基于假设检验和 ASM 状态机的生产过程决策问题研究

摘要

本文主要研究企业在实际生产中的决策问题，设计合理的抽样检测方案，并构造假设检验模型、多阶段单目标决策问题的递归 ASM 状态机模型，分别应用于产品的次品率确定问题和单/多流水线整体生产决策问题的求解。

对于问题一， 首先分析题目中所给两种情形，确定各自对应的原假设和对立假设，依据给出的显著性水平 α 建立**单侧假设检验模型**。在模型的求解中，根据批次总量 N 和样本容量 n 的关系，分别构建**超几何分布概率模型** $(n \geq 0.1N)$ 和二**项分布模概率型** $(n < 0.1N)$，并根据各自的累积密度函数完成假设检验问题当中条件概率的求解，对应解得 (N, n, k) 的合理值。对于**情形**一，解得**放宽检测**、**正常检测**、**严格检测策略分别为** $(N, 38, 1)$、$(N, 50, 2)$、$(N, 80, 4)$，并得到包括全检的四种检测方式的具体使用策略。对于**情形**二，解得**放宽检测**、**正常检测**、**严格检测策略分别为** $(N, 47, 7)$、$(N, 79, 11)$、$(N, 129, 17)$，并得到包括全检的四种检测方式的具体使用策略。

对于问题二， 首先分析题目中给出的生产流程当中的各步骤，将其明确为各系统状态，并确定状态之间的转化关系，构造 **ASM 状态机模型**。而后根据状态转移的终止状态和循环性转移特征，利用**递归**思路，将转化关系具体表现为递归方程，并利用**固定点迭代**算法求解递归当中的无限递归问题。最终根据整体的**多阶段单目标决策的递归 ASM 状态机模型**设计相应函数进行具体实现，通过对**批次平均利润**的计算比较求解题目中所给的各情形的**最佳生产决策**。针对题目中的前五种情形，采用 **[不检测零件 1，不检测零件 2，检测成品，拆解不合格成品]** 策略，对第六种情形，采用 **[不检测零件 1，不检测零件 2，不检测成品，不拆解不合格成品]**，可以获得最大利润。

对于问题三， 首先分析本题较问题二的差异：中间工序增多，状态转移复杂化。对于解空间较大、计算复杂度高的情形，可以使用**智能优化算法如遗传算法**，对决策向量进行迭代计算，寻找最优利润策略。在上述普适模型之外根据题给具体情境，可以进行问题的简化和拆解，对子问题仿照问题二中的 **ASM 状态机模型**进行递归求解。具体解得策略为 **[检测零件 1，检测零件 2，不检测零件 3，检测零件 4，检测零件 5，不检测零件 6，检测零件 7，不检测零件 8，检测半成品 1，检测半成品 2，检测半成品 3，不拆解半成品 1，不拆解半成品 2，不拆解半成品 3，检测成品，拆解成品]** 并获得最大利润 $W = 84.093$。

对于问题四， 首先明确给定的各零件、半成品、成品的次品率为问题一所给抽样检测模型样本对于总体的**次品率（统计量）估计值**，根据假设检验问题的显著性水平设置可知，支持原假设的相应统计量落在一个特定区间即**接收域**中。利用接收域对给定值进行修正，从而考察当次品率在某一特定范围内最优决策的选取。采取**概率权重**的方式综合考量各次品率情况的决策效力，并给出**最终决策**。

关键字：概率统计模型　假设检验　ASM 状态机　多阶段单目标决策优化　遗传算法

一、 问题重述

1.1 问题背景

在实际的生产生活当中，企业出于盈利最大化的考量，会密切关注产品生产销售全流程的各项支出情况。在产品制造和售卖过程中，质量成本也是影响企业支出的重要因素，其主要包括检测支出、制造支出、废弃支出和调换支出。从以上几个角度考量，次品率控制是影响企业相关支出的重要因素，因此需要设置合理的抽样检测方案，最大程度上完善产品的检测机制，在生产经营过程中对各项成本进行核算、分析、控制和监督 [1] 并据此完成相关决策，从而助力企业完成品牌质量建设和获取最大经济效益，实现长远发展。

1.2 问题重述

根据题目中给定的电子产品生产和销售过程中的相关变量（包含零配件次品率、各类成本和收益），解决下列问题：

问题 1 对于给定的零配件标称值（10%），利用抽样检测的方法判定该批零件的估计次品率。在检测次数尽可能少的前提下，根据不同的单向信度限定规则完成接受/拒收策略的判断。

问题 2 对于给定的两种零部件和成品的次品率以及相应装配、调换和销售流程，具体根据图 1规则设计相关决策模型，并基于此模型对题中所给表 1 中的具体情形进行决策分析。

问题 3 对于给定的 n 个零配件和 m 道工序中 t 个半成品和最终成品的次品率，同样根据问题二的规则设计相关决策模型，并基于此模型对题中所给表 2 中的具体情形进行决策分析。

问题 4 假设在问题 2 和问题 3 中，零配件、半成品和成品的次品率不再严格给定而是通过抽样检测方法确定，即使用类似于问题 1 中的抽样检测方法来估算各个环节的次品率。基于此假设，重新完成问题 2 和问题 3 中的决策分析。

二、 问题分析

2.1 问题一分析

对于问题一，要求根据给定零件的标称值和企业设定的置信度，给出一个检测次数尽可能少、企业成本尽可能小的抽样检测方案。在实际的生产问题中，抽样检测流程通

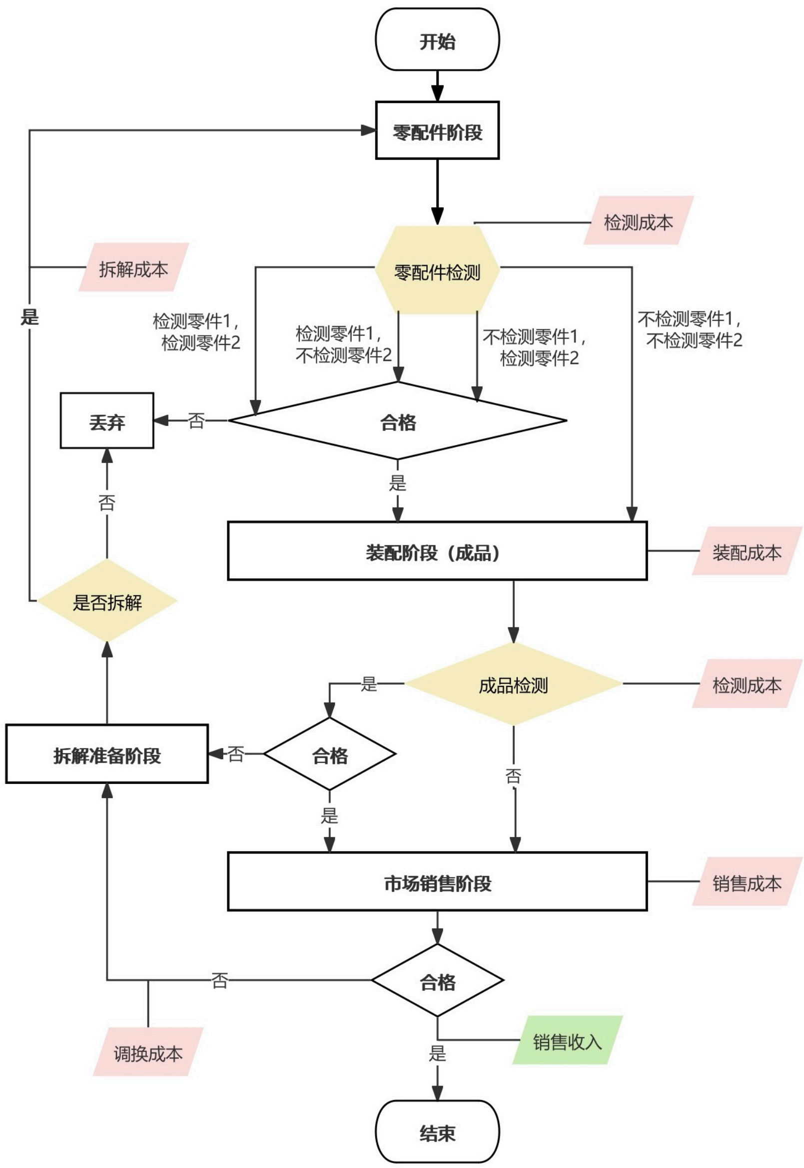


图 1　问题二流程图

常独立于生产、装配等流程，故在抽样方式的选择当中主要进行的是无放回抽样。由此根据批次零件个数和选取的样本个数，可以构建对应的概率统计模型。根据题目中所给的接受和拒绝原则及其显著性水平，建立相关假设并进行假设检验，并根据标称值确定批次产品实际次品率的分档区间，并据此分析其出现第一类风险的概率，求解出对应不同档位的检出次品临界值，从而给出可判定企业接受/拒绝策略的最小抽样检测方案。

2.2 问题二分析

对于问题二，要求根据题目给出的企业生产四阶段：零配件检测装配、成品检测、不合格成品拆解、成品销售，分析各阶段对应的执行情况和对应下一转移状态，从而根据不同零件状态、各类成本等完成企业生产流程决策。通过考察产品在过程中流动方向，

注意到流程中存在可能性的循环，这使得直接给出各类决策的评估函数较为困难，于是本文采取对各步骤进行函数化，并对整体进行递归逻辑建构的方法，完成对应 ASM 状态机的建立，确定各状态转移的逻辑关系，根据状态具体的执行路线计算获得某一特定决策的收益函数（售出获利-各类成本），从而给出最优决策。

2.3 问题三分析

对于问题三，需要给出能够求解任意道工序与任意个零配件的普适性决策模型，并应用于一个具体实例。当生产工序较为复杂时，会导致过程中的概率与成品利润期望难以用普通的概率统计方式求解，因此可对求解流程进行简化和划分成为小规模子问题，利用递归的 ASM 状态机模型对子问题利润函数进行求解，而后自底向上地计算问题整体利润，进而给出生产最优决策。同时较多的工序和零件数量会导致大规模的解空间，从而可以选取遗传算法来搜索解空间，对决策进行 01 编码后以利润函数作为适应度评估函数进行优化过程。

2.4 问题四分析

对于问题四，明确给定的次品率均是通过问题一所给抽样检测模型计算获得的数据，显然该数据同企业标称值拥有相似的特征即不能完全表征样品的实际次品率。对于抽样特征已知存在放宽检测、正常检测和严格检测三类且存在一定显著性水平，由此实际次品率可能在标称值左右进行浮动，由此在问题二、问题三模型的具体求解的过程当中，需要综合考察不同次品率取值的可能性，并根据其在接收域内的概率权重对具体执行的策略进行综合评价分析。

三、 模型假设

为简化问题，本文做出以下假设：

- 假设 1 在抽样检测中，出于实际生产流程的考量，进行的是不放回抽样。
- 假设 2 在生产过程当中各零件各工序的生产、检验均具有独立性，不存在条件关系。
- 假设 3 在产品量支持的前提下，售出次品可以一直调换至正品为止。
- 假设 4 对于同一生产线上的不同零件批，其次品率的变动趋势一致，如均低于/高于标称值。

四、 符号说明

符号	说明	单位
N	整体批次的零件个数	个
n	抽样检测样本中的零件个数	个
k	抽样检测样本中的不合格零件个数	个
X_i	记录单个零件是否为次品的随机变量	无单位
Y_i	记录样品中次品个数的随机变量	无单位
p_i	第 i 类零配件的次品概率	无单位
P_i	第 i 类半成品的次品概率	无单位
P_0	成品装配产生次品的概率	无单位
P	成品的正品概率	无单位
Cc_i	第 i 类零配件的购买单价	元/件
Cd_i	第 i 类零配件的检测成本	元/件
Cm	成品的装配成本	元/件
Cd	成品的检测成本	元/件
Cm_i	第 i 类半成品的装配成本	元/件
Cd_i	第 i 类半成品的检测成本	元/件
S	成品的市场售价	元/件
Lr	调换损失	元/件
Ls	成品拆解费用	元/件
Ls_i	第 i 类半成品拆解费用	元/件
W	产品的出售的净利润	元/件

五、 问题一的模型的建立和求解

5.1 模型建立

对于本题模型的建立，首先需要考察批次零件的总个数 N，对于不同的 N 值确定抽检样本量 n 的大小。

根据题意可以构造随机变量 X_i，X_i 表示在一次抽样试验当中，某个零件是否为次品，各随机变量 X_i 均独立同分布，服从同一个 0-1 分布 $B(1,p)$，即公式1中所示概率分布。

$$P(X=x)=p^x(1-p)^{1-x},\quad x\in\{0,1\} \tag{1}$$

各次伯努利试验的结果互不干扰。显然对于不同抽检样本量 n 和批次总体量 N，其

数量关系会对抽检样本的概率模型建立存在影响，具体的影响表现为:

- **样本占总体较大即 $n \geq 0.1N$ 时**: 抽样的不放回特征不能够被忽略，在整体抽样过程中，抽样的概率会随着抽样过程的变化而发生变化，设置随机变量 Y 表示在全体抽样样本中存在次品零件的个数，存在 $Y \sim h(N, D, n)$ 的概率分布，此时抽样方案的适宜概率统计模型为超几何分布模型。
- **样本占总体较小即 $n < 0.1N$ 时**: 抽样的样本大小 n 远远小于有限总体所含的个数 N，可以将实际问题的不放回抽样近似于有放回抽样进行处理，从而近似地认为所得到的样本为简单随机样本 $(X_1, X_2, \cdots, X_n)$，对于该简单随机样本 [2]，进行进一步抽象表示，设置随机变量 Y 表示在全体抽样样本中存在次品零件的个数，存在 $Y \sim B(n, p)$ 的概率分布规律，从而可以通过构建二项分布的累积模型完成该情况下的求解。

显然对于不同的样本和总体关系 n, N，应该构建不同的模型完成对应的求解。

5.1.1 超几何分布模型

对于超几何分布模型 $Y \sim h(N, M, n)$，其概率质量函数为:

$$P(Y = k) = \frac{\binom{M}{k}\binom{N-M}{n-k}}{\binom{N}{n}}, \quad k \in 0, 1, \cdots, m \tag{2}$$

在公式2中，N 为该批次总体的零件个数；M 为总体中次品的个数；n 为抽取的样本大小；而 k 则是样本中次品临界值。

在抽检问题当中需要计算次品数量小于等于次品数量临界值的概率，这是一个累积值，对于超几何分布可以通过下式3进行计算，求解累加和。

$$P(Y \leq k) = \sum_{i=0}^{k} P(X = i) = \sum_{i=0}^{k} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}} \tag{3}$$

5.1.2 二项分布的累积模型

对于二项分布模型 $Y \sim B(n, p)$，其概率质量函数为:

$$P(Y = k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{4}$$

在公式4中，n 为抽取的样本大小；k 为样本中次品临界值；而 p 则为该批零件总体真实的次品率。

显然 p 和生产厂家给出的标称值具有差异，但考虑到标称值的一定代表性，在本题的求解过程中，我们分别设置 $0.05, 0.08, 0.10, 0.12, 0.15$ 五档次品率对批次总体进行性质描述，并据此继续后续的计算。

同样地基于累积意义进行计算，根据二项分布的累积分布函数公式即式5计算获得次品数量小于等于次品数量临界值 k 的概率。

$$P(Y \le k) = \sum_{i=0}^{k} P(X=i) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i} \tag{5}$$

5.1.3 企业决策的假设检验模型

假设检验作为一种统计方法，主要用于根据样本数据对总体参数作出特定的推断。其基本思想是考察“小概率事件”在试验当中发生的可能性，利用带有反证思路的概率推断方法，完成对于样本数据特征的确定。

