

基于统计分析的电子产品生产多决策优化模型

摘要

本文研究企业生产过程中的决策问题。我们在分析成品生产工序的基础上编程实现了生产过程的模拟，并且通过遍历得到不同的决策组合带来的收益。基于不同决策组合的数据，通过数理分析得出较优的决策选取策略。在运用所得策略的基础上对多道工序的生产进行分解，从而得到更普适的多工序生产流程模拟模型。

对于问题一，我们首先根据零部件抽样符合**二项分布模型**，估计其样本与方差。同时通过标准化处理次品率，对其进行正态总体均值的假设检验，确定其为**单边检查**，采用**临界值判别法**求解出满足题意中显著性水平 α 的最小的 n 值。对于情形一， $n=98$ ；对于情形二， $n=60$ 。通过最小 n 值进行**随机抽样**，若抽样结果次品率依然无法满足置信度，则视为不合格，反之则接受这批零部件。

对于问题二，为了解决企业在生产中遇到的情况，我们将零配件是否检测，成品是否检测，不合格产品是否拆解作为决策变量，以获得最大利润为目标函数，建立**多阶段决策模型**。通过穷举**16种决策变量**的组合情况，为每种情况都做出了最优化的决策方案。最后我们模拟了工厂生产中的**资金流动过程**，对上述各个决策组合进行**结果验证**，所得决策方案保持一致。随后，我们计算出部分策略随着时间存在稳定亏损，部分检测策略之间存在优劣。为了得到决策方案与次品率之间的关系，我们进行了决策方案对于次品率的**灵敏度分析**。

对于问题三，我们先分析了多工序、多零配件的生产过程中，决策组合随着工序和零配件数量改变的变化趋势。基于问题二得出的推论，对第三问中明显不能提高利润和重复的决策组合进行排除。随后我们对零配件和中间产物流回上一工序的收益、损失进行计算，做出等效的只有**前向通道的加工流程图**。基于**等效加工流程图**，对所给生产流程进行编程模拟，遍历出最优决策方案，**收益为56.4元/件**。随后我们对多工序、多零配件的生产过程进行分解，通过不断迭代组件的总量和次品率，我们可以得到最终的收益。

对于问题四，根据问题二，三中对于次品率对决策方案产生影响的**灵敏度分析**，我们得知在一定区间内，次品率改变，决策方案也随之改变。因此，我们利用问题一的结论求出满足合理**置信区间**下的次品率，套用问题二，三中建立的模型，即可求出相应的决策方案。

关键字： 多阶段决策模型 统计分析 遍历穷举 假设检验 数理分析

一、问题重述

1.1 问题背景

电子产品是当前信息化时代重要的生产产品。在实际产品生产过程中，零配件的质量及其组装过程是影响产品质量的重要因素。企业需要指定合适的抽样检测方案和生产决策方案来提高收益、减少企业信誉折损。面对实际生产的复杂工序，我们需要建立适当的数学模型，来提高生产效益。

本文针对电子产品的生产流程中各阶段决策进行了讨论与研究，旨在协助实际生产，提高企业收益，以科学角度剖析产品生产问题。

1.2 问题要求

问题 1 从零配件的次品率监控出发，为两种零件设计抽样检测方法，使得在次品率小于 10% 的情况且满足以下两种情况：(1) 在不满足 95% 的置信度的情况下，拒收零件；(2) 满足 90% 的置信度的情况下，接收零件。并且设计出检验次数足够少的抽样检测方案。

问题 2 针对零配件测验与否、成品测验与否、不合格产品是否拆解的不同组合，通过计算在题目所给六种具体情形的盈利情况，确定决策方案并且说明决策的依据。

问题 3 根据图片给出的两道工序，八种零配件，已知零配件、半成品和成品的次品率，设计同问题二相同的决策，同时加上对于半成品拆解的判断，得到最终的盈利策略。最后推广到普适情况，即对 m 道工序， n 个零配件的成品组装策略。

问题 4 假设上述条件中的零配件、半成品和成品的次品率都是通过题一中的抽样方法得到的，重新设计产品工序决策，达到效益最大化。

二、问题分析

2.1 问题一分析

对于问题一，我们通过查找文献资料，明确该类零配件的次品出现概率往往符合正态分布 [1]，分析出在不同置信度下能够满足统计学规律的最小 n 值，进而通过最小化检验的次数，得到使得利益最大化的方法。抽样检查的结果符合**二项分布模型**，可以估计出标称值的方差与标准差。为了更好利用标准正态分布模型，我们将次品出现的概率进行了**标准化处理**，得到符合模型的不等式。根据题目中的置信度约束，我们对其进行了**方差已知时正态总体均值的假设检验**，确定其为**单边检查**，采用**临界值判别法**求解出满足题意中显著性水平 α 的最小的 n 值。

2.2 问题二分析

对于问题二，我们已知零配件一、二以及二者组装后成品的次品率、单价、检测成本，与不合格产品的调换损失和拆解费用等信息。要求我们对两种不同零配件是否进行检测、装配后的成品是否进行检测、不合格的成品是否进行拆解或回收等问题进行决策，得到最优的决策方案，以及最大的利润收入。因此，我们选择**多阶段决策模型**，求解出不同策略下不同的状态信息，通过对**状态信息的叠加**，进而对决策变量组合采取**穷举法**，代入目标方程，求出最优的决策方案及其利润。

2.3 问题三分析

对于问题三，我们已知零配件、半成品、成品的次品率以及对应的购买单价、检测成本等相关数据，并且决策的种类及后果和第二问保持一致。题目拓展了工序和零配件的数量，使决策方案的种类成指数型增长，增加模拟流程程序编写难度，无法通过程序遍历寻求最优方案。因此，我们结合了第二问所得推论，建立决策指定策略，简化求解流程。经过优化，我们设计了多层次生产流程模拟模型，找到应对此类多阶段决策问题的普适方法。

2.4 问题四分析

对于问题四，我们已知得到的次品率为第一问抽样所得，要求我们重新对第二第三问的问题进行分析。因此我们原先所认知的次品率和实际产品次品率应有一定的偏差，我们需要通过重新确定置信区间，重新确定次品率，进而指定相应的决策。

三、 模型假设

为简化问题，本文做出以下假设：

1. 独立性假设：假设零配件 1 和零配件 2 的合格情况相互独立
2. 检测准确性假设：假设检测过程是完全准确的，即检测结果完全反映了零配件的真实质量。
3. 成本稳定性假设：假设所有成本（购买成本、检测成本、装配成本、拆解费用、调换损失）在考虑的时间段内保持不变。
4. 市场需求假设：假设市场需求足够大，可以吸收所有合格的成品。
5. 生产能力假设：假设企业的生产能力足够，可以处理所有检测后的合格零配件。
6. 风险中性假设：假设企业在决策时不考虑风险，只基于期望利润最大化。
7. 本模型假设企业在购买零配件时，每种零配件购买总量一致。
8. 企业不考虑不合格半成品被丢弃后，部分合格半成品可能无法配对产生的闲置问题。

四、符号说明

符号	说明	单位
n	检验次数	次
$profit_{last}$	现阶段利润	元/件
x	次品数量	个
μ_0	假设检验均值	/
$TEST_{11}$	零件 1 决策变量	/
$TEST_{12}$	零件 1 决策变量	/
$TEST_2$	成品检测决策变量	/
$TEST_3$	次品拆解决策变量	/
c_1	第一阶段成本	元/件
c_2	第二阶段成本	元/件
c_3	第三阶段成本	元/件
s_1	第一阶段盈利	元/件
s_2	第二阶段盈利	元/件
s_3	第三阶段盈利	元/件

五、模型的建立和求解

5.1 问题一：基于总体均值假设检验的零配件检测方案设计

在问题一中，在题目要求的两种置信度下，采用抽样检测的方法，确定供应商提供的零配件的次品率不会超过某个标称值的最小抽样检验次数 n ，因此我们建立了**方差已知时正态总体均值的假设检验**，确定其为**单边检查**，并利用二项分布确定抽样次品率的均值和方差，采用**临界值判别法**求解出满足题意中显著性水平 α 的最小的 n 值。

5.1.1 样本量的假设

利用二项分布进行抽样检测设计。设置变量 n 为抽样检查的次数， x 为次品数量，故样本次品率为 x/n 。令 p_0 为标称值即 0.1， σ 为整体标准差。

5.1.2 二项分布模型与正态分布模型

如果在随机试验中，我们只对某随机事件是否发生感兴趣，其结果可以归类为发生与不发生两种情况，我们则可以称这类随机试验为伯努利试验，而考虑多次的伯努利试验，则我们可以称之为带有参数 n, p 的二项分布，记为 $X \sim B(N, P)$ ，其分布律为：

$$P\{X = k\} = C_n^k P^k (1-p)^{n-k}, k = 0, 1, \dots, n \quad (1)$$

由于零件的合格与否满足 (0-1) 分布, 其分布为:

$$P\{X = 0\} = 1 - p, \quad P\{X = 1\} = P \quad (2)$$

其方差为:

$$E(X) = 0 \cdot (1-p) + 1 \cdot p = p \quad (3)$$

$$E(X^2) = 0^2 \cdot (1-p) + 1^2 \cdot p = p \quad (4)$$

$$D(X) = E(X^2) - [E(X)]^2 = p - p^2 = p(1-p) \quad (5)$$

由 n, p 为参数的二项分布变量, 可分解成为 n 个相互独立且都服从以 p 为参数的 $(0-1)$ 分布的随机变量之和。由于 $E(X_k) = p, D(X_k) = p(1-p), k = 1, 2, \dots, n$, 故知

$$E(X) = E\left(\sum_{k=1}^n X_k\right) = \sum_{k=1}^n E(X_k) = np \quad (6)$$

又由于 X_1, X_2, \dots, X_n 相互独立, 得

$$D(X) = D\left(\sum_{k=1}^n X_k\right) = \sum_{k=1}^n D(X_k) = np(1-p) \quad (7)$$

即:

$$E(X) = np, \quad D(X) = np(1-p) \quad (8)$$

根据文献提供资料 [1], 电子配件的不合格概率应符合正态分布:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty \quad (9)$$

