基于贝叶斯分析的生产决策方案

摘要

企业生产商品的中间流程中，需要选择是否对各部分进行检测或拆解，一个合理的决策方案可以帮助企业有效地控制成本，提升整体的经济效益。

针对原料抽样检测方案的设计，首先通过对比 4 种常见抽样方法的优缺点，针对总体特征，最终选择简单随机抽样。建立与均值相关的枢轴变量，通过 u 检验进行区间检验，得到抽样次数的解析表达式，建立了基于区间估计的抽样次数模型。取不同的绝对精度，计算得到满足 95% 与 90% 置信度所需的抽样次数，并选择抽样次数变化转折处作为最优方案以平衡判断精度与抽样成本。最终确定取绝对精度为 0.025 时，需要分别抽样 390 次与 270 次以满足 95% 与 90% 的置信度，样本中次品数的接受域分别为 [0, 29.26] 与 [0, 17.78] 。此外，补充分析了本问题汇总区间估计两类错误发生的概率。

针对 2 个零件的生产决策分析，首先对应决策节点设置 4 个决策变量，借助生产参数与决策变量计算生产流程中的零件购买成本，零件装配成本，零件检测成本等 7 部分净收益的期望值。引入随生产决策变化的贝叶斯概率，结合净收益期望值建立最优收益模型。由于此处共有 16 种情况，直接采用遍历求解，得到每种情形最佳的决策方案均与实际情况的特征相符合。6 种情况中的最大单件收益期望依次为 17.29，9.60，15.43，11.80，14.03，19.24 元，在整体次品率最高的情况 2 取到最小，在次品率最低的情况 6 取到最大。

针对 m 道工序，n 个零配件情形的生产决策，首先将决策变量与各项净收益期望值进行一般化推广，并引入与半成品相关的净期望值，将模型更新为基于贝叶斯概率的多工序最优收益模型。为了加快方案数过多时的结果收敛速度，采用遗传算法进行求解，得到在 2 道工序，8 个零部件时的最佳决策组合为：所有成品均不予检测，对生产过程中每一个零部件与半成品进行检测，对所有不合格半成品与退还成品进行拆解处理。该方案对应的最大单件收益期望为 58.83 元/件。使用遍历算法验算结果相同，表明该方案即为全局最优解。

针对基于抽样检测的生产决策分析，为模拟真实抽样的情形，制定了给予蒙特卡罗方法的随机抽样模拟方案，发现抽样得到的次品率服从以真实次品率为均值的正态分布。进而利用样品检测信息，制定了结合抽样的综合决策方案。采用 500 次随机抽样模拟生成的次品率组合，采用相同算法计算，得到问题 2 中情况 1，情况 3，情况 4 得到的最优方案与理想方案全数符合，情况 6 所得方案有 93.2% 与理想方案吻合。问题 3 得到的决策方案与理想方案完全一致，且模拟得到的最大单件收益期望为 64.98 元/件。

最后，通过改变零件次品率与随机模拟次数，分别验证了模型的灵敏性与稳定性，并对模型的优缺点进行了分析与展望。

关键字： 生产决策　贝叶斯概率　遗传算法　区间估计　蒙特卡罗模拟

一、 问题重述

1.1 问题背景

在工业产品制造过程中，企业管理者通常需要制定合理而有效的策略，确保生产过程中的每一个环节都得到严格的质量检测和管理。通过系统地衡量和分析检测、拆解、次品处理以及退换货等环节中的成本与效益，企业不仅可以确保产品的质量符合标准，还能在控制成本的同时优化各个阶段的经济效益。这些措施对于企业在竞争激烈的市场中保持竞争力、提高盈利水平具有至关重要的意义。

1.2 问题提出

（1）**原料抽样检测方案设计：**

设计一种针对零配件原料的抽样检测方案，帮助企业决定是否采购对应批次原料。要求在保证检测结果准确性的前提下，尽量减少检测次数，控制检测成本，并根据所制定的方案给出置信度分别为 95% 与 90% 的两种情况的具体结果。

（2）**生产过程中的成本控制决策：**

考虑生产各环节中零部件与装配成品的检测，次品的拆解以及调换次品产生的调换损失等问题，设计一种综合以上因素的成本控制决策方案，实现整体经济利益的最大化，并根据表 1 中的情形给出具体的解决方案。

（3）**多工序生产过程的决策优化：**

在零配件、半成品与成品的次品率均为已知的情况下，对成本控制决策方案进行拓展，制定 m 道工序、n 个零配件时的生产决策方案，并针对 m=2 、n=8 的情形给出具体的解决方案。

（4）**基于抽样检测的生产决策优化：**

在零配件、半成品与成品的次品率均为通过抽样检测得到的情况下，重新制定两个零部件与 m 道工序、n 个零配件时的决策方案。

二、 问题分析

2.1 对原料抽样检测方案设计的分析

为了确定零部件货物的次品率是否满足供应商声称的标称值，需要首先确定科学的抽样方法，选出具有代表性的样本。要确定不同置信度的置信区间，关键是构造合理的枢轴变量进行区间估计，得到取样次数 z 的解析表达式 [1]。根据不同的显著性水平与

绝对精度 [2]，可计算出不同的取样次数 z，取其中 z 变化最剧烈处作为结果，即可得到兼顾估计精度与成本控制的取样方案。

2.2 对生产过程中的成本控制决策的分析

为了表述各环节中的不同决策，可设置对应的决策变量。依靠决策变量与基础数据，可计算生产销售各环节的成本与收益，包括零件购买成本，装配成本，零件检测成本，成品检测成本，商品销售收益，成品拆解收益，成本丢弃成本与调换成本。将这些净收益进行组合 [3]，可以构造出收益函数，计算出生产中的收益期望。决策变量取不同的数值组合时对应的收益期望不同，找出其中期望最大者作为最佳决策方案，对应的期望即为最大收益期望。代入表格中的各项参数，即可算得不同生产情境下的具体决策方案。

2.3 对多工序生产过程的决策优化的分析

对于 m 个零部件，n 个流程的生产流程，需要在前一个问题的基础上对各项费用进行拓展定义，并增加对于半成品的拆解收益，丢弃成本等费用的讨论。基于拓展的各项费用变量，采用类似的方法可构造新的收益函数。对决策变量取不同的数值组合，比较方案之间的收益差异，可得到其中的最佳决策方案与对应的收益期望。代入表格中的各项参数，即可算得 m=2，n=8 的情境下的具体决策方案。针对 m 与 n 较大的情况，可以预见需要计算的情况数较多，需要采用更加智能的算法进行优化求解。

2.4 对基于抽样检测的生产决策优化的分析

在实际生产应用中，次品率一定是通过抽样检测估计得到的，计算结果一定会受到抽样的随机性影响。当样品中的次品分布较为极端，则可能会导致计算得到的最佳方案产生不同。因此，为了模拟真实的抽样，可以考虑使用计算机进行多次抽样模拟，将所得次品率代回问题 2 与问题 3 的模型，即可得到此时基于抽样检测的生产优化决策与对应收益，分析抽样数据随机性对理想结果的影响。

三、 模型的假设

为了建立更精确合理的数学模型，本文根据实际情况建立了一些合理的假设以及条件约束，具体的假设如下所示。

假设一：购买阶段抽样检测的总数足够大；

假设二：拆解过程不会对零部件造成损坏；

假设三：不合格成品中的调换损失是指除调换次品之外的损失；

假设四：在不对原料进行抽样的情况下，认为其次品率不变；

假设五：若组成不合格成品的零部件均预先被检测为合格品，拆解不合格品后得到的合格零部件会单独存放，下次使用无需检验。

四、 符号说明

本文建立模型的过程中主要涉及以下变量，变量及说明如表1。

表 **1**　变量及其说明

符号	含义	单位
Z	每批零部件的总数量	-
z	样本容量	-
S^2	样本方差	-
σ^2	总体方差	-
α	显著性水平	-
λ	区间估计绝对精度	-
p_n	第 n 种零部件的次品率	-
$C_{买}$	零件购买成本	元/件
$C_{装}$	零件装配成本	元/件
$C_{检零}$	零件检查成本	元/件
$B_{拆}$	成品拆解收益	元/件
$C_{检成}$	成品检验成本	元/件
$C_{丢}$	成品丢弃成本	元/件
$C_{调}$	商品调换成本	元/件
W	商品单价	元/件
E	单件商品收益期望	元/件

五、 问题一模型的建立与求解

5.1 抽样方法的选择

通常，对整体的抽样主要有简单随机抽样，分层抽样，系统抽样与整群抽样四种方法，下面分别陈述其定义与优缺点，以便结合题目情形选择合适的采样方法。

（1）简单随机抽样

从总体中随机地抽取 z 个样本，每个样本被抽取的概率相等，且每次抽取相互独立。

优点：简单直观，可以保证采样的随机性，理论上能最好地反映总体特征。

缺点：样本数较大时编号抽样过程较为繁琐，且在样本较为分散时不便于实地调查。

（2）分层抽样

将总体按照某些特征分成若干层，然后从每一层中独立地进行简单随机抽样。

优点：可保证样本在各层中的代表性，提高估计的精度，可对不同层次单独分析。

缺点：需要对总体有一定的了解，以便进行合理分层，且分层过程可能较为复杂。

（3）系统抽样

先将总体中的个体编号，然后按照一定的间隔抽取样本。

优点：操作相对简单，且可以保证样本在总体中的均匀分布，具有一定的代表性。

缺点：对于存在周期性的总体会出现偏差，对总体的了解程度要求较高。

（4）整群抽样

将总体分成若干个群，随机抽取若干个群作为样本，对选中群中所有个体进行调查。

优点：总体分布分散，群内个体集中时抽样和调查操作方便，可提升调查成本。

缺点：样本集中在某些特定的群中时代表性不足，群间差异较大时估计精度较低。

本题中，企业在购买零部件时，对该批零部件没有了解，故无法对其总体进行有效的分层，分层抽样不适用于此类情形。同理，由于对总体了解程度较低，无法确定系统抽样的合适抽样间隔。该零部件样本也不存在集群分布的特征，故不应采用整群抽样。

相较而言，简单随机抽样可使抽样结果具有足够的随机性，能够最好地反映总体的特征，帮助企业对对应批次的货物做出偏差最小的判断。此外，简单随机抽样具有操作简便的特点，取样成本相对较低，可以帮助企业更好地控制成本。综上分析，此处采用简单随机抽样最为合适。

5.2 基于区间估计的抽样次数模型

假设每批零部件的数量足够大，总数为 Z，定义 K_i 为

$$K_i = \begin{cases} 0 & \text{第 i 个非次品} \\ 1 & \text{第 i 个为次品} \end{cases} \tag{1}$$

假设总体中次品数为 Z_{bad}，则 K_i 是由 Z_{bad} 个 1 与 $Z-Z_{\text{bad}}$ 个 0 组成的序列，其包含了该批货物中的所有零部件的优劣信息。根据 K_i 的定义，其平均值 $\overline{K}$ 应为一个在 0 与 1 之间的与总体中次品分布相关的值。我们可利用 K_i 写出这批零部件总体方差 σ^2 为

$$\sigma^2=\frac{1}{Z-1}\sum_{i=1}^{Z}\left(K_i-\overline{K}\right)^2 \tag{2}$$

其中 $\overline{K}=\dfrac{1}{Z}\sum_{i=1}^{z}K_i$。

从总体 $\{K_1,K_2,...,K_Z\}$ 中简单随机抽样，抽取 z 个样本，命为 $\{k_1,k_2,...,k_n\}$，同理可写出样本方差 S^2 为

$$S^2=\frac{1}{z-1}\sum_{i=1}^{z}\left(k_i-\bar{k}\right)^2 \tag{3}$$

在简单随机抽样中，样本方差 S^2 是总体方差 σ^2 的无偏估计 [1]。此处可以引入新变量

$$\mathrm{v}(k)=\frac{1}{z}\left(1-\frac{z}{Z}\right)\sigma^2 \tag{4}$$

它是 $\overline{k}$ 的均方偏差 $\mathrm{V}(\overline{k})$ 的无偏估计。k 有近似分布 $N(\overline{K},V(\overline{k}))$，$\dfrac{\overline{k}-\overline{K}}{\sqrt{\mathrm{v}(\overline{k})}}$ 的渐近分布为标准正态分布正态分布 $N(0,1)$[1]。由此，可得 $\overline{K}$ 的接收域区间估计为

$$1-\alpha\approx P\left\{\frac{\overline{k}-\overline{K}}{\sqrt{\mathrm{v}(\overline{k})}}\leq u_{1-\alpha}\right\} \tag{5}$$

其中 $u_{1-\alpha}$ 为概率 $1-\alpha$ 对应的正态分布临界值。由此可得到 $\overline{K}$ 的接受域区间估计为

$$\left[\overline{k}-u_{1-\alpha}\sqrt{\mathrm{v}(\overline{k})},+\infty\right] \tag{6}$$

设区间估计的绝对精度为 λ，相对精度为 ε，满足

$$\begin{cases}\lambda=\overline{K}\varepsilon\\ |\overline{k}-\overline{K}|<\lambda\\ \left|\dfrac{\overline{k}-\overline{K}}{\overline{K}}\right|\leqslant\varepsilon\end{cases} \tag{7}$$

在 1-α 的置信度下，应满足

$$P\left\{\frac{\overline{k}-\overline{K}}{\overline{K}}\leq\varepsilon\right\}=1-\alpha \tag{8}$$

与区间估计的结果进行对比，可得

$$\lambda=\overline{K}\varepsilon=u_{1-\alpha}\sqrt{\mathrm{v}(\overline{k})} \tag{9}$$

将 $\mathrm{V}(\overline{k})$ 的表达式入方程 (9)，可得

$$\lambda^2=(\overline{K}\varepsilon)^2=(u_{1-\alpha})^2\,\frac{1}{z}\left(1-\frac{z}{Z}\right)\sigma^2 \tag{10}$$

移项即可得到关于样本容量 z 的表达式

$$z=\frac{\left(u_{1-\alpha}\right)^2\sigma^2}{\lambda^2+\dfrac{1}{Z}\left(u_{1-\alpha}\right)^2\sigma^2} \tag{11}$$

当总体数量 Z 足够大时，$\dfrac{1}{Z}$ 趋近于 0，可将式子简化为

$$z=\frac{\left(u_{1-\alpha}\right)^2\sigma^2}{\lambda^2} \tag{12}$$

从该表达式中可以看出，若已知总体的方差 σ^2 与绝对精度 λ，即可计算出满足要求的样本容量 z。由于等式右侧不一定为整数，而取样个数 z 必定是整数，由此可将方程（12）改写为

$$z=[\frac{\left(u_{1-\alpha}\right)^2\sigma^2}{\lambda^2}] \tag{13}$$

此即基于区间估计的抽样次数模型。

5.3 实际样本容量求解与分析

现利用上述模型对两种情况进行求解。

Step1: 初始化参数

题目中给定两种零部件的次品率标称值 10%，其与从货物中检测出次品的概率相等，故设 p=10%。

Step2: 估计总体方差

在计算 z 的过程中，需要用到总体的方差 σ^2。对于一个未知的总体，企业无法对其进行全部抽样检测计算得到真实的总体方差。为了对总体方差进行大致的估计，此处将供应商声称的标称值 p=10% 看作实际值，利用二项分布的公式计算 σ^2 的经验值，即

$$\sigma^2\approx\frac{Z}{Z-1}p(1-p)\approx p(1-p)=0.09 \tag{14}$$

Step3: 确定绝对精度与抽样次数

其他计算 z 所需的参数都确定后，不同的绝对精度 λ 会对 z 的大小产生显著影响。取用的精度越高，取样结果对货物整体情况的反映越准确全面，但是对应抽样次数与抽样成本也会随之增加。过高的精度可以减少因不合格零件产生的损失，但也会大大增加抽样成本，不利于提升整体的经济效益。因此，针对不同的精度要求，需要找到对应高最优的绝对精度 λ。