假设检验的首要步骤式提出假设，假设具体包含以下两种分类，并且这两类在对同一问题的分析当中同时存在，呈完全对立、二者择一的关系。

- **原假设** H_0：通常是不希望被否定的假设，根据实际问题而提出，在假设检验过程中需要被检验，直到有足够的证据将其拒绝前都被默认为真。
- **备择假设** H_1：与原假设完全对立的假设，表示在本次试验当中样本数据和总体数据的数据特征有着显著的差异，必须从实际数据中得到支持，否则不被认定正确。

在假设建立后，我们应当对支持原假设还是备择假设进行判别，但需要注意到的是样本作为从总体中随机抽取的部分数据，即使数量足够多，也无法对总体分布进行完全的表征，即：基于样本数据构建的对总体的决策会存在一定风险，其中包含两类风险：

- **I 类风险** $P($ **拒绝** $H_0|H_0$ **成立** $)$，即原假设 H_0 原本是正确的，但是根据样本观测值错误地作出了拒绝原假设 H_0 的决定，出现了弃真错误的概率。
- **II 类风险** $P($ **不拒绝** $H_0|H_1$ **成立** $)$ 即原假设 H_0 原本是不正确的，但是根据样本观测值错误地作出了不拒绝原假设 H_0 的决定，出现了取伪错误的概率。

显然上述两类错误同样呈现此消彼长的态势，当样本容量 n 确定后，犯第一类错误的概率减小时，第二类错误出现的概率则相应增大。但在实际生产问题当中，由于原假设通常不希望被轻易否定，故第一类错误的影响更严重，通常进行优先控制。具体方式是设置相应的显著性水平 α，通过显著性水平检验的方式尽可能降低犯第一类错误的概率，从而使得统计结论的正确性得到进一步保证。

根据显著性检验逻辑，可以确定相应统计量的临界值，完成问题拒绝域的构造，对于不同单侧检验方向，有不同拒绝域：

- **左侧检验**：拒绝域位于检验统计量的左侧，即较小值。有 $H_0: \theta \le \theta_0$，其拒绝域 W_1 形如 $W_1 = \{(x_1, x_2, \cdots, x_n) : \hat{\theta} - \theta_0 \ge c_1\}$
- **右侧检验**：拒绝域位于检验统计量的右侧，即较大值。有 $H_0: \theta \ge \theta_0$，其拒绝域 W_1 形如 $W_1 = \{(x_1, x_2, \cdots, x_n) : \hat{\theta} - \theta_0 \le c_2\}$

对于题目中给出的两种情形，分别分析其内含原假设和对应的对立假设，给出拒绝

域的相应形式，并根据显著性水平确定临界值，最终完成拒绝域的构造，求解对应假设检验问题并给出决策：

情形一：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 ：

原假设 H_0: 该批零配件的次品率超过 10%，备择假设 H_1: 该批零配件的次品率不超过 10%，对于上述假设的显著性水平 α 设定为 0.05。

该问题具体为左侧假设检验问题，未知参数为总体次品的概率 p，构造拒绝域形如式6

$$\hat{p} \geq \frac{k}{n} \tag{6}$$

而后根据下式7，带入 $\alpha = 0.05$，p_l 分别为 $10\%, 12\%, 15\%$ 分档进行计算，根据不同的概率统计模型对样本量 n 和样本次品临界值 k 进行求解。

$$P(X \leq k \mid p = p_l) \leq \alpha \tag{7}$$

对于超几何分布模型，综合式7和式3得下列具体计算公式如式8

$$P(Y \leq k \mid p = p_l) = \sum_{i=0}^{k} P(X = i) = \sum_{i=0}^{k} \frac{\binom{N \times p_l}{i}\binom{N - N \times p_l}{n-i}}{\binom{N}{n}} \tag{8}$$

对于二项分布累积模型，综合式7和式5得下列具体计算公式如式9

$$P(Y \leq k \mid p = p_l) = \sum_{i=0}^{k} P(X = i) = \sum_{i=0}^{k} \binom{n}{i} p_l^i (1 - p_l)^{n-i} \tag{9}$$

最终根据计算获得的 $\hat{p}$ 与拒绝域的关系完成假设的选取与认定，并给出对应计算求解得到的 N, n, k 参数值。

情形二：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 ：

原假设 H_0：该批零件的次品率不超过 10%，备择假设 H_1：该批零配件的次品率超过 10%，对于上述假设的显著性水平 α 设定为 0.10。

该问题具体为右侧假设检验问题，未知参数为总体次品的概率 p，构造拒绝域形如式10

$$\hat{p} \leq \frac{k}{n} \tag{10}$$

然后根据下式11，带入 $\alpha = 0.10$，p_r 分别为 $5\%, 8\%, 10\%$ 分档进行计算，根据不同的概率统计模型对样本量 n 和样本次品临界值 k 进行求解。

$$P(X > k \mid p = p_r) \leq \alpha \tag{11}$$

对于超几何分布模型，综合式11和式3得下列具体计算公式如式12

$$P(Y > k \mid p = p_r) = 1 - \sum_{i=0}^{k} P(X = i) = 1 - \sum_{i=0}^{k} \frac{\binom{N \times p_r}{i}\binom{N - N \times p_r}{n-i}}{\binom{N}{n}} \tag{12}$$

对于二项分布累积模型，综合式11和式5得下列具体计算公式如式13

$$P(Y > k \mid p = p_r) = 1 - \sum_{i=0}^{k} P(X = i) = 1 - \sum_{i=0}^{k} \binom{n}{i} p_r^i (1 - p_r)^{n-i} \tag{13}$$

最终根据计算获得的 $\hat{p}$ 与拒绝域的关系完成假设的选取与认定，并给出对应计算求解得到的 N, n, k 参数值。

5.2 模型求解

根据模型的整体构建，对于情形一和情形二，均按照下述步骤进行求解，最终确定最佳的抽样检测方案即 (N, n, k) 三元组，用以表征不同批次数量情况下，选取的抽样检测样本量和其对应的次品检出临界值，用以明确零件接受或拒收的相应抉择。

Step1： 根据构建的假设检验模型，遍历各可能的 (N, n, k) 元组，计算对应原假设的显著性水平 α。

Step2： 根据单侧假设检验的性质，筛选元组。对于情形一，出于支持拒收零配件的目的，尽可能地筛选在同一抽样样本大小下，其计算所得假设的显著性水平尽可能小对应的数据元组；而对于情形二，出于支持接收零配件的目的，尽可能筛选在同一抽样样本大小，其计算所得假设的显著性水平尽可能接近设定值 0.10 对应的数据元组。

Step3： 依据筛选出的各元组值，构建在不同批次实际次品率给定值 p 的情况下以批次总零件数 N 为横坐标，以样本大小 n 为纵坐标的最小抽样图，展示不同总体数情况下样本容量的变化。并给出各样本容量下的次品检出临界值 k 以供查表判断整批零件的接收和拒绝策略 [3]。

5.3 求解结果

5.3.1 情形一：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。

情形一下，信度大于 95%，即显著性水平小于 5% 时总体数与最小检测样本策略的关系如附件【问题 1_ 超几何分布.xlsx】所示，截取部分如表 1。在不同假设次品率下，总体数 N 与最小样本量在超几何分布下的散点图如图 2

表 1　情形一求解结果

CDF，即 P(X<=k)	总体数 N	总体中次品数 D	样本量 n	临界次品量 k
0.045614035	20	3	17	1
0.041353383	21	3	18	1
0.037662338	22	3	19	1
0.034443817	23	3	20	1
0.031620553	24	3	21	1
0.029130435	25	3	22	1
0.026923077	26	3	23	1
0.041880342	27	4	20	1
0.037606838	28	4	21	1
0.03389331	29	4	22	1
0.047509579	30	4	22	1
0.043159066	31	4	23	1

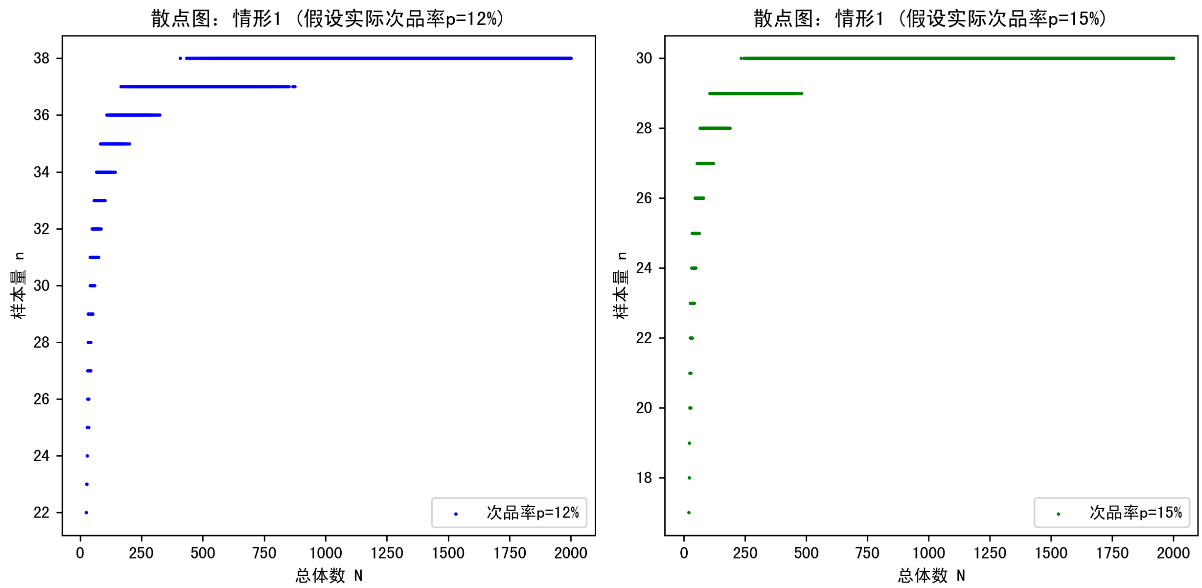


图 2　情形一超几何分布情况下散点图

由散点图可以看出，在超几何分布的情形下，最小检测样本量 n 在总体数 N 较大时，对于两种假设次品率的情况分别趋近于 38 和 30，其对应的临界次品量 k 值均为 1，且均满足附件【问题 1_ 二项分布.xlsx】二项分布情形下的置信度要求，可以验证超几何分布在样本数较大时近似于二项分布。因此 38 是临界次品量选择的最小值。考虑从 38 开始，基于实际生产的策略需要，在显著性水平限制下对不同检测样本量进行临界

次品量区间检测，找到检测成本与误判风险的平衡。

取样本总数足够大时，即 2000 时 n 与 k 的组合情况如表 2。

表 2　检测样本数 n 与满足情形一要求的临界次品数 k 的组合

(a) 假设次品率 p 为 12%

检测样本数 n	临界次品数 k
38	1
50	1,2
80	1,2,3,4
125	1,2,3,4,5,6,7,8

(b) 假设次品率 p 为 15%

检测样本数 n	临界次品数 k
38	1
50	1,2,3
80	1,2,3,4,5,6
125	1,2,3,4,5,6,7,8,9,10,11,12

观察表可以知道，当检测样本数 n 较小时，其对应的 k 的取值与 k 的范围也较小，尽管满足信度要求，但在实际生产中偶然因素导致的误判风险较大，选择临界次品数 k 可以有较大取值的检测样本数 n 可以有效降低这样的风险，但是，较大的检测样本数同时也会产生较大的检测成本。综合考虑实际生产中的检测成本与误判风险，我们定义检测分为全检、正常、严格与放宽四种情形，如下表 3。

表 3　四种检测情况的策略

	检测样本数 n	临界次品数 k
全部检测	批次零件个数 N	-
放宽检测	38	1
正常检测	50	2
严格检测	80	4

并对情形一制定以下策略：

1. **对于检测样本数少于 80 的情况**：
   - **全部检测**：当批次零件数小于 38 时
   - **放宽检验**：零配件数在 38 和 50 之间
   - **正常检验**：零配件数在 50 和 80 之间
2. **对于检测样本数大于 80 的情况**：
   - **开始**：使用正常检验
   - **正常检验—严格检验**：连续抽验的 5 批中有 2 批被拒收
   - **严格检验—正常检验**：连续抽验 5 批都被允收
   - **正常检验—放宽检验**：连续抽验 10 批都被允收

- **放宽检验—正常检验**：只要有一批抽查被拒收

5.3.2 情形二：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。

情形二下，信度大于 90%，即显著性水平小于 10% 时总体数与最小检测样本策略的关系如附件【问题 1_ 超几何分布.xlsx】所示，截取部分如表 4。在不同假设次品率下，

表 4　情形二求解结果

1-CDF，即 P(X>k)	总体数 N	总体中次品数 D	样本量 n	临界次品量 k
0.090909091	21	2	7	1
0.093333333	22	2	8	1
0.095238095	25	2	9	1
0.093596059	30	3	6	1
0.091733871	32	3	15	2
0.093582888	34	3	16	2
0.095187166	35	3	5	1
0.090196078	36	3	18	2
0.096728307	37	4	3	1
0.099682679	40	4	4	1
0.095191073	41	4	6	1

总体数 N 与最小样本量 n 在超几何分布下的散点图如图 3。

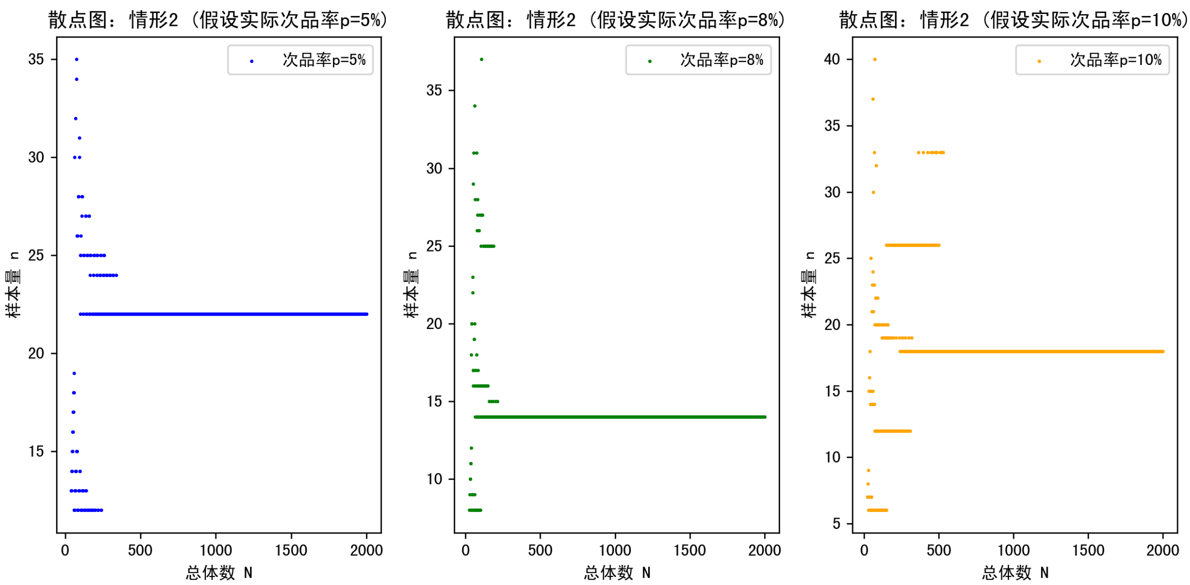


图 3　情形二超几何分布情况下散点图

由散点图可以看出，在超几何分布的情形下，最小检测样本量 n 在总体数 N 较大时，对于三种假设次品率的情况分别趋近于 22、14、18，其对应的临界次品量 k 值分别为 2，2，3，且均满足附件【问题 1_ 二项分布.xlsx】中二项分布情形下的置信度要求, 可以验证超几何分布在样本数较大时近似于二项分布。较大的样本量可以减少弃真错误，综合考虑实际次品率与小总体数的二项分布性不显著的情况，选择散点图中的最高点 40 作为临界次品量选择的起始值。