同时中心极限定理指出: 当样本量足够大时, 无论原始数据分布如何, 样本均值的分布都将接近正态分布。这是使用正态分布来近似二项分布的基础。

5.1.3 假设检验与数值求解

对于情况一: 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。故:

$$H_0 = \mu_0 \leq 0.1; H_1 = \mu_0 > 0.1$$

通过以大数定律为基础的临界值判别法得到下面等式:

$$P\left(z = \frac{x - np_0}{\sqrt{np_0(1-p_0)}}\right) \leq z_{0.05}$$

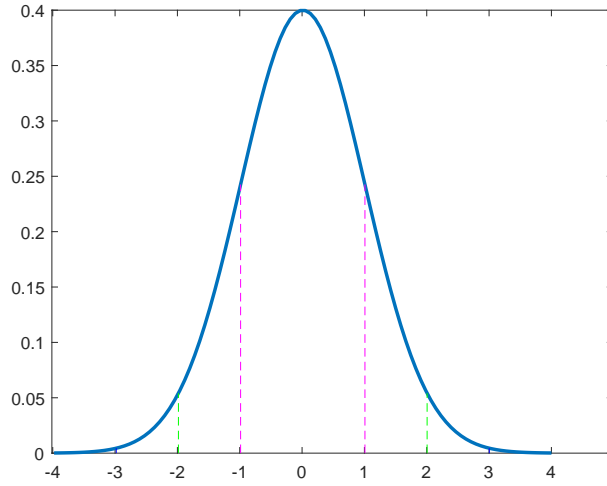


图 1 正态分布图像

经过推导可以得到：

$$n = \left(\frac{z \sqrt{p_0(1-p_0)}}{\frac{x}{n} - p_0} \right)^2$$

令 $x/n - p_0$ 为误差量，则可接受的误差 e 为 5%，代入上式得：

$$n = \left(\frac{z \sqrt{0.01(1-0.01)}}{e} \right)^2 \quad (10)$$

故当满足以下条件时：

$$z_\alpha = z_{0.05} = 1.645$$

通过代入上式方程计算，我们可以得到最小的检验次数 n ：

$$n = 98$$

对于情况二：在 90% 的信度下认定零配件次品率不超过标称值，则接受这批零配件。故：

$$H_0 = \mu_0 \leq 0.1 ; H_1 = \mu_0 > 0.1$$

同理，令 $x/n - p_0$ 为误差量，则可接受的误差 e 为 5%，代入上式得：

$$n = \left(\frac{z \sqrt{0.01(1-0.01)}}{e} \right)^2 \quad (11)$$

故当满足以下条件时：

$$z_\alpha = z_{0.1} = 1.29$$

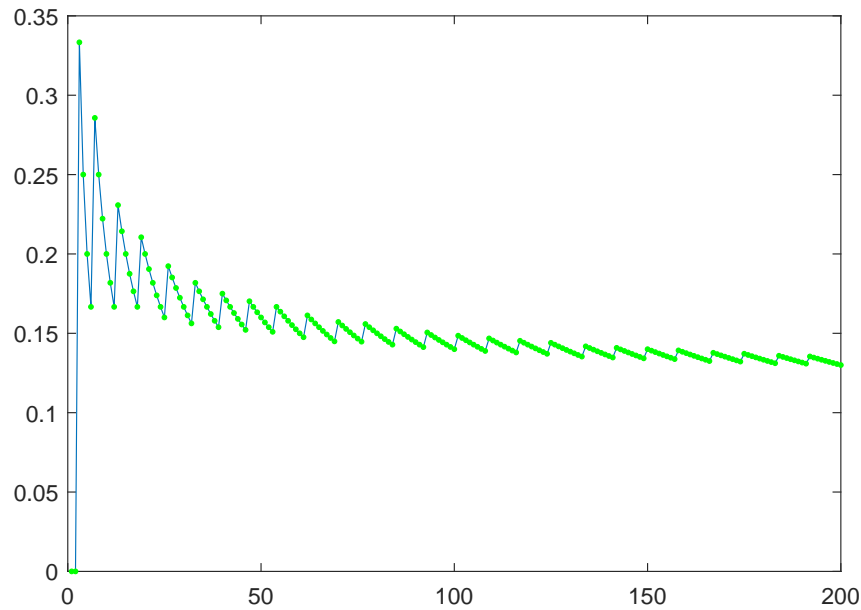


图2 抽样次数与得到不合格比例关系

通过代入上式方程计算，我们可以得到最小的检验次数 n ：

$$n = 60$$

5.1.4 基于随机抽样的抽样检测方案设计

通过以上数据，我们可以明确满足相应置信区间的最小抽样次数，如果抽样次数小于最小抽样次数，则其置信度不够，该批零部件不一定满足质量要求；而过多抽样检测则会造成产品合格率检验成本的增加。

在此基础上，我们可以采用**随机抽样**，即根据规定的置信度，从每批零部件中随机抽取最小样本量 n ，最小样本量根据不同置信度确定。如果抽样样本超过了 n ，却无法满足置信度条件，则我们可以认为这批零部件不满足合格资质。如问题一中的情形一，应抽取 98 组样品，进行分别检验，如若次品率超过 15%，则可以选择拒绝该批次的零部件。

5.2 问题二：产品生产线决策方案

5.2.1 多阶段决策模型

在第二问中，决策方案涉及各个工序的检测与成本控制利益最大化问题，是一种典型的多阶段决策问题。多阶段决策是指决策者在整个决策过程中，需要做出时间上先后有别的多项决策。[2] 这类决策通常比单阶段决策更为复杂，因为它要求决策者考虑多个阶段之间的相互影响和依赖关系。[3]

决策过程被明确地划分为若干个阶段，每个阶段都有相应的决策点。各阶段的决策是相互依赖的，后一阶段的决策往往受到前一阶段决策结果的影响。由于多阶段决策过程中存在不确定性，因此还需要制定适合不同状态的灵活策略，以应对可能出现的各种情况。

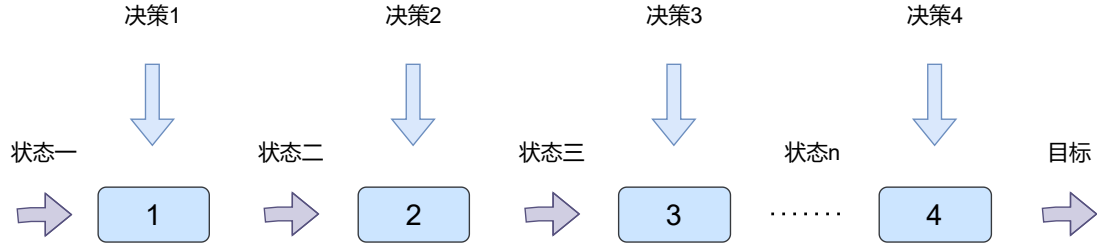


图3 决策与状态转换

本题可以通过多阶段决策问题进行求解，对于产品最大的收益的决策过程中，我们不只做了一次决策，对应的决策变量有：零件1是否需要检测、零件2是否需要检测、组装后的成品是否需要检测、不合格的成品是否需要拆解回收。

我们需要在每个阶段中做出决策，每一次决策会影响到当前阶段的状态，最终使产品收益达到最大。

在本题中，我们分析了四个阶段的状态变量，状态变量在多阶段决策过程中起到桥梁作用，一方面链接不同的决策变量，另一方面影响后续阶段的状态。下面我们详细介绍本题的决策变量及每个阶段的状态变量。

(1) 零件的检验

零件1的单位检测成本 t_{c1} ，零件2的单位检测成本 t_{c2} ，零件1的次品率 d_{r1} ，零件2的次品率 d_{r2} ，成品的次品率 d_{r3} ，零件1起始数量 p_{01} ，零件2起始数量 p_{02} ，零件1的单价 $value1$ ，零件2的单价 $value2$ 。经过决策零件1、2后是否进行检测得到的成本和盈利分别为 c_1 与 s_1 。

	待组装的成品数量 p_{n1}	合格的成品数量 p_{n2}	零件1检测成本 tt_{c1}	零件2检测成本 tt_{c2}
零件1&2都检测	$\min\{(1 - d_{r1}) \cdot p_{01}, (1 - d_{r2}) \cdot p_{02}\}$	$(1 - d_{r3}) \cdot p_{n1}$	$t_{c1} \cdot p_{01}$	$t_{c2} \cdot p_{02}$
零件1&2都不检测	$\min\{p_{01}, p_{02}\}$	$(1 - d_{r1}) \cdot (1 - d_{r2}) \cdot (1 - d_{r3}) \cdot p_{n1}$	0	0
零件1检测零件2不检测	$\min\{p_{02}, (1 - d_{r1}) \cdot p_{01}\}$	$(1 - d_{r2}) \cdot (1 - d_{r3}) \cdot p_{n1}$	$t_{c1} \cdot p_{01}$	0
零件1不检测零件2检测	$\min\{p_{01}, (1 - d_{r2}) \cdot p_{02}\}$	$(1 - d_{r1}) \cdot (1 - d_{r3}) \cdot p_{n1}$	0	$t_{c2} \cdot p_{02}$

表1 零部件状态变量表

通过对表格中相关规律的总结，零件 1 的检测成本：

$$tt_{c1} = t_{c1} \cdot p_{01} \cdot TEST_{11}$$

零件 2 的检测成本：

$$tt_{c2} = t_{c2} \cdot p_{02} \cdot TEST_{12}$$

经过上述分析我们可以得到

$$c_1 = tt_{c1} + tt_{c2} + value1 + value2$$

$$s_1 = 0$$

(2) 组装后成品的检验

对于成品是否检测的问题，我们同样设立了如下变量，单位成品检测成本 t_{c4} ，单位调度损失成本 t_{c5} ，单位市场价格 t_{c6} ，单位拆解费用 t_{c7} 。

	成品检测成本 tt_{c4}	成品装配成本 tt_{c3}	调换损失费用 tt_{c5}
成品检测	$tt_{c4} \cdot p_{n1}$	$p_{n1} \cdot t_{c3}$	0
成品不检测	0	$p_{n1} \cdot t_{c3}$	$p_{n3} \cdot t_{c5}$

表 2 成品检测决策变量表

由表格我们可清楚知悉需要拆解的数量：

$$p_{n3} = p_{n1} - p_{n2}$$

进而得到成品的检测成本：

$$tt_{c4} = t_{c4} \cdot p_{n1} \cdot TEST_2$$

经过上述分析我们可以得到第二轮盈利：

$$s_2 = p_{n2} \cdot t_{c6}$$

及第二轮所需成本：

$$c_2 = tt_{c5} + tt_{c3} + tt_{c4}$$

(3) 回收拆解问题

在回收拆解问题上，我们定义第三轮成本为 c_3 ，盈利为 s_3 ，拆解所需费用为 tt_{c6} ，现阶段利润为 $profit_{last}$ 。


```

1  计算成品构成中分别含有一至三个次品零配件的总量
2  p_zero=part1_good*part2_good/(max{(part1_good+part1_bad),(
    part2_good+part2_bad)});
3  p_once, p_two...
4
5  outcome= 各个零配件购买单价
6  根据 零配件检测策略 分别计算合格成品和不合格成品的量。
7  if 不检测任何零配件
8      good_product= (1-成品次品率) *p_zero
9      bad_product  =sum(p_zero,p_once,p_two)-p_zero*成品次品率
10 else if 检测零配件1
11     good_product=...
12     bad_product  =....
13     outcome+=零配件1检测费用
14 ...
15
16 根据 成品检测策略 计算费用支出
17 outcome+= (good_product+bad_product) *装配成本
18 if 检测成品
19     outcome+= (good_product+bad_product) *检测成本
20 else
21     outcome+=bad_product*调换损失
22
23 income+=good_product*市场售价
24
25 根据 不合格成品拆解策略 决定是否计算拆解的配件的含量
26 if 拆解不合格成品
27     outcome+=bad_product*拆解费用
28     part1_good=p_zero*成品次品率+部分p_once
29     part1_bad=p_two*成品次品率+部分p_once
30     part2_good=...
31     part2_bad=...
32 回到开始进行循环
33 ....

