

题目

B 题 生产过程中的决策问题

某企业生产某种畅销的电子产品，需要分别购买两种零配件（零配件 1 和零配件 2），在企业将两个零配件装配成成品。在装配的成品中，只要其中一个零配件不合格，则成品一定不合格；如果两个零配件均合格，装配出的成品也不一定合格。对于不合格成品，企业可以选择报废，或者对其进行拆解，拆解过程不会对零配件造成损坏，但需要花费拆解费用。请建立数学模型，解决以下问题：

问题 1 供应商声称一批零配件（零配件 1 或零配件 2）的次品率不会超过某个标称值。企业准备采用抽样检测方法决定是否接收从供应商购买的这批零配件，检测费用由企业自行承担。请为企业设计检测次数尽可能少的抽样检测方案。

如果标称值为 10%，根据你们的抽样检测方案，针对以下两种情形，分别给出具体结果：

- (1) 在 95% 的信度下认定零配件次品率超过标称值，则拒收这批零配件；
- (2) 在 90% 的信度下认定零配件次品率不超过标称值，则接收这批零配件。

问题 2 已知两种零配件和成品次品率，请为企业生产过程的各个阶段作出决策：

- (1) 对零配件（零配件 1 和/或零配件 2）是否进行检测，如果对某种零配件不检测，这种零配件将直接进入装配环节；否则将检测出的不合格零配件丢弃；
- (2) 对装配好的每一件成品是否进行检测，如果不检测，装配后的成品直接进入市场；否则只有检测合格的成品进入市场；
- (3) 对检测出的不合格成品是否进行拆解，如果不拆解，直接将不合格成品丢弃；否则对拆解后的零配件，重复步骤(1)和步骤(2)；
- (4) 对用户购买的不合格品，企业将无条件予以调换，并产生一定的调换损失（如物流成本、企业信誉等）。对退回的不合格品，重复步骤(3)。

请根据你们所做的决策，对表 1 中的情形给出具体的决策方案，并给出决策的依据及相应的指标结果。

问题 3 对 m 道工序、 n 个零配件，已知零配件、半成品和成品的次品率，重复问题 2，给出生产过程的决策方案。图 1 给出了 2 道工序、8 个零配件的情况，具体数值由表 2 给出。

图 1 两道工序、8 个零配件的组装情况

表 2 企业在生产中遇到的情况（问题 3）

针对以上这种情形，给出具体的决策方案，以及决策的依据及相应指标。

问题 4 假设问题 2 和问题 3 中零配件、半成品和成品的次品率均是通过抽样检测方法（例如，你在问题 1 中使用的方法）得到的，请重新完成问题 2 和问题 3。

附录 说明

- (1) 半成品、成品的次品率是将正品零配件（或者半成品）装配后的产品次品率；
- (2) 不合格成品中的调换损失是指除调换次品之外的损失（如：物流成本、企业信誉等）。
- (3) 购买单价、检测成本、装配成本、市场售价、调换损失和拆解费用的单位均为元/件

题目简析：

问题背景：该题目涉及企业生产管理，尤其是在产品零部件采购和生产的各个环节中做出最佳决策。重点是如何通过检测和拆解策略最小化成本，同时控制产品次品率。

关键问题：

- 设计合理的抽样检测方案，帮助企业判断零部件的次品率是否超过标称值。
- 在生产过程中决定是否对零部件和成品进行检测，以及是否拆解不合格品。
- 综合考虑多个工序和零部件的情况，给出最优生产决策。

建模思路：

- 利用统计学中的抽样检验理论，设计低成本的抽样方案。
- 建立一个成本收益模型，综合考虑零件的次品率、检测成本、拆解费用和市场售价。
- 对于多工序生产，使用动态规划或线性规划等方法，优化整个生产流程的决策。