考虑从大于等于 40 开始，基于实际生产的策略需要，对不同检测样本量进行其满足置信度要求对临界次品量区间检测，找到检测成本与误判风险的平衡。

取样本总数足够大时，即 2000 时 n 与 k 的组合情况（由于更倾向于接收，置信度在 90% 右侧时，越接近 90% 越好）如表 5

表 5　检测样本数 n 与满足情形二要求的临界次品数 k 的组合

(a) 假设次品率 p 为 5%

检测样本数 n	临界次品数 k
49	4
78	6
125	9

(b) 假设次品率 p 为 8%

检测样本数 n	临界次品数 k
49	6
79	9
129	14

(c) 假设次品率 p 为 10%

检测样本数 n	临界次品数 k
47	7
79	11
129	17

观察表可以知道，当检测样本数 n 较小时，其对应的 k 的取值也较小，尽管满足信度要求，但在实际生产中偶然因素导致的误判风险较大，选择临界次品数 k 可以有较大取值的检测样本数 n 可以有效降低这样的风险，但是，较大的检测样本数同时也会产生较大的检测成本。综合考虑实际生产中的检测成本与误判风险，我们定义检测分为全检、正常、严格与放宽四种情形，如下表 6

表 6　四种检测情况的策略

	检测样本数 n	临界次品数 k
全部检测	批次零件个数 N	-
放宽检测	47	7
正常检测	79	11
严格检测	129	17

并对情形二制定以下策略：

1. **对于检测样本数少于 129 的情况**：
   - **全部检测**：当批次零件数小于 47 时
   - **放宽检验**：零配件数在 47 和 79 之间

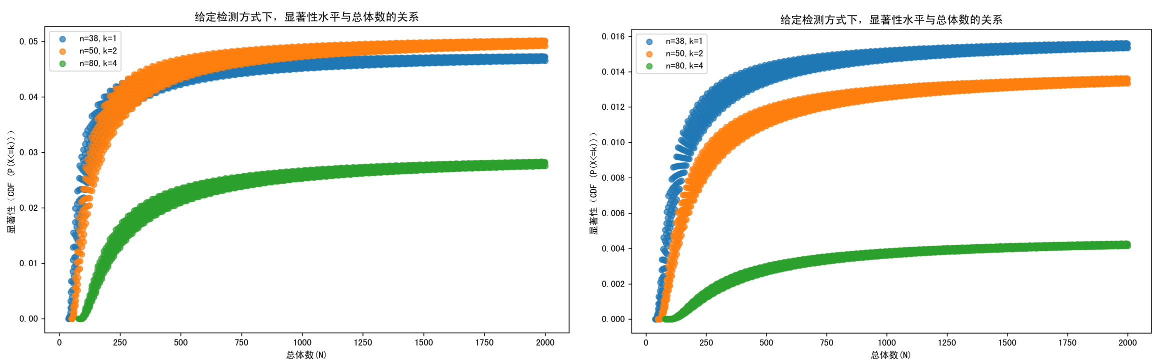
- **正常检验**：零配件数在 79 和 129 之间

2. **对于检测样本数大于 129 的情况**：
   - **开始**：使用正常检验
   - **正常检验—严格检验**：连续抽验的 5 批中有 2 批被拒收
   - **严格检验—正常检验**：连续抽验 5 批都被允收
   - **正常检验—放宽检验**：连续抽验 10 批都被允收
   - **放宽检验—正常检验**：只要有一批抽查被拒收

5.4 模型检验

5.4.1 模型直接检验

对于模型一，将除了全部检测外的三种检测策略：$(n = 38.k = 1)(n = 50, k = 2)(n = 80, k = 3)$ 带入每一种样本数量下的情形进行显著性分析，结果如图 4a图 4b, 满足显著性小于 0.05 即置信度大于 95% 的要求。



(a) 假设 $p = 12\%$　　**(b) 假设** $p = 15\%$

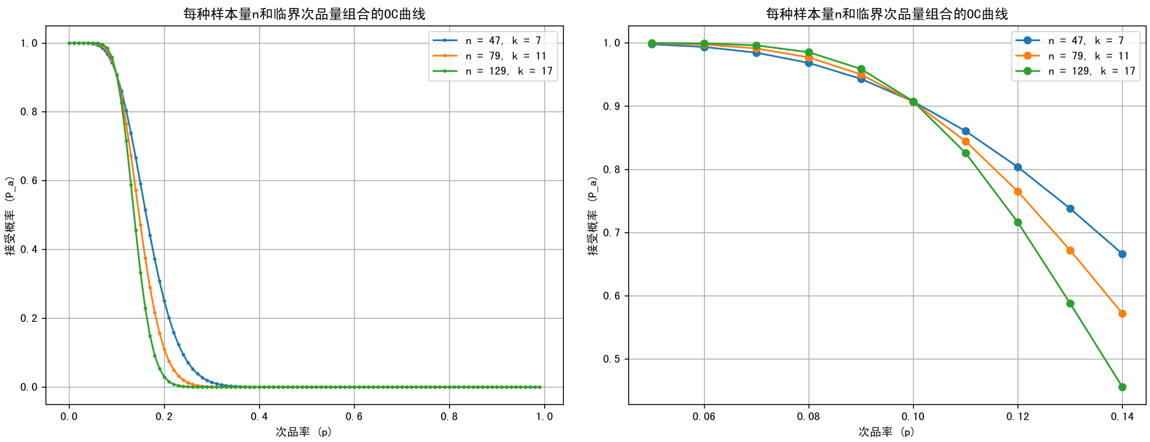
图 4　情形一显著性验证

OC 曲线用来衡量一个批次的产品被接收的概率与该批次产品不合格率之间的关系。而情形二考虑的是，在一定置信度下接受零配件，因此可以使用 OC 曲线来检验情形二下严格、正常和放宽检测对于接受率的影响，绘制 OC 曲线如图 5a图 5b

由 OC 曲线可以看出，在 $p = 0.1$，即次品率等于标称值时，三种检测方式的接受度均在 0.91 左右，满足情形二的要求。当 $p > 0.1$ 时，三种检测方式接受率的关系是 $P_{严格} < P_{正常} < P_{放宽}$，符合对于三种检测方式作用的预期。

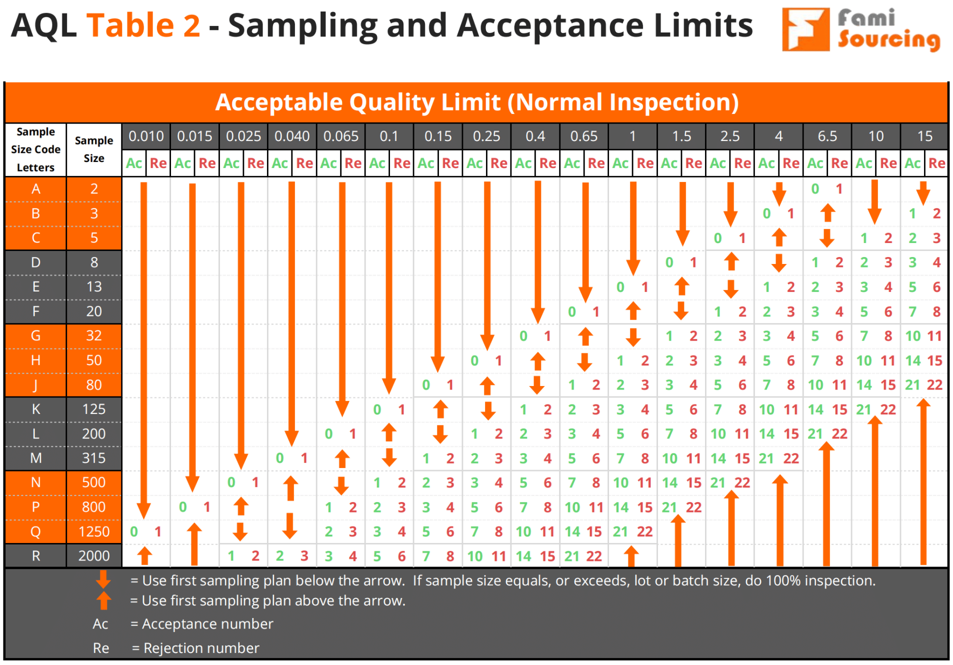
5.4.2 实际生产辅助检验

对于零件抽检的实际生产任务，在 GB/T 2828《计数抽样检验程序》[4] 中给出了按接收质量限（AQL）检索的检验抽样计划和对应查找表如图6所示。



(a) **全局图**（$0 \le p \ge 1$） **(b)** **局部图**（$0.05 \le p \ge 0.14$）

图 5 情形二 OC 曲线



AQL Table 2 - Sampling and Acceptance Limits

Fami Sourcing

Acceptable Quality Limit (Normal Inspection)

Sample Size Code Letters	Sample Size	0.010	0.015	0.025	0.040	0.065	0.1	0.15	0.25	0.4	0.65	1	1.5	2.5	4	6.5	10	15
		Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re	Ac Re
A	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓
B	3	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	1 2
C	5	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	1 2	2 3
D	8	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	1 2	2 3	3 4
E	13	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6
F	20	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8
G	32	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11
H	50	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15
J	80	↓	↓	↓	↓	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22
K	125	↓	↓	↓	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑
L	200	↓	↓	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑	↑
M	315	↓	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑	↑	↑
N	500	↓	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑	↑	↑	↑
P	800	↓	0 1	↑	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑	↑	↑	↑	↑
Q	1250	0 1	↑	↓	↓	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑	↑	↑	↑	↑	↑
R	2000	↑	↑	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	↑	↑	↑	↑	↑	↑	↑

↓ = Use first sampling plan below the arrow. If sample size equals, or exceeds, lot or batch size, do 100% inspection.
↑ = Use first sampling plan above the arrow.
Ac = Acceptance number
Re = Rejection number

图 6 按接收质量限（AQL）的检索表

对于本题，可以类比于设置重缺陷 AQL 为 10 的抽样检验情形，根据查表所得不同的样本大小 n 和对应次品检出临界值 k 的具体值。并与本文模型计算所得结果进行比较，可以看到其在数量上可以实现较好的吻合。

其中少量差异存在的主要原因是：

- **对于情形一**：更侧重于拒收相应批次零件，相当于强化 AQL 的限制，即实际期待 AQL 较给定值 10% 更小，所以对应的检出临界值也小于同 n 值 $AQL = 10\%$ 的 k_{gb}。

- **对于情形二**：更侧重于接收相应批次零件，相当于弱化 AQL 的限制，即实际期待 AQL 较给定值 10% 更大，所以对应的检出临界值也大于同 n 值 $AQL = 10\%$ 的 k_{gb}。

六、问题二的模型的建立和求解

6.1 模型建立

6.1.1 系统流程分析

首先对企业生产决策涉及的四个流程进行分析：

1. **零配件检测阶段**：在这一阶段对各类零件抽样样本的全体进行检测。将各类零配件的次品率理解为理论上直接控制整批零件的次品个数的值，如式1所示可以计算每个零件为次品的可能性。检测出的次品直接丢弃，不进入后续阶段【流程出口 1】。
2. **成品装配与检测阶段**：进入这一阶段的零件既可能未经过上一阶段的检测，也可能已经完成检测，同时由于在装配过程中也会导致新的次品产生，成品总体次品率由上述两个阶段共同决定。合格成品进入市场之后直接完成销售，经由终止条件判定流向【流程出口 2】，而经检测的不合格产品则流向不合格拆解阶段。显然如果不进行检测，则成品全部进入市场，按照成品合格概率，部分流入【流程出口 2】，其余则进入不合格调换阶段。
3. **不合格拆解阶段**：对于已经明确的不合格产品，需要进行拆解判断：如果不进行拆解则直接丢弃【流程出口 1】；而若确定进行拆解，则需要花费拆解成本，同时拆解得到的合格零件进行回收，放回整体零件当中，等待重新进入零配件检测阶段，重复上述流程直至流程出口。
4. **不合格调换阶段**：对于已经出售并明确不合格性质的次品，企业实行无条件调换方案，具体而言是将不合格拼用成品装配与检测阶段中获得的新成品进行对调，并付出过程中的调换成本。对于回收的不合格品，进入不合格拆解阶段，并继续执行流程。需要注意的是，在调换过程中并不能完全保证给出的新品为正品，故可能重复多次调换过程。

根据上述流程分析，可以明确流程具有以下特征：

- **明确的流程出口**：对于上述各流程，确定的流程出口只有两类，第一类出口是对零件进行丢弃，与该出口对应的是各零件成本的损失 $\sum Ci$；第二类出口是正品的成功售出，与该出口对应的收支情况是成品的市场售价 S 和售出零件的成本 $\sum Ci$。
- **流程循环性**：除了上述明确的流程出口之外，系统整体存在着循环过程，主要体现在调换和拆解逻辑部分：对于不合格成品既可以在检测时发现其不合格并选择拆解，也可以在其已经流入市场之后被调换退回并抉择是否需要进行拆解。在决定拆解后均需要从起点（即零配件检测阶段）重新进入流程。并且由于调换的无条件性，可

能存在无穷多次调换，上述循环以及循环的损失是难以直接精确计量的。

6.1.2 递归逻辑构建

根据上述两大特征，对于本题模型的具体构建，本文尝试使用递归思路完成，逐步骤的给出递归函数，并据此完成各状态设计和说明，构建整体 ASM 状态机系统。

- **递归终止条件-明确的流程出口**：在递归中，终止条件决定递归何时停止，尽可能避免无限循环的出现。本题中给出了较为明确的终止条件和其对应的收支，能够较好地计算同批次零件生产过程中不同决策的平均利润。
- **递归调用特征-流程循环性**：递归可以通过调用自身实现相应的循环，这和流程中的拆解和调换逻辑较为相似，此外流程的无穷循环也可以通过递归深度的设置完成。

显然采用递归思维可以有效地模拟上述循环流程，但需要注意的是，即使设置了合理的递归出口，由于调换无穷多次的可能性，系统中依旧可能存在着无限循环，显然这样是不利于计算的，于是本文采用固定点迭代法对调换逻辑进行特殊处理，具体处理如下：

面对如图7所示循环，可以利用固定点迭代法计算获得该循环最终逼近的支出数值，即利用数值逼近的方式，从某一初始值 $assumeW$ 出发，根据递归关系反复迭代，直至结果收敛为止，该值即为调换步骤对于最终值的影响。

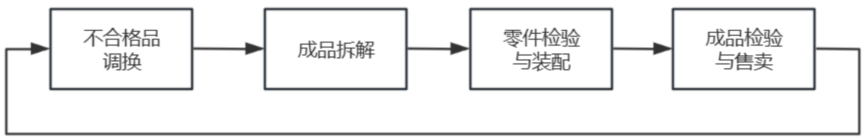


图 7　调换可能导致的循环

根据上述内容建立各步骤递归函数并完成整体递归的设计如下：

- **STEP1: 零配件检测阶段**　由于零配件 1 零配件 2 的检测过程相对独立，故可以按照式14对零配件 1 和零配件 2 的检测过程的相关决策进行模拟，并根据相应概率计算本阶段的支出影响。

$$Step1() = \begin{cases} -Cd_1 - Cd_2 + (1-p_1)\times(1-p_2)\times Step2(), & \text{零件 1 和 零件 2 均被检测} \\ -Cd_1 + (1-p_1)\times Step2(), & \text{仅零件 1 被检测} \\ -Cd_2 + (1-p_2)\times Step2(), & \text{仅零件 2 被检测} \\ Step2(), & \text{零件 1 和 零件 2 均不被检测} \end{cases} \tag{14}$$

