

基于混合整数线性规划的机械零件生产与投资计划问题的研究及制定

摘要

在当今竞争激烈的制造环境中，有效的生产计划、库存管理和维护计划对于实现运营和最大化盈利至关重要。该工厂生产七种机械零部件，使用五种不同能力和维护需求的机床运作，并实现在六个月的规划周期内获得最大化总利润，同时满足生产需求、库存限制和设备维护需求。因而我们的目标是开发一个综合的数学模型，将生产计划、库存控制和设备维护计划整合起来，以在指定的规划周期内实现最高可能的利润。

针对问题一，我们构建了线性规划（LP）模型。以 为决策变量，确定目标函数，即就是最大化利润，存在多个约束条件，例如零件销售上线，设备维修，机床可用工时，以及最终库存量应为 60，库存上限等等。题目中的约束条件以及目标函数均可以用线性方程表示，通过对线性规划（LP）模型求解，最终得到最大利润为 92931 元，以及满足条件的生产规划计划。

针对问题二，该工厂要求在考虑投资扩大生产的情况下，给出不同投资数量下应购买哪些机床设备以及增加的利润。对于每种情况，基于问题一的模型，在此基础上迭代搜索方法，将所有情况列举出，并求出增加一台及两台三台时的利润，可得增加一台时增加最大利润增加的利润是 13320 元，增加两台时最大增加得利润是 18630 元，当增加数量为三台时最大增加的利润是 20931 元。

针对问题三，本文构建了一个（0，1）规划模型，在问题一线性规划（LP）模型的基础上，保证不增加新设备，并且不影响设备维修的情况，寻找一个最佳维修方案，使得最终利润最大值。设备维修与否可以看成一个（0，1）规划模型进行求解，在三层迭代搜索中，通过对利润最大值的不断更新，计数器 $count_{mj}$ 的搜索，成功找到最大利润的安排方案，此时利润应为 101120 元。

关键词：优化模型；线性规划；迭代搜索；整数规划；敏感性分析

目 录

基于混合整数线性规划的机械零件生产与投资计划问题的研究及制定	0
摘 要	0
一、问题重述	2
二、模型假设	2
三、问题分析	2
3.1 对问题一的分析	2
3.2 对问题二的分析	3
3.3 对问题三的分析	3
四、符号说明	4
五、模型建立与求解	4
5.1 问题一的模型建立与求解	4
5.2 问题二的模型建立和求解	6
5.3 问题三的模型建立和求解	8
六、模型的评价与改进	10
6.1 模型的评价	10
6.2 模型的改进	10
七、参考文献	11
八、附录	12

一、问题重述

某工厂生产七种机械零件产品，该厂有五种机床设备，数量有限但不同。一种零件需要多个工序，每个工序需要不同的机床设备进行，但生产零件过程中各个工序没有次序要求。工厂对于零件的库存也有所要求，另外需要考虑不同月份由不同的机床需要整修，在此期间，接受整修的机床不能用来进行生产活动。不在整修时间段的机床，每月每天都有固定的工作时长。

1. 制定一个六个月的生产和库存计划，在满足题目条件的基础上，可以获得最大化利润的生产销售库存计划。

2. 该工厂考虑投资扩大生产，分别考虑多购买一台、两台和三台新的机床设备时，如何选择购买机床才能达到利润的最大化。求出购买计划和所增加的利润

3. 该工厂考虑是否可以通过不购买新设备，用改变机床设备的整修方案来提高总利润，构造一个新的机床设备整修计划数学模型，使得总利润尽可能大。

二、模型假设

为了对模型进行合理简化，本文建立了以下的模型假设：

1. 假设本题所提供的附件数据均真实有效，且具有相当的计算分析价值。
2. 假设工厂用于生产库存销售的机床设备和零件不会因为无关变量的影响。
3. 假设生产能力、销售限制、库存成本等与生产数量和销售数量之间的关系是线性的。
4. 假设模型所提及的时间周期内，生产和销售相关的基本参数（如生产成本、销售价格、库存成本等）保持恒定不变。
5. 假设工厂新购入的机床与原有机床在生产方面不存在差异