```

```

34
35 profit=income-outcome
36 if flag_assemble_test==1:
37     income -= p_out * assemble_cost;
38     if flag_assemble_test == 1:
39         income -= p_out *assemble_test_cost
40     else:
41         income -= p_bad_product * Replace_cost;
42
43 income += p_good_product * Price
44
45
46 if flag_dissemble==1:
47     进行多轮模拟:
48     for 每轮:
49         计算出拆解零件的总量及次品率, 重复上述步骤计算出新
           一轮的收益情况
50
51
52 输出 income
53 如果 if_dissemble_cost:
54     输出每轮模拟的 income

```

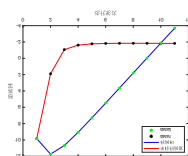
5.2.3 生产流程模型推论

推论一: 如果中间产物的不合格部分需要被拆解, 那么在前一阶段中, 该中间产物的组成部分需要经过全部检验。

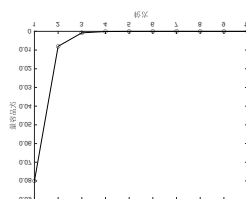
不合格的成品在多次拆解之后会导致一部分次品零配件不断被装配并拆解, 造成趋近于定值的稳定损失, 并且我们可以求出这一部分的固定损失。设初始可配对总量为 d , 未检测配件次品率为 k , 拆解费用为 $apart$, 部分检测费用为 $test$, 组装费用为 $assem$, 配件总量为 n_i , 第 i 件未被检测的配件次品率为 $rate_i$ 。我们可得固定损失 $lost$:

$$lost = d \times \max\{rate_i\} \times \{apart + test + assem\}$$

反之, 当我们拆解的同时, 对上一层级进行全部检验, 那么我们可以将拆解的零配件所损失的钱进行求和:



(a) 每轮利润与循环次数关系图



(b) 次品含量随着循环次数变化

图 5

$$cost_{loop} = \sum_{n=1}^{\infty} rate_{final}^{n+1} \cdot \min\{n_i\} \cdot \{apart + test + assem\}$$

同时这一部分次品在循环次数接近无限时，次品量趋近于零。

推论二：如果不考虑匹配失败的零配件损失，则选择检测费用少的组合收益最大化。如果不考虑匹配失败的零配件损失，并且两个零配件的检测费用一致，选择检测成本低的零配件利润更高。

5.2.4 决策结果分析

我们规定 $xxxx$ 为选择决策的简单表达，其中的含义为：每一位可以由 0 和 1 来表达，0 代表放弃检查或放弃拆解，反之 1 代表选择检查与拆解；第一位代表着零部件 1 选择检查与否，第二位代表零部件 2 选择检查与否，第三位代表是否对成品进行检查，而第四位则代表是否对不合格产品进行回收。不同的决策会导致收益的不同，表 4 罗列了各情况下的收益最高值，如情况一当决策选择检查零部件 1 和 2，不检查成品，并且对不合格成品进行回收时可以达到收益最大化，最高收益可以达到 15.8 元每件商品。其他情况同理。

每一种情况的具体演算数据将在附件中体现。

	情况一	情况二	情况三	情况四	情况五	情况六
最高收益 (元/件)	15.8	8.599	13.4	11.4	6.968	18.5
选择决策	1101	1101	1101	1111	0100	0000

表 4 不同情况决策方案与对应收益

对于情况 5 的决策方案，若按照我们模型建立的假设进行求解，得出 0100 决策方

案为最佳决策方案；但是基于零件 1 和 2 次品率的差异，对零件 1 购买数量进行下调，使其等于 $\frac{8}{9}$ 的零件 2 的数量再次进行分析，得出 1101 为最佳决策方案。

5.2.5 灵敏度分析

控制模型中另外两个次品率不变，只改变其中一个次品率，使其从零开始以 0.005 的步长遍历至 0.9。利用构建的以利益最大为目标函数的多目标检测模型中得到对应的 191 种决策方案储存在 Excel 表格中。

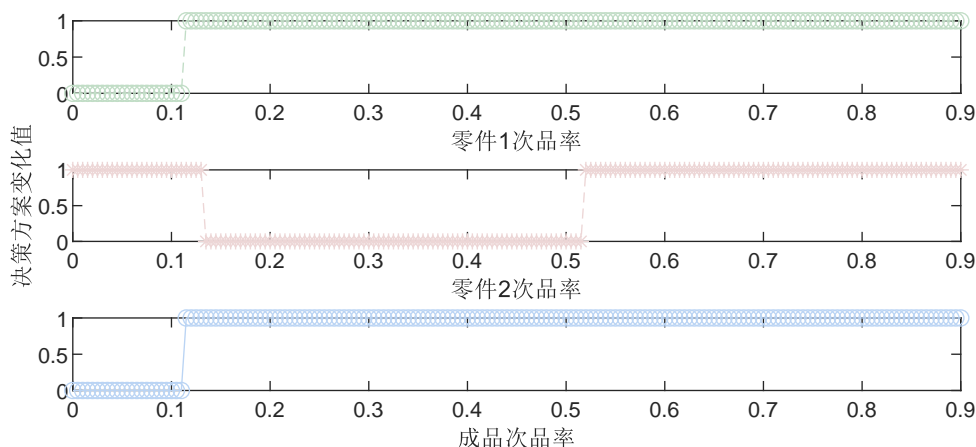


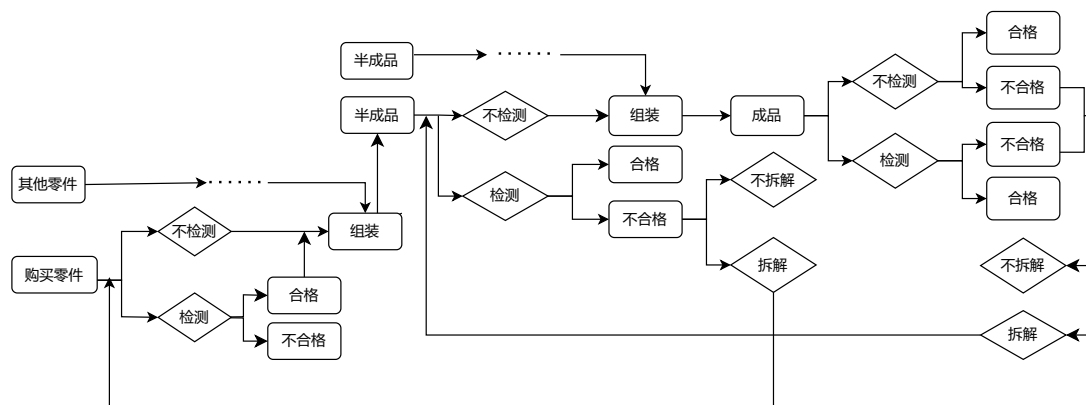
图 6 灵敏度分析

经过分析可得，原有的方案决策在原有次品率附近的区间仍基本为最优决策方案，但是对于偏离较大的区间部分决策会发生翻转，产生新的决策方案。同时，本次灵敏度分析具有较大的现实意义，对零配件次品率和成品次品率的灵敏度进行分析可知，当次品率存在较大偏移时，除了更换零件供应商和更换加工工厂之外也可以通过调整决策方案来减少损失。

5.3 问题三：多层次的生产流程模拟

5.3.1 多层次的生产流程的建立

针对题意，我们对生产流程进行抽象得到了以下流程图



观察流程图可知，组装流程的复杂度较高。每个零件都有独立的检测决策，如果该零件选择检测，则只有合格的零件流向半成品组装，反之则全部进入半成品组装。每个组装完成的半成品都有独立的检测和拆解决策，存在不检测、检测不拆解和检测拆解三种决策可能，每个决策都会影响进入后一阶段的半成品总量和次品率。最终的成品也面临检测和拆解的决策，直接影响取得收益的成品总量和相关金额支出。

通过对各阶段概率的计算，结合排列组合，我们可以得到初始抉择样本复杂度高，不易求解，完全排列组合存在 27648 种可能性。当成品的组装变得复杂，各阶段决策的排列组合可能性太多，无法用穷举法进行解决。

5.3.2 多层次的生产流程的优化

根据问题二得到的结论，我们可以假设，当我们在某一步采取了拆解的决策的时候，这个物件的组成部分必然全部都经过了检验。

那么在中间产物进行检测并且拆解的决策的时候，我们可以进行如下推导：

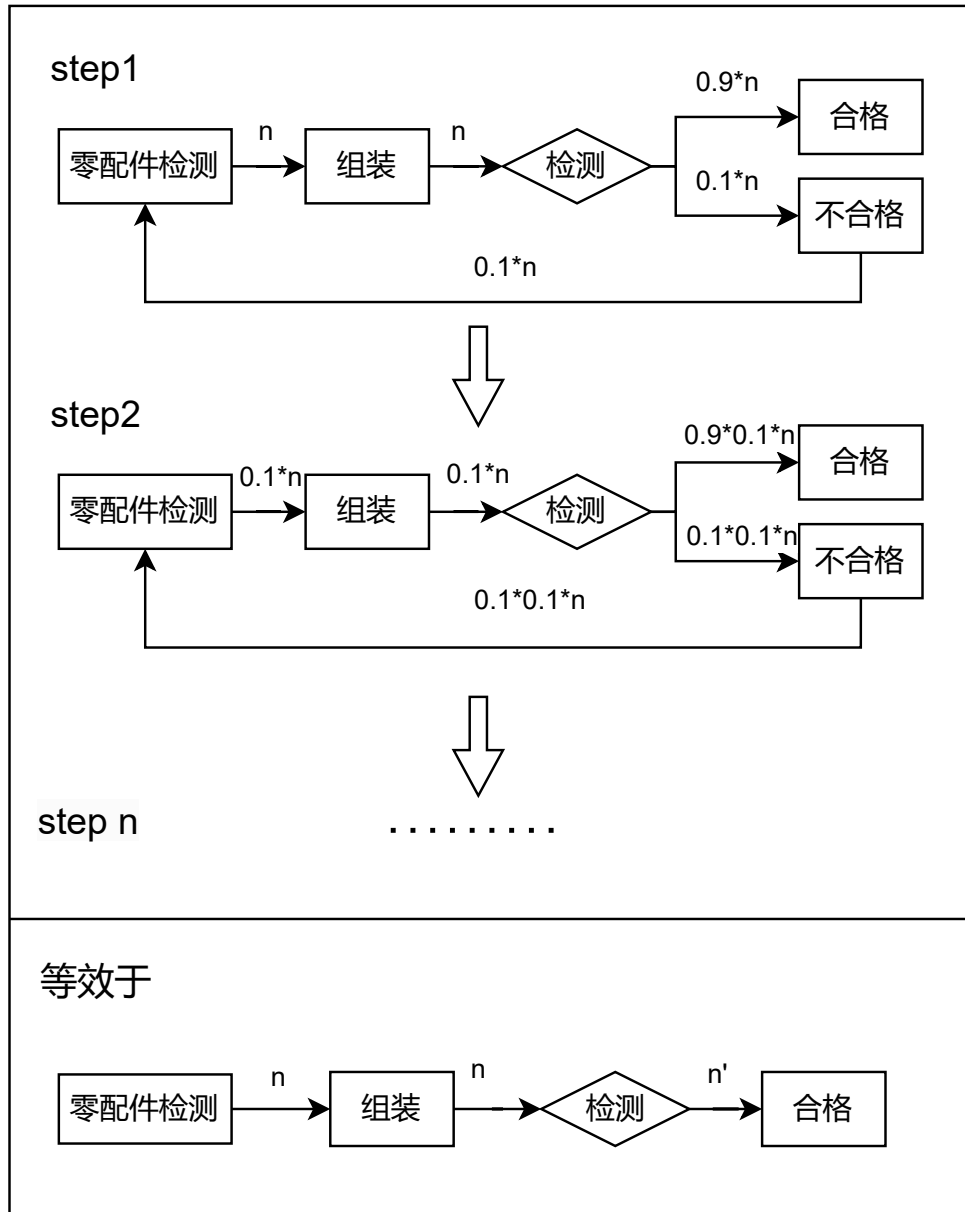


图 8 计算流程步骤图

$$\begin{aligned}
 n' &= \frac{9}{10}n + \frac{9}{100}n + \frac{9}{1000}n + \dots \\
 &= \sum_{k=0}^{\infty} \left(\frac{1}{10}\right)^k \frac{9}{10}n = n
 \end{aligned}$$