- **STEP2: 成品装配与检测阶段** 首先根据式15计算该阶段的成品正品率 P 的具体值，然后对当前零件装配为成品、成品检验和售出的过程中的相关决策进行模拟如式16，并根据相应概率计算本阶段的支出影响。

$$P = \begin{cases} (1-p_1)\times(1-p_2)\times(1-P_0), & \text{零件 1 和 零件 2 均不被检测} \\ (1-p_1)\times(1-P_0), & \text{仅零件 1 不被检测} \\ (1-p_2)\times(1-P_0), & \text{仅零件 2 不被检测} \\ (1-P_0), & \text{零件 1 和 零件 2 均被检测} \end{cases} \tag{15}$$

$$Step2() = \begin{cases} -Cm - Cd + P\times S + (1-P)\times Step3(), & \text{检测成品} \\ Step4(), & \text{不检测成品} \end{cases} \tag{16}$$

- **STEP3: 不合格拆解阶段** 对对当前确定为不合格品的产品作拆解决策模拟如式17，并根据相应概率计算本阶段的支出影响。

$$Step3() = \begin{cases} -Ls + Iteration(), & \text{拆解不合格品} \\ 0, & \text{不拆解不合格品} \end{cases} \tag{17}$$

- **STEP4: 市场销售与调换阶段** 对装配得到的成品进行销售模拟，并对应给出可能的调换结果如式18，根据相应概率计算本阶段的支出影响，在本步骤中采用了 Step2 中计算得到的成品正品率 P 值。

$$Step4() = P\times S + (1-P)\times(-Lr + Step3()) \tag{18}$$

- **Iteration(): 迭代逻辑** 依据固定点迭代方法，首先给定 $assumeW$ 相当于初始的调换步骤收支猜测值，每次通过重复进入函数 $Step1()$ 计算结果 $rea1\_res$，然后取新旧结果的平均值，直到两者差值足够小（达到收敛条件 $abs(real\_res - assumeW) < 18e^{-6}$）逐渐逼近最终的解。

6.1.3 ASM 系统状态机模型的构建

根据上述各阶段定义和递归方程，本文构建如下图所示 ASM 状态机，

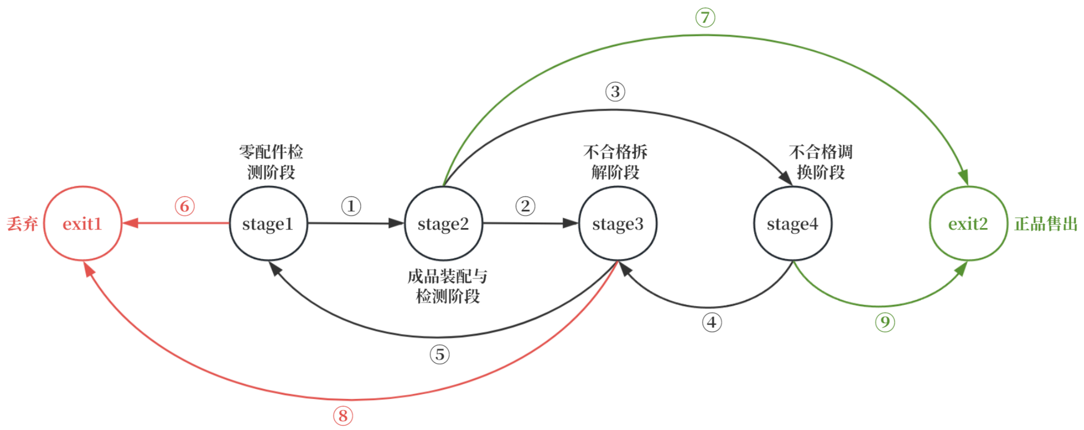


图 8　ASM 系统状态机

首先介绍模型当中涉及的状态，即系统在某一特定时刻的情况，表征此时系统的模式。同节 6.1.1 系统流程分析中介绍的四段流程一致，包含 $stage1$ 零配件检测，这同时也是起始状态，$stage2$ 成品装配与检测，$stage3$ 不合格拆解，$stage4$ 不合格调换四个状态。在四个状态之外还包括两个状态出口（终止状态），分别为 $exit1$ 丢弃和 $exit2$ 正品售出。

上述状态间、状态与出口间的状态转移关系的整体流程控制逻辑①至⑨解释如下：

- ①零件不进行检测或零件进行检测且质量合格
- ②成品进行检测且质量不合格
- ③成品不进行检测
- ④不合格品流入市场售卖给用户
- ⑤成品不合格但进行拆解成为独立的零件
- ⑥零件检测且不合格
- ⑦成品检测且合格
- ⑧成品不合格且不拆解
- ⑨合格产品售出

ASM 系统状态机模型的构建能够清楚地描述所有可能的决策路径和状态转移，同时展示可能存在的循环问题，有助于明确决策路径，优化决策过程。

6.2 模型求解

根据上述过程中建立的递归性质的 ASM 系统状态机模型进行求解：

Step1： 根据递归方程完成各阶段递归函数的设计和确定，完成整体递归流程的确定并给出相应的递归出口。

Step2： 带入题目所给表一中的具体生产情况，对于不同给定的策略，求取其平均利润，并选取其中最大值所对应的决策策略作为该情况下企业生产具体策略。

6.3 求解结果

对问题二中 6 种情形求解得到的策略四元组结果如表 7

表 7　问题二求解结果

情况 \ 策略	是否检测零件 1	是否检测零件 2	是否检测成品	是否拆解不合格成品	利润（元/件）
1	FALSE	FALSE	TRUE	TRUE	19.7956031
2	FALSE	FALSE	TRUE	TRUE	11.65623511
3	FALSE	FALSE	TRUE	TRUE	19.7956031
4	FALSE	FALSE	TRUE	TRUE	13.60935798
5	FALSE	FALSE	TRUE	TRUE	18.93826337
6	FALSE	FALSE	FALSE	FALSE	24.58675

6.4 模型检验

6.4.1 对于决策合理性的检验

分析根据题目条件给出的相应决策，可以较为明显地看到对于第六种情形（零件的次品率较低、成本中等同时拆解费用极高），在此场景下成品的不次品率可以通过公式16进行计算，$P = 95\% \times 95\% \times 95\% \approx 0.857$ 值较大，存在次品调换的可能性较小，同时拆解费用远超直接将零件进行废弃的费用，故最终选择对零件 1、零件 2 和成品均不进行检验，同时直接丢弃不合格产品的策略从而获得高度近似于最大利润 $W_{max} = S - Cc_1 - Cc_2 - Cm = 28$ 的利润 $W = 24.58675$。

6.4.2 对于模型可解释性的检验

为了验证上述模型的可解释性，除了对问题二的 6 种情形进行求解外，还需要对其他一般性情形（不同零件 1 次品率、零件 2 次品率、成品次品率、检测成本、拆解费用、调换费用）遍历求解利润最大时的策略组合, 并依据获得的大量数据进行总体关联性分析。

- **次品率与是否检测的趋势分析**：附表 1、附表 2分别给出了零件 2 和成品次品率与是否检测对应的最优解的数量，可以看出当次品率逐渐增加时，对应最优解不检测情况与检测情况的比值逐渐降低 (如附图 1、附图 2)，即次品率越高，设计出的模型更倾向于检测，筛选出次品，在检测成本一定时减少后续费用，满足对模型的直观理解。

- **拆解费用与是否拆解的趋势分析**：附表 3给出了成品拆解费用与是否拆解的最优解的数量，可以看出当成品拆解费用逐渐增加时，对应最优解下成品不拆解情况与拆解情况的比值逐渐升高 (如附图 3)，即拆解费用越高，设计出的模型更倾向于不拆解，直接丢弃，满足对模型的直观理解。
- **检测成本与是否检测的趋势分析**：附表 4、附表 5分别给出了零件 2 和成品的检测成本与是否检测对应的最优解的数量，可以看出当检测成本逐渐增加时，对应最优解不检测情况与检测情况的比值逐渐升高 (如附图 4、附图 5)，即检测成本越高，设计出的模型更倾向于不检测，降低成本，满足对模型的直观理解。
- **调换损失与成品是否检测的趋势分析**：附表 6给出了成品的调换损失与是否检测对应的最优解的数量，可以看出当调换损失逐渐增加时，对应最优解不检测情况与检测情况的比值逐渐降低 (如附图 6)，即调换成本越高，设计出的模型更倾向于检测，降低成本，满足对模型的直观理解。

七、 问题三的模型的建立和求解

7.1 模型建立

7.1.1 对于多零件多工序的生产决策模型推广

相比较问题二，问题三涉及的变量数量较多（$n+t+4$ 个决策变量，其中 n 是零配件数量，t 是半成品数量)，以及大量可能的组合方案。随着零配件和半成品数量的增加，组合呈指数级增长，导致解空间非常庞大。因此在 t 和 n 的一般情况下，考虑使用遗传算法进行全局搜索，寻找最优生产策略。

对决策变量定义如下：

1. **检测策略**：
   - $x_i \in \{0,1\}$, $i \in [1,n]$：表示是否对零配件 i 进行检测，1 表示检测，0 表示不检测。
   - $y_i \in \{0,1\}$, $i \in [1,t]$：表示是否对半成品 i 进行检测，1 表示检测，0 表示不检测。
   - $z \in \{0,1\}$：表示是否对成品进行检测，1 表示检测，0 表示不检测。
2. **拆解策略**：
   - $d_{j,k} \in \{0,1\}$, $j \in [1,t], k \in [1,j-1]$：表示是否将半成品 j 拆解为更前一级的半成品或零部件，1 表示拆解，0 表示不拆解，其中 j 表示当前半成品，k 表示可能的前一级半成品或零部件。
   - $d \in \{0,1\}$：表示是否对成品进行检测，1 表示检测，0 表示不检测。
3. **调换策略**：
   - $e \in \{0,1\}$ ：表示是否对不合格成品进行调换，1 表示调换，0 表示不调换。

目标是最小化总成本，包括以下几部分：

$$
\begin{aligned}
\text{Minimize } f(x,y,z,d,e) = & \underbrace{\sum_{i=1}^{n} C_{Ci}}_{\text{购买成本}} \\
& + \underbrace{\left(\sum_{i=1}^{n} x_i \cdot C_{di} + \sum_{i=1}^{t} y_i \cdot C_{di} + z \cdot C_d\right)}_{\text{检测成本}} \\
& + \underbrace{\left(\sum_{i=1}^{t} C_{mi} + C_m\right)}_{\text{装配成本}} \\
& + \quad \underbrace{(d_e \cdot L_s)}_{\text{成品拆解为半成品的成本}} \\
& + \quad \underbrace{\left(\sum_{j=1}^{t}\sum_{k=1}^{j-1} d_{j,k} \cdot L_{si}\right)}_{\text{半成品拆解为更早期的半成品或零部件的成本}} \\
& + \underbrace{(e \cdot L_r)}_{\text{调换损失}}
\end{aligned} \tag{19}
$$

约束条件如下：

- 检测决策约束：

$$x_i, y_i, z \in \{0,1\} \tag{20}$$

- 拆解策略约束：

$$d_{j,k}, d_e \in \{0,1\} \tag{21}$$

$$d_{j,k} \le d_e,\ \forall j,k \tag{22}$$

- 调换策略约束：

$$e \in \{0,1\} \tag{23}$$

$$e + d_e \le 1 \tag{24}$$

7.1.2 对于生产实际问题的模型简化

节 7.1.1 中所给出的遗传算法模型作为模拟自然选择和遗传机制的智能优化算法，主要用于解决复杂的优化和搜索问题，其对应的解空间一般具有高维性和非线性性特征。然而对于本题两道工序八个零件的具体决策问题，其解空间的特征较为清晰，并不需要采用遗传算法进行计算。

故本文选择对实际的问题进行简化，仿照问题二的求解思路，将多道工序的转换成多个子问题求解，从而自底向上地完成分析和计算，具体方式如下：

- **对于第一道工序**：分别将（零件 1、零件 2、零件 3 装配得半成品 1）、（零件 4、零件 5、零件 6 装配得半成品 2）、（零件 7、零件 8 装配得半成品 3）视作第一层子问题。
- **对于第二道工序**：将（半成品 1、半成品 2、半成品 3 装配得成品）视作第二层子问题。

上述简化的整体逻辑是，出于解决工程实际问题的目的，仿照工业生产上的多道流水线多道工艺模式，对整体问题进行拆分，在本题中由上述两层子问题的逐层解决导向最终整体的决策，第一层子问题求解完成后，可以获得各半成品的次品率和对应平均成本，从而进行第二层的计算分析，求解最终决策。

在模型建立的过程中，我们可以仿照模型二的流程，通过确定递归逻辑，明确状态机的状态及其转移条件构建适用于三零件的多阶段单目标决策模型。

整体子问题决策模型在模型二适用于二零件的多阶段单目标决策模型上的增补内容为：

1. **递归方程的扩充**：零配件个数的增加导致在 STEP1 零配件检测阶段的概率计算复杂度增加，具体如下式25所示

$$Step1() = \begin{cases} Step2(), & \text{零件 1/2/3 均不检测} \\ -C_{d1} + (1-p_1) \times Step2(), & \text{仅检测零件 1} \\ -C_{d2} + C_1 - p_2) \times Step2(), & \text{仅检测零件 2} \\ -C_{d3} + (1-p_3) \times Step2(), & \text{仅检测零件 3} \\ -C_{d1} - C_{d2} + (1-p_1) \times (1-p_2) \times Step2(), & \text{仅检测零件 1/2} \\ -C_{d1} - C_{d3} + (1-p_1) \times (1-p_3) \times Step2(), & \text{仅检测零件 1/3} \\ -C_{d2} - C_{d3} + (1-p_2) \times (1-p_3) \times Step2(), & \text{仅检测零件 2/3} \\ -C_{d1} - C_{d2} - C_{d3} + (1-p_1) \times (1-p_2) \times (1-p_3) \times Step2(), & \text{零件 1/2/3 全部检测} \end{cases} \tag{25}$$

2. **ASM 状态机存在状态扩充**：三零件决策系统的状态机的各状态如图9所示

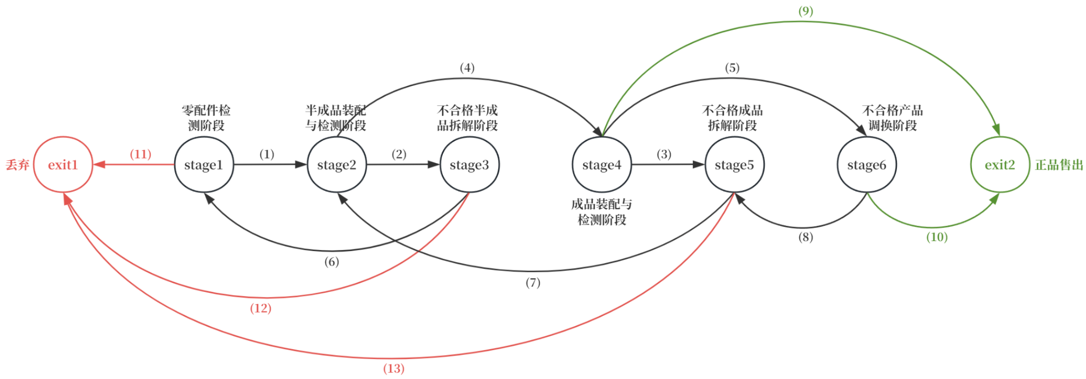


图 9　三零件 ASM 系统状态机

实际角度上，除了明显增加的半成品检测和拆解状态外，对于各状态内部的具体情形也有着相应的增加，如对于具体半成品 1 的检测状态等。

3. **ASM 状态机状态转移条件扩充**：在模型二相关转移条件之外，新增关于半成品阶段的转移条件内容，包括不合格半成品拆解与否的判断等。

7.2 模型求解

7.2.1 遗传算法的模型求解

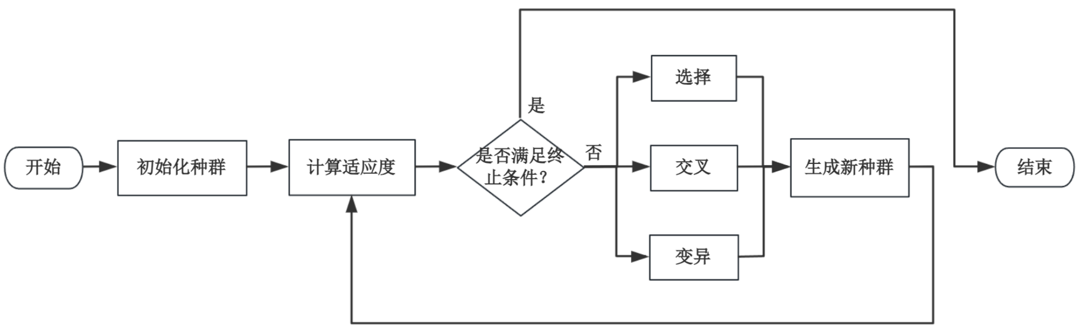


图 10　遗传算法求解流程图

Step1：编码　染色体表示为一个长度为 $n+t(t+1)/2+2$ 的二进制向量：

$$\text{Chromosome} = [x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_t, d_{1,2}, d_{2,3}, \ldots, d_{t-1,t}, z, d_e, e] \tag{26}$$