做题思路

问题1：抽样检测方案的设计

目标是为企业设计抽样检测方案，以尽可能少的检测次数判定零配件次品率是否超过标称值。

- 假设检测过程服从二项分布，即抽取一定数量的样本，根据样本中不合格零配件的数量推断整体次品率。
- 根据标称值和信度水平，应用统计学中的假设检验理论。这里可以采用假设检验的双侧检验，设置：
零假设：次品率不超过标称值
备择假设：次品率超过标称值
- 检验方案设计：
通过计算置信区间，结合企业的次品率要求以及信度水平（95%和90%），确定需要的样本量。
可采用样本量计算公式来确定最少的检测次数，确保在给定信度下可以得出正确结论。

计算公式（论文需体现）：其他公式问chat

2. 样本量计算公式：使用以下公式计算样本量：

$$\text{样本量} = \frac{(z_{\alpha/2})^2 \cdot p \cdot (1 - p)}{\text{误差范围}^2}$$

其中， $z_{\alpha/2}$ 是对应置信水平的z值， p 是标称次品率（如10%），误差范围可以设定为5%。

参考代码：

（需修改过查重）

```
import numpy as np
import scipy.stats as stats

def binomial_test(sample_size, defective_items, defective_rate,
confidence_level):
    """
    使用二项分布进行假设检验，判断是否接受这批零配件。
    :param sample_size: 样本大小
```

```

:param defective_items: 样本中次品的数量
:param defective_rate: 标称次品率（如 10%）
:param confidence_level: 检验的置信水平
:return: 是否接收零配件
"""

# 计算显著性水平
alpha = 1 - confidence_level

# 使用新的 binomtest 进行检验，计算拒绝零假设的 p 值
result = stats.binomtest(defective_items, sample_size, defective_rate,
alternative='greater')

# 判断是否拒绝零假设
if result.pvalue < alpha:
    return f"拒收零配件 (p-value: {result.pvalue:.5f})"
else:
    return f"接受零配件 (p-value: {result.pvalue:.5f})"

def calculate_min_sample_size(defective_rate, confidence_level, error_margin):
    """
    计算最小样本量，满足置信水平和误差率要求
    :param defective_rate: 标称次品率
    :param confidence_level: 置信水平
    :param error_margin: 允许的误差率
    :return: 最小样本量
    """

    # 根据置信水平计算 Z 分数
    z_score = stats.norm.ppf(1 - (1 - confidence_level) / 2)

    # 计算最小样本量的公式
    sample_size = (z_score**2 * defective_rate * (1 - defective_rate)) /
(error_margin**2)

    # 返回向上取整后的样本量
    return int(np.ceil(sample_size))

# 设定标称次品率和置信水平
defective_rate = 0.10 # 标称次品率为 10%
confidence_level_95 = 0.95 # 95%的置信水平
confidence_level_90 = 0.90 # 90%的置信水平
error_margin = 0.05 # 允许的误差率为 5%

# 计算最小样本量
sample_size_95 = calculate_min_sample_size(defective_rate, confidence_level_95,
error_margin)
sample_size_90 = calculate_min_sample_size(defective_rate, confidence_level_90,
error_margin)

print(f"95% 置信水平下的最小样本量: {sample_size_95}")
print(f"90% 置信水平下的最小样本量: {sample_size_90}")

# 假设我们检测到样本中的次品数量
detected_defective_items = 12 # 假设样本中有 12 个次品

# 对两种置信水平进行二项分布假设检验

```

```
result_95 = binomial_test(sample_size_95, detected_defective_items,
defective_rate, confidence_level_95)
result_90 = binomial_test(sample_size_90, detected_defective_items,
defective_rate, confidence_level_90)

print(f"95% 置信水平结果: {result_95}")
print(f"90% 置信水平结果: {result_90}")
```