三、问题分析

3.1 对问题一的分析

为了制定一个六个月的生产和库存计划以最大化利润，需要对销售需求进行分析和预测，考虑生产成本和库存成本，分析库存约束，应用优化算法建

立线性规划模型，保证其满足一月份到五月份其每种零件的库存都不超过 100 个，且在六月份时，每种零件的库存均为 60 个。根据题目信息，可以得到，每种零件在每一月份有不同的销售上限，需保证销售量不超过销售上限。其次，该工厂的机床设备的工作时间为每月 22 天，每天 8 小时，且每种零件的生产需经过多种工序，各工序所需的时间机床设备各有不同。通过综合考虑这些因素，可以建立一个有效的计划，以最大化利润并确保生产、销售和库存之间的协调与平衡。最后，本文选择利用 python 语言进行编程实现，求出最优计划。

3.2 对问题二的分析

针对问题二，题目为生产计划优化问题。与第一问进行类比，该问通过购买一台两台三台的新机床设备来影响利润和生产销售库存计划。重新调整生产计划以适应新增设备的生产能力，结合优化模型，同时计算新增设备的预期收益，综合分析这些因素，制定出合理的机床设备购买计划，达到不同条件的最大化利润。最后，本文基于问题一的模型利用迭代搜索算法的内容进行求解，利用问题一所使用的程序，通过改变机床设备的数量达到计划的实施，从而求出达到最大化利润的机床设备购买计划。

3.3 对问题三的分析

该题目标是通过合理调配不同月份的机床维修策略，以实现最大化利润。这个问题包含了时间、机床类型和机床数量的多维度约束，需要通过线性规划模型进行建模，并通过整数规划确保机床的维修状态只能是开或关（即(0,1)整数规划），本文通过采取嵌套三层循环来遍历所有可能的机床使用策略。最外层循环迭代月份，中间层循环遍历不同的机床类型，最内层循环则考虑每种机床在当前月份使用的个数。在每次内层循环中，根据当前机床的维修情况（(0,1)整数规划），构建对应的线性规划问题，并求解得到在当前月份和机床维修情况下的最大利润。

四、符号说明

符号	符号意义	备注/单位
i	零件的种类	
A_{ij}	各月各零件的销售矩阵	
B_{im}	每种零件各工序的时间矩阵	
C_{mj}	每月设备数的矩阵	
$count_{mj}$	计数器	
j	生产的月份	
x_{ij}	第 i 种零件在第 j 个月的生产量	个
y_{ij}	第 i 种零件在第 j 个月的库存量	个
s_{ij}	第 i 种零件在第 j 个月的销售量	个
p_i	第 i 种零件的利润	元
t_{im}	机床 m 加工单个零件 i 的时间	小时/个
H_{mj}	第 m 种机床在第 j 个月的数量	个
Sum_{max}	最大总利润	元
T_m	单个机床 m 每个月的工时上限	小时
Z_m	第 m 种机床购入的个数	个
N	新增机床的总量	个

注：符号说明中未提到的符号，在本文中有具体解释

五、模型建立与求解

5.1 问题一的模型建立与求解

5.1.1 建模思路

对于问题一，考虑到生产、库存与销售之间的紧密关联，它们之间的关系呈现出明显的线性特征。生产成本、销售收入以及库存持有成本等因素，在大多数情况下，都与相关变量呈线性关系。因此，我们选择了线性规划模型作为解决问题的有效工具。线性规划模型能够精准地描述这些线性关系，并通过优化算法找到最佳的生产、库存和销售策略，从而实现利润最大化或成本最小化的目标。

5.1.2 模型的建立

问题一要求求解以每种产品每月的生产数量，每月的库存数量和每种零件每月的销售数量为决策变量，在以下多种约束条件下，确立目标函数，即写出决策变量与目标函数之间的关系式，即：

$$Sum_{max} = \sum_{i=1}^7 \sum_{j=1}^6 s_{ij} * p_i - 0.5 * \sum_{i=1}^7 \sum_{j=1}^6 y_{ij}$$