其中：

- x_i: 零配件检测决策变量。
- y_i: 半成品检测决策变量。
- $d_{j,k}$: 半成品拆解决策变量。
- z: 成品检测决策变量。

- d_e: 成品拆解决策变量。
- e: 调换决策变量。

Step2：种群初始化 随机生成一组染色体作为初始种群，种群大小设定为 N（通常在几十到几百之间），每个染色体长度为 $n + m(m-1)/2 + 2$。

Step3：适应度函数 适应度函数用于评估每个染色体的优劣，公式为：

$$\text{Fitness} = \frac{1}{1 + f(x, y, z, d, e)} \tag{27}$$

其中，$f(x, y, z, d, e)$ 是目标函数，即总成本函数。

Step4：选择、交叉、变异 使用轮盘赌选择——根据适应度的比例随机选择个体，适应度越高的染色体被选择的概率越大，或锦标赛选择——从种群中随机选择两个或多个个体进行竞争，选出适应度最高的个体进行选择操作; 接着使用单点交叉或多点交叉进行交叉操作，产生新的个体；最后在较低的概率下变异，即使用位翻转操作对某些位随机选择翻转（0 变 1 或 1 变 0）。

Step5：终止条件 设定算法的终止条件，包括：

- 达到预设的最大代数（如 100 代或 500 代）。
- 种群的适应度值在若干代内没有显著提高时终止。

7.2.2 流水线嵌套子问题简化模型求解

同 6.2 节一致，根据上建立的递归性质的 ASM 系统状态机模型对子问题进行逐层求解：

Step1： 根据递归方程完成各阶段递归函数的设计和确定，完成整体递归流程的确定并给出相应的递归出口。

Step2： 带入题目所给表中的第一道工序的具体生产情况，对于不同给定的策略，求取其计算得到的半成品的次品率和平均成本，并选取其中最小平均成本所对应的决策策略作为该子问题的决策。

Step3： 对各半成品子问题求解完成后，进行第二道工序的求解，对于不同给定的策略，求取计算得到的最大平均利润，并确定策略。

Step4： 根据 Step2、Step3 进行拼接，完成整体策略的制定。

7.3 求解结果

对问题三中 2 道工序，8 个零部件的一种具体情形求解得到的策略十六元组结果如表 8

表 8　问题三求解结果

零件 1~8 是否检测	半成品 1~3 是否检测	半成品 1~3 是否拆解	成品是否检测	成品是否拆解
T, T, F, T, T, F, T, F	T, T, T	F, F, F	T	T

在该策略下，利润期望最大为 $W = 84.09333173794123$ 元/件。

7.4 模型检验

7.4.1 对于决策合理性的检验

分析根据题目条件给出的相应决策，可以较为明显地看到对于问题三种的情形（零件的次品率较低、成本中等同时拆解费用极高），在此场景下半成品的次品率可以分布通过问题二中公式16进行计算，$P_1 = P_2 = 1 - 90\% \times 90\% \times 90\% \times 90\% \approx 0.3439, P_3 = 1 - 90\% \times 90\% \times 90\% \approx 0.271$ 值较大，同时半成品拆解后，后续组装、检测费用远超直接将零件进行废弃的费用，故最终选择对零件中检测成本为 1 的进行检测，检测成本为 2 的不进行检测，同时直接丢弃不合格半成品。通过分布算得的半成品次品率，进一步计算成品的次品率 $P = 1 - 72.9\% \times 65.61\% \times 65.61\% \approx 0.6862$，值较大，因此成品需要检测。而成品的售价相比较前序成本较大，丢弃成品的损失较大，因此选择拆解不合格成品，以提高利润。最终按照模型策略获得接近于没有次品的最大利润 $W_{max} = 112$ 的利润 $W = 84.09333173794123$。

7.4.2 对于模型可解释性的检验

为了验证上述模型的可解释性，除了对问题中种情形进行求解外，还需要对其他一般性情形（不同零件次品率、半成品次品率、成品次品率、各级检测成本、各级拆解费用、各级调换费用）遍历求解利润最大时的策略组合, 并依据获得的大量数据进行总体关联性分析。

- **次品率与是否检测的趋势分析**：附表 7、附表 8分别给出了半成品 1 和成品次品率与是否检测对应的最优解的数量，可以看出当次品率逐渐增加时，对应最优解不检测情况与检测情况的比值逐渐降低，即次品率越高，设计出的模型更倾向于检测，筛选出次品，在检测成本一定时减少后续费用，满足对模型的直观理解。
- **拆解费用与是否拆解的趋势分析**：附表 9给出了半成品 1 拆解费用与是否拆解的最优解的数量，可以看出当半成品拆解费用逐渐增加时，对应最优解下半成品不拆解

情况与拆解情况的比值逐渐升高，即拆解费用越高，设计出的模型更倾向于不拆解，直接丢弃，满足对模型的直观理解。

- **检测成本与是否检测的趋势分析**：附表 10、附表 11分别给出了半成品 2 和成品的检测成本与是否检测对应的最优解的数量，可以看出当检测成本逐渐增加时，对应最优解不检测情况与检测情况的比值逐渐升高，即检测成本越高，设计出的模型更倾向于不检测，降低成本，满足对模型的直观理解。
- **调换损失与成品是否检测的趋势分析**：附表 12给出了成品的调换损失与是否检测对应的最优解的数量，可以看出当调换损失逐渐增加时，对应最优解不检测情况与检测情况的比值逐渐降低、，即调换成本越高，设计出的模型更倾向于检测，降低成本，满足对模型的直观理解。

八、 问题四的模型的建立和求解

8.1 模型建立

8.1.1 抽样检测模型对零件次品率影响的修正

基于问题一当中的最优抽样检测模型和问题二三中的多阶段单目标决策模型构建本题的综合评价模型，其核心是对于决策模型当中使用的各零件的次品率特征进行修改。

根据问题一的分析可知，在抽样检验过程中，其构造的假设检验概率模型具有显著性水平 α 特征，即该模型给出的是在某一特定信度下接收/拒绝对特定统计量 θ 的结论。故对于给出的某个次品率具体值，其在抽样检测的过程中实际对应着是一个范围更大的结论值域 $[\theta_1, \theta_2]$，由于信度水平的存在该值域内的各值均能够支撑原有假设对特定统计量 θ 的估计即 $\theta = \hat{\theta}$，故称上述值域为原假设的接收域。并且对于接收域当中的不同值，其对应的出现概率存在着一定的差异，越靠近给定估计值 $\hat{\theta}$ 的值其出现的概率越大，相反越接近接收域边界的值其出现概率越小。

在问题一中出现的两种情形分别倾向于拒绝和接收本批零件，均为单侧假设检验问题，而本题中涉及的则是双侧假设检验可能遇到的情况，故可以选定一特定的显著性水平，据此计算获得对应假设的拒绝域，从而反向求解其接收域，并根据接收域的范围进行计算。

根据上述分析，本文尝试将接收域中的各次品率值带入问题二三中构建的决策模型进行计算，分别得出相应最优决策，并对各次品率值的出现概率进行归一化，形成最终决策考量时各决策的考察权重，从而较为整体的获得对于使用基于假设检验模型的抽样检测方案所得概率值的生产决策问题的最优策略。

8.2 模型求解

Step1： 根据问题一，确定本题对各批次抽样检测样本量 n 的大小：综合考察情形一和情形二的检测策略，最终选取样本容量大小为 $n=50$，对应显著性水平 $\alpha=0.05$。

Step2： 对于抽样得到的样本，各零件的次品率 p 相互独立并均服从于式1所示 0-1 分布，表征样本总体次品个数的随机变量 X 在 $n = 50$ 的情况下执行由二项分布 $X \sim B(n,p)$ 近似为正态分布 $X \sim N(np, np(1-p))$ 的近似步骤，计算近似正态的期望和标准差可得 $\mu=5, \sigma^2=4.5$。

Step3： 根据给定的显著性水平 $\alpha=0.05$，其对应双侧假设检验问题的每侧概率为 0.025，查找标准正态分布 Z 的临界值为 $Z_{0.025} \approx 1.96$。将原有随机变量作标准正态化操作得标准正态分布 $\Phi(X)=\frac{x-\mu}{\sigma}$ 接收域为 $[Z_{0.025}, Z_{0.975}]$，据此求解对于原随机变量 X 的接收域 $[x_1, x_2]$。

Step4： 根据求解获得的抽样样本次品个数接收域，计算获得样本次品率值域，对于次品率值域的不同值 p_i，根据随机变量 X 的概率密度函数，求解各值的出现概率 $P(X=np_i)$。

Step5： 对上述各值的概率进行归一化后作为对应样本次品率带入问题二三模型求解所得最优方案在整体方案决策当中的影响权重，求解平均利润期望，选取与该利润值最接近的平均利润对应的生产决策方案作为最终的决策方案。

8.3 求解结果

利用近似正态分布的双侧假设检验模型和给定显著性水平值求解得到的该抽样检测样本实际检出次品率 $\frac{X}{n}$ 的接收域, 具体计算数据在附件【问题 4_ 模型修正.xlsx】中给出。

将接收域内各可能的次品率带入问题三中给出的具体计算模型中, 求解对应情况下的最优决策和其对应的平均利润，根据公式 $Weight_i = P(p=p_i)/\sum_0^i P(p=p_i)$ 计算各次品率的权重，并将该权重与对应次品率的最优决策平均利润相乘，取得整体的利润期望，并根据该利润对应查找最接近的决策方案。

8.3.1 问题二结果修正

问题二中的六种情形给出了多个标称次品率：$5\%, 10\%, 20\%$，其分别对应的实际次品率置信区间为 $[0\%, 11.1\%]$，$[1.69\%, 18.3\%]$ 和 $[8.69\%, 31.31\%]$，各次品率出现的对应概率分别如下表9所示：

对下列各情形的结果修正具体进行求解，得到下列修正后的结果：

表 9　5%, 10%, 20% 对应接收域内次品概率值

(a) 5%

Interval	**Probability**
[0, 0.01]	0.044804
[0.01, 0.02]	0.068012
[0.02, 0.03]	0.093011
[0.03, 0.04]	0.114595
[0.04, 0.05]	0.127199
[0.05, 0.06]	0.127199
[0.06, 0.07]	0.114595
[0.07, 0.08]	0.093011
[0.08, 0.09]	0.068012
[0.09, 0.10]	0.044804
[0.10, 0.11]	0.026591

(b) 10%

Interval		**Probability**
[0.02, 0.03]	[0.17, 0.18]	0.019807
[0.03, 0.04]	[0.16, 0.17]	0.029170
[0.04, 0.05]	[0.15, 0.16]	0.040647
[0.05, 0.06]	[0.14, 0.15]	0.053593
[0.06, 0.07]	[0.13, 0.14]	0.066861
[0.07, 0.08]	[0.12, 0.13]	0.078926
[0.08, 0.09]	[0.11, 0.12]	0.088156
[0.09, 0.10]	[0.10, 0.11]	0.093168

(c) 20%

Interval		**Probability**
[0.09, 0.10]	[0.30, 0.31]	0.012635
[0.10, 0.11]	[0.29, 0.30]	0.017256
[0.11, 0.12]	[0.28, 0.29]	0.022844
[0.12, 0.13]	[0.27, 0.28]	0.029313
[0.13, 0.14]	[0.26, 0.27]	0.036460
[0.14, 0.15]	[0.25, 0.26]	0.043957
[0.15, 0.16]	[0.24, 0.25]	0.051371
[0.16, 0.17]	[0.23, 0.24]	0.058191
[0.17, 0.18]	[0.22, 0.23]	0.063895
[0.18, 0.19]	[0.21, 0.22]	0.068005
[0.19, 0.20]	[0.20, 0.21]	0.070158

表 10　问题二求解结果修正

策略 / 情况	是否检测零件 1	是否检测零件 2	是否检测成品	是否拆解不合格成品	利润
1 $p_1 = p_2 = 0.1$	FALSE	FALSE	TRUE	TRUE	19.7956031
2 $p_1 = p_2 = 0.2$	FALSE	FALSE	TRUE	TRUE	11.65623511
3 $p_1 = p_2 = 0.1$	FALSE	FALSE	TRUE	TRUE	19.7956031
4 $p_1 = p_2 = 0.2$	FALSE	FALSE	TRUE	TRUE	13.60935798
5 $p_1 = 0.11$ $p_2 = 0.21375$	FALSE	FALSE	TRUE	TRUE	18.12613
6 $p_1 = p_2 = 0.05$	FALSE	FALSE	FALSE	FALSE	24.58675

8.3.2 问题三结果修正

问题三中 10% 标称次品率对应的实际次品率置信区间为为 [1.69%, 18.3%]，各次品率出现的对应概率如表11。

表 11 接收域内次品率概率值

Interval		Probability
[0.02, 0.03]	[0.17, 0.18]	0.019807
[0.03, 0.04]	[0.16, 0.17]	0.029170
[0.04, 0.05]	[0.15, 0.16]	0.040647
[0.05, 0.06]	[0.14, 0.15]	0.053593
[0.06, 0.07]	[0.13, 0.14]	0.066861
[0.07, 0.08]	[0.12, 0.13]	0.078926
[0.08, 0.09]	[0.11, 0.12]	0.088156
[0.09, 0.10]	[0.10, 0.11]	0.093168

表 12 问题三求解结果

对应次品率	零件 1-8 是否检测	半成品 1-3 是否检测	半成品 1-3 是否拆解	成品是否检测	成品是否拆解	对应利润
0.13	T, T, F, T, T, F, T, F	T, T, T	F, F, F	T	T	84.80819

8.4 模型检验

本题涉及的模型在节 6.4，节 7.4 中已经完成检验，在此处仅简要地针对模型可解释性中的灵敏度特征进行重述。

分析节 8.3 中给出的结论可以得到，当带入次品率从标称值转变为范围值后，模型三的最终决策变动较模型二中更为明显，这也与模型二、模型三自身的灵敏度相关，模型三中变量更为丰富、可转移的状态数量更多，故对于细微变化的反馈更明显；而模型二中变量较少，状态数量显著降低，对于变化的反应较为迟钝。

但上述两者都能针对于不同的变量特征作出合理的决策，从而指导生产实践的开展。

九、 模型的评价

9.1 模型的优点

- 优点 1：抽样检测考虑放宽检测、正常检测、加严检测三种检测方式，并进行策略组合，满足实际生产中误判风险与检测成本的平衡。
- 优点 2：问题三中模型适用于工序数 m 和零部件数 n 的不同情况，并根据问题的实际规模选择流水线子问题分解或遗传算法，具有较高普适性。

9.2 模型的缺点

- 缺点 1：问题二中模型将不同状态表征为递归，并进行状态转移，在策略选择较多的实际情况下时间复杂度可能较高。
- 缺点 2：模型三中，对问题进行子问题拆解，可能会降低策略选择的整体性。

9.3 模型的推广

本题设计的抽样模型实现了误判风险与检测成本的平衡，可以进一步应用到医疗诊断（如药物测试）、金融（如信用评分模型的风险评估）、环境监控（如污染物抽样监测）等领域，具有较高的可推广性。

假设检验与 ASM 状态机的结合对于企业生产策略的选择也有较高的普适性，可以推广到不同产品类型、不同流水线数量的企业。

参考文献

[1] 周成. 价值链视角下企业战略成本管理研究[J/OL]. 全国流通经济, 2024(11):189-192. DOI: 10.16834/j.cnki.issn1009-5292.2024.11.031.