问题2：生产过程中的决策

为生产过程中不同阶段做出决策，包含零配件检测、成品检测、不合格成品处理等。

1. 零配件检测：

计算检测成本与不检测带来的潜在风险损失（如不合格零配件进入装配过程会导致更多的成品不合格）。

通过分析次品率、检测成本等因素，决定是否对零配件进行检测。若检测成本较高且次品率较低，可能选择不检测。

2. 成品检测：

需要权衡成品的市场售价、检测成本以及调换不合格成品的损失。通过建模计算各决策方案下的总成本，找出最优的检测方案（检测或不检测）。

3. 不合格成品拆解：

分析拆解费用与次品率，判断拆解零配件的价值是否高于直接报废不合格成品。对于拆解费用高但零配件较贵的情况，可以选择拆解，否则直接报废。

4. 退回的不合格品处理：

同样使用成本效益分析，对退回产品的处理（拆解或丢弃）做出最优决策。

整体模型：成本效益分析模型

（总成本=检测成本+装配成本+次品带来的损失合格品的收益）

使用动态规划算法：

由于生产过程具有多个阶段，且每个阶段的决策都会影响后续阶段，可使用动态规划来逐步求解最优策略。

具体步骤：

1. 零配件1的检测：

- 如果检测零配件1，成本增加为2元。如果不检测，后续可能出现次品。

成本1=min(检测成本1+次品损失1,不检测的损失)

成本1=min(检测成本1+次品损失1,不检测的损失) $\text{成本1} = \min(\text{检测成本}1 + \text{次品损失}1, \text{不检测的损失})$

成本1=min(检测成本1+次品损失1,不检测的损失)

2. 零配件2的检测：

- 如果检测零配件2，成本增加为3元。如果不检测，同样可能出现次品。

成本2=min(检测成本2+次品损失2,不检测的损失)

成本2=min(检测成本2+次品损失2,不检测的损失) $\text{成本2} = \min(\text{检测成本}2 + \text{次品损失}2, \text{不检测的损失})$

成本2=min(检测成本2+次品损失2,不检测的损失)

3. 成品检测：

- 对成品进行检测，成本增加为3元，可能避免高额的调换或拆解费用。若不检测，直接将成品推向市场，则可能面临调换或拆解的高额成本。

成品成本=min(检测成本成品+拆解或调换成本,不检测的调换损失)

成品成本=min(检测成本成品+拆解或调换成本,不检测的调换损失)

$$\text{成品成本} = \min(\text{检测成本成品} + \text{拆解或调换成本}, \text{不检测的调换损失})$$

4. 拆解决策：

- 如果不合格成品的拆解费用较低（如情境3，拆解成本为5元，调换损失为30元），则应选择拆解。

拆解成本=min(拆解成本,直接丢弃的损失)

拆解成本=min(拆解成本,直接丢弃的损失)

$$\text{拆解成本} = \min(\text{拆解成本}, \text{直接丢弃的损失})$$

伪代码：

```
# 动态规划表格，用于存储每个状态下的最优成本
dp = [[[-1 for _ in range(2)] for _ in range(2)] for _ in range(2)]

# 定义递归函数
def min_cost(i, j, k):
    # 如果这个状态已经计算过，直接返回
    if dp[i][j][k] != -1:
        return dp[i][j][k]

    # 如果检测零配件1
    if i == 1:
        cost_i = 2 + min_cost(0, j, k)
    else:
        cost_i = 次品损失_零配件1 + min_cost(0, j, k)

    # 如果检测零配件2
    if j == 1:
        cost_j = 3 + min_cost(i, 0, k)
    else:
        cost_j = 次品损失_零配件2 + min_cost(i, 0, k)

    # 如果检测成品
    if k == 1:
        cost_k = 3 + min(拆解成本, 调换损失)
    else:
        cost_k = 调换损失

    # 总成本是三者中的最小值
    dp[i][j][k] = min(cost_i, cost_j, cost_k)
    return dp[i][j][k]
```