对于置信度 95% 与 90% 的情况，取不同的 λ 计算对应的样本容量 z，结果如图1与表2所示。

表 **2**　不同绝对精度 λ 对应的样本数 z

显著性水平 α	λ=0.015	0.020	0.025	0.030	0.035	0.040	0.045	0.050
0.1	657	370	237	165	121	93	73	60
0.05	1083	609	390	271	199	153	121	98

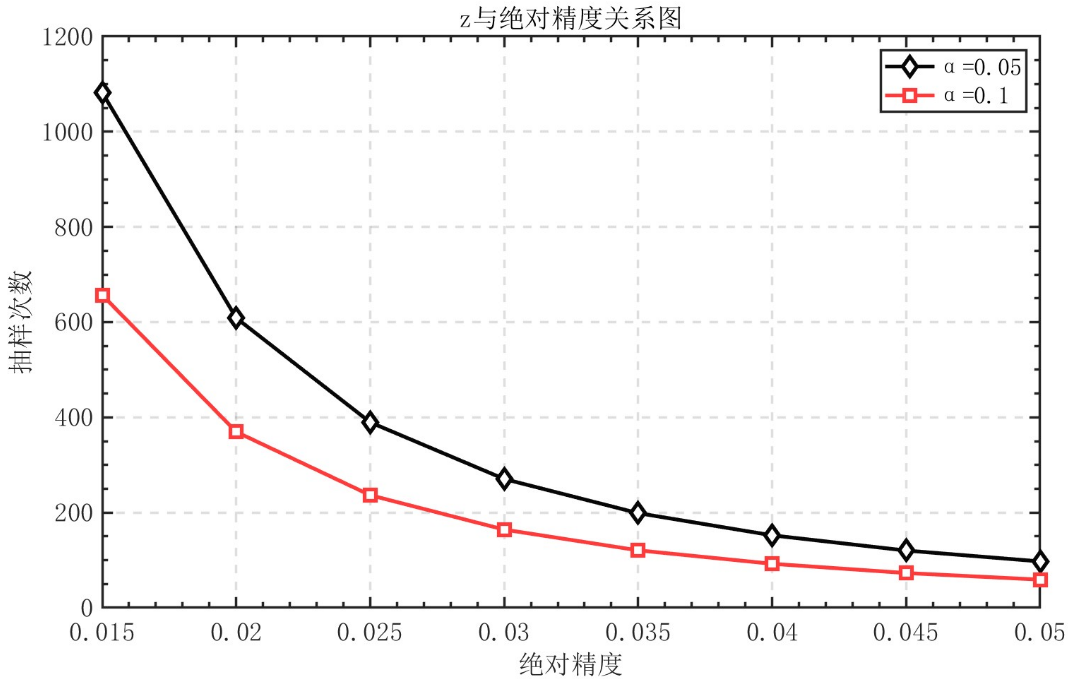

图 **1**　不同绝对精度 λ 对应的样本数 z

从理论层面预测，计算结果应符合以下三条规律：

（1）在相同精度要求下，置信度高的结果应需要更多的抽样次数来保证结果的可信度。

（2）随着精度的提升（λ 越小，精度越高），增加同等精度所需增加的抽样次数会更多，因为精度的提升已接近饱和。

（3）随着精度的提升，做出不同置信度的判断所需抽样次数差距会变大，置信度高者 z 增加更快。

从图中可以看出，置信度 95% 代表的折线始终在置信度 90% 折线上方。随着精度的上升，两条折线的距离逐渐变大，且增加的速度都在增加，与理论预测契合度较高。当绝对精度 λ=0.015 时，对应的抽样次数分别为 657 与 1083，此时的抽样次数偏多，会导致抽样成本较高。对于 λ<0.015 的情况，z 会随着 λ 的减小，与 $\frac{1}{\lambda^2}$ 成正比迅速上升。为了平衡判断精度与抽样成本，应选择图中转折较大处作为最终方案，可以看出，在 λ=0.025 左右，z 随 λ 增加而下降的速度开始变，此处取 λ=0.025 作为置信度 95% 与 90%

时的相对精度最合适，对应的取样次数分别为 390 与 237。

Step4: 计算置信区间

在 z 次抽样中，设其中的次品数为 z_{bad}，则 $z_{\text{bad}} \sim B(z,p)$。此处的取样次数 z 均较大，直接用二项分布计算较为困难，不便确定拒绝域，故根据中心极限定理构造检验统计量

$$\frac{z_{\text{bad}} - zp}{\sqrt{zp(1-p)}}$$

该统计量近似服从标准正态分布 $N(0,1)$，对应的拒绝域为

$$\frac{z_{\text{bad}} - zp}{\sqrt{zp(1-p)}} > u_\alpha$$

代入 p=10% 与对应的显著性水平 α，即可解得对应的拒绝域。

结果显示，置信度在 95%，取样次数为 390 时，接受域为 $[0, 29.26]$。这说明 390 个简单随机取样样本中次品数超过 29 时，即可有 95% 的把握认为该批零部件的次品率超过 10%，应拒收该批货物。若其中次品数少于 29，则没有足够的理由拒绝该批货物，应予以接收。同理在置信度为 90%，取样次数为 270 时，计算得到接受域为 $[0, 17.78]$。这说明 390 个简单随机取样样本中次品数小于 14 时，即可有 90% 的把握认为该批零部件的次品率低于 10%，应予以接收。

5.4 实际情况假设检验错误概率分析

上述的模型是建立在假设检验的区间检验上的，而在这个过程中，可能会存在两种可能的错误判断，即假设检验中的第一类错误与第二类错误。第一类错误是拒绝了正确的结论，第二类错误是指接受了一个错误的结论，二者对应概率如图2所示，图中的概率密度函数均经过归一化，积分对应的面积与其概率相等。下面，对实际计算的两种情况进行错误概率的分析。

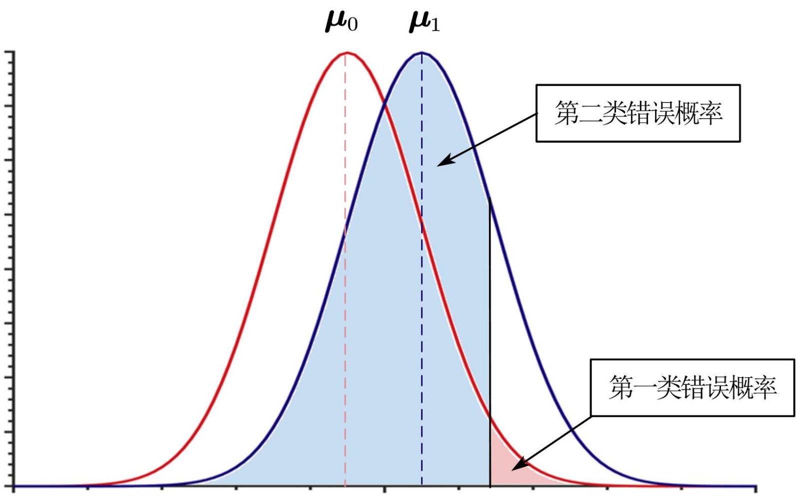

图 **2**　假设检验中的两类错误概率

（1）置信度为 95%

要求在 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，故设定显著性水平 α=0.05。取绝对精度 λ=0.025，该值可根据对估计精度的实际需求变化。

为了检验零部件的次品率是否低于标称值 10%，建立假设 H_0 与 H_1 为

$$H_0 : p \leqslant 10\% \quad H_1 : p > 10\%$$

为方便计算与分析，将单侧假设检验 H_0 与 H_1 转换为等效假设 H_0' 与 H_1'，即

$$H_0' : p = 10\% \quad H_1' : p > 10\%$$

根据 m_i 的分布特征分析可知，若 H_0' 成立，p=10%，则 $\sum_{i=1}^{z} k_i$ 服从二项分布 $B(z, 0.1)$，此时构造统计量

$$\frac{\sum_{i=1}^{z} k_i - zp}{\sqrt{zp(1-p)}}$$

由中心近似定理，该统计量近似服从标准正态分布 $N(0,1)$。此时，拒绝 H_0 的条件为

$$\frac{\overline{k} - p}{\frac{\sigma}{\sqrt{z}}} > u_{0.05} \tag{15}$$

对于该判断，我们认为其发生第一类错误的概率

$$\alpha = 5\%$$

假设其犯第二类错误时真实的均值为 μ，则发生第二类错误的概率

$$\beta = \int_{-\infty}^{\frac{u_{0.95}\sqrt{\frac{p(p+1)}{z}}+0.1-\mu}{\sqrt{p(p+1)}}} \phi(x)dx$$

其中 $\phi(x)$ 为标准正态分布的概率密度函数，β 的大小取决于真实的均值 μ 与样本容量 z。

（2）置信度为 90%

参照置信度为 95% 时相同的思路，同样取 λ=0.025，此时的显著性水平 α=0.1，建立相同的基本假设 H_0 与 H_1 并进行转换。若 H_0' 成立，p=10%，则 $\sum_{i=1}^{n} k_i$ 服从二项分布 $B(z, 0.1)$，此时构造相同的服从标准正态分布 $N(0,1)$ 的检验统计量。此时，接受 H_0' 的条件为

$$\frac{\overline{k} - p}{\frac{\sigma}{\sqrt{z}}} < 0.1 \tag{16}$$

其犯第一类错误的概率是

$$\alpha = 10\%$$

假设其犯第二类错误时真实的均值为 μ，则发生第二类错误的概率

$$\beta = \int_{-\infty}^{\frac{u_{0.9}\sqrt{\frac{p(p+1)}{z}}+0.1-\mu}{\sqrt{p(p+1)}}} \phi(x)dx$$

其大小同样取决于真实均值 μ 与样本容量 z 的大小。

六、 问题二模型建立与求解

问题二在主要研究在生产过程中的决策问题，其生产流程图如图3所示。

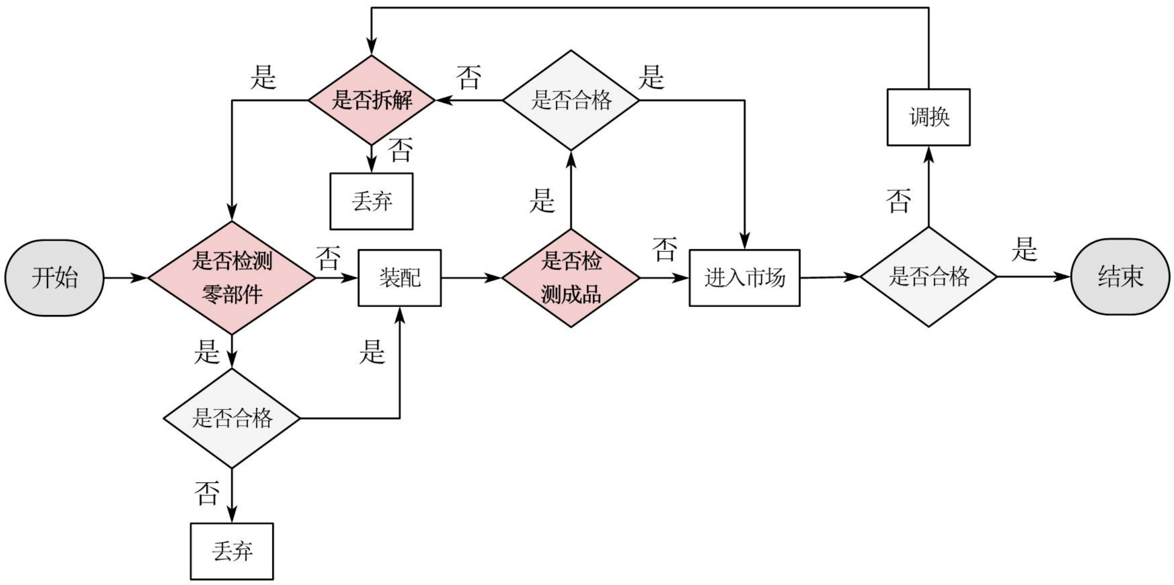

图 **3** 生产决策流程图

6.1 决策变量的设立

在生产过程中，中间环节涉及到的选择主要有 4 个：

（1）是否检测零件 1；（2）是否检测零件 2；（3）是否检测装配后的成品；（4）成品中检测出的次品是否要拆解。

不同的选择组合会带来不同的收益，我们的目标就是通过找到特定情况下的最佳决策方案，实现整体经济利益的最大化。为了在收益计算过程中定量体现不同决策带来的影响，对应 4 个选择，分别设定 4 个决策变量为

$$t_1 = \begin{cases} 0 & \text{不检测零件1} \\ 1 & \text{检测零件 1} \end{cases} \quad t_0 = \begin{cases} 0 & \text{不检测成品} \\ 1 & \text{检测成品} \end{cases}$$
$$t_2 = \begin{cases} 0 & \text{不检测零件2} \\ 1 & \text{检测零件 2} \end{cases} \quad s_1 = \begin{cases} 0 & \text{不拆解成品} \\ 1 & \text{拆解成品} \end{cases} \tag{17}$$

这四个变量取不同的值相互组合，组成一个大小为 4 的数组 t_1, t_2, t_0, s_1，该数组代表了在对应情况下的不同环节的所有决策信息。如数组 1100 代表对零件 1 和零件 2 都进行检测，但是对装配好的成品不进行检测，最后若成品被顾客退回，选择不拆解成品，直接丢弃。各决策点之间的关系图如图4所示。

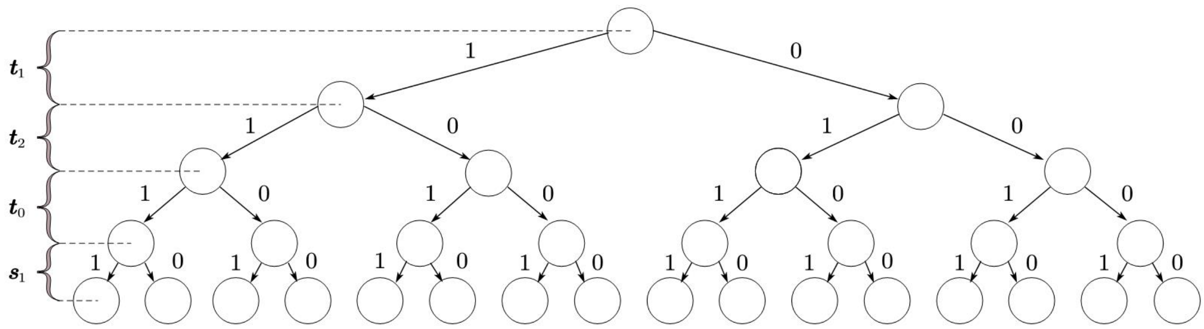

图 **4**　不同决策情况的关系图

6.2 各部分收益的计算

中间环节的决策不仅会决定是否增加对应的检测成本或拆解成本，还会影响后续环节中的收益。比如，若在生产过程中对每个零件和组装后的成品都进行了检测，若结果均为合格，则出售商品后一定不存在调换损失。