已知每种零件在每个月份有不同的销售上限，由题目中所给的数据，可以得到以下约束条件：

$$\begin{cases} s_{ij} \leq y_{ij-1} + x_{ij} \\ s_{ij} \leq A_{ij} \\ s_{ij} \geq 0 \end{cases}$$

该工厂的工人每天操作机床工作八小时，同时为了简化问题，每月都工作 22 天，同时考虑机床整修，可得出如下限制条件：

$$\begin{cases} \sum_{i=1}^7 t_{im} x_{ij} \leq 22 * 8 (m = 1, 2, \dots, 5; j = 1, 2, \dots, 6) \\ t_{im} = B_{im} \end{cases}$$

各种机床的实际工作的数量由总数减去整修的机床的数，因此，机床的设计及数量由此可得：

$$H_{mj} = C_{mj}$$

由题意可以得出，要求其库存满足从一月份到五月份，库房中每种零件的库存都不超过 100 个，且在六月份时，每种零件的库存均为 60 个。可以得出以下约束条件：

$$\begin{cases} y_{ij} = y_{ij-1} + x_{ij} - s_{ij} (j = 1, 2, \dots, 6) \\ y_{ij} \leq 100 \\ y_{i6} = 60 \\ y_{i0} = 0 \end{cases}$$

5.1.3 模型的求解

对模型建立之后，本文通过 python 进行求解，导入所需要 pulp 库（python 源代码见附录 1），经过代码运行，即求得最大利润是 92931 元，决策方案如（表 5.1）所示（详细运行结果见附录 2）。

表 5.1 最大利润时生产计划（个）

	零件 1	零件 2	零件 3	零件 4	零件 5	零件 6	零件 7
1 月	474	0	200	0	700	300	0
2 月	500	345	300	300	500	200	250
3 月	200	500	400	200	500	0	300
4 月	300	400	0	400	300	500	100
5 月	0	200	300	200	900	200	0
6 月	400	300	0	300	0	0	0

5.2 问题二的模型建立和求解

5.2.1 建模思路

问题二考虑投资扩大生产，假设每种机床投资相同。题目需要确定在购买一台、两台和三台新机床的情况下，应该购买哪些机床设备以及增加的利润。考虑到生产过程中的各种约束条件和目标函数的线性特性，可以将这个问题有效地建模为一个线性规划问题。为了解决这个问题，本文构建一个线性规划模型。通过这个模型，可以精确地分析出在不同投资规模下，购买哪些机床设备能够带来最

大的利润增长。这不仅有助于工厂做出明智的投资决策，还能确保工厂的生产扩张计划能够在满足所有约束条件的同时，实现利润的最大化。

5.2.2 模型建立

首先，定义一个新的决策变量 $Z_m(m = 1,2, \dots, 5)$ ，且满足条件

$$\sum_{m=1}^5 Z_m = N(N = 1,2,3)$$

接着，继续使用问题一中所使用的目标函数，决策变量以及约束条件。

5.2.3 模型的求解和分析

在扩大生产规模的过程中，逐步增加机床数量是一个关键的策略。为了找到在不同机床增加数量下生产利润的最大化方案，本文将采用枚举的方法，通过对各个机床进行变量赋值，来逐一计算并比较各种组合下的利润情况。

具体来说，文章会分别考虑增加一台、两台和三台机床时的情景。针对每种情况，都会基于第一问中的 Python 代码进行变量赋值运算。通过这种方式，可以精确地找到在每种机床数量增加情况下，能够带来最大利润的生产方案。

接下来，将分别对增加一台、两台和三台机床时的最优解进行求解和分析。这一过程不仅能够帮助我们明确在不同投资规模下的最优机床配置，还能够为我们提供扩大生产规模的具体路径和策略。（增加一、二、三台机床时，所得利润见附录 3）结果如下（表 5.2）：

表 5.2 机床分配方式及利润增加

	机床 1	机床 2	机床 3	机床 4	机床 5	最大利 润	增加利 润
加一台	0	0	0	0	1	106251	13320
加二台	1	0	0	0	1	111561	18630
加三台	2	0	0	0	1	113826	20931