[2] 同济大学数学科学学院. 概率论与数理统计[M]. 北京: 高等教育出版社, 2023.

[3] 王军虎. 基于假设检验的区间估计必要样本容量确定[J/OL]. 统计与决策, 2023, 39(21):29-33. DOI: 10.13546/j.cnki.tjyjc.2023.21.005.

[4] 中国标准化研究院. 计数抽样检验程序第 1 部分：按接收质量限（AQL）检索的逐批检验抽样计划: GB/T 2828.1-2012[R]. 北京: 中国国家标准化管理委员会, 2012.

附录 A　文件列表

文件名	功能描述
q1.py	问题 1 求解，使用假设检验分别计算超几何分布与二项分布下的样品量与次品临界值
q2.py	问题 2 求解，使用 ASM 状态机方法计算每种情形下的最佳决策
q3.py	问题 3 求解，使用递归子问题方法对不同的 m 和 n 进行拓展
q4_3.py	问题 4 中使用修正模型重写求解问题 3
q4_21.py	问题 4 中使用修正模型重写求解问题 2 中的情形 1
q4_22.py	问题 4 中使用修正模型重写求解问题 2 中的情形 2
q4_23.py	问题 4 中使用修正模型重写求解问题 2 中的情形 3
q4_24.py	问题 4 中使用修正模型重写求解问题 2 中的情形 4
q4_25.py	问题 4 中使用修正模型重写求解问题 2 中的情形 5
q4_26.py	问题 4 中使用修正模型重写求解问题 2 中的情形 6
问题 1_ 超几何分布.xlsx	问题 1 中使用超几何分布概率模型计算情形 1、2 下的样品量与次品临界值
问题 1_ 二项分布.xlsx	问题 1 中使用二项分布概率模型计算情形 1、2 下的样品量与次品临界值
问题 2_ 决策依据.xlsx	问题 2 中对每种情形的每种决策的利润计算结果
问题 4_ 模型修正.xlsx	问题 4 中在对模型修正后重新计算问题 2、3 的结果

附录 B 表格

表 1 零件 2 次品率与是否检测对应的最优解的数量

次品率 \ 是否检测	False	True
0.0	267750	0
0.1	251364	16386
0.2	230802	36948
0.3	210715	57035
0.4	191570	76180
0.5	172862	94888
0.6	191570	113316
0.7	191570	129093
0.8	191570	143499

表 2 成品次品率与是否检测对应的最优解的数量

次品率 \ 是否检测	False	True
0.0	204941	36034
0.1	196420	44555
0.2	186637	54338
0.3	176228	64747
0.4	165397	75578
0.5	152172	88803
0.6	138107	102868
0.7	125814	115161
0.8	115417	125558
0.9	105630	135345

表 3　不合格成品拆解费用与是否拆解对应的最优解的数量

拆解费用 \ 是否拆解	False	True
1	362963	118987
11	442500	39450
21	473417	8533
31	481255	695
41	481950	0

表 4　零件 2 检测成本与是否检测对应的最优解的数量

检测成本 \ 是否检测	False	True
1	47249	94501
2	54281	87469
3	63207	78543
4	71947	69803
5	80237	61513
6	88859	52891
7	97081	44669
8	105115	36635
9	112723	29027
10	116261	25489
11	119917	21833
12	124033	17717
13	127229	14521
14	129747	12003
15	132721	9029
16	134832	6918
17	136966	4784

表 5　成品检测成本与是否检测对应的最优解的数量

检测成本 \ 是否检测	False	True
1	28096	86654
2	33247	81503
3	39106	75644
4	46351	68399
5	52860	61890
6	57401	57349
7	60886	53864
8	65635	49115
9	69535	45215
10	74815	39935
11	78390	36360
12	83392	31358
13	87376	27374
14	92228	22522
15	94145	20605
16	95824	18926
17	98756	15994
18	99655	15095
19	101135	13615
20	103115	11635
21	104815	9935

表 6　调换损失与成品是否检测对应的最优解的数量

调换损失 \ 是否检测	False	True
1	465732	16218
11	422893	59057
21	312757	169193
31	212376	269574
41	153005	328945

表 7　半成品 1 次品率与是否检测对应的最优解的数量

次品率 \ 是否检测	False	True
0.0	1874272	0
0.1	1759548	114702
0.2	1615814	258636
0.3	1475005	399245
0.4	1340990	533120
0.5	1210034	664216
0.6	1341130	793044
0.7	1341130	903651
0.8	1341130	1004493

表 8　成品次品率与是否检测对应的最优解的数量

次品率 \ 是否检测	False	True
0.0	1721520	302664
0.1	1649904	374280
0.2	1567776	456420
0.3	1480326	543864
0.4	1389312	634860
0.5	1278228	745956
0.6	1160088	864108
0.7	1056840	967332
0.8	969516	1054668
0.9	887280	1136916

表 9　半成品 1 拆解费用与是否拆解对应的最优解的数量

拆解费用 \ 是否拆解	False	True
5	2823107	928210
10	3443205	308720
15	3682655	66690
20	3755941	5500
25	3759240	0

表 10　半成品 2 检测成本与是否检测对应的最优解的数量

检测成本 \ 是否检测	False	True
1	368542	737509
2	423402	682307
3	492015	612635
4	561172	544471
5	625850	479809
6	692094	412554
7	756032	348418
8	819897	285753
9	878239	226410
10	906835	198815
11	935351	170297
12	967457	138192
13	991387	113262
14	1012023	93623
15	1035222	70426
16	1051690	53960
17	1068333	37215

表 11　成品检测成本与是否检测对应的最优解的数量

检测成本＼是否检测	False	True
1	219156	675902
2	259329	635727
3	304027	589183
4	361538	533512
5	412308	482742
6	447328	447320
7	474915	419139
8	511953	383097
9	542373	352677
10	583557	311493
11	611442	283608
12	649461	244576
13	681533	213517
14	718378	175672
15	733331	160719
16	747427	147623
17	770293	124753
18	777309	117741
19	789177	106197
20	803497	90735
21	817557	77493

表 12　调换损失与成品是否检测对应的最优解的数量

调换损失＼是否检测	False	True
1	3632710	126950
11	3298581	460647
21	2439485	1319712
31	1657185	2102677
41	1193439	2565771