为了更清晰地计算各个环节的净收益，我们将其分为 7 部分：

（1）零件成本 $C_{买}$；（2）装配成本 $C_{装}$；（3）零件检测成本 $C_{检零}$；（4）拆解成品收益 $B_{拆}$；（5）成品检测成本 $C_{检成}$；（6）成品丢弃成本 $C_{丢}$；（7）调换成本 $C_{调}$。

将这 6 部分组合，即可得到该方案对应的整体经济收益，对比不同的方案即可得到最佳方案。下面对不同部分的净收益的期望分别进行计算。

（1）零件成本 $C_{买}$

对于两个零部件组成的成品，可以将其零件成本 $C_{买}$ 表达为

$$C_{买} = a_1 + a_2 \tag{18}$$

其中 a_1 为购买 1 个零件 1 的成本，a_2 为购买 1 个零件 2 的成本。

（2）装配成本 $C_{装}$

对于两个零部件组成的成品，生产过程中不会产生半成品，只需要一次装配即可获得成品，故可以将其装配成本 B 表达为

$$C_{装} = b \tag{19}$$

其中 b 为装配 1 次所消耗的成本。

（3）零件检测成本 $C_{检零}$

零件成本 $C_{\text{买}}$ 与装配成本 $C_{\text{装}}$ 均为生产过程中的固有成本，只要零部件的单价与装配一次的成本固定，$C_{\text{买}}$ 与 $C_{\text{装}}$ 就均为定值。与之不同的是，零件检测成本的数值取决于是否选择检测零件，$C_{\text{检零}}$ 应为关于 t_1 与 t_2 的函数。

在初步检测过程中，零件检测成本 $C_{\text{检零}\,1}$ 为

$$C_{\text{检零}\,1} = c_1 t_1 + c_2 t_2 \tag{20}$$

其中 c_1 为检测 1 次零件 1 的成本，c_2 为检测 1 次零件 2 的成本。

当检测结果显示零件为合格品，则进入下一步装配环节。若检测结果为次品，则应该将该次品零部件丢弃，对应的损失就是检测费用 c_n 以及购买该不合格零件所需费用 a_n。由于不合格零件做出来的成品必然是不合格品，我们需要用一个合格零件替代该次品进入后续装配环节。为了得到该合格零件，需要继续进行检测，直到检测到一个合格品。下一个零件的检测结果仍未未知，其为不合格品的概率与该零件的次品率 p_n 相等，第 i 次重复检测出来不合格品的概率为 p_n^i。每个不合格零件都会带来损失 $a_n + c_n$，故重复检测所需成本的期望可写作

$$C_{\text{检零}\,2} = t_1 \sum_{i=1}^{\infty} p_1^i \left(a_1 + c_1\right) + t_2 \sum_{i=1}^{\infty} p_2^i \left(a_2 + c_2\right) \tag{21}$$

此概率的求解可通过等比数列的求和简化，最终可将用于检测零件成本的费用 $C_{\text{检零}}$ 写作

$$C_{\text{检零}} = c_1 t_1 + c_2 t_2 + t_1 \frac{p_1}{1 - p_1} \left(a_1 + c_1\right) + t_2 \frac{p_2}{1 - p_2} \left(a_2 + c_2\right) \tag{22}$$

（4）拆解收益 $B_{\text{拆}}$

在检测出一个成品为不合格品后，可选择将其丢弃或拆解。若选择拆解，返还的原料可以弥补之前购买原来的费用 $C_{\text{买}}$。根据本文假设，若组成该不合格品的零部件均已经过检验为合格品，则下次使用无需再次检验，可以为第二轮生产节省检测成本 $C_{\text{检零}}$，$C_{\text{检零}}$ 应算进拆解收益中。若该轮的零部件未经检验，则 C 中对应项为 0，不会影响结果。因此，拆解一个成品的收益 $B_{\text{拆}}$ 可写作

$$B_{\text{拆}} = (A + C - e) s_1 \tag{23}$$

其中 e 为拆解一个成品所需的价格。

（5）丢弃成本 $C_{\text{丢}}$

由于丢弃不合格成品后，其中的零部件不再能够再次使用。用于购买和检测零部件，组装与检测成品的价格都已在其他部分有所计算，故

$$C_{\text{丢}} = 0 \tag{24}$$

（6）成品检测成本 $C_{\text{检成}}$

设检测一个成品需要花费 d 元，则检测成本

$$C_{检成} = dt_0 \tag{25}$$

（7）调换成本 $C_{调}$

需要明确的是，只有没经过检测的成品才会出现被调换的情况，其概率为 $1-t_0$，调换一个成品会产生 ω 的调换成本。除调换成本之外，还需要给客户提供一个合格品作为补偿，该合格品会在后续生产中得到，其成本在后续生产中计算，此处不作讨论。综上，调换一个不合格品会消耗调换成本 $C_{调}$ 为

$$C_{调} = (1-t_0)\omega \tag{26}$$

6.3 基于贝叶斯的概率的最优收益模型

与传统预计相比，贝叶斯概率引入了先验概率，通过结合先验知识和新的观测数据来更新对事件发生概率的估计。在该问题中，我们可以利用贝叶斯概率来计算成品为合格品的概率，以生产过程中的决策作为观测数据，影响最终概率的大小。设零件 1，零件 2 与成品的次品率分别为 p_1，p_2 与 p_0，良品率 q_1，q_2，q_0 与对应的次品率和为 1。

设最后的成品为合格品的概率为 θ，用各部分良品率与次品率来表示 θ，有

$$\theta = \frac{q_1q_2q_0}{(1-p_1t_1)(1-p_2t_2)} \tag{27}$$

θ 的取值会随着决策的改变而改变。其中分子 $q_1q_2q_0$ 为三部分的良品率相乘得到的先验概率，表示在各环节都不经检验，即 t_1, t_2 均为 0 时，成品为合格品的概率 $\theta = q_1q_2q_0$。中间若对零件进行检验，可以保证参与制作的该零件是合格品，可以作为观测数据影响最终的概率。例如，两个零件均参与检验时，t_1，t_2 均为 1，成品为合格品的概率 $\theta = \dfrac{q_1q_2q_0}{(1-p_1)(1-p_2)}$。

得到成品的合格率后，我们可以借助前文对各部分净收益的分析，写出一轮生产中的总收益期望

$$E = -C_{买} - C_{装} - C_{检零} - C_{检成} + (1-\theta)(B_{拆} - C_{丢} - C_{调}) + \theta W \tag{28}$$

其中 W 是一个商品的售价。$B_{拆} - C_{丢} - C_{调}$ 是在成品不合格时涉及的费用，θW 表示合格产品出售后带来的收入。

由此，可以将基于贝叶斯概率的最优收益模型总结为

$$
\max E
$$

$$
\text{s.t.}\begin{cases}
E=-C_{买}-C_{装}-C_{检零}-C_{检成}+(1-\theta)\left(B_{拆}-C_{丢}-C_{调}\right)+\theta W \\
C_{买}=a_1+a_2 \\
C_{装}=b \\
C_{检零}=C_{检零}=c_1t_1+c_2t_2+t_1\frac{p_1}{1-p_1}\left(a_1+c_1\right)+t_2\frac{p_2}{1-p_2}\left(a_2+c_2\right) \\
B_{拆}=(A+C-e)s_1 \\
C_{丢}=0 \\
C_{检成}=dt_0 \\
C_{调}=\left(1-t_0\right)\omega \\
\theta=\dfrac{q_1q_2q_0}{\left(1-p_1t_1\right)\left(1-p_2t_2\right)}
\end{cases} \tag{29}
$$

6.4 基于 0-1 规划的最佳方案求解

确定最合理的决策方案，本质上是确定能够使整体经济效益最大的 $\{t_1,t_2,t_0,s_1\}$ 数组。由于这四个决策变量的取值都只有 0 或 1，故这是一个 0-1 整数规划问题。下面陈述问题的求解流程。

Step1: 初始化参数

不同的生产过程中的参数的选择是不同的，故首先需要结合实际情况，将计算所需要用到的参数进行初始化。其中包括各零件的单价 a_1，a_2，装配成本 b，各环节检测的成本 c_1，c_2，d，各部分的次品率 p_1，p_2，p，拆解的成本 e，单个成品售价 W 与调换成本 ω。

Step2: 组合决策变量

由于此问题设计的决策变量数量较少，无需采用启发式算法进行求解，故考虑遍历所有可能情况精确求解。将 4 个决策变量分别取不同的值，得到共 2^4 种可能的方案用于后续计算。

Step3: 计算各方案的收益期望值

代入不同的 $\{t_1,t_2,t_0,s_1\}$ 组合，根据

$$
E=-C_{买}-C_{装}-C_{检零}-C_{检成}+(1-\theta)(B_{拆}-C_{丢}-C_{调})+\theta W
$$

依次计算各方案对应的收益期望。

Step4: 比较获取最佳方案

依次比较各方案的收益期望，若方案的收益期望大于当前的最大收益期望，则将其令为新的最大收益期望，对应方案作为最优方案。

6.5 不同生产条件的计算结果与分析

题目中表 1 共给出了企业在实际生产过程中遇到的 6 种不同情况，将其中的数据代入上述求解流程，可以得到最佳的收益期望与对应的方案如表3，图5与图6所示，图5中方案 1-16 分别对应 0000-1111。

表 **3**　不同生产情况下的最大收益期望与对应方案

生产情况	1	2	3	4	5	6
最大单件收益期望（元）	17.29	9.60	15.43	11.80	14.03	19.24
决策变量组合 $\{t_1, t_2, t_0, s_1\}$	1001	1101	1011	1111	0101	1000

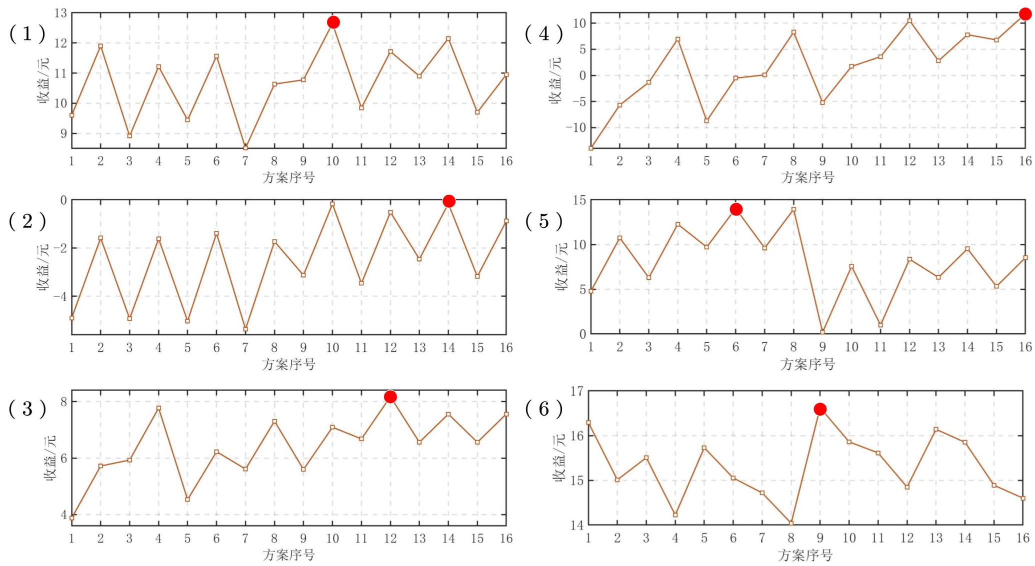

图 **5**　不同决策方案下的收益期望

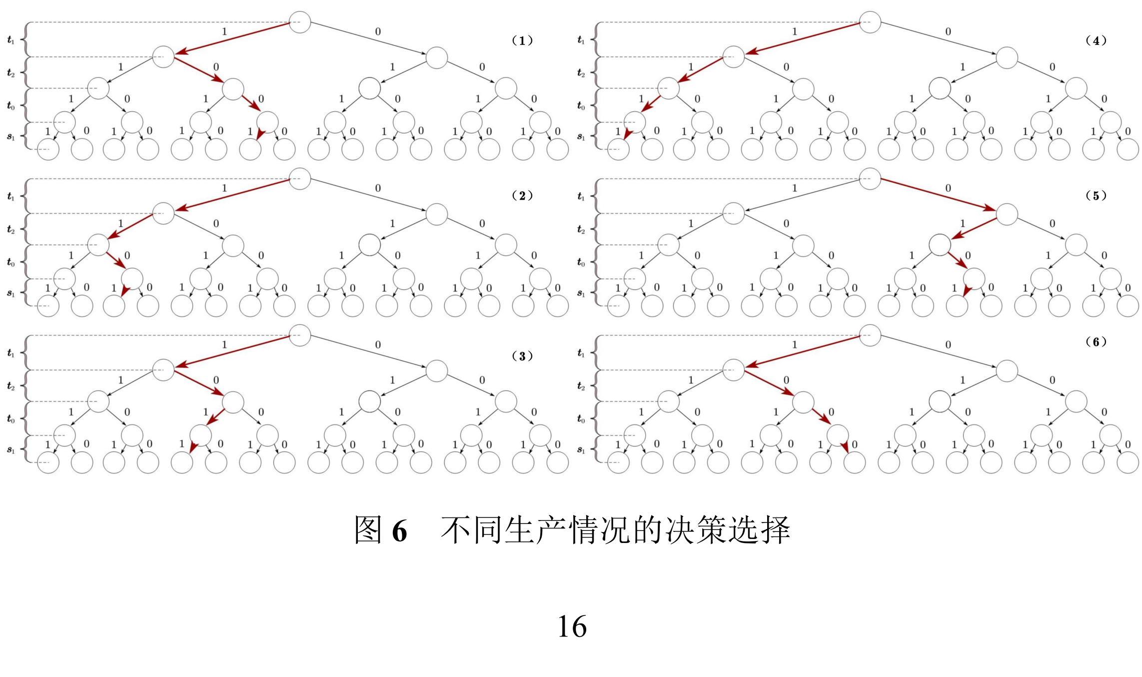

图 **6**　不同生产情况的决策选择

从结果中可以看出，次品率对商品的单件最大收益期望影响巨大，这是因为此时的各部分次品率最低，大大提升了成品的合格率。情况 6 的各项次品率均为 5% 最低，其单件商品的最大收益期望取到最大为 19.24 元。情况 2 每部分的次品率均为情况 1 的两倍，其最大收益期望最小为 9.60 元。情况 1 的次品率是情况 2 的一半，其他情况均一致，可通过选取合适的方案使最大单件收益期望提升 80.1% 至 17.29 元。