值得注意的是，在这种决策组合情况下，我们在利润计算的时候，需要减去问题二中所求的损失 $cost_{loop}$ 。

通过上述推论，我们可以将部分决策中的循环部分进行等效计算，可以得到一个只存在前向通道的计算流程，极好的降低了决策复杂性。

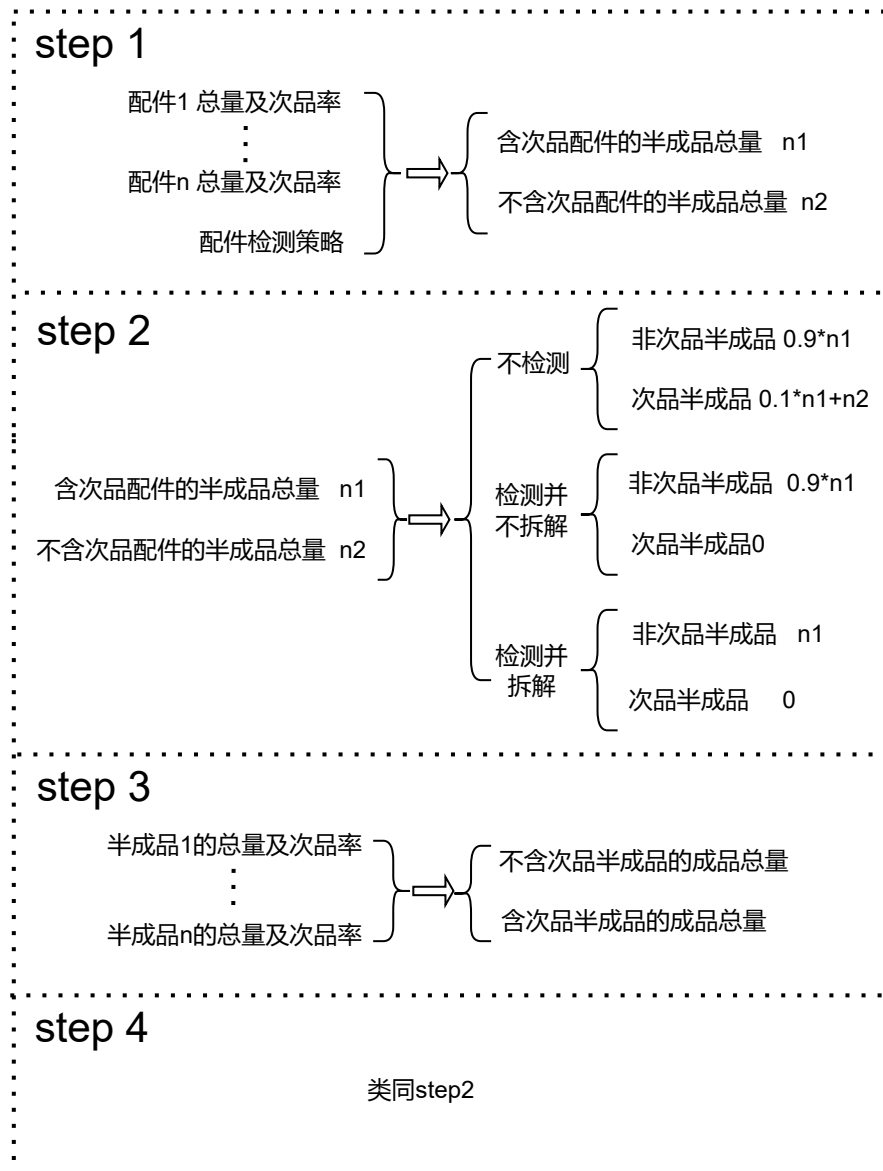


图 9 计算流程图

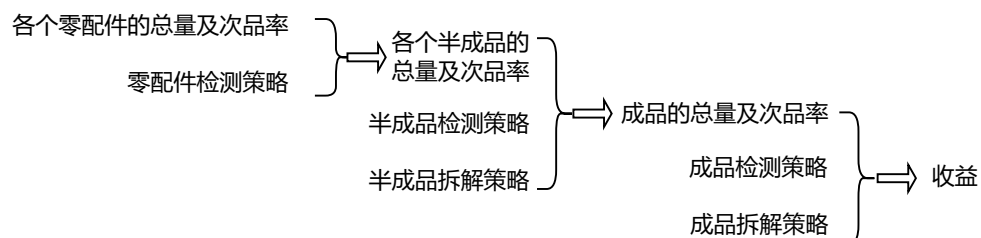


图 10 计算流程图

并且我们可以运用以上结论，推广得出 m 道工序的计算流程：

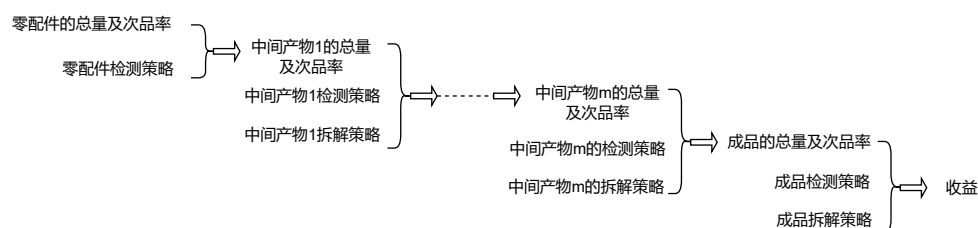


图 11 M 道工序的计算流程

5.3.3 模型计算与结论

综合推论二，我们可以得出，零配件 1-3 中我们只检测 1 个，检测零配件 1 收益最大。检测两种零配件，选择零配件 1、2 收益最大。半成品的次品含量并不受零配件检测个数影响。我们还发现半成品 1 和 2 的参数和构成零配件参数一致，在半成品 1、2 检测总数一致的情况下，选择半成品 1 或者半成品 2 在结果计算上保持一致。至此，我们可以直接排除一些明显不能提高利润或者重复的决策组合，计算得现有的决策组合只剩 714 种。

得到下一层级输入量的半成品含量和占比之后，问题三的后续求解和问题二一致。需要注意，在各个半成品的总量并不一致的情况下，生产成品总量取决于半成品总量的最小值。

经过程序模拟，我们最终得出在决策组合可以取得最大收益，为 56.4 元/件。

以下是各个决策的状态

零件1是否检验	零件2是否检验	零件3是否检验	零件4是否检验	零件5是否检验	零件6是否检验
是	是	是	是	是	是
零件7是否检验	零件8是否检验	半成品1是否检验	半成品2是否检验	半成品3是否检验	半成品1是否拆解
是	是	是	是	是	是
半成品2是否拆解	半成品3是否拆解	成品是否检验	成品是否拆解		
是	是	是	是		

图 12 问题三决策方案

5.4 问题四

5.4.1 置信区间的确认

根据第一问我们求得的 n 与 p_0, z, x 的关系：

$$n = \left(\frac{z \sqrt{p_0(1-p_0)}}{\frac{x}{n} - p_0} \right)^2$$

我们可以根据不同的置信度与 p_0 ，得到满足条件的最小 n 值，通过 n 值与 p_0 确定其置信区间，进而实现将样本的次品率到实际产品的次品率的转化。

由问题二与问题三得我们需要知道在样本次品率分别为 0.1, 0.2, 与 0.05 的情况下, 最小的样本抽样数, 以及其可推论的置信区间。根据大数定律, 我们可得:

$$P\left(-z_{\alpha/2} < \frac{nx - np_0}{\sqrt{np_0(1-p_0)}} < z_{\alpha/2}\right) \approx 1 - \alpha$$

$$-z_{\alpha/2} < \frac{nx - np_0}{\sqrt{np_0(1-p_0)}} < z_{\alpha/2}$$

通过不等式对换, 我们得到:

$$\left(n + z_{\alpha/2}^2\right)p^2 - \left(2nx + z_{\alpha/2}^2\right)p + nx^2 < 0$$

最后我们可以得到对应比例的置信区间:

$$p_1 = \frac{1}{2a} \left(-b + \sqrt{b^2 - 4ac}\right)$$

$$p_2 = \frac{1}{2a} \left(-b - \sqrt{b^2 - 4ac}\right)$$

代入本题中数据, 我们解得以下结果:

当样本次品率为 10%, 抽样数量为 98 时, 我们得到其置信区间为 (0.0604, 0.1611)

当样本次品率为 20%, 抽样数量为 173 时, 我们得到其置信区间为 (0.1548, 0.2545)

当样本次品率为 10%, 抽样数量为 98 时, 我们得到其置信区间为 (0.0188, 0.1266)

5.4.2 次品率决策方案

已知问题 2 的零配件、成品的次品率均是由抽样检测所得到的, 根据问题 2 中对于次品率对决策方案的灵敏度分析, 我们得知在一次区间内次品率改变, 决策方案也会随之改变。在问题二基于次品率概率求出的决策方案, 并不一定适用于通过抽样检测方法得到的次品率。因此结合上文得到的次品率置信区间, 将其代入第二问, 得到在置信区间内的决策是否改变的数据, 最后从新进行决策方案的确定。

情景	零件 1 置信度 为 95% 的置信 区间	在区间内 决策是否 改变	零件 2 置信度 为 95% 的置信 区间	在区间内 决策是否 改变	成品	在区间内 决策是否 改变	决策方案
1	(0.0601,0.1611)	0	(0.0601,0.1611)	0	(0.0601,0.1611)	0	1101
2	(0.1548,0.2545)	0	(0.1548,0.2545)	0	(0.1548,0.2545)	0	1101
3	(0.0604,0.0161)	0	(0.0604,0.0161)	0	(0.0604,0.0161)	1	1101 & 1111
4	(0.1548,0.2545)	0	(0.1548,0.2545)	0	(0.1548,0.2545)	0	1111
5	(0.0601,0.1611)	1	(0.0601,0.1611)	0	(0.0601,0.1611)	1	0100 & 1101 & 0100
6	(0.0188,0.1266)	1	(0.0188,0.1266)	1	(0.0188,0.1266)	0	0000 & 1000

表 5 第二问：次品率波动与决策方案关系

而对于第三问而言，并不需要考虑问题二中所述六种及以上的情况，因此，我们对满足于 95% 的置信度的置信区间采用了离散化处理，通过离散的点集得到对应好的每一个决策。当 $p=0.15$ 到 $p=0.1504$ 之间时，整体决策没有发生改变。

p	profit	零件检验与否	半成品检验与否	不合格半成品拆解与否	成品检验与否	不合格成品拆解与否
0.0604	67.263	1	1	1	1	1
0.1004	58.31	1	1	1	1	1
0.144	48.707	1	1	1	1	1
0.153	46.86	1	1	1	1	1

