_arm - arm[i]) * 5
    maintenance_costs_container = maintenance_costs_container + (initial_container -
container[i]) * 10
maintenance_costs_arm = maintenance_costs_arm - sum(add_arm) * 5

for i in range(len(weeks_arm)):
    print("第", weeks_arm[i] + 1, "周购买的机械臂数量为：", add_arm[i])
for i in range(len(weeks_container)):
    print("第", weeks_container[i] + 1, "周购买的容器艇数量为：", add_container[i])

expenses_arm = (initial_arm - 50) * 100
```

```
expenses_container = (initial_container - 13) * 200
training_expenses_arm = (initial_arm - 50) * 10 + math.ceil(sum(add_arm) / 10) * 5

total_cost = expenses_arm + expenses_container + training_expenses_arm + maintenance_costs_arm + maintenance_costs_container
print("总成本：", total_cost, "元")
```

附录二：问题二代码

```
import math
from openpyxl import    Workbook
from openpyxl import load_workbook
from openpyxl.utils import get_column_letter

container = [11, 5, 4, 7, 16, 6, 5, 7,
                    13, 6, 5, 7, 12, 5, 4, 6,
                    9, 5, 5, 11, 29, 21, 17, 20,
                    27, 13, 9, 10, 16, 6, 5, 7,
                    11, 5, 5, 6, 12, 7, 7, 10,
                    15, 10, 9, 11, 15, 10, 10, 16,
                    26, 21, 23, 36, 50, 45, 45, 49,
                    57, 43, 40, 44, 52, 43, 42, 45,
                    52, 41, 39, 41, 48, 35, 34, 35,
                    42, 34, 36, 43, 55, 48, 54, 65,
                    80, 70, 74, 85, 101, 89, 88, 90,
                    100, 87, 88, 89, 104, 89, 89, 90,
                    106, 96, 94, 99, 109, 99, 96, 102]

arm = []
for i in range(len(container)):
      arm.append(container[i] * 4)

initial_arm = 50
initial_container = 13
weeks_arm = []
weeks_container = []
maintenance_costs_arm, maintenance_costs_container = 0, 0
num1 = 0
num2 = 0
add_arm = []
add_container = []
weekly_damage_arm = []
weekly_damage_container = []
existing_arms = []
```

```
weekly_maintenance_arm = []
weekly_maintenance_container = []


v = []
total_cost = 0
a = 0
for i in range(len(container)):
    weekly_damage_container.append(int (container[i] * 0.2 + 0.5))
    weekly_damage_arm.append(4 * weekly_damage_container[i])
    if (i + 1 < len(arm)):
        s = arm[i] + arm[i + 1]
        if (s > initial_arm):
            num1 = s - initial_arm
            add_arm.append(num1)
            a = math.ceil(num1 / 10)
            v.append(num1 + a)
            weekly_maintenance_arm.append(initial_arm - arm[i] - a)
            initial_arm = s
            weeks_arm.append(i)
        else:
            add_arm.append(0)
            v.append(0)
            weekly_maintenance_arm.append(initial_arm - arm[i])

    if (container[i] > initial_container):
        num2 = container[i] - initial_container
        add_container.append(num2)
        initial_container = container[i]
        weeks_container.append(i - 1)
    else:
        add_container.append(0)
    weekly_maintenance_container.append(initial_container - container[i])

    initial_arm = initial_arm - weekly_damage_arm[i]

    if (i + 1 >= len(container)):
        break
    else:
        maintenance_costs_arm = maintenance_costs_arm + weekly_maintenance_arm[i] * 5
    maintenance_costs_container = maintenance_costs_container + (initial_container - container[i]) * 10
    initial_container = initial_container - weekly_damage_container[i]
```

```
        existing_arms.append(initial_arm - add_arm[i] + weekly_damage_arm[i])