最佳决策方案的选择也和实际的生产情况密切相关。在情况 3 和情况 4 中，调换损失最高。在零部件均合格的情况下，由成品次品率引起的调换损失的期望均高于其检测成本，故其最佳方案都选择了检测成品。在情况 6 中，拆解一个成品所需成本最高，故只有这个最佳方案没有选择拆解成品。在情况 1,2,3,4,6 中，零部件 1 的检测成本均低于零件 2，故其中最佳方案均选择检测零件 1，且次数多于检测零件 2 的次数。在情况 5 中，检验一次零件 1 的成本过高，故最终未选择检测零件 1。

以上分析都表明计算结果与理论分析高度契合。

七、 问题三模型的建立与求解

问题二的情境中，两个零部件可以通过一次装配直接成型，其决策理论方法的针对性较强，不便于推广应用于其他产品的生产中。因此，为了提升理论的实用性，我们将在问题二模型的基础上进行拓展，解决 n 个零件，m 道工序的生产情形中的最优决策问题。

7.1 决策变量的更新

在 n 个零件，m 道工序的情况下，生产过程中的决策组合可能性会因为流程的复杂性提升而大大增加，此时需要将问题二中的决策变量一般化，此时重新定义决策变量为

$$t_m = \begin{cases} 1 & \text{检测成品} \\ 0 & \text{不检测成品} \end{cases} \qquad s_m = \begin{cases} 1 & \text{拆解成品} \\ 0 & \text{不拆解成品} \end{cases} \tag{30}$$

$$t_i = \begin{cases} 1 & \text{检测零件 i} \\ 0 & \text{不检测零件 i} \end{cases} \quad (i \in \{1, 2, \ldots n\}) \tag{31}$$

$$t_{jl} = \begin{cases} 1 & \text{检测第}j\text{道工序后的第}l\text{个半成品} \\ 0 & \text{不检测第}j\text{道工序后的第}l\text{个半成品} \end{cases} \quad (j \in \{1, 2, \ldots m-1\}) \tag{32}$$

$$s_{jl} = \begin{cases} 1 & \text{拆解第}j\text{道工序后的第}l\text{个半成品} \\ 0 & \text{不拆解第}j\text{道工序后的第}l\text{个半成品} \end{cases} \quad (j \in \{1, 2, \ldots m-1\}) \tag{33}$$

与先前相同，这些决策变量的取值代表了在不同的决策节点做出的对应决策，可实现整体经济利益最大化的决策变量组合即为所求。

7.2 净收益变量的更新

在生产工序增加后，生产过程中会出现若干半成品，我们同样需要讨论其拆解或丢弃时对总费用的影响。因此，需要对先前的净收益变量进行扩展与更新。

其中，几个已有的变量可分别更新为

$$\begin{cases} C_{\text{买}} = \sum_{i=1}^{n} a_i \\ C_{\text{装}} = \sum_l \sum_{j=1}^{m} b_{jl} \\ C_{\text{调}} = (1 - t_m)\,\omega \\ C_{\text{检零}} = \sum_{i=1}^{n} c_i t_i + \sum_{i=1}^{n} \dfrac{t_i p_i}{1 - p_i}(a_i + c_i) \\ B_{\text{拆成}} = (A + C - e_m)\, s_m \\ C_{\text{丢成}} = 0 \end{cases} \tag{34}$$

其中 a_i 为购买 1 个零件 i 的成本，b_{jl} 为装配 1 次第 j 个工序中的第 l 个半成品所消耗的成本，c_i 为检测 1 个零件 i 的成本，$B_{\text{拆成}}$ 为拆解一个成品的净收益期望，$C_{\text{丢成}}$ 为丢弃一个成品的净收益期望。

下面引入与半成品相关的变量。需要提前声明的是，以下涉及到的含 $\sum_{jl}$ 或 $\prod_{jl}$ 的变量并非对于所有 j 与 l 求和，而是生产过程中第 j 道工序后第 l 个半成品的对应量求和。例如，$\sum_{jl} a_i$ 代表生产第 j 道工序的第 l 个半成品所需的零件成本之和，$\sum_{jl} b_{j_0 l_0}$ 为装配第 j 个工艺后第 l 个半成品过程中消耗的装配成本。

（1）半成品拆解成本 $C_{\text{拆半}}$

与拆解成品不同，拆解一个半成品后，只需用一个新的半成品替代即可继续生产，该过程中产生的损失 $C_{\text{拆}jl}$ 可写作

$$C_{\text{拆}jl} = \left(\sum_{jl} b_{j_0 l_0} + e_{jl} \right) s_{jl} t_{jl} \tag{35}$$

（2）半成品丢弃成本 $C_{\text{丢}jl}$

丢弃不合格半成品后，生产过程中用于购买和检测零部件，组装与检测成品的价格都会损失，而前文已经论述丢弃成品不会额外造成损失，故 $C_{\text{丢 jl}}$ 只计算丢弃半成品的成本，可写作

$$C_{\text{丢}jl} = \left[\sum_{jl} \left(a_i + c_i + \frac{t_i p_i}{1 - p_i}(a_i + c_i) \right) + \sum_{jl} b_{j_0 l_0} \right] (1 - s_{jl})\, t_{jl} \tag{36}$$

（3）（半）成品检测成本 $C_{\text{检成}}$

生产过程中对于成品和半成品的检测成本可写作

$$C_{\text{检成}} = \sum_l \sum_{j=1}^{m-1} d_{jl} t_{jl} + d_0 t_0 \tag{37}$$

其中 d_{jl} 为检测 1 个第 j 道工序中第 l 个半成品的成本，d_m 为检测 1 个成品的成本。

7.3 最优收益模型的更新

基于先前给予贝叶斯概率的最优收益模型，我们同样需要引入半成品相关变量进行更新。设第 i 个零件的次品率为 p_i，第 j 道工序后的第 l 个半成品的次品率为 p_{jl}，成品的次品率为 p_0，其对应的良品率分别为 q_i，q_{jl}，q_0。可基于贝叶斯概率，将各半成品的合格概率 θ_{jl} 写作

$$\theta_{jl} = \frac{\prod_{jl} q_i \prod_{jl} q_{j_0 l_0}}{\prod_{jl} (1 - p_i t_i)} \tag{38}$$

成品进入市场后无需调换的概率 θ 为

$$\theta = \frac{q_0 \prod_{i=1}^{n} q_i \prod_l \prod_{i=1}^{m-1} q_{ji}}{\prod_{i=1}^{n} (1 - p_i t_i) \prod_l \prod_{j=1}^{m-1} (1 - p_{jl} t_{jl})} \tag{39}$$

由此可以将最优收益函数期望更新为

$$\begin{aligned} E = & - C_{\text{买}} - C_{\text{装}} - C_{\text{检零}} - C_{\text{检成}} + (1 - \theta)(B_{\text{拆}} + C_{\text{丢}} - C_{\text{调}}) \\ & + \theta W - \sum_l \sum_{j=1}^{m-1} \frac{1 - \theta_{jl}}{\theta_{jl}} (C_{\text{拆}jl} + C_{\text{丢}jl}) \end{aligned} \tag{40}$$

由此，可确定多工序最优收益模型为

$$\max E$$

$$\text{s.t.} \begin{cases} E = -C_{\text{买}} - C_{\text{装}} - C_{\text{检零}} - C_{\text{检成}} + (1 - \theta)(B_{\text{拆}} + C_{\text{丢}} - C_{\text{调}}) \\ +\theta W - \sum_l \sum_{j=1}^{m-1} \dfrac{1 - \theta_{jl}}{\theta_{jl}} (C_{\text{拆}jl} + C_{\text{丢}jl}) \\ C_{\text{买}} = \sum_{i=1}^{n} a_i \\ C_{\text{装}} = \sum_l \sum_{j=1}^{m} b_{jl} \\ C_{\text{调}} = (1 - t_0)\,\omega \\ C_{\text{检零}} = \sum_{i=1}^{n} c_i t_i + \sum_{i=1}^{n} \dfrac{t_i p_i}{1 - p_i} (a_i + c_i) \\ B_{\text{拆成}} = (A + C - e_0)\, s_0 \\ C_{\text{丢成}} = 0 \\ C_{\text{拆}jl} = \left(\sum_{jl} b_{j_0 l_0} + e_{jl} \right) s_{jl} t_{jl} \\ C_{\text{丢}jl} = \left[\sum_{jl} \left(a_i + c_i + \dfrac{t_i p_i}{1 - p_i} (a_i + c_i) \right) + \sum_{jl} b_{j_0 l_0} \right] (1 - s_{jl})\, t_{jl} \\ C_{\text{检成}} = \sum_l \sum_{j=1}^{m-1} d_{jl} t_{jl} + d_m t_m \\ \theta_{jl} = \dfrac{\prod_{jl} q_i \prod_{jl} q_{j_0 l_0}}{\prod_{jl} (1 - p_i t_i)} \\ \theta = \dfrac{q_0 \prod_{i=1}^{n} q_i \prod_l \prod_{i=1}^{m-1} q_{ji}}{\prod_{i=1}^{n} (1 - p_i t_i) \prod_l \prod_{j=1}^{m-1} \left(1 - p_{jl} t'_{jl}\right)} \end{cases} \tag{41}$$

7.4 基于遗传算法的最佳方案求解

在该问题中，需要计算的决策变量组合数会随工序数 m 与零件数 n 的增加而指数式上升，故在 m 与 n 较大时，遍历算法的效率较低，此时就需要只能算法提升收敛速度。遗传算法（GA）是一种模拟自然进化过程的优化算法，可以用于求解复杂问题中的最优解，具有全局搜索能力强，通用性高等特点，可以用于此处对于最佳决策方案的求解。遗传算法的流程图如图7所示。

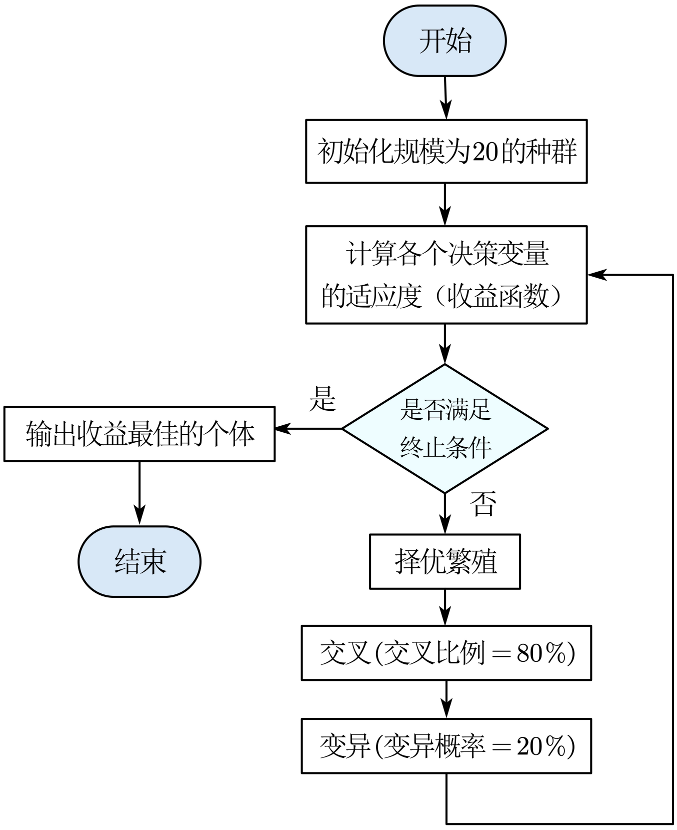

图 **7**　遗传算法流程图

Step 1: 初始化种群

设置初始种群规模为 20，其中个体代表一个可行的解，每个解都是由不同的决策变量（t_i、t_{jl}、s_{jl}）取值组成的方案。在 $m=2$，$n=8$ 的情况下，一共有 16 个决策变量。

Step 2: 设置适应度函数

大自然中个体的适应度函数衡量个体的优劣程度，即本题的适应度函数为

$$
\begin{aligned}
E = & -C_{\text{买}} - C_{\text{装}} - C_{\text{检零}} - C_{\text{检成}} + (1-\theta)\left(B_{\text{拆}} + C_{\text{丢}} - C_{\text{调}}\right) \\
& + \theta W - \sum_{l} \sum_{j=1}^{m-1} \frac{1-\theta_{jl}}{\theta_{jl}} \left(C_{\text{拆}jl} + C_{\text{丢}jl}\right)
\end{aligned} \tag{42}
$$

Step 3: 生成新种群

在迭代过程中，种群之间会发生选择，交叉与变异。其中，选择是根据适应度函数，选择优秀的个体作为父代进行繁殖，适应度值高的个体被赋予更高的概率以产生下一代。交叉是两个父代个体的基因相互交换以生成新的后代，扩充可行解空间。变异则是为了增加种群的多样性并防止陷入局部最优解，对新生成的后代进行变异操作。

设置交叉比例为 0.8，变异概率为 0.2，通过上述操作生成新一代的种群，新种群将取代旧种群继续进行迭代。

Step 4: 终止循环

当满足达到最大迭代次数 100 或收益函数收敛到某个适应度值（即最佳解不再显著变化）的条件则终止，选择适应度最优个体作为最终的解。

7.5 方案求解结果与分析

代入生产过程中的各项成本与收益参数，经过遗传算法的求解，可以得到最佳方案对应的决策变量数组为

$$
\begin{aligned}
&\{t_0,t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_{11},t_{12},t_{13},s_0,s_{11},s_{12},s_{13}\}\\
=&\{0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\}
\end{aligned}
$$

对应的最大收益期望 $E_{\max}=58.83$ 元/件。

该结果表明，在该生产条件下，为了实现整体利益的最大化，生产过程中每一个零部件与半成品都应该检测，若发现不合格的半成品都应进行拆解处理。所有未出售的成品均不予检测，对于客户退回的不合格成品，均进行拆解处理。

从生产参数中分析可以发现，所有零部件的检测成本均小于 2 元，半成品的检测成本也均为 4 元，远远小于商品售价 200 元与调换成本 40 元。花费相对较少的费用用于中间环节的检测可大大减少成品中次品的比例，可有效避免高昂的退换成本，故最佳方案中选择对每个零件和半成品进行检验是合理的。

所有经过检验后参与装配的半成品和零部件均为合格品，可认为成品中次品的占比与成品的次品率相等为 10%，此时直接出售产生退换成本的期望

$$
E_{\text{退}}=40\times 10\%=4
$$

而检测一个成品需要固定花费 6 元，高于退换成本的期望值，故此处选择不检测成品具有合理性。

7.6 基于遍历算法的结果检验

考虑到此处只有 16 个取值为 0 或 1 的决策变量，所有的可能组合有 $2^{16}=15536$ 种。若使用计算机遍历所有情况，仍能在较短的时间内得到全局最优解。为了检验遗传