附录 C　图片

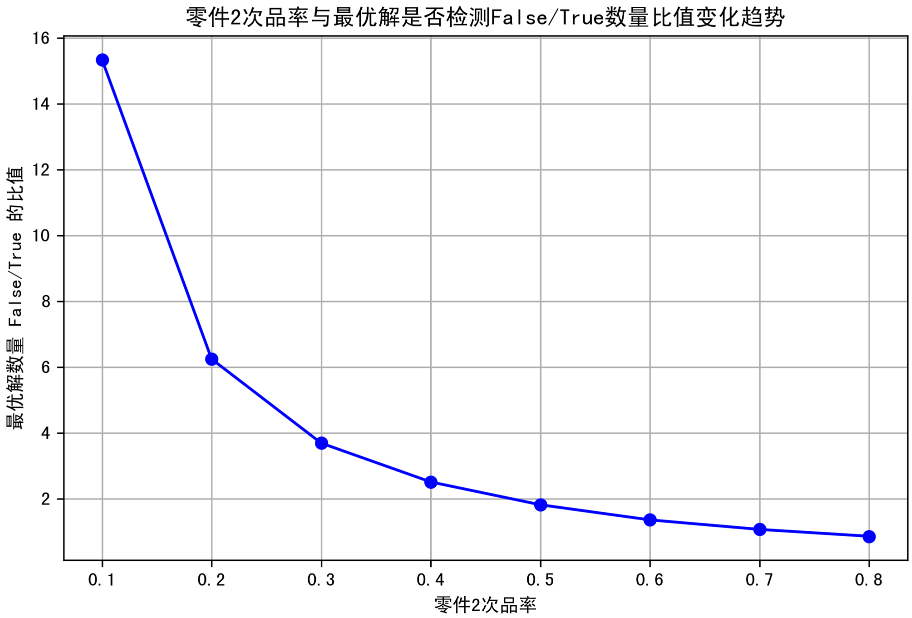


图 1　零件 2 次品率与最优解是否检测 False/True 数量比值变化趋势

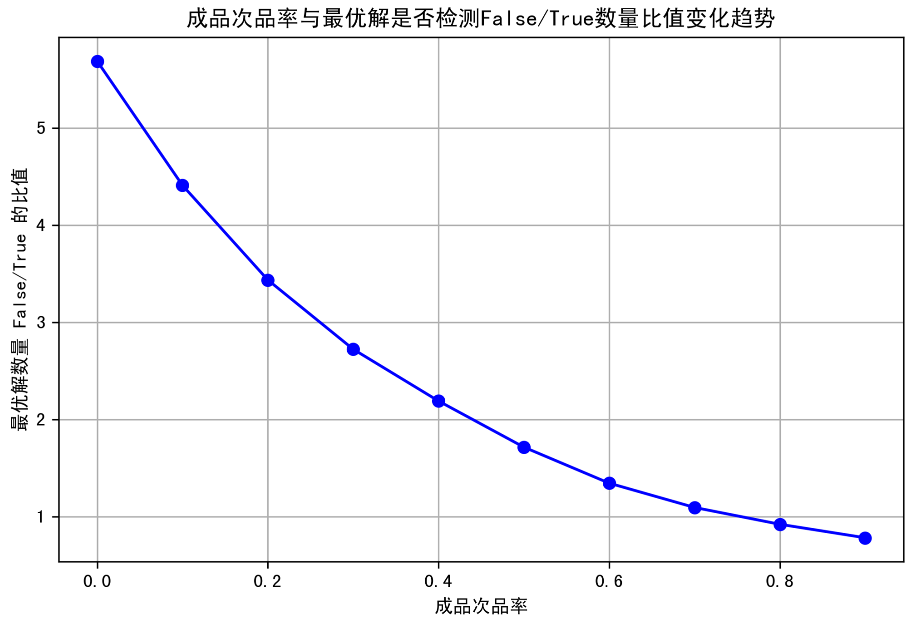


图 2　成品次品率与最优解是否检测 False/True 数量比值变化趋势

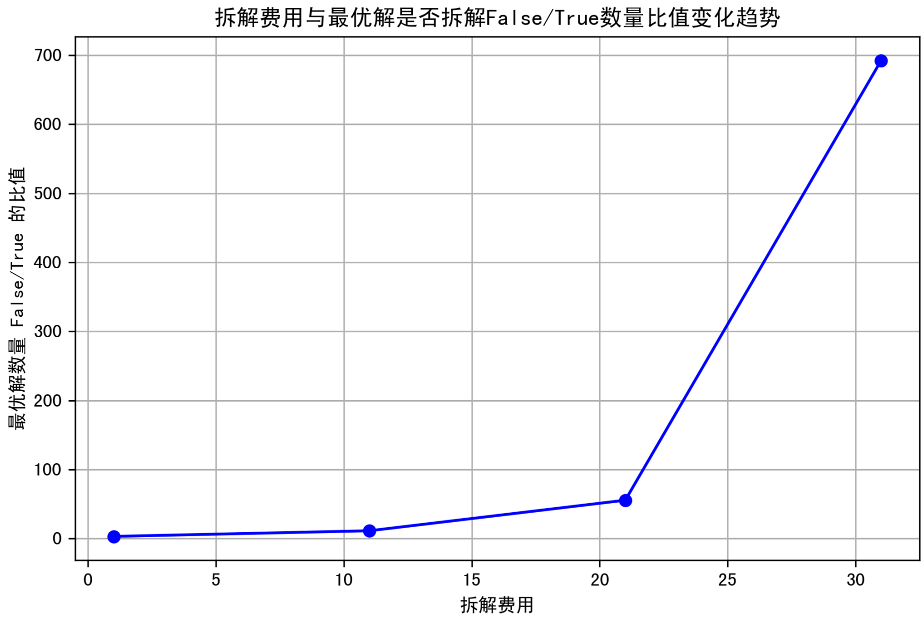


图 3　拆解费用与最优解是否拆解 False/True 数量比值变化趋势

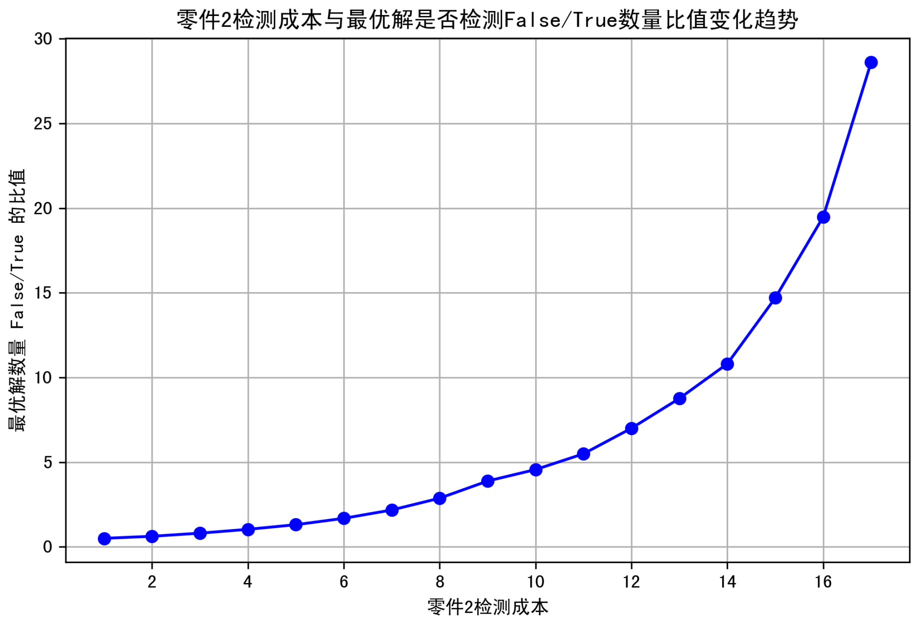


图 4　零件 2 检测成本与最优解是否检测 False/True 数量比值变化趋势

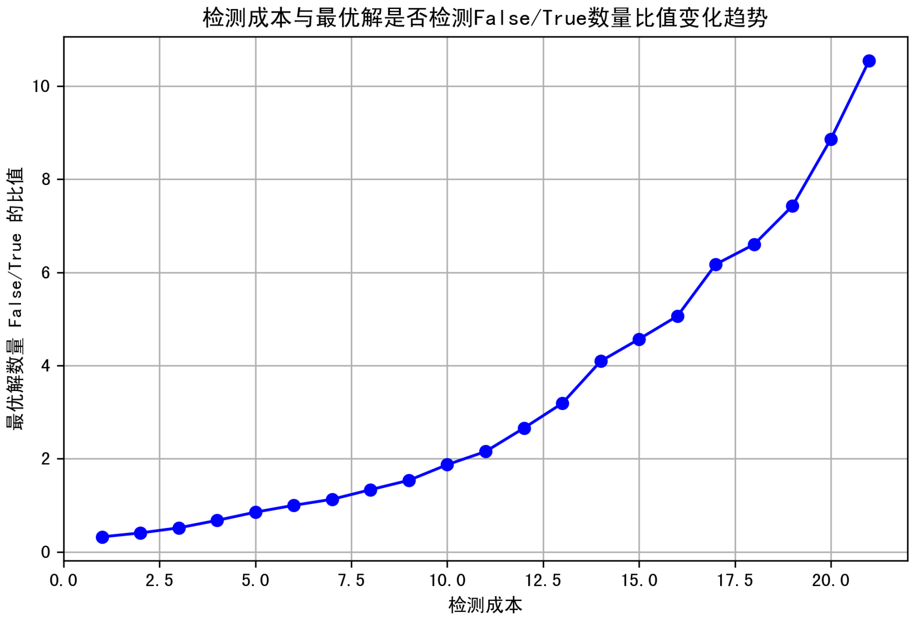


图 5　检测成本与最优解是否检测 False/True 数量比值变化趋势

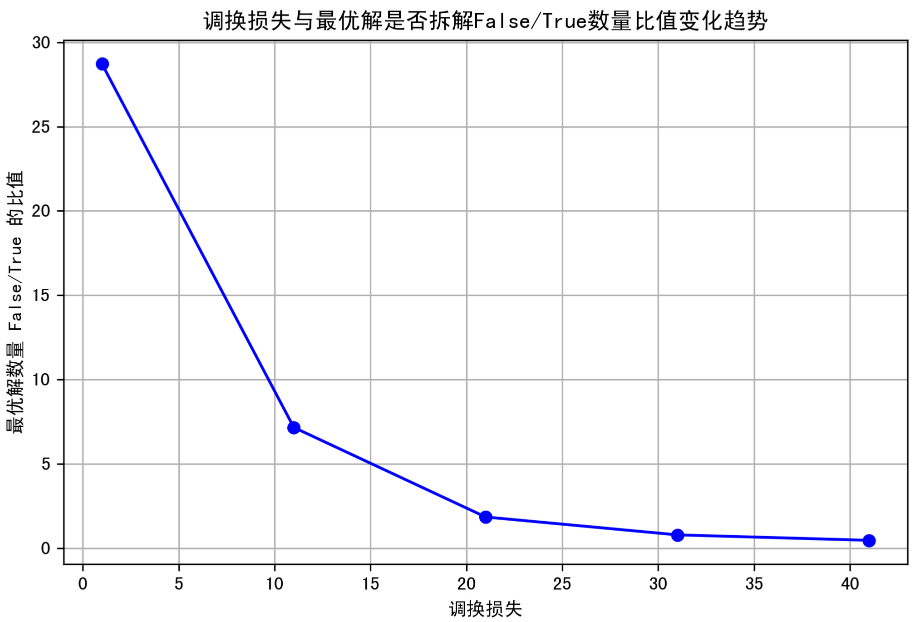


图 6　调换损失与最优解是否拆解 False/True 数量比值变化趋势

附录 D　代码

q1.py

```
import scipy.stats as stats
```

```python
import openpyxl

# 二项分布模型
def test_B():
    # 创建一个新的工作簿
    workbook = openpyxl.Workbook()

    # 创建第一个工作表
    sheet1 = workbook.active
    sheet1.title = "情形1,假设实际次品率p=12%"
    sheet1.append(['CDF， 即P(X<=k)', '样本量n', '临界次品量k'
])

    p = 0.12
    for n in range(1,150):    # 枚举样本量
        for k in range(n):  # 枚举临界值
            # 计算二项分布的累积分布函数 P(X <= k)
            cdf_value = stats.binom.cdf(k, n, p)
            if(cdf_value<=0.05):# and cdf_value>=0.04):    #
希望更接近0.05
                #print(cdf_value,n,k)
                sheet1.append([cdf_value,n,k])

    # 创建第一个工作表
    sheet1 = workbook.create_sheet(title="情形1,假设实际次品率
p=15%")
    sheet1.append(['CDF， 即P(X<=k)', '样本量n', '临界次品量k'
])

    p = 0.15
    for n in range(1,150):    # 枚举样本量
        for k in range(n):  # 枚举临界值
            # 计算二项分布的累积分布函数 P(X <= k)
            cdf_value = stats.binom.cdf(k, n, p)
            if(cdf_value<=0.05):# and cdf_value>=0.04):    #
```

```
希望更接近0.05
            #print(cdf_value,n,k)
            sheet1.append([cdf_value,n,k])

# 创建第二个工作表
sheet2 = workbook.create_sheet(title="情形2,假设实际次品率p=5%")
sheet2.append(['1-CDF, 即P(X>k)', '样本量n', '临界次品量k'])

p = 0.05
for n in range(1, 150):  # 枚举样本量
    for k in range(n):  # 枚举临界值
        # 计算二项分布的累积分布函数 P(X <= k)
        cdf_value = stats.binom.cdf(k, n, p)
        if (1 - cdf_value <= 0.1):#and 1 - cdf_value >=0.09):    # 希望更接近0.10
            #print(1-cdf_value, n, k)
            sheet2.append([1-cdf_value, n, k])

# 创建第二个工作表
sheet2 = workbook.create_sheet(title="情形2,假设p=8%")
sheet2.append(['1-CDF, 即P(X>k)', '样本量n', '临界次品量k'])

p = 0.08
for n in range(1, 150):  # 枚举样本量
    for k in range(n):  # 枚举临界值
        # 计算二项分布的累积分布函数 P(X <= k)
        cdf_value = stats.binom.cdf(k, n, p)
        if (1 - cdf_value <= 0.1):# and 1 - cdf_value >=0.09):    # 希望更接近0.10
            #print(1-cdf_value, n, k)
            sheet2.append([1-cdf_value, n, k])

```

```python
    # 创建第二个工作表
    sheet2 = workbook.create_sheet(title="情形2,假设p=10%")
    sheet2.append(['1-CDF， 即P(X>k)', '样本量n', '临界次品量k'])

    p = 0.10
    for n in range(1, 150):  # 枚举样本量
        for k in range(n):  # 枚举临界值
            # 计算二项分布的累积分布函数 P(X <= k)
            cdf_value = stats.binom.cdf(k, n, p)
            if (1 - cdf_value <= 0.1):# and 1 - cdf_value >=0.09):    # 希望更接近0.10
                #print(1-cdf_value, n, k)
                sheet2.append([1-cdf_value, n, k])

    # 保存文件
    workbook.save('问题1——二项分布.xlsx')

def test_H():
    # 创建一个新的工作簿
    workbook = openpyxl.Workbook()

    # 创建第一个工作表
    sheet1 = workbook.active
    sheet1.title = "情形1,假设实际次品率p=12%"
    sheet1.append(['CDF， 即P(X<=k)', '总体数M','总体中次品数D', '样本量n', '临界次品量k'])

    p = 0.12  # 次品率
    for M in range(1, 2001):  # 总体中元素的总数（比如零件总数）
        for n in range(1,M):  # 抽取的样本数
            D = int(M*p)
            flag = False
            for k in range(1,min(n,D)):  # 计算 P(X <= k)
```

```
            # 计算超几何分布的累积分布函数 P(X <= k)
            cdf_value = stats.hypergeom.cdf(k, M, D, n)
            if(cdf_value<=0.05):# and cdf_value>=0.04):
                #print(cdf_value,M,D,n,k)
                sheet1.append([cdf_value,M,D,n,k])
                flag = True
                break
        if flag:
            break

sheet1 = workbook.create_sheet(title="情形1,假设实际次品率p=15%")
sheet1.append(['CDF， 即P(X<=k)', '总体数M','总体中次品数D', '样本量n', '临界次品量k'])

p = 0.15  # 次品率
for M in range(1, 2001):  # 总体中元素的总数（比如零件总数）
    for n in range(1,M):  # 抽取的样本数
        D = int(M*p)
        flag = False
        for k in range(1,min(n,D)):  # 计算 P(X <= k)
            # 计算超几何分布的累积分布函数 P(X <= k)
            cdf_value = stats.hypergeom.cdf(k, M, D, n)
            if(cdf_value<=0.05):# and cdf_value>=0.04):
                #print(cdf_value,M,D,n,k)
                sheet1.append([cdf_value,M,D,n,k])
                flag = True
                break
        if flag:
            break

sheet1 = workbook.create_sheet(title="情形2,假设实际次品率p=5%")
sheet1.append(['CDF， 即P(X<=k)', '总体数M', '总体中次品数D
```

```
', '样本量n', '临界次品量k'])

p = 0.05  # 次品率
for M in range(1, 2001):  # 总体中元素的总数（比如零件总数）
    for n in range(1,M):  # 抽取的样本数
        D = int(M*p)
        flag = False
        for k in range(1,min(n,D)):  # 计算 P(X <= k)
            # 计算超几何分布的累积分布函数 P(X <= k)
            cdf_value = stats.hypergeom.cdf(k, M, D, n)
            if(1-cdf_value <= 0.10):# and cdf_value>=0.04):
                #print(cdf_value,M,D,n,k)
                sheet1.append([1-cdf_value,M,D,n,k])
                flag = True
                break
        if flag:
            break

sheet1 = workbook.create_sheet(title="情形2,假设实际次品率p=8%")
sheet1.append(['CDF，即P(X<=k)', '总体数M', '总体中次品数D', '样本量n', '临界次品量k'])

p = 0.08  # 次品率
for M in range(1, 2001):  # 总体中元素的总数（比如零件总数）
    for n in range(1,M):  # 抽取的样本数
        D = int(M*p)
        flag = False
        for k in range(1,min(n,D)):  # 计算 P(X <= k)
            # 计算超几何分布的累积分布函数 P(X <= k)
            cdf_value = stats.hypergeom.cdf(k, M, D, n)
            if(1-cdf_value <= 0.10):# and cdf_value>=0.04)
```

```
:
                    #print(cdf_value,M,D,n,k)
                    sheet1.append([1-cdf_value,M,D,n,k])
                    flag = True
                    break
            if flag:
                break

    sheet1 = workbook.create_sheet(title="情形2,假设实际次品率p=10%")
    sheet1.append(['CDF，即P(X<=k)', '总体数M', '总体中次品数D', '样本量n', '临界次品量k'])

    p = 0.1  # 次品率
    for M in range(1, 2001):  # 总体中元素的总数（比如零件总数）
        for n in range(1,M):  # 抽取的样本数
            D = int(M*p)
            flag = False
            for k in range(1,min(n,D)):  # 计算 P(X <= k)
                # 计算超几何分布的累积分布函数 P(X <= k)
                cdf_value = stats.hypergeom.cdf(k, M, D, n)
                if(1-cdf_value <= 0.10):# and cdf_value>=0.04):
                    #print(cdf_value,M,D,n,k)
                    sheet1.append([1-cdf_value,M,D,n,k])
                    flag = True
                    break
            if flag:
                break

    # 保存文件
    workbook.save('问题1——超几何分布.xlsx')

test_B()
```

```
test_H()
```

q2.py

```python
import openpyxl

# 情况1
a1 = 0.1
a2 = 4
a3 = 2
b1 = 0.1
b2 = 18
b3 = 3
c1 = 0.1
c2 = 6
c3 = 3
c4 = 56
d1 = 6
d2 = 5

# 零件1、2是否检测
def step1():
    global x1
    global x2
    if(x1==1 and x2==1):
        return -a3-b3+(1-a1)*(1-b1)*step3()  # 检测成本，同时
    有(1-a1)的概率能够进入后续
    elif(x1==1 and x2==0):
        return -a3 + (1 - a1) * step3()
    elif(x1==0 and x2==1):
        return -b3+(1-b1)*step3()
    else:
        return step3()  # 直接进入下一步

# 成品的装配和是否检测
def step3():
```

```python
    global x1
    global x2
    global x3
    if(x3==1):
        p = ((1-a1) if x1==0 else 1)*((1-b1) if x2==0 else 1)*(1-c1) # 正品率
        return -c2-c3+p*c4+(1-p)*step4()
    else:
        return step5()

# 是否拆解不合格品
def step4():
    global assumeW
    if(x4==1):
        return -d2+assumeW  # 拆解出两个零件，相当于又是一件成品的利润
    else:
        return 0

# 进入市场（当未检测成品时才能到达这一步）
def step5():
    global x1
    global x2
    p = ((1 - a1) if x1==0 else 1) * ((1 - b1) if x2==0 else 1) * (1 - c1)
    return p*c4+(1-p)*(-d1+step4())  # 同时损失了一件

# 对不同情形枚举各种决策
for x1 in range(2):
    for x2 in range(2):
        for x3 in range(2):
            for x4 in range(x3+1):  # 如果不检测，也拆解不了
                real_res = 40
                assumeW = 20
                while abs(real_res-assumeW)>=10e-6:
```

```
                    real_res = step1()
                    assumeW = (real_res+assumeW)/2
                print(bool(x1),bool(x2),bool(x3),bool(x4), real_res - a2 - b2)  # 真正的利润要减去两个零件的成本
```