#另长度为 104
add_arm.append(0)
add_container.remove(add_container[0])
add_container.append(0)
v.append(0)
existing_arms.append(existing_arms[102]-weekly_damage_arm[102]+add_arm[102])
weekly_maintenance_arm.append(existing_arms[len(existing_arms)-1] -
arm[len(arm)-1])

maintenance_costs_arm = maintenance_costs_arm +
weekly_maintenance_arm[len(weekly_maintenance_arm)-1] * 5
for i in range(len(add_arm)):
        print("第", i + 1, "周购买的操作手数量为：", add_arm[i])
for i in range(len(add_container)):
        print("第", i+1, "周购买的容器艇数量为：", add_container[i])

total_add_arm = sum(add_arm)
expenses_arm = sum(add_arm) * 100
expenses_container = sum(add_container) * 200
training_expenses_arm = sum(v) * 10
total_cost = expenses_arm + expenses_container + training_expenses_arm +
maintenance_costs_arm + maintenance_costs_container

print("总共购买的操作手数量为：%d 个" % sum(add_arm))
print("总共购买的容器艇数量为：%d 个"% sum(add_container))
print("总成本：%d 元"% total_cost)

#周数（104 周）
weeks = []
for i in range(0, 104):
        weeks1 = "第%d 周"% (i+1)
        weeks.append(weeks1)

#统计每周总成本
weeks_cost = []
for i in range(0, 104):

weeks_cost.append(add_container[i]*200+add_arm[i]*100+weekly_maintenance_arm[i]
*5+weekly_maintenance_container[i]*10+v[i]*10)
```

```
# 绘制表格
wb = Workbook()
sheet = wb.active
sheet.title='问题二：1-104 周的结果数据'
sheet.append(['周数', '购买的容器艇数量','现有的操作手数量', '购买的操作手数量','保养的操作手数量','保养的容器艇数量','参与训练的操作手数量','总成本（单位：元）
'])
for i in range(0,104):

sheet.append([weeks[i],add_container[i],existing_arms[i],add_arm[i],weekly_maintenanc
e_arm[i],weekly_maintenance_container[i],v[i],weeks_cost[i]])
sheet.append(["1-104 周总和
",sum(add_container),sum(existing_arms),sum(add_arm),sum(weekly_maintenance_arm)
,sum(weekly_maintenance_container),sum(v),sum(weeks_cost)])
wb.save('问题二：1-104 周的结果数据.xlsx')
wb = load_workbook('问题二：1-104 周的结果数据.xlsx')
sheet = wb.active