算法的求解结果是否陷入了局部最优解，我们采用 6.4 节中基于 0-1 规划的最佳方案求解思路，对 2 道工序，8 个零件的情况进行了遍历求解，得到全局最佳方案为

$$
\begin{aligned}
&\{t_0,t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_{11},t_{12},t_{13},s_0,s_{11},s_{12},s_{13}\}\\
=&\{0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\}
\end{aligned}
$$

对应的最大收益期望 $E_{\max}=58.83$ 元/件。该结果与遗传算法的求解结果完全一致，说明该情境中遗传算法所得即为全局最优解。

八、 问题四模型的建立与求解

本问要求我们在次品率均为抽样结果的情况下重新计算问题二与问题三，重新得到对应情况中的最优方案与判断指标。该问题的核心在于模拟真实生产情况中的次品率获取，分析由于抽样随机性导致的误差对于决策结果的影响程度。

8.1 随机抽样模拟方案

为了模拟真实生产情况中的次品率估计，我们采用计算机模拟的方式对假想总体进行简单随机抽样。下面以真实零件与成品的次品率均为 10%，置信度为 95%，绝对精度为 0.025 的情况举例说明抽样模拟的方法。

使用计算机生成编号 0-999 的共 1000 个零件 i，其中有 100 个零件为次品。根据问题一的结论，在绝对精度为 0.025 时，需要抽取 390 次以保证 95% 的置信度。因此，使用蒙特卡罗方法在其中随机抽样 500 次，每次抽出 390 个零件作为样品，并对这 390 个零件进行全数检测。设其中的次品数为 z_{ibad}，则此次抽样得到第 i 个零件的次品率 p_i' 为

$$
p_i'=\frac{z_{\text{ibad}}}{390} \tag{43}
$$

由此可模拟抽样得到每个零件的次品率，其中第 i 个零件的抽样结果中合格品的个数为 $390-z_{\text{ibad}}$。由于成品的次品率是指将正品零部件装配后的产品次品率，故应选出所有可参与组装正品零部件零件，组合可得到的成品数为

$$
z'=\min\{390-z_{\text{ibad}}\}\,(i=1,2,\cdots,n) \tag{44}
$$

这 z' 个成品是随机生成的，其合格率服从二项分布 $B(z',0.1)$。统计模拟成品中的次品数量 z'_{bad}，即可计算模拟得到的成品次品率为

$$
p_0'=\frac{z'_{\text{bad}}}{z'} \tag{45}
$$

经过上述步骤，即可实现对实际抽样情况的模拟，得到不精确的零件次品率 p_i' 与成品次品率 p_0'。半成品次品率抽样估计模拟流程同理。

以真实次品率为 10% 的情况为例分析，图8是蒙特卡罗模拟抽样得到的次品率 p' 的分布直方图。可以看出，经过多次简单随机抽样，得到的次品率主要聚集在真实值 10% 附近。经过 K-S 假设，可知其分布近似为以 10% 为均值的正态分布。

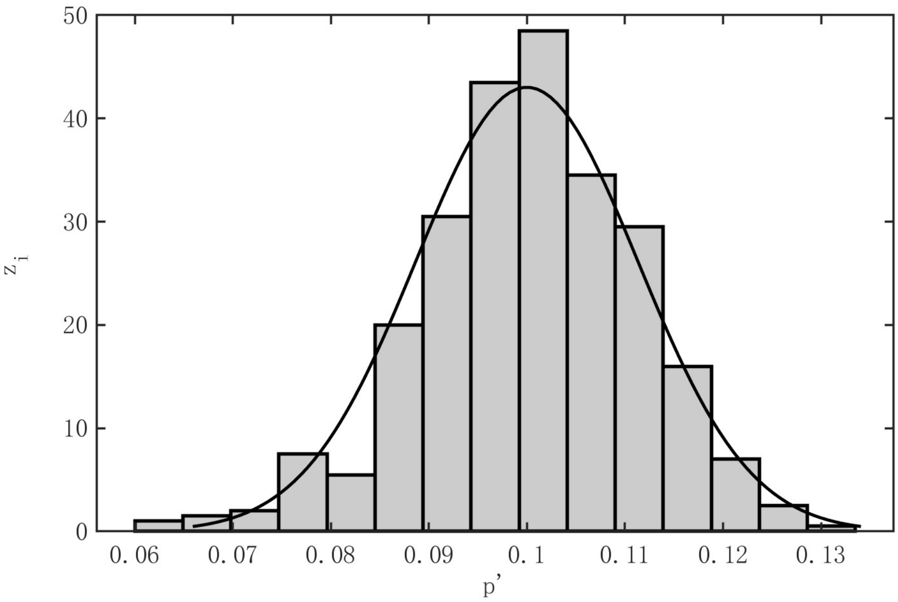

图 **8**　蒙特卡罗模拟抽样所得次品率分布

8.2 结合抽样的综合决策方案

由于抽样估计次品率的成本为必要开支，若充分利用抽样获取的信息，对初步发现的不合格成品进行处理，可以进一步提升整体的经济效益。

在实际抽样检测阶段，零件及（半）成品的次品率均为未知，被抽到的所有零件与组装好的成品均会经历检验，此时有关检测的决策变量 t_i,t_{jl},t_m 均为 1，关于成品与半成品拆解的决策变量 s_m，s_{jl} 尚未确定。由于抽样阶段的所有样品均需要检测，此处拆解一个成品带来的收益期望应写为

$$B_{拆} = \left(\sum_{i=1}^{n} \frac{a_i + c_i}{1 - p_i} - e_m \right) s_m \tag{46}$$

其中 $1 - p_i$ 表明只有在检测出来零部件为合格品时才会获益 $\sum_{i=1}^{n} \frac{a_i + c_i}{1 - p_i}$。为了使 $B_{拆}$ 最大化，若 $\sum_{i=1}^{n} \frac{a_i + c_i}{1 - p_i} > e$，应取 s_m=1，否则 s_m=0。

对于不合格半成品，由于抽样估计次品率时参与组装的都是正品零件，将其拆解后可以将正品零件重复利用，故 s'_{jl} 始终为 1，始终选择拆解不合格半成品。

以上内容即为充分利用抽样检测成本提升经济效益的方法。该部分决策只会影响是否拆解在估计次品率时遇到的不合格（半）成品，不会影响问题 2 与问题 3 的结果，故可看作是对实际生产的一个降本增效的建议。

经过对 p_i 的模拟，得到在任何情况下决策变量 s 均等于 1，即选择拆解遇到的所有次品。

由此可确定结合抽样的综合决策方案如图9所示。

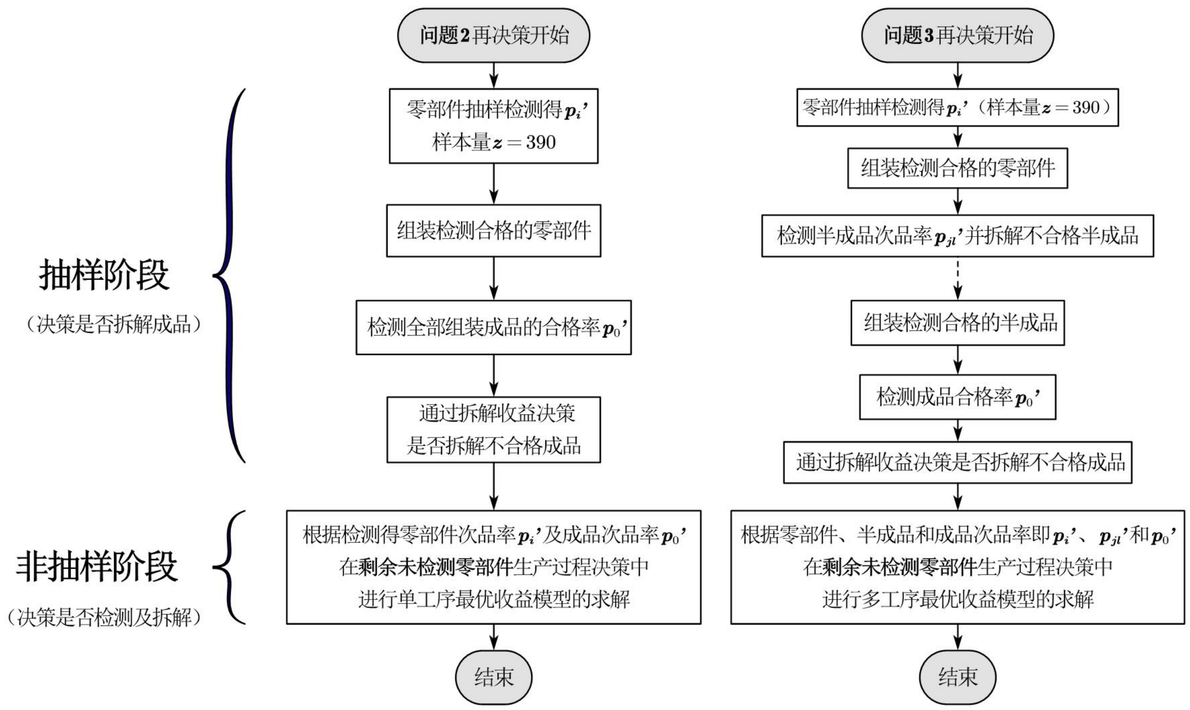

图 **9**　结合抽样的综合决策方案

8.3 问题 2 的重新计算

针对问题 2 中的 6 种情形，通过计算机模拟抽样得到 500 组不同的 p_1', p_2', p_0' 次品率组合，其中每个次品率都分别服从以真实值为均值的正态分布。将 p_1', p_2', p_0' 代入问题 2 中的最大收益期望模型计算，得到的理想方案变化如图10所示。

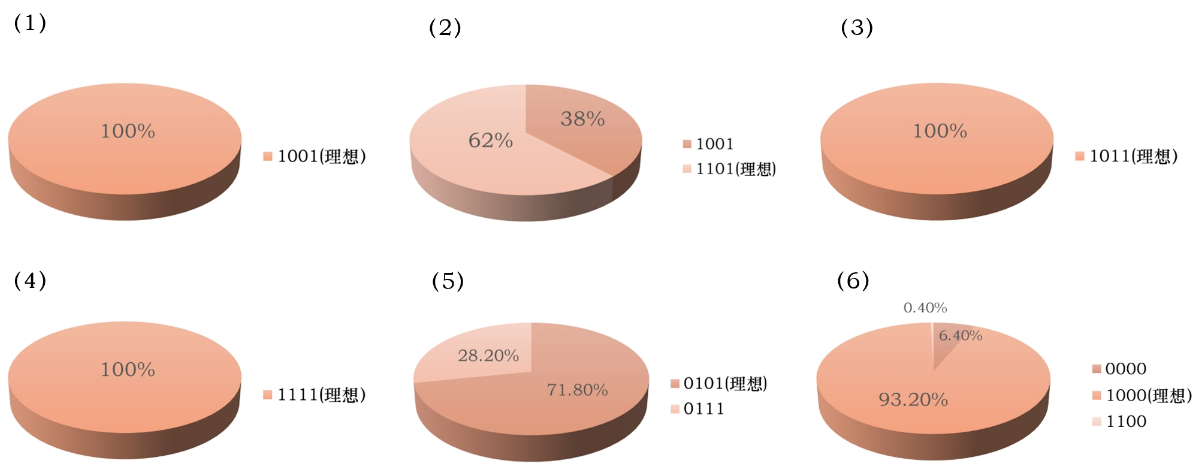

图 **10**　抽样所得次品率计算所得最佳方案分布

从图中可以看出，利用 500 次蒙特卡罗模拟抽样所得的模糊次品率计算，情况 1，情况 3 与情况 4 所得最优决策方案与理想情形完全一致，抽样带来的次品率的浮动对其生产决策无影响，显示出极强的稳定性。其他三种情况中，情况 6 的计算结果最为理想，理想方案占总方案数的 93.2%，同样表现出极强的抗干扰性。其中，情况 2 的理想方案占比最低，仅为 62.0%。

从原始数据的角度分析，情况 3 与情况 4 的调换损失最大，故在生产过程中应尽可能检测零件，提升成品的正品率。当次品率出现偏差，该趋势仍不会改变，故情况 3 与情况 4 对受次品率浮动的影响较小。与情况 1 相比，情况 2 的理想次品率更高，总体中次品的个数更多。在取样数相同的情况下，情况 2 更容易抽出次品分布极端的样本，故其结果稳定性小于情况 1 具有合理性。

总体而言，简单随机抽样的结果可以使得我们较大概率地选择最优决策路径，这映证了简单随机抽样方法的有效性。

8.4 问题 3 的重新计算

问题 3 中使用的次品率均为理想值，将模型中的对应部分次品率 p 改为通过抽样得到的次品率 p'，即可将多工序的最优收益模型更新为基于抽样检测的最优收益模型。对 2 道工序，8 个零部件中各零件，半成品与成品的次品率进行蒙特卡罗模拟，将模拟得到的计算结果代入模型。由于此处进行了大量的随机模拟，若采用遗传算法涉及到的变量过于繁杂，故此处考虑直接利用遍历算法进行求解，即可得到 500 组最优方案。

计算结果显示，500 组模拟结果计算得到的实现利益最大化的决策变量组合均为

$$\begin{aligned}&\{t_0,t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_{11},t_{12},t_{13},s_0,s_{11},s_{12},s_{13}\}\\&=\{0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1\}\end{aligned}$$

与问题三的结果相同，这说明问题三的生产情况具有较强的稳定性。

方案对应的最大收益期望 E 有所浮动，在

$$\begin{aligned}&\{p_0,p_1,p_2,p_3,p_4,p_5,p_6,p_7,p_8,p_{11},p_{12},p_{13}\}\\&=\{0.098,0.103,0.100,0.110,0.115,0.103,0.0970.105,0.097,0.104,0.100,0.065\}\end{aligned}$$

时取到最大值为 64.9752 元/件。

九、 模型的检验

9.1 灵敏性分析

在实际生产过程中，往往是通过抽样检测获得次品率的估计值，这个估计值可能会因为抽样的随机性而浮动。为了研究这种随机的浮动对于决策的影响，我们以问题 2 中

情况 2 的零件 1 次品率 p_1 为研究对象，使其从初始值 0.20 依次增加 0.01，零件 2 与成品的次品率不变，计算此时的最佳决策变量组合，结果如表4所示。

表 **4**　问题 **2** 情况 **2** 中零件 **1** 的次品率影响

零件 1 次品率 p_1	0.20	0.21	0.22	0.23	0.24	0.25	0.26
决策变量组合 $\{t_1, t_2, t_0, s_1\}$	1101	1101	1101	1001	1001	1001	1001

从表中可以看出，当 p_1=20%-22% 时，计算所得最佳方案均为理想方案 1101。当 p_1 达到 23% 及以上，最佳决策变量组合变为 1001。可见，在该生产条件下，只需要其中一个零件的次品率改变 3%，就会导致最佳理想组合改变，这充分说明我们的模型具有高度的精度与灵敏度。

9.2 稳定性分析

在进行蒙特卡罗随机模拟时，采用的是每次模拟抽样 500 次。为了验证模拟次数是否足够，我们将模拟次数改为不同的值，分别计算问题 2 中情况 2 的理想方案占比，结果如图11所示。