5.3 问题三的模型建立和求解

5.3.1 建模思路

整合题目所给维修要求，很容易得到五个机床分别需要的维修个数分别为 3, 2, 2, 1, 1。新建 $\text{Re}_{m6} = (3, 2, 2, 1, 1)$ ，其元素对应每台机床在年底应达到的维修次数，我们仅需在第六个月月末时，满足相应条件，便可以求解，在第一题线性规划的模型的基础上，建立 (0,1) 整数规划模型，新建变量 count_{mj} ，用来计数。

5.3.2 模型建立

根据题目描述，机床的维修状态可以简化为两个变量：维修或不维修，这恰好符合 (0, 1) 整数规划的要求。因此，我们可以构建一个 (0, 1) 整数规划模型，以确定每台机床在每个月份是否进行维修。如下（其中 G_{mj} 表示 m 号机床在 j 月是否维修）：

$$G_{mj} = \begin{cases} 0 & \text{未维修机床} \\ 1 & \text{维修机床} \end{cases} \quad (m = 1, 2, \dots, 5; j = 1, 2, \dots, 6)$$

当 $G_{mj}=1$ 时，采用计数器 count_{mj} 计数求和：

$$\text{count}_{mj} = \text{count}_{mj-1} + G_{mj}$$

此时我们仅需满足如下条件即可：

$$G_{m6} = (3, 2, 2, 1, 1) = \text{Re}_{m6}$$

有了这些条件和约束，本文在第一问的线性规划基础上，结合 (0, 1) 整数规划，通过迭代搜索算法来寻找满足条件的最大利润维修方式。在搜索过程中，去掉那些与问题无关的约束，以简化计算和提高效率。最终，通过迭代搜索，本文找到一种维修方式，使得在满足所有约束条件的前提下，实现利润的最大化。这样不仅能够帮助确定每台机床在每个月份是否进行维修，还能确保维修次数符合预定要求，从而实现整体利润的最大化。通过整合线性规划和整数规划的优势，能够得到一个既高效又准确的解决方案。

5.3.3 模型的求解和分析

在这种场景下，本文目标是通过合理调配不同月份的机床维修策略，以实现最大化利润。这个问题包含了时间、机床类型和机床数量的多维度约束，需要通过线性规划模型进行建模，并通过整数规划确保机床的维修状态只能是开或关（即 0 或 1）。

首先，建立线性规划模型。线性规划模型通常由目标函数和一系列约束条件组成。在这个问题中，目标函数是最大化总利润，约束条件包括机床的数量限制、生产能力的限制以及机床维修状态的整数约束（0 或 1）。

接下来，本文将嵌套三层循环来遍历所有可能的机床使用策略。最外层循环迭代月份，中间层循环遍历不同的机床类型，最内层循环则考虑每种机床在当前月份使用的个数。在这个过程中，利用前两问中建立的线性规划模型，通过调用线性规划求解器来求解当前策略下的最大利润。在每次内层循环中，根据当前机床的维修情况（0 或 1），构建对应的线性规划问题，并求解得到在当前月份和机床维修情况下的最大利润。

最后，本文将求解得到的利润与上次迭代的结果进行比较。如果是第一次迭代，将其与第一问中不考虑时间约束的线性规划结果进行比较。通过比较，可以确定当前策略是否比之前的策略更优。如果是更优的，就要更新最大利润和对应的机床维修策略。

当所有迭代都完成时，即可得到了在所有月份和机床类型下能够获得的最大利润，以及对应的机床使用策略。

通过迭代搜索（改变的部分 python 代码见附录 4），我们找到了最大利润应为 101220 元，维修安排如表 5.3 所示：

表 5.3 维修安排

	机床 1	机床 2	机床 3	机床 4	机床 5
1 月	1	0	0	0	0

2 月	1	1	0	0	1
3 月	0	1	0	1	0
4 月	1	0	1	0	0
5 月	0	0	0	0	0
6 月	0	0	1	0	0

六、模型的评价与改进

6.1 模型的评价

结合线性规划和 $(0, 1)$ 整数规划的方法来解决机床调度问题是一个复杂但高效的过程，该模型的建立与求解是系统的，合理的，具有相当的参考价值与意义，故其结果是较为正确的。其次，该模型中使用的算法和程序等是在充分考虑了所有约束条件后得到的，考虑内容较为全面，具有研究分析的意义。但是，在该模型的建立过程中，有许多不足之处。比如：机床设备会因为工作磨损等导致生产效率的降低。此外，零件的利润也不是一成不变的，其利润应随着时长需求的变化而变化。包括销售上限也应收到多方面因素的影响，例如市场的需求和加工零件的原材料的成本。