python代码：

```
# 定义递归动态规划函数
def min_cost(situation, i, j, k):
    # 获取当前情境下的数据
    data = situations[situation]
```

```

# 如果检测零配件1，计算检测成本和可能的次品损失
if i == 1:
    cost_i = data["零配件1检测成本"]
else:
    cost_i = data["零配件1次品率"] * data["市场售价"]

# 如果检测零配件2，计算检测成本和可能的次品损失
if j == 1:
    cost_j = data["零配件2检测成本"]
else:
    cost_j = data["零配件2次品率"] * data["市场售价"]

# 如果检测成品，计算检测成本和可能的拆解或调换成本
if k == 1:
    cost_k = data["成品检测成本"] + min(data["拆解费用"], data["调换损失"])
else:
    cost_k = data["调换损失"]

# 总成本计算
total_cost = cost_i + cost_j + cost_k + data["装配成本"]
return total_cost

# 测试每个情境的最优决策组合
for situation in range(1, 7):
    # 遍历所有可能的检测和不检测组合
    result = min_cost(situation, 1, 1, 1) # 假设所有阶段都检测
    print(f"情境 {situation} 的最优总成本为: {result}")

```

结果：

```

情境 1 的最优总成本为：19
情境 2 的最优总成本为：19
情境 3 的最优总成本为：19
情境 4 的最优总成本为：15
情境 5 的最优总成本为：22
情境 6 的最优总成本为：24

```

蒙特卡洛模拟

```

# 蒙特卡洛模拟函数
def monte_carlo_simulation(situation, simulations=1000):
    # 获取当前情境下的数据
    data = situations[situation]

    # 存储每次模拟的总成本
    total_costs = []

    # 进行多次模拟
    for _ in range(simulations):
        # 随机生成次品数量，使用伯努利分布模拟次品率
        parts1_defective = np.random.binomial(1, data["零配件1次品率"]) # 零配件1是
        # 否次品（0或1）

```

```

    parts2_defective = np.random.binomial(1, data["零配件2次品率"]) # 零配件2是
否次品 (0或1)
    product_defective = np.random.binomial(1, data["成品次品率"]) # 成品是
否次品 (0或1)

    # 零配件1的成本
    if parts1_defective:
        cost_parts1 = data["零配件1次品率"] * data["市场售价"] # 不检测时的次品损
失
    else:
        cost_parts1 = data["零配件1检测成本"] # 检测成本

    # 零配件2的成本
    if parts2_defective:
        cost_parts2 = data["零配件2次品率"] * data["市场售价"]
    else:
        cost_parts2 = data["零配件2检测成本"]

    # 成品的成本
    if product_defective:
        cost_product = min(data["拆解费用"], data["调换损失"]) # 拆解或调换的最小
损失
    else:
        cost_product = data["成品检测成本"] # 成品检测成本

    # 总成本 = 零配件1成本 + 零配件2成本 + 成品成本 + 装配成本
    total_cost = cost_parts1 + cost_parts2 + cost_product + data["装配成本"]
    total_costs.append(total_cost)

    # 返回总成本的均值和标准差
    return np.mean(total_costs), np.std(total_costs)

# 运行蒙特卡洛模拟
for situation in range(1, 7):
    mean_cost, std_cost = monte_carlo_simulation(situation)
    print(f"情境 {situation} 的平均总成本为: {mean_cost:.2f}, 标准差为:
{std_cost:.2f}")

```

情境 1 的平均总成本为: 14.89, 标准差为: 1.49
 情境 2 的平均总成本为: 18.25, 标准差为: 5.09
 情境 3 的平均总成本为: 14.88, 标准差为: 1.50
 情境 4 的平均总成本为: 14.94, 标准差为: 6.13
 情境 5 的平均总成本为: 19.28, 标准差为: 4.43
 情境 6 的平均总成本为: 14.35, 标准差为: 1.47