表 6 第三问：次品率波动对决策的影响

六、模型的评价

6.1 模型的优点

- 1、本模型不仅能适用于题目二、三所提供的生产情况，还能推广至 m 道工序、 n 个零配件的生产情况，进行决策分析，代码复用性高，可适用于大多数情况。
- 2、本文对于几个问题的模型建立具有可传递性，从问题 1 到问题 4 是对模型的层层递进，通过增加时间复杂程度来使模型由特殊到一般，模型层层递进，逐步深入。
- 3、本文在建立模型时，都对结果进行了合理性模拟，通过查找文献、模拟仿真，保证其准确性、科学性、合理性。
- 4、我们在问题 2 中的多阶段决策模型中进行了决策方案对零件、产品次品率的

灵敏度分析，为工厂在实际生产过程中做生产决策提供了有益的建议

6.2 模型的缺点

- 1、本模型没有考虑现实生产中各种批次的不合格零配件都回流的情况，只考虑同一轮次的零件回流之后的统计运算。
- 2、对于次品率相差较大的零件组合未进行分析研究

参考文献

- [1] 李剑焄, 汪悦, 张松, 等. 基于概率分布的元器件质量一致性控制[J]. 中国航天, 2020 (S1):93-98.
- [2] 刘义山, 李骥昭. 多阶段决策过程最优化问题研究[J]. 河南教育学院学报 (自然科学版), 2011, 20(02):8-11.
- [3] 朱培. 多阶段决策问题与中学建模[J]. 上海中学数学, 2003(06):40-43.
- [4] 曹胜炎. 单元内随机抽样法及其应用[J]. 黄牛杂志, 1990(03):7-12.
- [5] 朱国荣. 基于多属性决策的机械零件批次质量评判[J/OL]. 科技风, 2013(13):114. DOI: 10.19392/j.cnki.1671-7341.2013.13.093.

附录 A 文件列表

文件名	功能描述
<i>q1.m</i>	用于求解第一边问最小样本量函数
<i>question2.m</i>	用于求解第二问决策方案函数
<i>exhaustivity.m</i>	用于求解第二问决策方案的文件
<i>question_2_sensitivity.m</i>	零件次品率敏感性分析函数
<i>question_2_sensitivity1.m</i>	成品次品率敏感性分析函数
<i>exhaustivity_sensitivity.m</i>	用于求解第二问敏感性的文件
<i>myplot.m</i>	图形可视化文件
<i>question2_main.m</i>	第二问检测函数主程序
<i>new_round.m</i>	检测函数
<i>combinations_loop</i>	第三问主程序
<i>calculate_profit.m</i>	第三问函数文件
<i>combinations_loop_q4.m</i>	第四问的主程序
<i>calculate_profit_q4.m</i>	第四问函数文件
问题二数据.xlsx	各种情况下 16 种决策方案利润值
问题二敏感性分析.xlsx	各种情况下决策方案随次品率的变化

附录 B 代码

q1.1.m

```

1 p=0.1;
2 z=1.292;
3 x=1:1:100;
4 n=floor((z.*sqrt(p.*(1-p))./0.05).^2);

```

q1.2.m

```

1 n=98;
2 x=0.1;
3 a=n+1.645.^2;
4 b=-(2.*n.*x+1.645^2);
5 c=n.*x.^2;
6 p2=(-b-sqrt(b.*b-4.*a.*c))./2./a;

```

q2.1.m

```
1 function [income_new, r_bad_new1, r_good_new1, r_bad_new2,
   r_good_new2] = new_round(r_bad_1, r_good_1, r_bad_2, r_good_2
   ,if_part1_test,if_part2_test,if_assemble_test,
   if_disassemble_cost)
2
3 income_new=0;%最终收益
4 %零配件相关定义
5
6 part1_test=2;
7 part2_test=3;
8
9 %%成品相关定义
10 %r_Semi_f=0.1; %半成品组装次品率
11 r_f=0.1; %成品组装次品率
12 assemble_cost=6;
13 assemble_test=3;
14
15 disassemble_cost=5;
16 r_f=0.1;
17
18 sale=56;
19 exchange_cost=6;
20
21 %%标志位
22
23 p_zero=r_good_1*r_good_2/(r_bad_2+r_good_2); %成品中零配件12都
   不是坏的概率
24 p_once_1=r_bad_1*r_good_2/(r_bad_2+r_good_2); %成品中只有零
   件1是坏的概率
25 p_once_2=r_bad_2*r_good_1/(r_bad_2+r_good_2);
26 p_once=p_once_1+p_once_2; %成品中有一个坏的概率
27 p_two=r_bad_1*r_bad_2/(r_bad_2+r_good_2); %成品中都是
   坏的概率
28 p_out=0;
```

```

29 p_good_product=0;
30 p_bad_product=0;
31
32 r_good_new1=0;
33 %display(r_good_new1)
34 r_bad_new1=0;
35
36 r_good_new2=0;
37 r_bad_new2=0;
38
39 %%
40 if if_part1_test+if_part2_test==0
41     p_out=p_zero+p_once_1+p_once_2+p_two;    %p_out表示能够进入
        到下一循环的概率
42     p_good_product=p_zero*(1-r_f);
43     p_bad_product=(p_zero*(r_f)+p_once_1+p_once_2+p_two);
44
45     r_bad_new1=(p_once_1+p_two);
46     r_good_new1=p_bad_product-r_bad_new1;
47
48     r_bad_new2=(p_once_2+p_two);
49     r_good_new2=p_bad_product-r_bad_new2;
50
51 elseif if_part1_test+if_part2_test==2
52     p_out=min(r_good_1,r_good_2);
53     p_good_product=p_out*(1-r_f);
54     p_bad_product=(p_out*(r_f));
55
56     income_new=income_new-(r_good_1+r_bad_1)*part1_test-(
        r_good_2+r_bad_2)*part2_test;
57
58     r_bad_new1=0;
59     r_good_new1=p_bad_product;
60
61     r_bad_new2=0;

```



```

62     r_good_new2=p_bad_product;
63
64 elseif if_part1_test==1
65     p_out=p_zero+p_once_2;
66     p_good_product=p_zero*(1-r_f);
67     p_bad_product=(p_zero*(r_f)+p_once_2);
68
69     income_new=income_new-(r_good_1+r_bad_1)*part1_test;
70
71     r_bad_new1=0;
72     r_good_new1=p_bad_product;
73
74     r_bad_new2=(p_once_2);
75     r_good_new2=p_bad_product-r_bad_new2;
76
77 elseif if_part2_test==1
78     p_out=p_zero+p_once_1;
79     p_good_product=p_zero*(1-r_f);
80     p_bad_product=(p_zero*(r_f)+p_once_1);
81
82     income_new=income_new-(r_good_2+r_bad_2)*part2_test;
83
84     r_bad_new1=(p_once_1);
85     r_good_new1=p_bad_product-r_bad_new1;
86
87     r_bad_new2=0;
88     r_good_new2=p_bad_product;
89 end
90 %%
91 income_new=income_new-p_out*assemble_cost;
92
93 if if_assemble_test==1
94     income_new=income_new-p_out*assemble_test;
95 elseif if_assemble_test==0
96     income_new=income_new-p_bad_product*exchange_cost;

```

```

97 end
98 %%
99 income_new=income_new+p_good_product*sale;
100
101 if if_dissemble_cost==1
102     income_new=income_new-p_bad_product*dissemble_cost;
103 elseif if_dissemble_cost==0
104 end
105 end

```

q2.2.m

```

1 %用于求解第二问决策方案主函数(用于敏感性分析)
2 %生成决策变量矩阵
3 a=[0,1];
4 jiance11=[];
5 jiance12=[];
6 jiance2=[];
7 jiance3=[];
8 for i1=1:2
9     for i2=1:2
10         for i3=1:2
11             for i4=1:2
12                 c=[a(i1),a(i2),a(i3),a(i4)];
13                 jiance11=[jiance11,c(1)];
14                 jiance12=[jiance12,c(2)];
15                 jiance2=[jiance2,c(3)];
16                 jiance3=[jiance3,c(4)];
17             end
18         end
19     end
20 end
21
22 jiance=[];
23 jiance=[jiance;jiance11];
24 jiance=[jiance;jiance12];

```

```

25 jiance=[jiance;jiance2];
26 jiance=[jiance;jiance3];
27 liron_q1=zeros(11,16);
28 sum_liron=zeros(1,16);
29 juece=[];
30 % 求解最终结果
31 for dr2=0:0.005:0.90
32 for i=1:16
33 anss=question_2_mingan(jiance(1,i),jiance(2,i),jiance(3,i),
    jiance(4,i),dr2);
34 liron_q1(:,i)=anss';
35 sum_liron(i)=sum(anss);%利润和数组
36 end
37 [max_liron,index]=max(sum_liron);%求出零件次品率对应的最佳决策
    变量
38 juece=[juece,jiance(:,index)];%求出决策变量矩阵
39 end
40 juece=juece';
41
42 % % 求解最终结果
43 % for dr3=0:0.005:0.90
44 % for i=1:16
45 % anss=question_2_mingan1(jiance(1,i),jiance(2,i),jiance(3,i),
    jiance(4,i),dr3);
46 % liron_q1(:,i)=anss';
47 % sum_liron(i)=sum(anss);%利润和数组
48 % end
49 % [max_liron,index]=max(sum_liron);%求出零件次品率对应的最佳决
    策变量
50 % juece=[juece,jiance(:,index)];%求出决策变量矩阵
51 % end
52 % juece=juece';
53
54 % % 不可行，暂停不了
55 % % for dr1=0:0.01:0.90

```

```

56 % %      for dr2=0:0.01:0.90
57 % %          for dr3=0:0.01:0.90
58 % % for i=1:16
59 % % anss=question_2_mingan(jiance(1,i),jiance(2,i),jiance(3,i)
    ,jiance(4,i),dr1,dr2,dr3);
60 % % lirun_q1(:,i)=anss';
61 % % sum_lirun(i)=sum(anss);%利润和数组
62 % % end
63 % %
64 % % [max_lirun,index]=max(sum_lirun);%求出零件次品率对应的最佳
    决策变量
65 % % juece=[juece,jiance(:,index)];%求出决策变量矩阵
66 % %          end
67 % %      end
68 % % end

```

q2.3.m

```

1 %用于求解第二问决策方案函数(用于敏感性分析零件cipinlv)
2 function lirun_all=question_2_mingan(jiance11,jiance12,jiance2
    ,jiance3,dr2)
3 % %情况1
4 % dr1=0.1;
5 % % dr2=0.1;
6 % dr3=0.1;
7 % danjia1=4;
8 % danjia2=18;
9 % tc1=2;
10 % tc2=3;
11 % tc3=6;
12 % tc4=3;
13 % tc5=6;
14 % tc6=56;
15 % tc7=5;
16
17 % % % 情况2