#表格列宽自适应
lks = []
for i in range(1, sheet.max_column + 1):
    lk = 1
    for j in range(1, sheet.max_row + 1):
        sz = sheet.cell(row=j, column=i).value
        if isinstance(sz, str):
            lk1 = len(sz.encode('gbk'))
        else:
            lk1 = len(str(sz))
        if lk < lk1:
            lk = lk1
        # print(lk)
    lks.append(lk)
# 第二步：设置列宽
for i in range(1, sheet.max_column + 1):
    k = get_column_letter(i)
    sheet.column_dimensions[k].width = lks[i - 1] + 2
wb.close()
wb.save('问题二：1-104 周的结果数据.xlsx')
```

附录三：问题二代码生成的表格

周数	购买的容器艇数量	现有的操作手数量	购买的操作手数量	保养的操作手数量	保养的容器艇数量	参与训练的操作手数量	总成本（单位：元）

第 1 周	0	50	14	4	2	16	1600
第 2 周	0	56	0	36	6	0	240
第 3 周	0	52	0	36	6	0	240
第 4 周	8	48	44	15	2	49	6585
第 5 周	0	88	0	24	0	0	120
第 6 周	0	76	0	52	7	0	330
第 7 周	0	72	0	52	7	0	330
第 8 周	3	68	12	38	4	14	2170
第 9 周	0	76	0	24	0	0	120
第 10 周	0	64	0	40	4	0	240
第 11 周	0	60	0	40	4	0	240
第 12 周	5	56	20	26	1	22	3360
第 13 周	0	72	0	24	0	0	120
第 14 周	0	64	0	44	5	0	270
第 15 周	0	60	0	44	5	0	270
第 16 周	2	56	4	31	2	5	1025
第 17 周	0	56	0	20	0	0	100
第 18 周	0	48	0	28	2	0	160
第 19 周	6	44	20	22	1	22	3540
第 20 周	20	60	100	6	0	110	15130
第 21 周	0	152	48	31	0	53	5485
第 22 周	0	176	0	92	2	0	480
第 23 周	4	160	0	92	2	0	1280
第 24 周	11	148	40	64	0	44	6960
第 25 周	0	172	0	64	0	0	320
第 26 周	0	152	0	100	9	0	590
第 27 周	0	140	0	104	10	0	620
第 28 周	1	132	0	92	7	0	730
第 29 周	0	124	0	60	0	0	300
第 30 周	0	112	0	88	7	0	510
第 31 周	0	108	0	88	7	0	510
第 32 周	1	104	0	76	4	0	620
第 33 周	0	100	0	56	0	0	280
第 34 周	0	92	0	72	4	0	400
第 35 周	0	88	0	68	3	0	370
第 36 周	6	84	0	60	1	0	1510
第 37 周	0	80	0	32	0	0	160
第 38 周	0	72	0	44	3	0	250
第 39 周	2	68	0	40	2	0	620
第 40 周	7	64	36	20	0	40	5500
第 41 周	0	92	8	31	0	9	1045
第 42 周	0	88	0	48	2	0	260
第 43 周	3	80	0	44	1	0	830

第 44 周	6	72	32	24	0	36	4880
第 45 周	0	96	4	35	0	5	625
第 46 周	0	88	0	48	2	0	260
第 47 周	8	80	24	37	0	27	4455
第 48 周	13	96	72	24	0	80	10720
第 49 周	0	156	32	48	0	36	3800
第 50 周	6	168	8	83	0	9	2505
第 51 周	18	160	76	60	0	84	12340
第 52 周	21	216	128	59	0	141	18705
第 53 周	5	316	64	109	0	71	8655
第 54 周	9	340	20	158	0	22	4810
第 55 周	13	324	52	138	0	58	9070
第 56 周	18	340	84	135	0	93	13605
第 57 周	0	384	16	154	0	18	2550
第 58 周	3	356	0	184	3	0	1550
第 59 周	12	320	16	158	0	18	4970
第 60 周	17	304	80	120	0	88	12880
第 61 周	1	348	32	136	0	36	4440
第 62 周	8	340	0	168	0	0	2440
第 63 周	11	304	44	131	0	49	7745
第 64 周	16	316	72	128	0	80	11840
第 65 周	0	352	20	142	0	22	2930
第 66 周	5	332	0	168	1	0	1850
第 67 周	10	300	20	142	0	22	4930
第 68 周	15	288	68	117	0	75	11135
第 69 周	0	324	8	131	0	9	1545
第 70 周	3	292	0	152	3	0	1390
第 71 周	8	264	12	126	0	14	3570
第 72 周	14	248	60	102	0	66	9970
第 73 周	0	280	24	109	0	27	3215
第 74 周	9	272	8	135	0	9	3365
第 75 周	14	252	64	101	0	71	10415
第 76 周	21	288	104	105	0	115	16275
第 77 周	4	356	56	130	0	62	7670
第 78 周	16	368	40	172	0	44	8500
第 79 周	22	368	108	141	0	119	17095
第 80 周	28	432	148	157	0	163	22815
第 81 周	6	528	72	200	0	80	10200
第 82 周	18	536	40	252	0	44	9300
第 83 周	26	520	116	212	0	128	19140
第 84 周	33	576	168	219	0	185	26345
第 85 周	8	676	84	263	0	93	12245
第 86 周	17	680	28	321	0	31	8115

第 87 周	20	636	76	276	0	84	13820
第 88 周	28	640	120	268	0	132	20260
第 89 周	7	688	60	282	0	66	9470
第 90 周	18	668	32	316	0	36	8740
第 91 周	19	632	76	272	0	84	13600
第 92 周	33	636	136	266	0	150	23030
第 93 周	6	700	72	276	0	80	10580
第 94 周	18	688	24	329	0	27	7915
第 95 周	19	640	76	276	0	84	13620
第 96 周	34	644	140	270	0	154	23690
第 97 周	11	712	96	278	0	106	14250
第 98 周	17	724	36	336	0	40	9080
第 99 周	24	684	88	299	0	97	16065
第 100 周	30	696	136	286	0	150	22530
第 101 周	12	752	80	308	0	88	12820
第 102 周	17	744	36	344	0	40	9120
第 103 周	25	700	92	306	0	102	16750
第 104 周	0	716	0	308	0	0	1540
1-104 周总和	879	29770	3826	13102	131	4234	667560

附录四：问题三代码