2.3 算法：蒙特卡洛模拟

描述：次品率和生产过程中的不确定性较大，可以使用蒙特卡洛模拟，通过多次随机模拟不同次品率和生产过程，估计不同策略下的期望成本。

步骤：

1. 随机生成次品率和检测结果。
2. 对每个策略进行多次模拟，统计平均成本。
3. 比较不同策略的期望值，选择最优策略。

问题3：多工序、多零配件的生产决策

本问题在问题2的基础上增加了更多复杂度，要求考虑多道工序及更多零配件的生产决策。

1. 多工序影响：

各道工序会影响零配件和半成品的次品率，需要建立更为复杂的模型。通过马尔科夫链或蒙特卡洛模拟等方法预测不同工序下的质量演变，计算各工序的最优策略。

2. 多零配件决策：

不同零配件的次品率和成本不一，对每个零配件进行单独决策。根据总装配成本、检测成本和市场售价，优化整体生产链的成本结构。

问题4：重新分析次品率

在问题4中，假设问题2和问题3中的次品率是通过抽样检测方法得到的。这意味着需要重新结合问题1中设计的抽样检测方案，重新计算次品率并基于此调整各个决策。

1. 结合抽样检测数据：

- 使用问题1中得出的检测方案，得到更为精确的次品率估计值。
- 将这些估计值替换到问题2和问题3中的模型，重新做出决策。

2. 决策调整：

- 可能由于次品率估计值的变化，导致某些阶段的检测方案或处理策略需调整。需再次优化总体模型。

总结

该题的解题思路可以通过建立统计模型与成本分析模型来解决。关键在于：

- 问题1采用假设检验与抽样检测设计。
- 问题2和问题3结合成本效益分析，优化生产各环节的检测与处理方案。
- 问题4结合前面问题中的抽样检测结果，重新校正生产决策。

参考模型

本题的难度主要在于综合应用统计分析、生产决策优化、动态规划等多种模型和算法，处理多阶段、多零配件生产中的决策复杂性和不确定性。在解题过程中，合理选择并结合这些经典的参考模型，能够有效解决该问题中的多个挑战。

1. 假设检验模型

描述：用于问题1中抽样检测的设计。

参考模型：二项分布假设检验、正态分布近似、置信区间计算。

2. 成本效益分析模型

描述：用于问题2和问题3中的生产过程决策。

参考模型：在经济学和管理科学中的成本效益分析模型。

3. 动态规划模型

描述：用于分阶段的决策优化问题（如问题2和问题3），特别适合多阶段递归性问题。

参考模型：Bellman提出的动态规划框架，常用于求解多阶段决策问题。

4. 马尔科夫决策过程（MDP）

描述：用于多工序、多阶段、多零配件的生产过程决策。

参考模型：马尔科夫决策过程（MDP）是处理多阶段决策的标准工具之一。

5. 蒙特卡洛模拟

描述：用于应对生产过程中的不确定性，可以通过模拟大量次品率和生产过程，评估不同决策下的平均成本。

6. 启发式算法

描述：适用于多工序和多零配件的复杂决策问题。

参考模型：模拟退火算法、遗传算法等启发式搜索方法。

7. 贝叶斯更新模型

描述：用于问题4中基于新的检测数据更新次品率的估计。

参考模型：贝叶斯统计推断模型，通过新的数据不断调整先验概率。

动态规划、决策树

马尔科夫决策

模型改进部分：

第一问假设检验不一定是最好的，参考：

以假设检验为主体的方案已经是一个非常有效的统计工具，特别是在第一问中帮助企业判断零配件次品率是否超过标称值的场景中。为了使方案更加完善，可以结合其他方法进行补充，提升检测方案的准确性、灵活性和适用性。以下是几种可以与假设检验结合使用的补充方法：

1. 贝叶斯推断（Bayesian Inference）

贝叶斯推断可以作为假设检验的补充，特别是在数据逐步收集时。贝叶斯方法通过先验分布和新的检测数据，逐步更新对次品率的估计，适用于动态决策环境。

- 结合点：**在开始时使用假设检验设计初步抽样方案，随着检测数据的增加，可以使用贝叶斯推断不断更新次品率估计，从而调整决策。

- **优点：**能够随着检测数据逐步收集，动态更新决策，避免单次抽样的局限性。
- **应用场景：**如果企业进行多批次检测，可以通过贝叶斯方法结合每批次的检测结果来调整策略。

2. 控制图 (Control Charts)