```

```

18 % dr1=0.2;
19 % % dr2=0.2;
20 % dr3=0.2;
21 % danjia1=4;
22 % danjia2=18;
23 % tc1=2;%零件1检测成本
24 % tc2=3;%零件2检测成本
25 % tc3=6;%装配成本
26 % tc4=3;%成品检测成本
27 % tc5=6;%损失调换
28 % tc6=56;%市场售价
29 % tc7=5;%拆解费用
30
31 % %情况3
32 % dr1=0.1;
33 % % dr2=0.1;
34 % dr3=0.1;
35 % danjia1=4;
36 % danjia2=18;
37 % tc1=2;%零件1检测成本
38 % tc2=3;%零件2检测成本
39 % tc3=6;%装配成本
40 % tc4=3;%成品检测成本
41 % tc5=30;%损失调换
42 % tc6=56;%市场售价
43 % tc7=5;%拆解费用
44
45 % %情况4
46 % dr1=0.2;
47 % % dr2=0.2;
48 % dr3=0.2;
49 % danjia1=4;
50 % danjia2=18;
51 % tc1=1;%零件1检测成本
52 % tc2=1;%零件2检测成本

```

```

53 % tc3=6;%装配成本
54 % tc4=2;%成品检测成本
55 % tc5=30;%损失调换
56 % tc6=56;%市场售价
57 % tc7=5;%拆解费用
58
59 % %情况5
60 % dr1=0.1;
61 % % % dr2=0.2;
62 % dr3=0.1;
63 % danjia1=4;
64 % danjia2=18;
65 % tc1=8;%零件1检测成本
66 % tc2=1;%零件2检测成本
67 % tc3=6;%装配成本
68 % tc4=2;%成品检测成本
69 % tc5=10;%损失调换
70 % tc6=56;%市场售价
71 % tc7=5;%拆解费用
72
73 %情况6
74 dr1=0.05;
75 % dr2=0.05;
76 dr3=0.05;
77 danjia1=4;
78 danjia2=18;
79 tc1=2;%零件1检测成本
80 tc2=3;%零件2检测成本
81 tc3=6;%装配成本
82 tc4=3;%成品检测成本
83 tc5=10;%损失调换
84 tc6=56;%市场售价
85 tc7=40;%拆解费用
86
87 p01=1;

```

```

88 p02=1;
89 %单位零件1检测成本tc1,单位零件2检测成本tc2,单位成品装配成本tc3
90 %零件1次品率dr1,零件2次品率dr2
91 ttc1=tc1*p01;
92 ttc2=tc2*p02;
93 ttc1=ttc1*jiance11;
94 ttc2=ttc2*jiance12;
95 %零件单价danjia1,零件2单价danjian2;零件1value1, value2;
96 value1=danjia1*p01;
97 value2=danjia2*p02;
98 %零件1好的数量p01=p02,c零件1次品率dr1;
99 p011=p01*(1-dr1);
100 p022=p02*(1-dr2);
101 %带组装的成品的数量pn1
102 if(jiance11==1&&jiance12==1)
103     pn1=min(p011,p022);
104     pn2=pn1*(1-dr3);
105 elseif(jiance11==0&&jiance12==0)
106     pn1=min(p01,p02);
107     pn2=pn1*(1-dr1)*(1-dr2)*(1-dr3);
108 elseif jiance11==1&&jiance12==0
109     pn1=min(p011,p02);
110     pn2=pn1*(1-dr2)*(1-dr3);
111 else
112     pn1=min(p01,p022);
113     pn2=pn1*(1-dr1)*(1-dr3);
114 end
115
116 %第一轮决策得到成本和利润c1,s1
117 c1=ttc1+ttc2+value1+value2;
118 s1=0;
119
120 %成品检测2
121 %if(c1<tc4)
122 %jiance2=1;

```

```

123 %end
124 %if(jiance2==1)
125     %成品检测成本ttc3
126 %成品次品率dr3,好的成品数量pn2;
127
128 s2=pn2*tc6;
129 ttc3=pn1*tc3;%装配成本
130 ttc4=tc4*pn1;
131 ttc4=ttc4*jiance2;%检测成本
132
133 if jiance2==1
134 %第二轮赚的钱s2;
135 %成品检测ttc4
136     c2=ttc3+ttc4;
137     pn3=pn1-pn2; %需要拆的数量
138 %调换损失钱ttc5;
139 else
140     %坏的成品的数量pn3;
141     pn3=pn1-pn2;
142     ttc5=pn3*tc5;
143     c2=ttc5+ttc3+ttc4;
144 end
145 pn0=pn3;
146
147 %第三次决策不拆解
148 %需要拆解的数量pn3,拆解需要钱ttc6
149 if(jiance3==0)
150     s3=0;
151     c3=0;
152 else
153     ttc6=tc7*pn3;
154
155 s3=0;
156 c3=ttc6;
157 end

```



```

158
159 %第一轮利润
160 liron_last = s1+s2+s3-c1-c2-c3;
161 %ttc7存放迭代利润
162 ttc7=zeros(1,10);
163 if(jiance3==1)
164 for i=1:10
165 %迭代变量清零赋值
166 dr1=0.1;
167 dr2=0.1;
168 if(jiance11==1&&jiance12==1)
169     dr1=0;
170     dr2=0;
171 elseif(jiance11==0&&jiance12==0)
172     dr11=dr1;
173     dr1=dr11/pn3;
174     dr22=dr2;
175     dr2=dr22/pn3;
176 elseif(jiance11==1&&jiance12==0)
177     dr22=dr2*(1-dr1);
178     dr2=dr22/pn3;
179     dr1=0;
180 elseif(jiance11==0&&jiance12==1)
181     dr11=dr1*(1-dr2);
182     dr1=dr11/pn3;
183     dr2=0;
184
185
186
187
188 end
189 p01=pn3;
190 p02=pn3;
191
192 [ttc7(i),pn3]=liron_1(p01,p02,dr1,dr2,dr3,danjia1,danjia2,tc1

```

```

    ,tc2,tc3,tc4,tc5,tc6,tc7,jiance11,jiance12,jiance2,jiance3);
193 i=i+1;
194 end
195 end
196 lirun_all=[lirun_last,ttc7];
197 end

```

q3.1.m

```

1 function [income] = calculate_profit(p_test,s_test,s_dis,
    f_test,f_dis)
2
3 %假设购买数量一致。
4 %% 变量定义
5 income=0;%最终收益
6 %%标志位
7     %零配件相关定义
8 % r_part1=0.1;%零配件的次品率
9 % r_bad_part=0.1;
10 % r_good_part=0.9;
11
12 part_information=[
13     %每行分别是次品量、非次品量、购买单价、检测成本、是否检测
    (0: 不检测  1: 检测)
14     0.1, 0.9,  2,  1,  1; %零件1
15     0.1, 0.9,  8,  1,  1;
16     0.1, 0.9, 12, 2,  1;
17
18     0.1, 0.9,  2,  1,  1;
19     0.1, 0.9,  8,  1,  1;
20     0.1, 0.9, 12, 2,  1;
21
22
23     0.1, 0.9,  8,  1,  1;
24     0.1, 0.9, 12, 2,  1; %零件8
25     0.1, 1.0,  0,  0,  0;

```

```

26
27 ];
28 part_information_new=zeros(8,3,10);
29 %(行, 列)
30
31 semi_f_information=[
32     %每行分别是半成品组装次品率、装配成本、检测成本、拆解费用
    和
33     %是否检测 0: 不检测 1: 检测
34     %是否拆解 0: 不拆解 1: 拆解
35     %如果后者选择拆解前者必须选择检验
36     0.1, 8, 4, 6, 1, 0;
37     0.1, 8, 4, 6, 1, 0;
38     0.1, 8, 4, 6, 1, 0;
39 ];
40 if_f_test=1;%只改变成品的钱额计算
41 if_f_disassemble=0;
42
43 semi_f_product=zeros(3,3);%半成品1, 好的部分, 坏的部分, 加起来
    的含量
44
45 r_f=0.1;
46 f_assemble_cost=8;
47 f_assemble_test=6;
48 f_disassemble_cost=10;
49
50 sale=200;
51 exchange_cost=40;
52
53 %% 根据传入值进行赋值
54 for i=1:9
55     part_information(i,5)=p_test(i);
56 end
57 for i=1:3
58     semi_f_information(i,5)=s_test(i);

```

```

59     semi_f_information(i,6)=s_dis(i);
60 end
61 if_f_test=f_test;
62 if_f_dissemble=f_dis;
63
64 %% 去除亏本的组合和不存在的组合
65 if if_f_dissemble==1
66     for j=1:3
67         semi_f_information(j,5)=1;
68     end
69 end
70
71 for i=0:2
72     if semi_f_information(i+1,6)==1
73         semi_f_information(i+1,5)=1;
74         for j=1:3
75             part_information(i*3+j,5)=1;
76         end
77     end
78 end
79
80 p_semi=zeros(3,8);
81 I=ones(1,3);
82
83 for i=0:1:2
84     [M,I(i+1)] = min(part_information(1+i*3:3+i*3,1)+
85         part_information(1+i*3:3+i*3,2));
86 end
87 %I
88 %[M,I(3)]=min(part_information(1+6:3+6,1)+part_information
89     (1+6:3+6,2))
90 %% 计算每个半成品的构成含量
91 for i=0:2
92     if I(i+1)==1
93         p_semi(i+1,1)=part_information(1+i*3,2)*

```

```

part_information(2+i*3,2)/sum(part_information(2+i*3,1:2))*
part_information(3+i*3,2)/sum(part_information(3+i*3,1:2));
92     p_semi(i+1,2)=part_information(1+i*3,1)*
part_information(2+i*3,2)/sum(part_information(2+i*3,1:2))*
part_information(3+i*3,2)/sum(part_information(3+i*3,1:2));%
零件1坏
93     p_semi(i+1,3)=part_information(1+i*3,2)*
part_information(2+i*3,1)/sum(part_information(2+i*3,1:2))*
part_information(3+i*3,2)/sum(part_information(3+i*3,1:2));
94     p_semi(i+1,4)=part_information(1+i*3,2)*
part_information(2+i*3,2)/sum(part_information(2+i*3,1:2))*
part_information(3+i*3,1)/sum(part_information(3+i*3,1:2));
95     p_semi(i+1,5)=part_information(1+i*3,1)*
part_information(2+i*3,1)/sum(part_information(2+i*3,1:2))*
part_information(3+i*3,2)/sum(part_information(3+i*3,1:2));%
零件12坏
96     p_semi(i+1,6)=part_information(1+i*3,2)*
part_information(2+i*3,1)/sum(part_information(2+i*3,1:2))*
part_information(3+i*3,1)/sum(part_information(3+i*3,1:2));
97     p_semi(i+1,7)=part_information(1+i*3,1)*
part_information(2+i*3,2)/sum(part_information(2+i*3,1:2))*
part_information(3+i*3,1)/sum(part_information(3+i*3,1:2));
98     p_semi(i+1,8)=part_information(1+i*3,1)*
part_information(2+i*3,1)/sum(part_information(2+i*3,1:2))*
part_information(3+i*3,1)/sum(part_information(3+i*3,1:2));%
零件123坏
99     elseif I(i+1)==2
100         p_semi(i+1,1)=part_information(2+i*3,2)*
part_information(1+i*3,2)/sum(part_information(1+i*3,1:2))*
part_information(3+i*3,2)/sum(part_information(3+i*3,1:2));
101         p_semi(i+1,2)=part_information(2+i*3,2)*
part_information(1+i*3,1)/sum(part_information(1+i*3,1:2))*
part_information(3+i*3,2)/sum(part_information(3+i*3,1:2));%
零件1坏
102         p_semi(i+1,3)=part_information(2+i*3,1)*