6.2 模型的灵敏度分析

在以上模型中，本文以总利润为目标函数和多个约束条件建立了线性规划模型。其中，本文对单个零件的利润，单个零件不同工序的加工时长和机床设备的工作时间分别进行改变，即使用控制变量法探讨不同因素对目标函数的影响。将数据改变并填充入程序当中计算得出，相关参数的改变并不会对模型的最优解造成显著影响。分析几种变量可以得知，其与目标函数之间的关系都是线性的，对模型的影响足够的小。由此可以得出该模型受这三种因素影响较小，灵敏度较好。

此外，本文考虑了影子价格对目标函数所带来的影响，影子价格值得是加工零件所需的原材料的稀缺程度改变对于零件价格的变动。而影子价格在改题目中

较难预测，但可以得出的是，影子价格变化必定是非线性的，将其变化引入目标函数中，可以得出该模型的最优解发生改变。由此可知，该模型对于影子价格，即对原材料的稀缺程度和价格的灵敏度较差。

七、参考文献

- [1]王海龙,刘飘飘,王易平.Python 在数学建模中的应用初探[J].邯郸职业技术学院学报,2023.36(04):31-33+80.
- [2]谢承义.数学建模中线性规划与目标规划的比较[J].科技创业月刊,2011,24(16):128-130.
- [3]王斌.从线性规划谈中学数学建模[J].宁德师专学报(自然科学版),2008,(02):185-188.
- [4]葛志利.数学建模中最优化方法的研究[J].广西职业技术学院学报,2011,4(01):8-11.
- [5]曲孝海,梅汉飞.数学建模解的分析与检验[J].湖南文理学院学报(自然科学版),2012,24(03):12-14.

八、附录

附录 1

问题一（采取的 python 进行编程）

```
1. # 导入所需库
2. import pulp
3.
4. prob = pulp.LpProblem("Production_Planning", pulp.LpMaximize)
5.
6.
7. # 定义利润、机床工时和销售上限
8. Profits = [10, 9, 3, 5, 11, 9, 8]
9. MachineTimes = [
10.     [0.6, 0.1, 0.2, 0.05, 0],
11.     [0.7, 0.1, 0, 0.08, 0],
12.     [0, 0, 0.4, 0, 0.01],
13.     [0.3, 0.3, 0, 0.06, 0],
14.     [0.6, 0, 0.2, 0.1, 0.05],
15.     [0, 0.6, 0, 0, 0.08],
16.     [0.5, 0, 0.6, 0.08, 0.05]
17. ]
18. SalesLimits = [
19.     [600, 800, 200, 0, 700, 300, 200],
20.     [500, 600, 300, 300, 500, 200, 250],
21.     [200, 500, 400, 200, 500, 0, 300],
22.     [300, 400, 0, 400, 300, 500, 100],
23.     [0, 200, 300, 200, 900, 200, 0],
24.     [400, 300, 100, 300, 800, 400, 100]
25. ]
26. NumMachines = [6, 3, 4, 2, 1] # 五种机床各自的数量
27.
```

```

28. # 定义产品数量、月份和机床类型等常量
29. num_products = 7
30. num_months = 6
31. num_machine_types = len(NumMachines) # 假设 NumMachines 是一个定义了机床类型数量的列表
32.
33. # 定义机器设备数量和每月工作天数
34. working_days_per_month = 22 # 每月工作天数
35. working_hours_per_day = 8 # 每天工作小时数
36. total_months = 6 # 总月数
37.
38. # 定义每个月需要维修的设备
39. MaintenanceMonths = {
40.     1: {1: 2}, # 一月：2 台机床 1 整修
41.     2: {2: 1}, # 二月：1 台机床 2 整修
42.     3: {4: 1}, # 三月：1 台机床 4 整修
43.     4: {3: 1}, # 四月：1 台机床 3 整修
44.     5: {1: 1, 2: 1}, # 五月：1 台机床 1 和 1 台机床 2 整修
45.     6: {3: 1, 5: 1} # 六月：1 台机床 3 和 1 台机床 5 整修
46. }
47.
48. # 计算每月每种机床的可用工时
49. available_machine_hours = {
50.     month: {
51.         machine_type: (NumMachines[machine_type - 1]
52.             - maint.get(machine_type, 0)) * working_days_per_month * working_hours_per_day
53.     } for machine_type in range(1, len(NumMachines) + 1)
54. } for month, maint in MaintenanceMonths.items()
55.