```
import math
from openpyxl import    Workbook
from openpyxl import load_workbook
from openpyxl.utils import get_column_letter

container = [11, 5, 4, 7, 16, 6, 5, 7,
                    13, 6, 5, 7, 12, 5, 4, 6,
                    9, 5, 5, 11, 29, 21, 17, 20,
                    27, 13, 9, 10, 16, 6, 5, 7,
                    11, 5, 5, 6, 12, 7, 7, 10,
                    15, 10, 9, 11, 15, 10, 10, 16,
                    26, 21, 23, 36, 50, 45, 45, 49,
                    57, 43, 40, 44, 52, 43, 42, 45,
                    52, 41, 39, 41, 48, 35, 34, 35,
                    42, 34, 36, 43, 55, 48, 54, 65,
                    80, 70, 74, 85, 101, 89, 88, 90,
                    100, 87, 88, 89, 104, 89, 89, 90,
                    106, 96, 94, 99, 109, 99, 96, 102]

arm = []
for i in range(len(container)):
```

```
        arm.append(container[i] * 4)

initial_arm = 50
initial_container = 13
weeks_arm = []
weeks_container = []
maintenance_costs_arm, maintenance_costs_container = 0, 0
num1 = 0
num2 = 0
add_arm = []
add_container = []
weekly_damage_arm = []
weekly_damage_container = []
existing_arms = []

weekly_maintenance_arm = []
weekly_maintenance_container = []


v = []
total_cost = 0
a = 0
for i in range(len(container)):
        weekly_damage_container.append(int (container[i] * 0.1 + 0.5))
        weekly_damage_arm.append(4 * weekly_damage_container[i])
        if (i + 1 < len(arm)):
                s = arm[i] + arm[i + 1]
                if (s > initial_arm):
                        num1 = s - initial_arm
                        add_arm.append(num1)
                        a = math.ceil(num1 / 20)
                        v.append(num1 + a)
                        weekly_maintenance_arm.append(initial_arm - arm[i] - a)
                        initial_arm = s
                        weeks_arm.append(i)
                else:
                        add_arm.append(0)
                        v.append(0)
                        weekly_maintenance_arm.append(initial_arm - arm[i])

        if (container[i] > initial_container):
                num2 = container[i] - initial_container
                add_container.append(num2)
                initial_container = container[i]
```

```
            weeks_container.append(i - 1)
        else:
            add_container.append(0)
        weekly_maintenance_container.append(initial_container - container[i])
        initial_arm = initial_arm - weekly_damage_arm[i]

        if (i + 1 >= len(container)):
            break
        else:
            maintenance_costs_arm = maintenance_costs_arm + weekly_maintenance_arm[i] * 5
        maintenance_costs_container = maintenance_costs_container + (initial_container - container[i]) * 10
        initial_container = initial_container - weekly_damage_container[i]


        existing_arms.append(initial_arm - add_arm[i] + weekly_damage_arm[i])

add_arm.append(0)
add_container.remove(add_container[0])
add_container.append(0)
v.append(0)
existing_arms.append(existing_arms[102]-weekly_damage_arm[102]+add_arm[102])
weekly_maintenance_arm.append(existing_arms[len(existing_arms)-1] - arm[len(arm)-1])

maintenance_costs_arm = maintenance_costs_arm + weekly_maintenance_arm[len(weekly_maintenance_arm)-1] * 5
for i in range(len(add_arm)):
    print("第", i + 1, "周购买的操作手数量为：", add_arm[i])
for i in range(len(add_container)):
    print("第", i+1, "周购买的容器艇数量为：", add_container[i])

total_add_arm = sum(add_arm)
expenses_arm = sum(add_arm) * 100
expenses_container = sum(add_container) * 200
training_expenses_arm = sum(v) * 10
total_cost = expenses_arm + expenses_container + training_expenses_arm + maintenance_costs_arm + maintenance_costs_container

print("总共购买的操作手数量为：%d 个" % sum(add_arm))
print("总共购买的容器艇数量为：%d 个"% sum(add_container))
print("总成本：%d 元"% total_cost)
```