控制图是一种质量管理工具，适合持续监控生产过程中的次品率波动。假设检验通常用于单次抽样决策，而控制图可以用于长期的质量监控。

- **结合点：**在初步检测中使用假设检验进行批次接收决策，但在长期生产过程中可以使用控制图进行实时监控，确保生产过程的稳定性。
- **优点：**适合持续监控次品率，能够快速发现生产中的质量问题。
- **应用场景：**在抽样检测完成后，可以通过控制图监控后续生产，确保次品率维持在控制限内。

3. 蒙特卡洛模拟 (Monte Carlo Simulation)

蒙特卡洛模拟可以用于评估不同抽样策略在各种次品率情境下的效果。假设检验提供了一个判断标准，但通过模拟可以评估不同抽样方案在多种可能的现实情况下的表现。

- **结合点：**先用假设检验设计一个初步的检测方案，再通过蒙特卡洛模拟评估该方案的有效性。模拟不同次品率下的检测效果，找到可能的最优方案。
- **优点：**可以测试不同次品率下的检测效果，并通过大量随机模拟提高决策的稳健性。
- **应用场景：**适用于不确定性较大的情况下，例如供应商的次品率变化较大时，通过模拟找到最佳抽样检测方案。

4. 成本效益分析 (Cost-Benefit Analysis)

假设检验通常关注统计上的显著性，而成本效益分析则关注经济上的最优性。通过将假设检验和成本效益分析结合，企业可以在保证决策准确性的同时，最小化检测和处理成本。

- **结合点：**在假设检验设计抽样方案的基础上，增加一个成本分析模块，评估不同检测方案的经济性。企业不仅考虑次品率的判断，还需要考虑检测成本、不合格产品的处理成本以及潜在的市场损失。
- **优点：**综合考虑检测的经济性，帮助企业在决策时平衡成本和质量要求。
- **应用场景：**适用于对成本敏感的企业，可以通过对检测方案的经济分析找到最优的检测次数和检测策略。

5. 逐步检验 (Sequential Testing)

逐步检验是一种根据数据实时更新的检测方法，在检测到足够的证据时可以提前终止检测，减少不必要的检测次数。

- **结合点：**初期使用假设检验设计样本量，但在实际检测过程中，结合逐步检验方法，在检测过程中逐步判断是否可以提前结束抽样检测。如果证据足够支持接受或拒收零配件，可以提前做出决策。
- **优点：**可以减少不必要的检测次数，提高检测效率。
- **应用场景：**适用于次品率明显偏离标称值的情况，能够快速得出检测结果，避免过多检测开销。

6. 经验累积法 (Cumulative Sum Control Chart, CUSUM)

CUSUM控制图是一种监测过程偏差的工具，它结合历史数据和当前抽样检测数据，帮助企业监控次品率的变化趋势，适合长期监控质量问题。

- **结合点：**在假设检验的基础上，可以使用CUSUM方法积累不同批次的检测数据，并通过控制图监测次品率的变化趋势，及时调整生产策略。
- **优点：**通过积累数据，能够快速识别出过程中的质量变化趋势，适合长时间的生产监控。

- **应用场景：**适用于长期批次生产，帮助企业在持续监控中保持生产质量的稳定性。

7. 敏感性分析 (Sensitivity Analysis)

敏感性分析可以评估抽样检测方案在不同参数（如标称次品率、误差范围等）下的表现，以确保方案的鲁棒性。

- **结合点：**先用假设检验设计基本方案，再通过敏感性分析评估不同参数对方案结果的影响，确保方案在次品率波动情况下依然有效。
- **优点：**能够评估方案的鲁棒性，避免方案对某些参数过于敏感。
- **应用场景：**适用于不确定性较大的生产环境，尤其是供应商提供的次品率波动较大时。

总结：

在假设检验的基础上，可以结合贝叶斯推断、控制图、蒙特卡洛模拟、成本效益分析、逐步检验、CUSUM控制图和敏感性分析等方法，形成一个更全面的检测和决策方案。这些方法能够提高检测的灵活性、经济性和实时性，确保企业在不同情况下都能做出合理的决策。