```

```

56. # 填充没有整修数据的月份
57. for month in range(1, 7): # 假设有6个月，从1月到6月
58.     if month not in available_machine_hours:
59.         available_machine_hours[month] = {
60.             machine_type: NumMachines[machine_type
        - 1] * machine_hours_per_month
61.             for machine_type in range(1, len(NumM
        achines) + 1)
62.         }
63.
64. # 初始化库存量（一月初所有零件库存量为0）
65. Inventory = [[0] * 7 for _ in range(6)]
66. # 库存目标（六月底每种零件的库存量为60个）
67. InventoryTargets = [60] * 7
68.
69. # 设置线性规划问题
70. prob = pulp.LpProblem("Maximize_Profit", pulp.LpMaximize)
71.
72. # 定义决策变量
73. # 每个产品每月的生产数量
74. product_vars = pulp.LpVariable.dicts("Product", ((month, produc
        t) for month in range(1, num_months + 1) for product in
        range(num_products)), lowBound=0, cat='Continuous')
75. # 添加库存决策变量
76. InventoryVars = pulp.LpVariable.dicts("Inventory", ((month, produ
        ct) for month in range(1, num_months + 1) for product in
        range(num_products)), lowBound=0, cat='Continuous')
77. # 创建销售量变量字典，一月到六月，七种零件
78. SalesVars = pulp.LpVariable.dicts("Sales", ((i, j) for i in
        range(1, num_months + 1) for j in range(1, num_products
        + 1)), lowBound=0, cat='Integer')
79. # 目标函数：最大化总利润
80. prob += pulp.lpSum([Profits[product] * product_vars[(month,
        product)]]

```

```

81.                                     for month in range(1,
                                     num_months + 1)
82.                                     for product in range(num_products))]
83.
84. # 约束条件
85. # 1. 库存限制条件
86. for month in range(1, num_months + 1):
87.     for product in range(num_products):
88.         if month < 6: # 对于一月到五月
89.             prob += InventoryVars[(month, product)]
90.                 <= 100, f"InventoryLimit_{month}_{product}"
91.         else: # 对于六月
92.             prob += InventoryVars[(month, product)]
93.                 == 60, f"InventoryLimit_{month}_{product}"
94.
95. # 2. 销售上限约束
96. for month in range(1, num_months + 1):
97.     for product in range(num_products):
98.         prob += product_vars[(month, product)] <= SalesLimits[month-1][product], f"SalesLimit_{month}_{product}"
99.
100. # 3. 机床工时约束
101. for month in range(1, num_months + 1):
102.     for machine_type in range(1, num_machine_types + 1):
103.         prob += pulp.lpSum([MachineTimes[product][machine_type-1] * product_vars[(month, product)]
104.                               for product in range(num_products)]) <= available_machine_hours[month][machine_type], f"MachineTime_{month}_{machine_type}"
105.
106. # 求解问题
107. prob.solve()

```



```
106.  
107. # 输出结果  
108. print("生产决策: ")  
109. for month in range(1, num_months + 1):  
110.     print(f"月份 {month}:")  
111.     for product in range(num_products):  
112.         print(f"    产  
    品 {product+1}: {product_vars[(month, product)].value()} 个  
    ")  
113. print(f"总利润: {pulp.value(prob.objective)}")
```