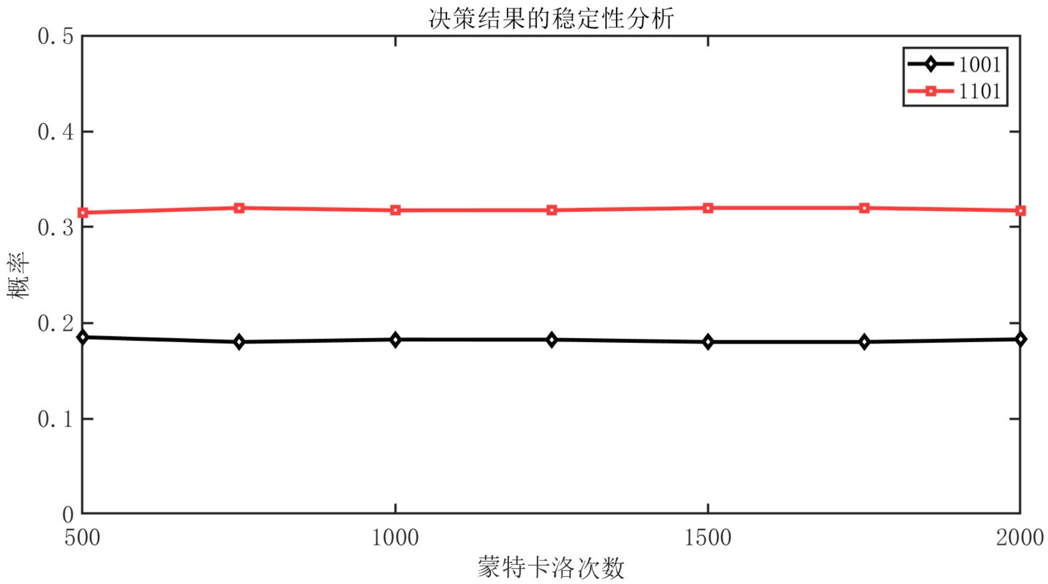

图 **11**　取不同模拟次数时的计算结果

从图中可以看出，在模拟次数取 500，1000，1500，2000 时，计算得到的理想方案占比基本不变，最大浮动为 1%。这充分说明 500 次的模拟已经接近饱和，能够较为准确地反映真实情况，我们的模型具有较好的稳定性。

十、 模型的评价与推广

10.1 模型的优点

（1）问题一的模型可以得到抽样次数的解析解，实际生产中可根据不同的精度要求得到对应的抽样方案。

（2）所有生产相关变量都参与建立决策模型，在实际使用过程中可以根据需要灵活调整参数，模型普适性强。

（3）问题三的通用模型对于各类商品的生产决策均适用，且配备了智能算法提升计算效率。

10.2 模型的缺点

（1）问题一取样时绝对精度的确定较为主观，取不同精度时会有不同的取样次数。

（2）对生产过程中零部件总数足够大的假设偏理想，没有考虑实际情况中次品率的动态变化。

10.3 模型的改进方向与应用推广

（1）若考虑了生产过程中零部件总数的实时变化，可以根据实际情况制定出实用性更强的决策方案。

（2）若将多轮次的生产作为整体进行考察，或许可以制定综合性更强的决策方案。

参考文献

[1] 李益. 抽样调查中样本量 n 的确定方法 [J]. 市场研究,2005,(12):54-57.

[2] 邢继娟, 葛含益, 卜广志. 仿真模型置信度分析方法研究 [J]. 军事运筹与系统工程,2005,(04):35-39.

[3] 熊立文. 贝叶斯决策理论与归纳逻辑 [J]. 北京师范大学学报 (社会科学版),2005,(02):108-113.

附录 A　支撑材料列表

以下是本论文求解过程中用到的所有代码文件与结果

CUMCM_B_1_n.m

CUMCM_B_2.m

CUMCM_B_3.m

CUMCM_B_3_2_ga.m

CUMCM_B_4_2_1.m

CUMCM_B_4_2_2.m

CUMCM_B_4_2_3.m

CUMCM_B_4_2_4.m

CUMCM_B_4_2_5.m

CUMCM_B_4_2_6.m

CUMCM_B_4_3_2_bianli.m

CUMCM_B_lingminxing_2.m

CUMCM_B_whendingxing_4_2.m

CUMCM_2_draw_6.m

list2_data.xls

result1_n.xls

result2_data.xls

result4_2_1_data.xls

result4_2_2_data.xls

result4_2_2_p_count.xls

result4_2_3_data.xls

result4_2_3_p_count.xls

result4_2_4_data.xls

result4_2_4_p_count.xls

result4_2_5_data.xls

result4_2_5_p_count.xls

result4_2_6_data.xls

result4_2_6_p_count.xls

result4_2_data.xls

附录 B　计算抽样样本数，以及不同置信度的拒绝域

```
function [n1,n2,x1,x2]=CUMCM_B_1_n
p=0.1%标称值

lamda=0.015:0.005:0.05;% 的取值范围,绝对精度
s2=p*(1-p);

u1=norminv(0.95,0,1);% =0.05
u2=norminv(0.9,0,1);% =0.1

n1=u1^2*s2./lamda.^2;
n2=u2^2*s2./lamda.^2;

plot(lamda,n1,'DisplayName',' =0.05')%相对精度
% for i=1:length(n1)
% text(lamda(i),n1(i),['(',num2str(lamda(i)),',',num2str(n1(i)),')']);
% end
hold on
plot(lamda,n2,'DisplayName',' =0.1')

%将结果写入excel
xlswrite('result1_n',[lamda;n1;n2]);


xlabel('绝对精度')
ylabel('抽样次数')
title('z与相对精度关系图')

%由图可以看出 1=0.004, 2=0.003时精度和经济效应最好
n_1=round(n1(3))
n_2=round(n2(3))

%计算各个的拒绝域
u_1=norminv(0.05,0,1);% =0.05
u_2=norminv(0.1,0,1);% =0.1
syms x;
equ=[(x-n_1*p)/sqrt(n_1*p*(1-p))==u_1];
x1=vpasolve(equ,x)
equ=[(x-n_2*p)/sqrt(n_2*p*(1-p))==u_2];
x2=vpasolve(equ,x)
```

附录 C　决策的选择

```
function [idr_best1,result_best]=CUMCM_B_2
%零配件1的基本条件：次品率、购买单价、检测成本
```

```
x1=[0.1 4 2
0.2  4 2
0.1  4 2
0.2  4 1
0.1  4 8
0.05 4 2];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.1 18 3
0.2  18 3
0.1  18 3
0.2  18 1
0.2  18 1
0.05 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.1 6 3 56
0.2  6 3 56
0.1  6 3 56
0.2  6 2 56
0.1  6 2 56
0.05 6 3 56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[6 5
6 5
30 5
30 5
10 5
10 40];

%定义整数规划的变量
t1=[0,1];%配件1是否检测
t2=[0,1];%配件2是否检测
t0=[0,1];%成品是否检测
s1=[0,1];%不合格是否拆解
idr=zeros(16,4,6);
ida=zeros(16,1,6);

result=zeros(6,16);%每个策略的优化函数值
for r=1:6 %六种不同情况
count=0;
for i=1:2 %t1的不同情况
for j=1:2 %t2的不同情况
for k=1:2 %t0的不同情况
for l=1:2 %s1的不同情况
%计数
count=count+1;
idr(count,:,r)=[i,j,k,l];
idr(count,:,r)=idr(count,:,r)-1;
```

```
%p为次品率，q为成品率
p1=x1(r,1);q1=1-x1(r,1);
p2=x2(r,1);q2=1-x2(r,1);
p=good(r,1);q=1-good(r,1);

theta=q1*q2*q/((1-p1*t1(i))*(1-p2*t2(j)));
ida(count,1,r)=theta;%用于存贮theta

%a为配件单价，c为配件检测成本，b为装配成本，d为检测成品成本
a1=x1(r,2);c1=x1(r,3);
a2=x2(r,2);c2=x2(r,3);
b=good(r,2);d=good(r,3);
%w为市场售价，h为调换损失，e为拆解费用
w=good(r,4);
h=n_good(r,1);
e=n_good(r,2);

%零件成本
A=a1+a2;
%装配成本
B=b;
%检测零件成本
C=c1*t1(i)+c2*t2(j)+t1(i)*(a1+c1)*p1/(1-p1)+t2(j)*(a2+c2)*p2/(1-p2);
%拆解收益
D=(A+C-e)*s1(l);
%丢弃收益为0
%成品检测调换费用
E=(1-t0(k))*h;
%成品检测
E1=d*t0(k);
%成品不检测的调换费用
F=(w-A-B-C-w)*(1-t0(k));

%每个策略的收益
result(r,count)=-A-B-C-E1+(1-theta)*(D-E)+theta*w;

end
end
end
end
end

%每个情况的最优策略以及对应值
result_best=zeros(6,1);
idr_best=zeros(6,1);
result
idr
```

```
[result_best,idr_best]=max(result,[],2)
for m=1:6
idr_best1(m,:)=idr(idr_best(m),:,m);
end
idr_best1

xlswrite('result2_data',result)
```

附录 D 寻找问题三最优决策策略 (遍历算法)

```
function [idr_best,result_best]=CUMCM_B_3
%8个零配件的基本信息
x=[.10 2 1
.10 8 1
.10 12 2
.10 2 1
.10 8 1
.10 12 2
.10 8 1
.10 12 2];
%3个半成品的基本信息
y=[.10 8 4 6
.10 8 4 6
.10 8 4 6];
%最终成品的基本信息
z=[.10 8 6 10 200 40];

%配件是否检测
t1=[0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1];
%半成品是否检测
t2=[0,1
0,1
0,1];
%半成品是否拆卸
s2=[0,1
0,1
0,1];
%成品是否检测
```

```
t0=[0,1];
%成品是否拆卸
s0=[0,1];

%储存结果的数组
result=zeros(2^16,1);
idr=zeros(2^16,16);
idx=zeros(1,8);
idy=zeros(1,3);
idz=zeros(1,2);
pp1=zeros(1,8);
count=0;

%次品率
p1=x(1,1);p2=x(2,1);p3=x(3,1);p4=x(4,1);
p5=x(5,1);p6=x(6,1);p7=x(7,1);p8=x(8,1);
p9=y(1,1);p10=y(2,1);p11=y(3,1);p=z(1,1);
pp1=[p1,p2,p3,p4,p5,p6,p7,p8];
%成功率
q1=1-x(1,1);q2=1-x(2,1);q3=1-x(3,1);q4=1-x(4,1);
q5=1-x(5,1);q6=1-x(6,1);q7=1-x(7,1);q8=1-x(8,1);
q9=1-y(1,1);q10=1-y(2,1);q11=1-y(3,1);q=1-z(1,1);

A=sum(x(:,2));%A为配件需要的总成本
B=sum(y(:,2))+z(2);%第一道工序半成品与成品的装配和
e=z(4);%成品拆解费用
h=z(6);%成品调换损失
w=z(5);%成品市场售价

%i代表检测，j代表拆卸
for i0=1:length(t0)
for j0=1:length(s0)
for i1=1:2
for i2=1:2
for i3=1:2
for i4=1:2
for i5=1:2
for i6=1:2
for i7=1:2
for i8=1:2
for i9=1:2
for i10=1:2
for i11=1:2
for j1=1:2
for j2=1:2
for j3=1:2
%储存结果的数组
```

```
%        result=zeros(2^16,1);
%        idr=zeros(2^16,16);
idx=zeros(1,8);
idy=zeros(1,3);
idz=zeros(1,2);
pp1=zeros(1,8);

count=count+1;
i0=i0-1;i1=i1-1;i2=i2-1;i3=i3-1;
i4=i4-1;i5=i5-1;i6=i6-1;i7=i7-1;
i8=i8-1;i9=i9-1;i10=i10-1;i11=i11-1;
j0=j0-1;j1=j1-1;j2=j2-1;j3=j3-1;
idr(count,:)=[i0,j0,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,j1,j2,j3];
idx=idr(count,3:10);
idy=idr(count,11:end);
idz=idr(count,1:2);
idx=idx';
idy=idy';
idz=idz';

%贝叶斯概率
theta1=q1*q2*q3*q9/((1-p1*i1)*(1-p2*i2)*(1-p3*i3));
theta2=q4*q5*q6*q10/((1-p4*i4)*(1-p5*i5)*(1-p6*i6));
theta3=q7*q8*q11/((1-p7*i7)*(1-p8*i8));
theta=q1*q2*q3*q4*q5*q6*q7*q8*q9*q10*q11*q/((1-p1*i1)*(1-p2*i2)*(1-p3*i3)*(1-p4*i4)*(1-p5*i5)*(1-p6*i6)*(1

%设立一个存放数组
pp1=pp1./(1-pp1);%存放检测的概率
pp1=pp1';
%8配件检测的总成本
C=sum(x(:,3).*idx+idx.*pp1.*(x(:,2)+x(:,3)));
%半成品拆解成本
D1=sum(y(1,2)+y(1,4))*i9*j1;%i代表检测，j代表拆卸
D2=sum(y(2,2)+y(2,4))*i10*j2;
D3=sum(y(3,2)+y(3,4))*i11*j3;
%半成品丢弃成本
G1=(sum(x(1:3,2)+x(1:3,3).*idx(1:3)+idx(1:3).*pp1(1:3).*(x(1:3,2)+x(1:3,3)))+y(1,2))*i9*(1-j1);
G2=(sum(x(4:6,2)+x(4:6,3).*idx(4:6)+idx(4:6).*pp1(4:6).*(x(4:6,2)+x(4:6,3)))+y(2,2))*i10*(1-j2);
G3=(sum(x(7:8,2)+x(7:8,3).*idx(7:8)+idx(7:8).*pp1(7:8).*(x(7:8,2)+x(7:8,3)))+y(3,2))*i11*(1-j3);
%成品拆解收益
D0=(A+C-z(4))*j0;
%成品调换费用
E=(1-i0)*z(6);
%成品检验费用
E1=sum(y(:,3).*idy(1:3))+z(3)*idz(1);
```

```
%计算策略最优函数值
result(count)=-A-B-C-E1-(1-theta1)*(D1+G1)/theta1-(1-theta2)*(D2+G2)/theta2-(1-theta3)*(D3+G3)/theta3+(1-t
i0=i0+1;i1=i1+1;i2=i2+1;i3=i3+1;
i4=i4+1;i5=i5+1;i6=i6+1;i7=i7+1;
i8=i8+1;i9=i9+1;i10=i10+1;i11=i11+1;
j0=j0+1;j1=j1+1;j2=j2+1;j3=j3+1;
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end

[result_best,idrr_best]=max(result);
idr_best=idr(idrr_best,:)
result_best
```