```

```

part_information(1+i*3,2)/sum(part_information(1+i*3,1:2))*
part_information(3+i*3,2)/sum(part_information(3+i*3,1:2));
103     p_semi(i+1,4)=part_information(2+i*3,2)*
part_information(1+i*3,2)/sum(part_information(1+i*3,1:2))*
part_information(3+i*3,1)/sum(part_information(3+i*3,1:2));
104     p_semi(i+1,5)=part_information(2+i*3,1)*
part_information(1+i*3,1)/sum(part_information(1+i*3,1:2))*
part_information(3+i*3,2)/sum(part_information(3+i*3,1:2));%
零件12坏
105     p_semi(i+1,6)=part_information(2+i*3,1)*
part_information(1+i*3,2)/sum(part_information(1+i*3,1:2))*
part_information(3+i*3,1)/sum(part_information(3+i*3,1:2));
106     p_semi(i+1,7)=part_information(2+i*3,2)*
part_information(1+i*3,1)/sum(part_information(1+i*3,1:2))*
part_information(3+i*3,1)/sum(part_information(3+i*3,1:2));
107     p_semi(i+1,8)=part_information(2+i*3,1)*
part_information(1+i*3,1)/sum(part_information(1+i*3,1:2))*
part_information(3+i*3,1)/sum(part_information(3+i*3,1:2));%
零件123坏
108     elseif I(i+1)==3
109         p_semi(i+1,1)=part_information(3+i*3,2)*
part_information(2+i*3,2)/sum(part_information(2+i*3,1:2))*
part_information(1+i*3,2)/sum(part_information(1+i*3,1:2));
110         p_semi(i+1,2)=part_information(3+i*3,2)*
part_information(2+i*3,2)/sum(part_information(2+i*3,1:2))*
part_information(1+i*3,1)/sum(part_information(1+i*3,1:2));%
零件1坏
111         p_semi(i+1,3)=part_information(3+i*3,2)*
part_information(2+i*3,1)/sum(part_information(2+i*3,1:2))*
part_information(1+i*3,2)/sum(part_information(1+i*3,1:2));
112         p_semi(i+1,4)=part_information(3+i*3,1)*
part_information(2+i*3,2)/sum(part_information(2+i*3,1:2))*
part_information(1+i*3,2)/sum(part_information(1+i*3,1:2));
113         p_semi(i+1,5)=part_information(3+i*3,2)*
part_information(2+i*3,1)/sum(part_information(2+i*3,1:2))*

```

```

part_information(1+i*3,1)/sum(part_information(1+i*3,1:2));%
零件12坏
114     p_semi(i+1,6)=part_information(3+i*3,1)*
part_information(2+i*3,1)/sum(part_information(2+i*3,1:2))*
part_information(1+i*3,2)/sum(part_information(1+i*3,1:2));
115     p_semi(i+1,7)=part_information(3+i*3,1)*
part_information(2+i*3,2)/sum(part_information(2+i*3,1:2))*
part_information(1+i*3,1)/sum(part_information(1+i*3,1:2));
116     p_semi(i+1,8)=part_information(3+i*3,1)*
part_information(2+i*3,1)/sum(part_information(2+i*3,1:2))*
part_information(1+i*3,1)/sum(part_information(1+i*3,1:2));%
零件123坏
117
118     end
119 end
120 % p_semi
121 %将组装失败的计入进去
122 for i=0:2
123
124     if sum(part_information(1+i*3:3+i*3,5))==0 %全部不检测
125         semi_f_product(1+i,1)=p_semi(1+i,1)*(1-
semi_f_information(1,1));
126         semi_f_product(1+i,2)=p_semi(1+i,1)*(
semi_f_information(1,1))+sum(p_semi(1+i,2:8));
127         income=income-sum(part_information(1+i*3,1:2))*
part_information(1+i*3,3)-sum(part_information(2+i*3,1:2))*
part_information(2+i*3,3)-sum(part_information(3+i*3,1:2))*
part_information(3+i*3,3);
128         part_information_new(1+i*3,1,1)=p_semi(i+1,1)*(
semi_f_information(1,1))+p_semi(1+i,3)+p_semi(1+i,4)+p_semi
(1+i,6); %好的成分
129         part_information_new(1+i*3,2,1)=p_semi(1+i,2)+p_semi
(1+i,5)+p_semi(1+i,7)+p_semi(1+i,8);
130
131         part_information_new(2+i*3,1,1)=p_semi(i+1,1)*(

```

```

semi_f_information(1,1))+p_semi(1+i,2)+p_semi(1+i,4)+p_semi
(1+i,7);
132     part_information_new(2+i*3,2,1)=p_semi(1+i,3)+p_semi
(1+i,5)+p_semi(1+i,6)+p_semi(1+i,8);
133
134     part_information_new(3+i*3,1,1)=p_semi(i+1,1)*(
semi_f_information(1,1))+p_semi(1+i,2)+p_semi(1+i,3)+p_semi
(1+i,5);
135     part_information_new(3+i*3,2,1)=p_semi(1+i,4)+p_semi
(1+i,6)+p_semi(1+i,7)+p_semi(1+i,8);
136
137     elseif sum(part_information(1+i*3:3+i*3,5))==3    %全部都
检测
138         % [x,y]=min(part_information(1+i*3:3+i*3,1)+
part_information(1+i*3:3+i*3,2));
139         semi_f_product(1+i,1)=min(part_information(1+i*3:3+i
*3,2))*(1-semi_f_information(1,1));    %全部都组装成功
140         semi_f_product(1+i,2)=min(part_information(1+i*3:3+i
*3,2))*(semi_f_information(1,1));
141         income=income-sum(part_information(1+i*3,1:2))*
part_information(1+i*3,3)-sum(part_information(2+i*3,1:2))*
part_information(2+i*3,3)-sum(part_information(3+i*3,1:2))*
part_information(3+i*3,3);
142         income=income-sum(part_information(1+i*3,1:2))*
part_information(1+i*3,4)-sum(part_information(2+i*3,1:2))*
part_information(2+i*3,4)-sum(part_information(3+i*3,1:2))*
part_information(3+i*3,4);
143         part_information_new(1+i*3,1,1)=semi_f_product(1+i,2);
144         part_information_new(1+i*3,2,1)=0;
145
146         part_information_new(2+i*3,1,1)=semi_f_product(1+i,2);
147         part_information_new(2+i*3,2,1)=0;
148
149         part_information_new(3+i*3,1,1)=semi_f_product(1+i,2);
150         part_information_new(3+i*3,2,1)=0;

```



```

151     elseif sum(part_information(1+i*3:3+i*3,5))==1 %由上一题
推论可得查检测成本和购买单价最少的    检测1最合适
152         % if i~=2
153         semi_f_product(1+i,1)=p_semi(i+1,1)*(1-
semi_f_information(1,1)); %全部都组装成功
154         semi_f_product(1+i,2)=p_semi(i+1,1)*(
semi_f_information(1,1))+p_semi(i+1,3)+p_semi(i+1,4)+p_semi(
i+1,6);
155         income=income-sum(part_information(1+i*3,1:2))*
part_information(1+i*3,3)-sum(part_information(2+i*3,1:2))*
part_information(2+i*3,3)-sum(part_information(3+i*3,1:2))*
part_information(3+i*3,3);
156         income=income-sum(part_information(1+i*3,1:2))*
part_information(1+i*3,4);%检查第一个收益比较大
157         part_information_new(1+i*3,1,1)=p_semi(i+1,1)*(
semi_f_information(1,1))+p_semi(i+1,3)+p_semi(i+1,4)+p_semi(
i+1,6);
158         part_information_new(1+i*3,2,1)=0;
159
160         part_information_new(2+i*3,1,1)=p_semi(i+1,1)*(
semi_f_information(1,1))+p_semi(i+1,4);
161         part_information_new(2+i*3,2,1)=p_semi(i+1,3)+p_semi(i
+1,6);
162
163         part_information_new(3+i*3,1,1)=p_semi(i+1,1)*(
semi_f_information(1,1))+p_semi(i+1,3);
164         part_information_new(3+i*3,2,1)=p_semi(i+1,4)+p_semi(i
+1,6);
165
166     elseif sum(part_information(1+i*3:3+i*3,5))==2
167         if i~=2
168
169         semi_f_product(1+i,1)=min(part_information(1+i*3:3+i
*3,2))*part_information(3+i*3,2)*(1-semi_f_information(1,1))
; %全部都组装成功

```

```

170         semi_f_product(1+i,2)=min(part_information(1+i*3:3+i
*3,2))*part_information(3+i*3,2)*(semi_f_information(1,1))+
min(part_information(1+i*3:3+i*3,2))*part_information(3+i
*3,1);
171         income=income-sum(part_information(1+i*3,1:2))*
part_information(1+i*3,3)-sum(part_information(2+i*3,1:2))*
part_information(2+i*3,3)-sum(part_information(3+i*3,1:2))*
part_information(3+i*3,3);
172         income=income-sum(part_information(1+i*3,1:2))*
part_information(1+i*3,4)-sum(part_information(2+i*3,1:2))*
part_information(2+i*3,4);
173         part_information_new(1+i*3,1,1)=semi_f_product(1+i,2);
174         part_information_new(1+i*3,2,1)=0;
175
176         part_information_new(2+i*3,1,1)=semi_f_product(1+i,2);
177         part_information_new(2+i*3,2,1)=0;
178
179         part_information_new(3+i*3,1,1)=min(part_information
(1+i*3:3+i*3,2))*part_information(3+i*3,2)*(
semi_f_information(1,1));
180         part_information_new(3+i*3,2,1)=min(part_information
(1+i*3:3+i*3,2))*part_information(3+i*3,1);
181         elseif i==2
182         semi_f_product(1+i,1)=min(part_information(1+i*3:3+i
*3,2))*(1-semi_f_information(1,1));    %全部都组装成功
183         semi_f_product(1+i,2)=min(part_information(1+i*3:3+i
*3,2))*(semi_f_information(1,1));
184         income=income-sum(part_information(1+i*3,1:2))*
part_information(1+i*3,3)-sum(part_information(2+i*3,1:2))*
part_information(2+i*3,3)-sum(part_information(3+i*3,1:2))*
part_information(3+i*3,3);
185
186         income=income-sum(part_information(1+i*3,1:2))*
part_information(1+i*3,4)-sum(part_information(2+i*3,1:2))*
part_information(2+i*3,4);    %检查前两个收益比较大

```

```

187
188     part_information_new(1+i*3,1,1)=semi_f_product(1+i,2);
189     part_information_new(1+i*3,2,1)=0;
190
191     part_information_new(2+i*3,1,1)=semi_f_product(1+i,2);
192     part_information_new(2+i*3,2,1)=0;
193
194     part_information_new(3+i*3,1,1)=semi_f_product(1+i,2);
195     part_information_new(3+i*3,2,1)=p_semi(i+1,4);
196     end
197
198     end
199
200 end
201 %% 将半成品的含量计算出来
202 for i=1:3
203     semi_f_product(i,3)=semi_f_product(i,1)+semi_f_product(i
204         ,2);
205 end
206 % semi_f_product
207 % part_information_new(:, :, 1)
208 % p_out=0;
209 % p_good_product=0;
210 % p_bad_product=0;
211
212 %% 计算成本 得到最新的半成品含量
213 for i=0:2
214     income=income-semi_f_product(i+1,3)*semi_f_information(i
215         +1,2);%装配的钱
216
217     if semi_f_information(i+1,5)==0
218
219         elseif semi_f_information(i+1,6)==0 %检测半成品并且把次品
220             丢弃
221
222         income=income-semi_f_product(i+1,3)*semi_f_information