```
#周数（104 周）
weeks = []
for i in range(0, 104):
    weeks1 = "第%d 周"% (i+1)
    weeks.append(weeks1)

#统计每周总成本
weeks_cost = []
for i in range(0, 104):

weeks_cost.append(add_container[i]*200+add_arm[i]*100+weekly_maintenance_arm[i]
*5+weekly_maintenance_container[i]*10+v[i]*10)

# 绘制表格
wb = Workbook()
sheet = wb.active
sheet.title='问题三：1-104 周的结果数据'
sheet.append(['周数', '购买的容器艇数量','现有的操作手数量', '购买的操作手数量','保
养的操作手数量','保养的容器艇数量','参与训练的操作手数量','总成本（单位：元）
'])
for i in range(0,104):

sheet.append([weeks[i],add_container[i],existing_arms[i],add_arm[i],weekly_maintenanc
e_arm[i],weekly_maintenance_container[i],v[i],weeks_cost[i]])
sheet.append(["1-104 周总和
",sum(add_container),sum(existing_arms),sum(add_arm),sum(weekly_maintenance_arm)
,sum(weekly_maintenance_container),sum(v),sum(weeks_cost)])
wb.save('问题三：1-104 周的结果数据.xlsx')
wb = load_workbook('问题三：1-104 周的结果数据.xlsx')
sheet = wb.active

#表格列宽自适应
lks = []
for i in range(1, sheet.max_column + 1):
    lk = 1
    for j in range(1, sheet.max_row + 1):
        sz = sheet.cell(row=j, column=i).value
        if isinstance(sz, str):
            lk1 = len(sz.encode('gbk'))
        else:
            lk1 = len(str(sz))
        if lk < lk1:
            lk = lk1
        # print(lk)
```