附录 E　寻找问题三最优决策策略 (遗传算法)

```
function [solution,objectiveValue]=CUMCM_B_3_2_ga
% Create optimization variables
i053 = optimvar("i0","Type","integer","LowerBound",0,"UpperBound",1);
i161 = optimvar("i1","Type","integer","LowerBound",0,"UpperBound",1);
i253 = optimvar("i2","Type","integer","LowerBound",0,"UpperBound",1);
i353 = optimvar("i3","Type","integer","LowerBound",0,"UpperBound",1);
i453 = optimvar("i4","Type","integer","LowerBound",0,"UpperBound",1);
i553 = optimvar("i5","Type","integer","LowerBound",0,"UpperBound",1);
i653 = optimvar("i6","Type","integer","LowerBound",0,"UpperBound",1);
i753 = optimvar("i7","Type","integer","LowerBound",0,"UpperBound",1);
i853 = optimvar("i8","Type","integer","LowerBound",0,"UpperBound",1);
i953 = optimvar("i9","Type","integer","LowerBound",0,"UpperBound",1);
i1053 = optimvar("i10","Type","integer","LowerBound",0,"UpperBound",1);
i1153 = optimvar("i11","Type","integer","LowerBound",0,"UpperBound",1);
j053 = optimvar("j0","Type","integer","LowerBound",0,"UpperBound",1);
j153 = optimvar("j1","Type","integer","LowerBound",0,"UpperBound",1);
```

```
j253 = optimvar("j2","Type","integer","LowerBound",0,"UpperBound",1);
j353 = optimvar("j3","Type","integer","LowerBound",0,"UpperBound",1);

% Set initial starting point for the solver
initialPoint53.i0 = zeros(size(i053));
initialPoint53.i1 = zeros(size(i161));
initialPoint53.i2 = zeros(size(i253));
initialPoint53.i3 = zeros(size(i353));
initialPoint53.i4 = zeros(size(i453));
initialPoint53.i5 = zeros(size(i553));
initialPoint53.i6 = zeros(size(i653));
initialPoint53.i7 = zeros(size(i753));
initialPoint53.i8 = zeros(size(i853));
initialPoint53.i9 = zeros(size(i953));
initialPoint53.i10 = zeros(size(i1053));
initialPoint53.i11 = zeros(size(i1153));
initialPoint53.j0 = zeros(size(j053));
initialPoint53.j1 = zeros(size(j153));
initialPoint53.j2 = zeros(size(j253));
initialPoint53.j3 = zeros(size(j353));

% Create problem
problem = optimproblem("ObjectiveSense","Maximize");

% Define problem objective
problem.Objective =
    fcn2optimexpr(@CUMCM_B_3_ga,i053,i161,i253,i353,i453,i553,i653,i753,i853,...
i953,i1053,i1153,j053,j153,j253,j353);

% 设置非默认求解器选项
options4 = optimoptions("ga","PlotFcn","gaplotbestindiv");

% Display problem information
show(problem);

% Solve problem
[solution,objectiveValue,reasonSolverStopped] = solve(problem,initialPoint53,...
"Solver","ga","Options",options4);

% Display results
solution
reasonSolverStopped
objectiveValue

% 清除变量
clearvars i053 i161 i253 i353 i453 i553 i653 i753 i853 i953 i1053 i1153 j053 j153 j253
    j353 initialPoint53...
```

```
options4 reasonSolverStopped objectiveValue



function result=CUMCM_B_3_ga(i0,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,j0,j1,j2,j3)


%8个零配件的基本信息
x=[.10 2 1
.10 8 1
.10 12 2
.10 2 1
.10 8 1
.10 12 2
.10 8 1
.10 12 2];
%3个半成品的基本信息
y=[.10 8 4 6
.10 8 4 6
.10 8 4 6];
%最终成品的基本信息
z=[.10 8 6 10 200 40];

%配件是否检测
t1=[0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1];
%半成品是否检测
t2=[0,1
0,1
0,1];
%半成品是否拆卸
s2=[0,1
0,1
0,1];
%成品是否检测
t0=[0,1];
%成品是否拆卸
s0=[0,1];



%次品率
```

```
p1=x(1,1);p2=x(2,1);p3=x(3,1);p4=x(4,1);
p5=x(5,1);p6=x(6,1);p7=x(7,1);p8=x(8,1);
p9=y(1,1);p10=y(2,1);p11=y(3,1);p=z(1,1);
pp1=[p1,p2,p3,p4,p5,p6,p7,p8];
%成功率
q1=1-x(1,1);q2=1-x(2,1);q3=1-x(3,1);q4=1-x(4,1);
q5=1-x(5,1);q6=1-x(6,1);q7=1-x(7,1);q8=1-x(8,1);
q9=1-y(1,1);q10=1-y(2,1);q11=1-y(3,1);q=1-z(1,1);

A=sum(x(:,2));%A为配件需要的总成本
B=sum(y(:,2))+z(2);%第一道工序半成品与成品的装配和
e=z(4);%成品拆解费用
h=z(6);%成品调换损失
w=z(5);%成品市场售价


%储存结果的数组
% result=zeros(2^16,1);
idr=zeros(1,16);
idx=zeros(1,8);
idy=zeros(1,3);
idz=zeros(1,2);
pp1=zeros(1,8);
count=1;
%         count=count+1;
%         i0=i0-1;i1=i1-1;i2=i2-1;i3=i3-1;
%         i4=i4-1;i5=i5-1;i6=i6-1;i7=i7-1;
%         i8=i8-1;i9=i9-1;i10=i10-1;i11=i11-1;
%         j0=j0-1;j1=j1-1;j2=j2-1;j3=j3-1;
idr(count,:)=[i0,j0,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,j1,j2,j3];
idx=idr(count,3:10);
idy=idr(count,11:end);
idz=idr(count,1:2);
idx=idx';
idy=idy';
idz=idz';

%贝叶斯概率
theta1=q1*q2*q3*q9/((1-p1*i1)*(1-p2*i2)*(1-p3*i3));
theta2=q4*q5*q6*q10/((1-p4*i4)*(1-p5*i5)*(1-p6*i6));
theta3=q7*q8*q11/((1-p7*i7)*(1-p8*i8));
theta=q1*q2*q3*q4*q5*q6*q7*q8*q9*q10*q11*q/((1-p1*i1)*(1-p2*i2)*(1-p3*i3)*(1-p4*i4)*(1-p5*i5)*(1-p6*i6)*(1

%设立一个存放数组
pp1=pp1./(1-pp1);%存放检测的概率
pp1=pp1';
%8配件检测的总成本
```

```
C=sum(x(:,3).*idx+idx.*pp1.*(x(:,2)+x(:,3)));
%半成品拆解成本
D1=sum(y(1,2)+y(1,4))*i9*j1;%i代表检测，j代表拆卸
D2=sum(y(2,2)+y(2,4))*i10*j2;
D3=sum(y(3,2)+y(3,4))*i11*j3;
%半成品丢弃成本
G1=(sum(x(1:3,2)+x(1:3,3).*idx(1:3)+idx(1:3).*pp1(1:3).*(x(1:3,2)+x(1:3,3)))+y(1,2))*i9*(1-j1);
G2=(sum(x(4:6,2)+x(4:6,3).*idx(4:6)+idx(4:6).*pp1(4:6).*(x(4:6,2)+x(4:6,3)))+y(2,2))*i10*(1-j2);
G3=(sum(x(7:8,2)+x(7:8,3).*idx(7:8)+idx(7:8).*pp1(7:8).*(x(7:8,2)+x(7:8,3)))+y(3,2))*i11*(1-j3);
%成品拆解收益
D0=(A+C-z(4))*j0;
%成品调换费用
E=(1-i0)*z(6);
%成品检验费用
E1=sum(y(:,3).*idy(1:3))+z(3)*idz(1);

result=-A-B-C-E1-(1-theta1)*(D1+G1)/theta1-(1-theta2)*(D2+G2)/theta2-(1-theta3)*(D3+G3)/theta3+(1-theta)*(

end
end
```

附录 F 利用第一问的抽样求解第四问中的第二问中的第一种情况

```
function [count_idr,S1]=CUMCM_B_4_2_1
%问题二的第1种情况
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.1 4 2];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.1 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.1 6 3 56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[6 5];

%定义整数规划的变量
t1=[0,1];%配件1是否检测
t2=[0,1];%配件2是否检测
t0=[0,1];%成品是否检测
s1=[0,1];%不合格是否拆解



%进行蒙特卡洛随机，减少抽样随机性
M=500;
result_best=zeros(M,1);%每个策略的最好的优化函数值
idr_best=zeros(M,4);
```

```
P_m=zeros(M,3);
S1=zeros(M,1);
n=390;%之前第一问的结果,抽样样本数
N=1000;%随机的总数

z1=zeros(1,N);
z2=zeros(1,N);
idn1=zeros(1,N*x1(1));
idn2=zeros(1,N*x2(1));
idn1=randperm(N,N*x1(1));%抽取次品的索引
idn2=randperm(N,N*x2(1));%抽取次品的索引
z1(idn1)=1;%把次品的值设成1
z2(idn2)=1;%把次品的值设成1
for m=1:M
% m=1;
%零配件1的基本条件: 次品率、购买单价、检测成本
x1=[0.1 4 2];
%零配件2的基本条件: 次品率、购买单价、检测成本
x2=[0.1 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.1 6 3 56];
%不合格成品的基本条件: 调换损失, 拆解费用
n_good=[6 5];

%提前开数组, 提高速度
%     z1=zeros(1,N);
%     z2=zeros(1,N);
%     idn1=zeros(1,N*x1(1));
%     idn2=zeros(1,N*x2(1));
idn0_1=zeros(1,n);
idn0_2=zeros(1,n);
z1_n=zeros(1,n);
z2_n=zeros(1,n);

%     idn1=randperm(N,N*x1(1));%抽取次品的索引
%     idn2=randperm(N,N*x2(1));%抽取次品的索引
%     z1(idn1)=1;%把次品的值设成1
%     z2(idn2)=1;%把次品的值设成1

idn0_1=randperm(N,n);%抽取随机样本的索引
idn0_2=randperm(N,n);%抽取随机样本的索引
z1_n=z1(idn0_1);%配件1抽取的随机样本
z2_n=z2(idn0_2);%配件2抽取的随机样本
p_1=sum(z1_n)/length(z1_n);%配件1的样品次品率
p_2=sum(z2_n)/length(z2_n);%配件2的样品次品率
z0=zeros(1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));
z0=binornd(1,good(1),1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));%样品生成的成品
```

```
p_0=sum(z0)/length(z0);%样本成品的次品率
%替换掉原来的次品率
x1(1)=p_1;
x2(1)=p_2;
good(1)=p_0;
P_m(m,:)=[p_1,p_2,p_0];
%对抽样时成本进行最优决策，选择其是否拆卸，因为t1，t2，t0均为0
A0=x1(2)/p_1+x2(2)/p_2;
C0=x1(3)/p_1+x2(3)/p_2;
E0=n_good(2);
if (A0+C0-E0)>0
S1_0=1;
else
S1_0=0;
end
S1(m)=S1_0;
%    P_mm=P_m(m,:);

result=zeros(1,16);%每个器件的每个策略函数值
idr=zeros(16,4);
count=0;
for i=1:2 %t1的不同情况
for j=1:2 %t2的不同情况
for k=1:2 %t0的不同情况
for l=1:2 %s1的不同情况
%计数
count=count+1;
idr(count,:)=[i,j,k,l];
%p为次品率，q为成品率
p1=x1(1);q1=1-x1(1);
p2=x2(1);q2=1-x2(1);
p=good(1);q=1-good(1);

theta=q1*q2*q/((1-p1*t1(i))*(1-p2*t2(j)));
%                  ida(count,1,r)=theta;%用于存贮theta

%a为配件单价，c为配件检测成本，b为装配成本，d为检测成品成本
a1=x1(2);c1=x1(3);
a2=x2(2);c2=x2(3);
b=good(2);d=good(3);
%w为市场售价，h为调换损失，e为拆解费用
w=good(4);
h=n_good(1);
e=n_good(2);

%零件成本
A=a1+a2;
```

```
%装配成本
B=b;
%检测零件成本
C=c1*t1(i)+c2*t2(j)+t1(i)*(a1+c1)*p1/(1-p1)+t2(j)*(a2+c2)*p2/(1-p2);
%拆解收益
D=(A+C-e)*s1(l);
%丢弃收益为0
%成品检测调换费用
E=(1-t0(k))*h;
%成品检测
E1=d*t0(k);
%成品不检测的调换费用
F=(w-A-B-C-w)*(1-t0(k));

%每个策略的收益
result(count)=-A-B-C-E1+(1-theta)*(D-E)+theta*w;
%
    result(r,count)=-a1-a2-c1*t1(i)-c2*t2(j)-b-d*t0(k)-(1-theta)*((e-a1-a2-t1(i)*c1-t2(j)*c2)*s1(l)+(1-t0(
end
end
end
end
%每个情况的最优策略以及对应值
idrr_best=zeros(1,1);
[result_best(m),idrr_best]=max(result,[],2);
idr_best(m,:)=idr(idrr_best,:)-1;%得到每个器件得到的最佳决策


end

RR=[P_m,result_best,idr_best];
%写入excel中
xlswrite('result4_2_1_data.xls',RR)

%第二问真实的策略
idr_right =[1 0 0 1
1 1 0 1
1 0 1 1
1 1 1 1
0 1 0 1
1 0 0 0];
idr1_right=idr_right(1,:);


%找出idr_best中有多少个不同的值
[diff_idr,~,~]=unique(idr_best,'rows')
d_l=size(diff_idr,1);
```

```
% count_right=zeros(1,5);

%count_bad用于计数
count_idr=zeros(d_l,5);%
if d_l>1
for a=1:d_l
if isequal(idr1_right,diff_idr(a,:))6
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
else
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
end
end
else
count_idr=[size(idr_best,1),diff_idr];
end
count_idr
S1
%    histogram(P_m(:,1),15)
%    hold on
histfit(P_m(:,1),15)
axis square tight
alpha(0.3)
hold on
% normplot(P_m(:,1))
% kstest(P_m(:,1))
```