```

```

(i+1,3);%检测的钱
219     semi_f_product(i+1,2)=0;
220     semi_f_product(i+1,3)=semi_f_product(i+1,1);
221
222     elseif semi_f_information(i+1,6)==1 %检测半成品并且把次品
拆解
223 %如果拆解，前一阶段必须完全测验，否则会造成负反馈持续亏钱
224     income=income-semi_f_product(i+1,3)*semi_f_information
(i+1,3);%第一次检测的钱
225     semi_f_product(i+1,1)=semi_f_product(i+1,1)+
semi_f_product(i+1,2); %所有次品最终都会转化成非次品
226     semi_f_product(i+1,2)=0;
227     income=income-10/9*semi_f_product(i+1,2)*(sum(
part_information(1+i*3:3+i*3,4))+sum(semi_f_information(i
+1,2:4)));
228     end
229 end
230
231 % semi_f_product
232
233 f_product=zeros(1,8);
234 [M,U] = min(semi_f_product(:,3));
235
236 if U==1
237     f_product(1)=semi_f_product(1,1)*semi_f_product(2,1)*
semi_f_product(3,1)/semi_f_product(2,3)/semi_f_product(3,3);
238     f_product(2)=semi_f_product(1,2)*semi_f_product(2,1)*
semi_f_product(3,1)/semi_f_product(2,3)/semi_f_product(3,3);
239     f_product(3)=semi_f_product(1,1)*semi_f_product(2,2)*
semi_f_product(3,1)/semi_f_product(2,3)/semi_f_product(3,3);
240     f_product(4)=semi_f_product(1,1)*semi_f_product(2,1)*
semi_f_product(3,2)/semi_f_product(2,3)/semi_f_product(3,3);
241     f_product(5)=semi_f_product(1,2)*semi_f_product(2,2)*
semi_f_product(3,1)/semi_f_product(2,3)/semi_f_product(3,3);
242     f_product(6)=semi_f_product(1,1)*semi_f_product(2,2)*

```

```

243     semi_f_product(3,2)/semi_f_product(2,3)/semi_f_product(3,3);
244     f_product(7)=semi_f_product(1,2)*semi_f_product(2,1)*
245     semi_f_product(3,2)/semi_f_product(2,3)/semi_f_product(3,3);
246     f_product(8)=semi_f_product(1,2)*semi_f_product(2,2)*
247     semi_f_product(3,2)/semi_f_product(2,3)/semi_f_product(3,3);
248 elseif U==2
249     f_product(1)=semi_f_product(1,1)*semi_f_product(2,1)*
250     semi_f_product(3,1)/semi_f_product(1,3)/semi_f_product(3,3);
251     f_product(2)=semi_f_product(1,2)*semi_f_product(2,1)*
252     semi_f_product(3,1)/semi_f_product(1,3)/semi_f_product(3,3);
253     f_product(3)=semi_f_product(1,1)*semi_f_product(2,2)*
254     semi_f_product(3,1)/semi_f_product(1,3)/semi_f_product(3,3);
255     f_product(4)=semi_f_product(1,1)*semi_f_product(2,1)*
256     semi_f_product(3,2)/semi_f_product(1,3)/semi_f_product(3,3);
257     f_product(5)=semi_f_product(1,2)*semi_f_product(2,2)*
258     semi_f_product(3,1)/semi_f_product(1,3)/semi_f_product(3,3);
259     f_product(6)=semi_f_product(1,1)*semi_f_product(2,2)*
260     semi_f_product(3,2)/semi_f_product(1,3)/semi_f_product(3,3);
261     f_product(7)=semi_f_product(1,2)*semi_f_product(2,1)*
262     semi_f_product(3,2)/semi_f_product(1,3)/semi_f_product(3,3);
263     f_product(8)=semi_f_product(1,2)*semi_f_product(2,2)*
264     semi_f_product(3,2)/semi_f_product(1,3)/semi_f_product(3,3);
265 elseif U==3
266     f_product(1)=semi_f_product(1,1)*semi_f_product(2,1)*
267     semi_f_product(3,1)/semi_f_product(2,3)/semi_f_product(1,3);
268     f_product(2)=semi_f_product(1,2)*semi_f_product(2,1)*
269     semi_f_product(3,1)/semi_f_product(2,3)/semi_f_product(1,3);
270     f_product(3)=semi_f_product(1,1)*semi_f_product(2,2)*
271     semi_f_product(3,1)/semi_f_product(2,3)/semi_f_product(1,3);
272     f_product(4)=semi_f_product(1,1)*semi_f_product(2,1)*
273     semi_f_product(3,2)/semi_f_product(2,3)/semi_f_product(1,3);
274     f_product(5)=semi_f_product(1,2)*semi_f_product(2,2)*
275     semi_f_product(3,1)/semi_f_product(2,3)/semi_f_product(1,3);
276     f_product(6)=semi_f_product(1,1)*semi_f_product(2,2)*
277     semi_f_product(3,2)/semi_f_product(2,3)/semi_f_product(1,3);

```

```

261     f_product(7)=semi_f_product(1,2)*semi_f_product(2,1)*
      semi_f_product(3,2)/semi_f_product(2,3)/semi_f_product(1,3);
262     f_product(8)=semi_f_product(1,2)*semi_f_product(2,2)*
      semi_f_product(3,2)/semi_f_product(2,3)/semi_f_product(1,3);
263 end
264 %
265 % semi_f_product
266 % f_product
267 %sum(f_product)
268 income=income-sum(f_product)*f_assemble_cost;
269 p_good_product=f_product(1)*(1-r_f);
270 p_bad_product=f_product(1)*(r_f)+sum(f_product(2:8));
271 if if_f_test==1
272     income=income-sum(f_product)*f_assemble_test;
273 elseif if_f_test==0
274     income=income-p_bad_product*exchange_cost;
275 end
276
277 income=income+p_good_product*sale;
278 if if_f_dissemble==1 %如果拆解，则前一层级必须全部检验，否则
      会造成负反馈。
279     income=income+f_product(1)*(r_f)*sale;
280
281     if if_f_test==1
282         income=income-10/9*p_bad_product*(f_dissemble_cost+sum(
      semi_f_information(:,3))+f_assemble_cost+f_assemble_test);
283         % income=income-sum(f_product)*f_assemble_test;
284     elseif if_f_test==0
285         income=income-10/9*p_bad_product*(f_dissemble_cost+sum(
      semi_f_information(:,3))+f_assemble_cost+r_f*exchange_cost);
286         %income=income-p_bad_product*exchange_cost;
287     end
288     % income_new(1)=income;
289
290 end

```

```

291
292 % income
293 end

```

q4.2.m

```

1  clc;clear;
2  income=zeros(1,4000);
3  p_test=zeros(1,9);
4  p_test_possibility=[    %24
5      0,0,0,0,0,0,0,0,0;
6      1,0,0,0,0,0,0,0,0;
7      1,0,0,1,0,0,0,0,0;
8      1,0,0,1,1,0,0,0,0;
9      1,0,0,1,1,1,0,0,0;
10     1,1,0,1,1,0,0,0,0;
11     1,1,0,1,1,1,0,0,0;
12     1,1,1,1,1,1,0,0,0;
13     0,0,0,0,0,0,1,0,0;
14     1,0,0,0,0,0,1,0,0;
15     1,0,0,1,0,0,1,0,0;
16     1,0,0,1,1,0,1,0,0;
17     1,0,0,1,1,1,1,0,0;
18     1,1,0,1,1,0,1,0,0;
19     1,1,0,1,1,1,1,0,0;
20     1,1,1,1,1,1,1,0,0;
21     0,0,0,0,0,0,1,1,0;
22     1,0,0,0,0,0,1,1,0;
23     1,0,0,1,0,0,1,1,0;
24     1,0,0,1,1,0,1,1,0;
25     1,0,0,1,1,1,1,1,0;
26     1,1,0,1,1,0,1,1,0;
27     1,1,0,1,1,1,1,1,0;
28     1,1,1,1,1,1,1,1,0;
29 ];
30 s_test=zeros(1,3);

```

```

31 s_test_possibility=[
32     0,0,0;
33     1,0,0;
34     0,0,1;
35     1,1,0;
36     1,0,1;
37     1,1,1,
38 ];
39 s_dis=zeros(1,3);
40 s_dis_possibility=[
41     0,0,0;
42     1,0,0;
43     0,0,1;
44     1,1,0;
45     1,0,1;
46     1,1,1;
47 ];
48 f_dis=0;
49 f_test=0;
50 % n=0;
51
52 n=1;
53 income_out=zeros(4000,17);
54 % [income_out(n,1)] = calculate_profit(p_test,s_test,s_dis,
55     f_test,f_dis);
56 income_out(n,1:9)=p_test;
57 income_out(n,10:12)=s_test;
58 income_out(n,13:15)=s_dis;
59 income_out(n,16)=f_test;
60 income_out(n,17)=f_dis;
61 ratio=0.0604:0.01:0.1611;
62 result=zeros(100,20);
63 hh=0;
64 for r=0.14:0.001:0.155;
65 for i=0:1

```



```

65     for j=0:1
66         for k=1:6
67             for x=1:6
68                 for y=1:24
69
70                     n=n+1;
71                     [income(n)] = calculate_profit(
p_test_possibility(y,:),s_test_possibility(x,:),
s_dis_possibility(k,:),j,i,0.1);
72                     income_out(n,1:9)=p_test_possibility(y,:);
73                     income_out(n,10:12)=s_test_possibility(x
,:);
74                     income_out(n,13:15)=s_dis_possibility(k,:)
;
75                     income_out(n,16)=j;
76                     if j==1
77                         income_out(n,10:12)=ones(1,3);
78                     end
79                     for a=1:3
80                         if s_dis_possibility(k,a)==1
81                             income_out(n,a*3-2:a*3)=ones(1,3);
82                         end
83                     end
84                     income_out(n,17)=i;
85
86                 end
87             end
88         end
89     end
90 end
91 [x,y]=max(income);
92 income_out(y,:);
93 n=0;
94 hh=hh+1;
95 result(hh,1)=r;

```

```

96 result(hh,2)=max(income);
97 result(hh,3:19)=income_out(y,:);
98 income=zeros(1,4000);
99 income_out=zeros(4000,17);
100 end

```

附录 C 表格

决策方案	1101	1111	1100	0000	1000	1110
初始利润	11.97	9.81	12.42	11.198	10.398	10.26
迭代1次利润	3.447	3.231	0	0	0	0
迭代2次利润	0.3447	0.3231	0	0	0	0
迭代3次利润	0.03447	0.03231	0	0	0	0
迭代4次利润	0.003447	0.003231	0	0	0	0
迭代5次利润	0.0003447	0.0003231	0	0	0	0
迭代6次利润	3.45E-05	3.23E-05	0	0	0	0
迭代7次利润	3.45E-06	3.23E-06	0	0	0	0
迭代8次利润	3.45E-07	3.23E-07	0	0	0	0
迭代9次利润	3.45E-08	3.23E-08	0	0	0	0
迭代10次利润	3.45E-09	3.23E-09	0	0	0	0
sum	15.8	13.4	12.42	11.198	10.398	10.26

图 13 图片/情况 1 利润前五决策方案表格