附录 2

运行结果

生产决策:

月份 1:

产品 1: 473.33333 个

产品 2: 0.0 个

产品 3: 200.0 个

产品 4: 0.0 个

产品 5: 700.0 个

产品 6: 300.0 个

产品 7: 0.0 个

月份 2:

产品 1: 500.0 个

产品 2: 344.28571 个

产品 3: 300.0 个

产品 4: 300.0 个

产品 5: 500.0 个

产品 6: 200.0 个

产品 7: 250.0 个

月份 3:

产品 1: 200.0 个

产品 2: 500.0 个

产品 3: 400.0 个

产品 4: 200.0 个

产品 5: 500.0 个

产品 6: 0.0 个

产品 7: 300.0 个

月份 4:

产品 1: 300.0 个

产品 2: 400.0 个

产品 3: 0.0 个

产品 4: 400.0 个

产品 5: 300.0 个

产品 6: 500.0 个

产品 7: 100.0 个

月份 5:

产品 1: 0.0 个

产品 2: 200.0 个

产品 3: 300.0 个

产品 4: 200.0 个

产品 5: 900.0 个

产品 6: 200.0 个

产品 7: 0.0 个

月份 6:

产品 1: 400.0 个

产品 2: 300.0 个

产品 3: 0.0 个

产品 4: 300.0 个

产品 5: 0.0 个

产品 6: 0.0 个

产品 7: 0.0 个

总利润: 92931.90469 元

附录 3

机床	1	2	3	4	5	
加一 台	1					98061.42861
		1				92931.90469
			1			92931.90469
				1		92931.90469
					1	106251.9047
加两 台	2					100362.8571
		2				92931.90469
			2			92931.90469
				2		92931.90469
					2	106251.9047
	1	1				98061.42861
	1		1			98061.42861
	1			1		98061.42861
	1				1	111561.4286
		1	1			92931.90469
		1		1		92931.90469
		1			1	106251.9047
			1	1		92931.90469
			1		1	106251.9047
				1	1	106251.9047
加三 台	3					102625.7143
		3				92931.90469

			3			92931.90469
				3		92931.90469
					3	106251.9047
	2	1				100362.8571
	2		1			100362.8571
	2			1		100362.8571
	2				1	113862.8571
		2	1			92931.90469
		2		1		92931.90469
		2			1	106251.9047
	1	2				98061.42861
			2	1		92931.90469
			2		1	106251.9047
		1	2			92931.90469
	1		2			98061.42861
				2	1	106251.9047
			1	2		92931.90469
		1		2		92931.90469
	1			2		98061.42861
				1	2	106251.9047
			1		2	106251.9047
		1			2	106251.9047
	1				2	111561.4286
	1	1	1			98061.42861
	1	1		1		98061.42861
	1	1			1	111561.4286
	1		1	1		98061.42861
	1		1		1	111561.4286
	1			1	1	111561.4286
		1	1	1		92931.90469
		1	1		1	106251.9047

		1		1	1	106251.9047
			1	1	1	106251.9047

附录 4

```

1. # 机床数量
2. num_machines = {
3.     1: 6,
4.     2: 3,
5.     3: 4,
6.     4: 2,
7.     5: 1
8. }
9.
10. # 需要维修的机床数量
11. maintenance_needed = {
12.     1: 3,
13.     2: 2,
14.     3: 2,
15.     4: 1,
16.     5: 1
17. }
18.
19.
20. # 初始化决策变量（这里假设维修是整月进行的，因此是二进制的）
21. maintenance_vars = pulp.LpVariable.dicts("Maintenance",
22.                                           ((machine_type, month) for machine_t
23.                                           type in maintenance_needed for month in
24.                                           range(1, months + 1)), cat='Binary')
25.
26. prob.solve()
27.
28. # 输出结果
29. print("Status:", pulp.LpStatus[prob.status])
30. for v in prob.variables():
31.     print(v.name, "=", v.varValue)

```