```
        lks.append(lk)
# 第二步：设置列宽
for i in range(1, sheet.max_column + 1):
        k = get_column_letter(i)
        sheet.column_dimensions[k].width = lks[i - 1] + 2
wb.close()
wb.save('问题三：1-104 周的结果数据.xlsx')
```

附录五：问题三代码生成的表格

周数	购买的容器艇数量	现有的操作手数量	购买的操作手数量	保养的操作手数量	保养的容器艇数量	参与训练的操作手数量	总成本（单位：元）
第 1 周	0	50	14	5	2	15	1595
第 2 周	0	60	0	40	7	0	270
第 3 周	0	56	0	40	7	0	270
第 4 周	6	56	36	26	4	38	5350
第 5 周	0	88	0	24	0	0	120
第 6 周	0	80	0	56	8	0	360
第 7 周	0	76	0	56	8	0	360
第 8 周	2	72	8	43	5	9	1555
第 9 周	0	76	0	24	0	0	120
第 10 周	0	72	0	48	6	0	300
第 11 周	0	68	0	48	6	0	300
第 12 周	3	64	12	35	3	13	2135
第 13 周	0	72	0	24	0	0	120
第 14 周	0	68	0	48	6	0	300
第 15 周	0	64	0	48	6	0	300
第 16 周	0	64	0	40	4	0	240
第 17 周	0	60	0	24	0	0	120
第 18 周	0	56	0	36	3	0	210
第 19 周	5	52	12	31	2	13	2505
第 20 周	19	60	100	11	0	105	14905
第 21 周	0	156	44	37	0	47	5055
第 22 周	0	188	0	104	5	0	570
第 23 周	0	180	0	112	7	0	630
第 24 周	7	172	16	91	2	17	3645
第 25 周	0	180	0	72	0	0	360
第 26 周	0	168	0	116	11	0	690
第 27 周	0	164	0	128	14	0	780
第 28 周	0	160	0	120	12	0	720
第 29 周	0	156	0	92	5	0	510
第 30 周	0	148	0	124	13	0	750

第 31 周	0	144	0	124	13	0	750
第 32 周	0	140	0	112	10	0	660
第 33 周	0	136	0	92	5	0	510
第 34 周	0	132	0	112	10	0	660
第 35 周	0	128	0	108	9	0	630
第 36 周	0	124	0	100	7	0	570
第 37 周	0	120	0	72	0	0	360
第 38 周	0	116	0	88	4	0	480
第 39 周	1	112	0	84	3	0	650
第 40 周	6	108	0	68	0	0	1540
第 41 周	0	104	0	44	0	0	220
第 42 周	0	96	0	56	3	0	310
第 43 周	0	92	0	56	3	0	310
第 44 周	5	88	16	43	0	17	2985
第 45 周	0	100	0	40	0	0	200
第 46 周	0	92	0	52	3	0	290
第 47 周	5	88	16	47	2	17	3025
第 48 周	12	100	68	32	0	72	10080
第 49 周	0	160	28	54	0	30	3370
第 50 周	2	176	0	92	2	0	880
第 51 周	15	168	68	72	0	72	10880
第 52 周	18	228	116	78	0	122	16810
第 53 周	0	328	52	125	0	55	6375
第 54 周	5	360	0	180	0	0	1900
第 55 周	9	340	36	158	0	38	6570
第 56 周	13	356	68	156	0	72	10900
第 57 周	0	404	0	176	0	0	880
第 58 周	0	380	0	208	8	0	1120
第 59 周	1	364	0	204	7	0	1290
第 60 周	12	348	36	170	0	38	7230
第 61 周	0	368	12	159	0	13	2125
第 62 周	0	360	0	188	4	0	980
第 63 周	6	344	4	175	1	5	2535
第 64 周	12	332	56	149	0	59	9335
第 65 周	0	368	4	159	0	5	1245
第 66 周	0	352	0	188	6	0	1000
第 67 周	2	336	0	180	4	0	1340
第 68 周	11	320	36	154	0	38	6950
第 69 周	0	340	0	148	0	0	740
第 70 周	0	320	0	180	8	0	980
第 71 周	0	304	0	168	5	0	890
第 72 周	10	292	16	151	1	17	4535
第 73 周	0	292	12	123	0	13	1945

第 74 周	1	288	0	152	4	0	1000
第 75 周	11	276	40	130	0	42	7270
第 76 周	16	300	92	123	0	97	13985
第 77 周	0	376	36	154	0	38	4750
第 78 周	10	388	20	195	1	21	5195
第 79 周	16	388	88	167	0	93	13765
第 80 周	22	456	124	189	0	131	19055
第 81 周	0	552	48	229	0	51	6455
第 82 周	9	568	8	287	2	9	4145
第 83 周	18	548	88	247	0	93	14565
第 84 周	25	608	136	261	0	143	21335
第 85 周	0	708	52	301	0	55	7255
第 86 周	6	720	0	364	2	0	3040
第 87 周	11	684	28	330	0	30	6950
第 88 周	19	676	84	311	0	89	14645
第 89 周	0	724	24	322	0	26	4270
第 90 周	7	708	0	360	3	0	3230
第 91 周	10	672	36	318	0	38	7570
第 92 周	24	672	100	311	0	105	17405
第 93 周	0	736	36	318	0	38	5570
第 94 周	4	732	0	376	5	0	2730
第 95 周	10	696	20	339	0	21	5905
第 96 周	25	680	104	314	0	110	18070
第 97 周	1	748	60	321	0	63	8435
第 98 周	8	764	0	380	0	0	3500
第 99 周	14	724	48	345	0	51	9835
第 100 周	20	736	96	335	0	101	16285
第 101 周	1	792	40	354	0	42	6390
第 102 周	7	788	0	392	0	0	3360
第 103 周	16	748	44	361	0	47	9875
第 104 周	0	752	0	344	0	0	1720