q3.py

```
import openpyxl

# a: 零配件 b: 半成品 c: 成品
a11 = 0.1
a12 = 2
a13 = 1
a21 = 0.1
a22 = 8
a23 = 1
a31 = 0.1
a32 = 12
a33 = 2
a41 = 0.1
a42 = 2
a43 = 1
a51 = 0.1
a52 = 8
a53 = 1
a61 = 0.1
a62 = 12
a63 = 2
a71 = 0.1
a72 = 8
a73 = 1
a81 = 0.1
a82 = 12
a83 = 2
b11 = 0.1
b12 = 8
```

```python
b13 = 4
b14 = 6
b21 = 0.1
b22 = 8
b23 = 4
b24 = 6
b31 = 0.1
b32 = 8
b33 = 4
b34 = 6
c1 = 0.1
c2 = 8
c3 = 6
c4 = 10
d1 = 200
d2 = 40

x = [0,0,0,0,0,0,0,0,  0,0,0, 0,0,0, 0,0]   # 分别是零件1~8是否检测、半成品1~3是否检测、半成品1~3是否拆解、成品是否检测、成品是否拆解

### 半成品1
# 零件1是否检测
def step11():
    return -a13*x[0]-a23*x[1]-a33*x[2]+((1-a11) if x[0]==1 else 1)*((1-a21) if x[1]==1 else 1)*((1-a31) if x[2]==1 else 1)*step12()

# 半成品是否检测
def step12():
    if x[8]==1: # 如果检测，有装配成本、检测费用，还需要进入是否拆解的判断
        p=((1-a11) if x[0]==0 else 1)*((1-a21) if x[1]==0 else 1)*((1-a31) if x[2]==0 else 1)*(1-b11)
        return -b12-b13+(1-p)*step13()
```

```python
    else: # 不检测，只有装配成本
        return -b12

# 半成品是否拆解
def step13():
    if x[11]==1:
        return -b14+assumeW1
    else:
        return 0

### 半成品2
# 零件1是否检测
def step21():
    return -a43*x[3]-a53*x[4]-a63*x[5]+((1-a41) if x[3]==1 else 1)*((1-a51) if x[4]==1 else 1)*((1-a61) if x[5]==1 else 1)*step22()

# 半成品是否检测
def step22():
    if x[9]==1: # 如果检测，有装配成本、检测费用，还需要进入是否拆解的判断
        p=((1-a41) if x[3]==0 else 1)*((1-a51) if x[4]==0 else 1)*((1-a61) if x[5]==0 else 1)*(1-b21)
        return -b22-b23+(1-p)*step23()
    else: # 不检测，只有装配成本
        return -b22

# 半成品是否拆解
def step23():
    if x[12]==1:
        return -b24+assumeW2
    else:
        return 0

### 半成品3
```

```
def step31():
    return -a73*x[6]-a83*x[7]+((1-a71) if x[6]==1 else 1)*((1-a81) if x[7]==1 else 1)*step32()

# 半成品是否检测
def step32():
    if x[10]==1: # 如果检测，有装配成本、检测费用，还需要进入是否拆解的判断
        p=((1-a71) if x[6]==0 else 1)*((1-a81) if x[7]==0 else 1)*(1-b31)
        #print("p",p)
        return -b32-b33+(1-p)*step33()
    else: # 不检测，只有装配成本
        return -b32

# 半成品是否拆解
def step33():
    if x[13]==1:
        return -b34+assumeW3
    else:
        return 0

#### 成品
# 半成品是否检测
def step1():
    return ((1-half[0][1]) if x[8]==1 else 1)*((1-half[1][1]) if x[9]==1 else 1)*((1-half[2][1]) if x[10]==1 else 1)*step2()

# 成品是否检测
def step2():
    if x[14]==1: # 如果检测，有装配成本、检测费用，还需要进入是否拆解的判断
        p=((1-half[0][1]) if x[8]==0 else 1)*((1-half[1][1]) if x[9]==0 else 1)*((1-half[2][1]) if x[10]==0 else 1)*(1-c1
```

```
    )
            return -c2-c3+p*d1+(1-p)*step3()
    else: # 不检测，只有装配成本
            return -c2 + step4()

# 成品是否拆解
def step3():
    if x[15]==1:
        return -c4+assumeW
    else:
        return 0

# 市场销售阶段
def step4():
    p=((1-half[0][1]) if x[8]==0 else 1)*((1-half[1][1]) if x
   [9]==0 else 1)*((1-half[2][1]) if x[10]==0 else 1)*(1-c1)
    return p * d1 + (1 - p) * (-d2 + step3())

bestw = -52364
bestx = []
i = 0
for x1 in range(2):
    x[0] = x1
    for x2 in range(2):
        x[1] = x2
        for x3 in range(2):
            x[2] = x3
            for x4 in range(2):
                x[3] = x4
                for x5 in range(2):
                    x[4] = x5
                    for x6 in range(2):
                        x[5] = x6
                        for x7 in range(2):
                            x[6] = x7
```

```python
                        for x8 in range(2):
                            x[7] = x8
                            for x9 in range(2):
                                x[8] = x9
                                for x10 in range(2):
                                    x[9] = x10
                                    for x11 in range(2):
                                        x[10] = x11
                                        for x12 in range(x9+1):
                                            x[11] = x12
                                            for x13 in range(x10+1):
                                                x[12] = x13
                                                for x14 in range(x11 + 1):
                                                    x[13] = x14
                                                    for x15 in range(2):
                                                        x[14] = x15
                                                        for x16 in range(x15+1):
                                                            x[15] = x16

        half = []  # 存放成本1~3的成本、次品率

        # 半成品1
        real_res = 40
```

```
    assumeW1 = -1100
    while abs(real_res - assumeW1) >= 10e-6:
        real_res = step11()
        assumeW1 = (real_res + assumeW1) / 2
    real_res -= (a12 + a22 + a32)

    p = (1 - a11) * (1 - a21) * (1 - a31) * (1 - b11)
    if x[8] == 1:
        p1 = p
    else:
        p1 = ((1 - a11) if x[0] == 1 else 1) * (
        (1 - a21) if x[1] == 1 else 1) * (
             (1 - a31) if x[2] == 1 else 1)
    half.append([-real_res,1-p/p1])

    real_res = 40
    assumeW2 = 20
    while abs(real_res - assumeW2) >= 10e-6:
```

```
        real_res = step21()
        assumeW2 = (real_res + assumeW2) / 2
    real_res -= (a42 + a52 + a62)

    p = (1 - a41) * (1 - a51) * (1 - a61) * (1 - b21)
    if x[9] == 1:
        p1 = p
    else:
        p1 = ((1 - a41) if x[3] == 1 else 1) * (
        (1 - a51) if x[4] == 1 else 1) * (
            (1 - a61) if x[5] == 1 else 1)
    half.append([-real_res, 1-p / p1])

    real_res = 40
    assumeW3 = 20
    while abs(real_res - assumeW3) >= 10e-6:
        real_res = step31()
        assumeW3 = (real_res + assumeW3) / 2
    real_res -= (a72 + a82)
```

```
    p = (1 - a71) * (1 - a81) * (1 - b31)
    if x[10] == 1:
        p1 = p
    else:
        p1 = ((1 - a71) if x[6] == 1 else 1) * (
        (1 - a81) if x[7] == 1 else 1)
    half.append([-real_res, 1-p / p1])

    real_res = 40
    assumeW = 20
    while abs(real_res - assumeW) >= 10e-6:
        real_res = step1()
        assumeW = (real_res + assumeW) / 2

    if bestw<(real_res-half[0][0]-half[1][0]-half[2][0]):
        bestw = real_res - half[0][0] - half[1][0] - \
        half[2][0]
        bestx = [element for element in x]
```

```
print(bestx,bestw)
```

q4_21.py

```
# 情况1   0.02~0.18
a1 = 0.02
a2 = 4
a3 = 2
b1 = 0.02
b2 = 18
b3 = 3
c1 = 0.02
c2 = 6
c3 = 3
c4 = 56
d1 = 6
d2 = 5

# # 可以先假定成本的值，然后逐渐逼近的方法求

# 零件1是否检测
def step1():
    global x1
    global x2
    if(x1==1 and x2==1):
        return -a3-b3+(1-a1)*(1-b1)*step3()  # 检测成本，同时有(1-a1)的概率能够进入后续
    elif(x1==1 and x2==0):
        return -a3 + (1 - a1) * step3()
    elif(x1==0 and x2==1):
        return -b3+(1-b1)*step3()
    else:
        return step3()  # 直接进入下一步

# 成品的装配和是否检测
def step3():
```

```python
    global x1
    global x2
    global x3
    if(x3==1):
        p = ((1-a1) if x1==0 else 1)*((1-b1) if x2==0 else 1)
  *(1-c1) # 正品率
        return -c2-c3+p*c4+(1-p)*step4()
    else:
        return step5()

# 是否拆解不合格品
def step4():
    global assumeW
    if(x4==1):
        return -d2+assumeW  # 拆解出两个零件，相当于又是一件成
  品的利润
    else:
        return 0

# 进入市场（当未检测成品时才能到达这一步）
def step5():
    global x1
    global x2
    p = ((1 - a1) if x1==0 else 1) * ((1 - b1) if x2==0 else
  1) * (1 - c1)
    return p*c4+(1-p)*(-d1+step4())  # 同时损失了一件

# 对不同情形枚举各种决策
for i in range(17):
    bestw = -45645
    bestx = []
    for x1 in range(2):
        for x2 in range(2):
            for x3 in range(2):
                for x4 in range(x3+1):  # 如果不检测，也拆解不
```

```
    了
                    real_res = 40
                    assumeW = 20
                    while abs(real_res-assumeW)>=10e-6:
                        real_res = step1()
                        assumeW = (real_res+assumeW)/2
                    real_res -=(a2 + b2)
                    if real_res>bestw:
                        bestw = real_res
                        bestx = [bool(x1),bool(x2),bool(x3),bool(x4)]
    print(bestx,bestw)

    a1+=0.01
    b1+=0.01
    c1+=0.01
```

q4_3.py

```
import openpyxl

# a：零配件 b：半成品 c：成品
a11 = 0.02
a12 = 2
a13 = 1
a21 = 0.02
a22 = 8
a23 = 1
a31 = 0.02
a32 = 12
a33 = 2
a41 = 0.02
a42 = 2
a43 = 1
a51 = 0.02
a52 = 8
```

```
a53 = 1
a61 = 0.02
a62 = 12
a63 = 2
a71 = 0.02
a72 = 8
a73 = 1
a81 = 0.02
a82 = 12
a83 = 2
b11 = 0.02
b12 = 8
b13 = 4
b14 = 6
b21 = 0.02
b22 = 8
b23 = 4
b24 = 6
b31 = 0.02
b32 = 8
b33 = 4
b34 = 6
c1 = 0.02
c2 = 8
c3 = 6
c4 = 10
d1 = 200
d2 = 40

x = [0,0,0,0,0,0,0,0,  0,0,0, 0,0,0, 0,0]   # 分别是零件1~8是否检测、半成品1~3是否检测、半成品1~3是否拆解、成品是否检测、成品是否拆解
# assumeW = 0

### 半成品1
```

```python

# 零件1是否检测
def step11():
    return -a13*x[0]-a23*x[1]-a33*x[2]+((1-a11) if x[0]==1 else 1)*((1-a21) if x[1]==1 else 1)*((1-a31) if x[2]==1 else 1)*step12()

# 半成品是否检测
def step12():
    if x[8]==1: # 如果检测，有装配成本、检测费用，还需要进入是否拆解的判断
        p=((1-a11) if x[0]==0 else 1)*((1-a21) if x[1]==0 else 1)*((1-a31) if x[2]==0 else 1)*(1-b11)
        #print("p",p)
        return -b12-b13+(1-p)*step13()
    else: # 不检测，只有装配成本
        return -b12

# 半成品是否拆解
def step13():
    if x[11]==1:
        return -b14+assumeW1
    else:
        return 0


### 半成品2
# 零件1是否检测
def step21():
    return -a43*x[3]-a53*x[4]-a63*x[5]+((1-a41) if x[3]==1 else 1)*((1-a51) if x[4]==1 else 1)*((1-a61) if x[5]==1 else 1)*step22()

# 半成品是否检测
def step22():
```

```
    if x[9]==1: # 如果检测，有装配成本、检测费用，还需要进入是否拆解的判断
        p=((1-a41) if x[3]==0 else 1)*((1-a51) if x[4]==0 else 1)*((1-a61) if x[5]==0 else 1)*(1-b21)
        #print("p",p)
        return -b22-b23+(1-p)*step23()
    else: # 不检测，只有装配成本
        return -b22

# 半成品是否拆解
def step23():
    if x[12]==1:
        return -b24+assumeW2
    else:
        return 0


### 半成品3
def step31():
    return -a73*x[6]-a83*x[7]+((1-a71) if x[6]==1 else 1)*((1-a81) if x[7]==1 else 1)*step32()

# 半成品是否检测
def step32():
    if x[10]==1: # 如果检测，有装配成本、检测费用，还需要进入是否拆解的判断
        p=((1-a71) if x[6]==0 else 1)*((1-a81) if x[7]==0 else 1)*(1-b31)
        #print("p",p)
        return -b32-b33+(1-p)*step33()
    else: # 不检测，只有装配成本
        return -b32

# 半成品是否拆解
def step33():
```

```python
    if x[13]==1:
        return -b34+assumeW3
    else:
        return 0

#### 成品
# 半成品是否检
def step1():
    return ((1-half[0][1]) if x[8]==1 else 1)*((1-half[1][1]) if x[9]==1 else 1)*((1-half[2][1]) if x[10]==1 else 1)*step2()

# 成品是否检测
def step2():
    if x[14]==1: # 如果检测，有装配成本、检测费用，还需要进入是否拆解的判断
        p=((1-half[0][1]) if x[8]==0 else 1)*((1-half[1][1]) if x[9]==0 else 1)*((1-half[2][1]) if x[10]==0 else 1)*(1-c1)
        #print("p",p)
        return -c2-c3+p*d1+(1-p)*step3()
    else: # 不检测，只有装配成本
        return -c2 + step4()

# 成品是否拆解
def step3():
    if x[15]==1:
        return -c4+assumeW
    else:
        return 0

# 市场销售阶段
def step4():
    p=((1-half[0][1]) if x[8]==0 else 1)*((1-half[1][1]) if x[9]==0 else 1)*((1-half[2][1]) if x[10]==0 else 1)*(1-c1)
```

```
    return p * d1 + (1 - p) * (-d2 + step3())

workbook = openpyxl.Workbook()
sheet1 = workbook.active
sheet1.title = "问题3重写"
sheet1.append(["次品率","是否检测零件1","是否检测零件2","是否检测零件3","是否检测零件4","是否检测零件5","是否检测零件6","是否检测零件7","是否检测零件8","是否检测半成品1",
               "是否检测半成品2","是否检测半成品3","是否拆解半成品1","是否拆解半成品2","是否拆解半成品3","是否检测成品","是否拆解成品","利润"])

for i in range(17):
    bestw = -52364
    bestx = []
    for x1 in range(2):
        x[0] = x1
        for x2 in range(2):
            x[1] = x2
            for x3 in range(2):
                x[2] = x3
                for x4 in range(2):
                    x[3] = x4
                    for x5 in range(2):
                        x[4] = x5
                        for x6 in range(2):
                            x[5] = x6
                            for x7 in range(2):
                                x[6] = x7
                                for x8 in range(2):
                                    x[7] = x8
                                    for x9 in range(2):
                                        x[8] = x9
                                        for x10 in range(2):
                                            x[9] = x10
```

```
                                        for x11 in range(2):
                                            x[10] = x11
                                            for x12 in range(x9+1):
                                                x[11] = x12
                                                for x13 in range(x10+1):
                                                    x[12] = x13
                                                    for x14 in range(x11 + 1):
                                                        x[13] = x14
                                                        for x15 in range(2):
                                                            x[14] = x15
                                                            for x16 in range(x15+1):
                                                                x[15] = x16

                                                                half = []  # 存放成本1~3的成本、次品率

                                                                # 半成品1
                                                                real_res = 40
                                                                assumeW1 = -1100
                                                                while abs(real_res - assumeW1) >= 10e-6:
```

```
    real_res = step11()
    # print(real_res)
    assumeW1 = (real_res + assumeW1) / 2
# print(real_res)
real_res -= (a12 + a22 + a32)

p = (1 - a11) * (1 - a21) * (1 - a31) * (1 - b11)
if x[8] == 1:
    p1 = p
else:
    p1 = ((1 - a11) if x[0] == 1 else 1) * (
    (1 - a21) if x[1] == 1 else 1) * (
         (1 - a31) if x[2] == 1 else 1)
half.append([-real_res,1-p/p1])

real_res = 40
assumeW2 = 20
while abs(real_res - assumeW2) >= 10e-6:
```

```
        real_res = step21()
        assumeW2 = (real_res + assumeW2) / 2
    real_res -= (a42 + a52 + a62)

    p = (1 - a41) * (1 - a51) * (1 - a61) * (1 - b21)
    if x[9] == 1:
        p1 = p
    else:
        p1 = ((1 - a41) if x[3] == 1 else 1) * (
        (1 - a51) if x[4] == 1 else 1) * (
            (1 - a61) if x[5] == 1 else 1)
    half.append([-real_res, 1-p / p1])

    real_res = 40
    assumeW3 = 20
    while abs(real_res - assumeW3) >= 10e-6:
        real_res = step31()
        assumeW3 = (real_res + assumeW3) / 2
    real_res -= (a72 + a82)
```

```
    p = (1 - a71) * (1 - a81) * (1 - b31)
    if x[10] == 1:
        p1 = p
    else:
        p1 = ((1 - a71) if x[6] == 1 else 1) * (
        (1 - a81) if x[7] == 1 else 1)
    half.append([-real_res, 1-p / p1])

    real_res = 40
    assumeW = 20
    while abs(real_res - assumeW) >= 10e-6:
        real_res = step1()
        assumeW = (real_res + assumeW) / 2

    if bestw<(real_res-half[0][0]-half[1][0]-half[2][0]):
        bestw = real_res - half[0][0] - half[1][0] - \
        half[2][0]
        bestx = [element for element in x]
```

```
    print(bestx,bestw)
    l = [bool(e) for e in bestx]
    l.append(bestw)  # type: ignore
    l.insert(0,a11)  # type: ignore
    sheet1.append(l)
    a11+=0.01
    a21+=0.01
    a31+=0.01
    a41+=0.01
    a51+=0.01
    a61+=0.01
    a71+=0.01
    a81+=0.01
    b11+=0.01
    b21+=0.01
    b31+=0.01
    c1+=0.01

workbook.save("问题4.xlsx")
```