附录 G　利用第一问的抽样求解第四问中的第二问中的第 2 种情况

```
function [count_idr,S1]=CUMCM_B_4_2_2
%问题二的第2种情况
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.2 4 2];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.2 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.2 6 3 56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[6 5];
```

```
%定义整数规划的变量
t1=[0,1];%配件1是否检测
t2=[0,1];%配件2是否检测
t0=[0,1];%成品是否检测
s1=[0,1];%不合格是否拆解



%进行蒙特卡洛随机，减少抽样随机性
M=500;
result_best=zeros(M,1);%每个策略的最好的优化函数值
idr_best=zeros(M,4);
P_m=zeros(M,3);
S1=zeros(M,1);
n=390;%之前第一问的结果,抽样样本数
N=1000;%随机的总数

z1=zeros(1,N);
z2=zeros(1,N);
idn1=zeros(1,N*x1(1));
idn2=zeros(1,N*x2(1));
idn1=randperm(N,N*x1(1));%抽取次品的索引
idn2=randperm(N,N*x2(1));%抽取次品的索引
z1(idn1)=1;%把次品的值设成1
z2(idn2)=1;%把次品的值设成1
for m=1:M
% m=1;
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.2 4 2];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.2 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.2 6 3 56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[6 5];

%提前开数组，提高速度
%     z1=zeros(1,N);
%     z2=zeros(1,N);
%     idn1=zeros(1,N*x1(1));
%     idn2=zeros(1,N*x2(1));
idn0_1=zeros(1,n);
idn0_2=zeros(1,n);
z1_n=zeros(1,n);
z2_n=zeros(1,n);

%     idn1=randperm(N,N*x1(1));%抽取次品的索引
```

```
%     idn2=randperm(N,N*x2(1));%抽取次品的索引
%     z1(idn1)=1;%把次品的值设成1
%     z2(idn2)=1;%把次品的值设成1

idn0_1=randperm(N,n);%抽取随机样本的索引
idn0_2=randperm(N,n);%抽取随机样本的索引
z1_n=z1(idn0_1);%配件1抽取的随机样本
z2_n=z2(idn0_2);%配件2抽取的随机样本
p_1=sum(z1_n)/length(z1_n);%配件1的样品次品率
p_2=sum(z2_n)/length(z2_n);%配件2的样品次品率
z0=zeros(1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));
z0=binornd(1,good(1),1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));%样品生成的成品
p_0=sum(z0)/length(z0);%样本成品的次品率
%替换掉原来的次品率
x1(1)=p_1;
x2(1)=p_2;
good(1)=p_0;
P_m(m,:)=[p_1,p_2,p_0];
%对抽样时成本进行最优决策，选择其是否拆卸，因为t1，t2，t0均为0
A0=x1(2)/p_1+x2(2)/p_2;
C0=x1(3)/p_1+x2(3)/p_2;
E0=n_good(2);
if (A0+C0-E0)>0
S1_0=1;
else
S1_0=0;
end
S1(m)=S1_0;
%     P_mm=P_m(m,:);

result=zeros(1,16);%每个器件的每个策略函数值
idr=zeros(16,4);
count=0;
for i=1:2 %t1的不同情况
for j=1:2 %t2的不同情况
for k=1:2 %t0的不同情况
for l=1:2 %s1的不同情况
%计数
count=count+1;
idr(count,:)=[i,j,k,l];
%p为次品率，q为成品率
p1=x1(1);q1=1-x1(1);
p2=x2(1);q2=1-x2(1);
p=good(1);q=1-good(1);

theta=q1*q2*q/((1-p1*t1(i))*(1-p2*t2(j)));
%                  ida(count,1,r)=theta;%用于存贮theta
```

```
%a为配件单价，c为配件检测成本，b为装配成本，d为检测成品成本
a1=x1(2);c1=x1(3);
a2=x2(2);c2=x2(3);
b=good(2);d=good(3);
%w为市场售价，h为调换损失，e为拆解费用
w=good(4);
h=n_good(1);
e=n_good(2);

%零件成本
A=a1+a2;
%装配成本
B=b;
%检测零件成本
C=c1*t1(i)+c2*t2(j)+t1(i)*(a1+c1)*p1/(1-p1)+t2(j)*(a2+c2)*p2/(1-p2);
%拆解收益
D=(A+C-e)*s1(l);
%丢弃收益为0
%成品检测调换费用
E=(1-t0(k))*h;
%成品检测
E1=d*t0(k);
%成品不检测的调换费用
F=(w-A-B-C-w)*(1-t0(k));

%每个策略的收益
result(count)=-A-B-C-E1+(1-theta)*(D-E)+theta*w;
%
    result(r,count)=-a1-a2-c1*t1(i)-c2*t2(j)-b-d*t0(k)-(1-theta)*((e-a1-a2-t1(i)*c1-t2(j)*c2)*s1(l)+(1-t0(
end
end
end
end
%每个情况的最优策略以及对应值
idrr_best=zeros(1,1);
[result_best(m),idrr_best]=max(result,[],2);
idr_best(m,:)=idr(idrr_best,:)-1;%得到每个器件得到的最佳决策
%        result_best___=result_best(m);
%        idr_best___=idr_best(m,:);

end

RR=[P_m,result_best,idr_best];
xlswrite('result4_2_2_data.xls',RR)

%第二问真实的策略
```

```
idr_right =[1 0 0 1
1 1 0 1
1 0 1 1
1 1 1 1
0 1 0 1
1 0 0 0];
idr2_right=idr_right(2,:);


%找出idr_best中有多少个不同的值
[diff_idr,~,~]=unique(idr_best,'rows')
d_l=size(diff_idr,1);
% count_right=zeros(1,5);

%count_bad用于计数
count_idr=zeros(d_l,5);%
diff_p=zeros(M,3,d_l);
if d_l>1
for a=1:d_l
if isequal(idr2_right,diff_idr(a,:))
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
else
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
diff_p(idx~=0,:,a)=P_m(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
end
end
else
count_idr=[size(idr_best,1),diff_idr];
end
count_idr

p_count=count_idr(:,1)./M
diff_p(:,:,1);
S1

xlswrite('result4_2_2_p_count.xls',[p_count,count_idr])
% xlswrite('result4_2_2_p_count.xls',[same_p(:,:,)])
```

附录 H　利用第一问的抽样求解第四问中的第二问中的第 3 种情况

```
function [count_idr,S1]=CUMCM_B_4_2_3
%问题二的第3种情况
%零配件1的基本条件: 次品率、购买单价、检测成本
x1=[0.1 4 2];
%零配件2的基本条件: 次品率、购买单价、检测成本
x2=[0.1 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.1 6 3 56];
%不合格成品的基本条件: 调换损失, 拆解费用
n_good=[30 5];

%定义整数规划的变量
t1=[0,1];%配件1是否检测
t2=[0,1];%配件2是否检测
t0=[0,1];%成品是否检测
s1=[0,1];%不合格是否拆解




%进行蒙特卡洛随机, 减少抽样随机性
M=500;
result_best=zeros(M,1);%每个策略的最好的优化函数值
idr_best=zeros(M,4);
P_m=zeros(M,3);
S1=zeros(M,1);
n=390;%之前第一问的结果,抽样样本数
N=1000;%随机的总数

z1=zeros(1,N);
z2=zeros(1,N);
idn1=zeros(1,N*x1(1));
idn2=zeros(1,N*x2(1));
idn1=randperm(N,N*x1(1));%抽取次品的索引
idn2=randperm(N,N*x2(1));%抽取次品的索引
z1(idn1)=1;%把次品的值设成1
z2(idn2)=1;%把次品的值设成1
for m=1:M
% m=1;
%零配件1的基本条件: 次品率、购买单价、检测成本
x1=[0.1 4 2];
%零配件2的基本条件: 次品率、购买单价、检测成本
x2=[0.1 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.1 6 3 56];
```

```
%不合格成品的基本条件：调换损失，拆解费用
n_good=[30 5];

%提前开数组，提高速度
%    z1=zeros(1,N);
%    z2=zeros(1,N);
%    idn1=zeros(1,N*x1(1));
%    idn2=zeros(1,N*x2(1));
idn0_1=zeros(1,n);
idn0_2=zeros(1,n);
z1_n=zeros(1,n);
z2_n=zeros(1,n);

%    idn1=randperm(N,N*x1(1));%抽取次品的索引
%    idn2=randperm(N,N*x2(1));%抽取次品的索引
%    z1(idn1)=1;%把次品的值设成1
%    z2(idn2)=1;%把次品的值设成1

idn0_1=randperm(N,n);%抽取随机样本的索引
idn0_2=randperm(N,n);%抽取随机样本的索引
z1_n=z1(idn0_1);%配件1抽取的随机样本
z2_n=z2(idn0_2);%配件2抽取的随机样本
p_1=sum(z1_n)/length(z1_n);%配件1的样品次品率
p_2=sum(z2_n)/length(z2_n);%配件2的样品次品率
z0=zeros(1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));
z0=binornd(1,good(1),1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));%样品生成的成品
p_0=sum(z0)/length(z0);%样本成品的次品率
%替换掉原来的次品率
x1(1)=p_1;
x2(1)=p_2;
good(1)=p_0;
P_m(m,:)=[p_1,p_2,p_0];
%对抽样时成本进行最优决策，选择其是否拆卸，因为t1，t2，t0均为0
A0=x1(2)/p_1+x2(2)/p_2;
C0=x1(3)/p_1+x2(3)/p_2;
E0=n_good(2);
if (A0+C0-E0)>0
S1_0=1;
else
S1_0=0;
end
S1(m)=S1_0;
%    P_mm=P_m(m,:);

result=zeros(1,16);%每个器件的每个策略函数值
idr=zeros(16,4);
count=0;
```

```
for i=1:2 %t1的不同情况
for j=1:2 %t2的不同情况
for k=1:2 %t0的不同情况
for l=1:2 %s1的不同情况
%计数
count=count+1;
idr(count,:)=[i,j,k,l];
%p为次品率，q为成品率
p1=x1(1);q1=1-x1(1);
p2=x2(1);q2=1-x2(1);
p=good(1);q=1-good(1);

theta=q1*q2*q/((1-p1*t1(i))*(1-p2*t2(j)));
%                  ida(count,1,r)=theta;%用于存贮theta

%a为配件单价，c为配件检测成本，b为装配成本，d为检测成品成本
a1=x1(2);c1=x1(3);
a2=x2(2);c2=x2(3);
b=good(2);d=good(3);
%w为市场售价，h为调换损失，e为拆解费用
w=good(4);
h=n_good(1);
e=n_good(2);

%零件成本
A=a1+a2;
%装配成本
B=b;
%检测零件成本
C=c1*t1(i)+c2*t2(j)+t1(i)*(a1+c1)*p1/(1-p1)+t2(j)*(a2+c2)*p2/(1-p2);
%拆解收益
D=(A+C-e)*s1(l);
%丢弃收益为0
%成品检测调换费用
E=(1-t0(k))*h;
%成品检测
E1=d*t0(k);
%成品不检测的调换费用
F=(w-A-B-C-w)*(1-t0(k));

%每个策略的收益
result(count)=-A-B-C-E1+(1-theta)*(D-E)+theta*w;
%
    result(r,count)=-a1-a2-c1*t1(i)-c2*t2(j)-b-d*t0(k)-(1-theta)*((e-a1-a2-t1(i)*c1-t2(j)*c2)*s1(l)+(1-t0(
end
end
end
```

```
end
%每个情况的最优策略以及对应值
idrr_best=zeros(1,1);
[result_best(m),idrr_best]=max(result,[],2);
idr_best(m,:)=idr(idrr_best,:)-1;%得到每个器件得到的最佳决策
%        result_best___=result_best(m);
%        idr_best___=idr_best(m,:);

end

% P_m
% result_best1=mode(result_best,1)
% idr_best1=mode(idr_best,1)
% P_m_best=mode(P_m,1)
% %
RR=[P_m,result_best,idr_best];
xlswrite('result4_2_3_data.xls',RR)


%第二问真实的策略
idr_right =[1 0 0 1
1 1 0 1
1 0 1 1
1 1 1 1
0 1 0 1
1 0 0 0];
idr3_right=idr_right(3,:);


%找出idr_best中有多少个不同的值
[diff_idr,~,~]=unique(idr_best,'rows')
d_l=size(diff_idr,1);
% count_right=zeros(1,5);

%count_bad用于计数
count_idr=zeros(d_l,5);%
diff_p=zeros(M,3,d_l);
if d_l>1
for a=1:d_l
if isequal(idr3_right,diff_idr(a,:))
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
else
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
```

```
diff_p(idx~=0,:,a)=P_m(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
end
end
else
count_idr=[size(idr_best,1),diff_idr];
end
count_idr
p_count=count_idr(:,1)./M
diff_p;
S1

xlswrite('result4_2_3_p_count.xls',[p_count,count_idr])
```

附录 I 利用第一问的抽样求解第四问中的第二问中的第 4 种情况

```
function [count_idr,S1]=CUMCM_B_4_2_4
%问题二的第4种情况
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.2 4 1];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.2 18 1];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.2 6 2 56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[30 5];

%定义整数规划的变量
t1=[0,1];%配件1是否检测
t2=[0,1];%配件2是否检测
t0=[0,1];%成品是否检测
s1=[0,1];%不合格是否拆解



%进行蒙特卡洛随机，减少抽样随机性
M=500;
result_best=zeros(M,1);%每个策略的最好的优化函数值
idr_best=zeros(M,4);
P_m=zeros(M,3);
S1=zeros(M,1);
n=390;%之前第一问的结果,抽样样本数
N=1000;%随机的总数
```

```
z1=zeros(1,N);
z2=zeros(1,N);
idn1=zeros(1,N*x1(1));
idn2=zeros(1,N*x2(1));
idn1=randperm(N,N*x1(1));%抽取次品的索引
idn2=randperm(N,N*x2(1));%抽取次品的索引
z1(idn1)=1;%把次品的值设成1
z2(idn2)=1;%把次品的值设成1
for m=1:M
% m=1;
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.2 4 1];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.2 18 1];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.2 6 2 56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[30 5];

%提前开数组，提高速度
%     z1=zeros(1,N);
%     z2=zeros(1,N);
%     idn1=zeros(1,N*x1(1));
%     idn2=zeros(1,N*x2(1));
idn0_1=zeros(1,n);
idn0_2=zeros(1,n);
z1_n=zeros(1,n);
z2_n=zeros(1,n);

%     idn1=randperm(N,N*x1(1));%抽取次品的索引
%     idn2=randperm(N,N*x2(1));%抽取次品的索引
%     z1(idn1)=1;%把次品的值设成1
%     z2(idn2)=1;%把次品的值设成1

idn0_1=randperm(N,n);%抽取随机样本的索引
idn0_2=randperm(N,n);%抽取随机样本的索引
z1_n=z1(idn0_1);%配件1抽取的随机样本
z2_n=z2(idn0_2);%配件2抽取的随机样本
p_1=sum(z1_n)/length(z1_n);%配件1的样品次品率
p_2=sum(z2_n)/length(z2_n);%配件2的样品次品率
z0=zeros(1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));
z0=binornd(1,good(1),1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));%样品生成的成品
p_0=sum(z0)/length(z0);%样本成品的次品率
%替换掉原来的次品率
x1(1)=p_1;
x2(1)=p_2;
good(1)=p_0;
```

```
P_m(m,:)=[p_1,p_2,p_0];
%对抽样时成本进行最优决策，选择其是否拆卸，因为t1，t2，t0均为0
A0=x1(2)/p_1+x2(2)/p_2;
C0=x1(3)/p_1+x2(3)/p_2;
E0=n_good(2);
if (A0+C0-E0)>0
S1_0=1;
else
S1_0=0;
end
S1(m)=S1_0;
%     P_mm=P_m(m,:);

result=zeros(1,16);%每个器件的每个策略函数值
idr=zeros(16,4);
count=0;
for i=1:2 %t1的不同情况
for j=1:2 %t2的不同情况
for k=1:2 %t0的不同情况
for l=1:2 %s1的不同情况
%计数
count=count+1;
idr(count,:)=[i,j,k,l];
%p为次品率，q为成品率
p1=x1(1);q1=1-x1(1);
p2=x2(1);q2=1-x2(1);
p=good(1);q=1-good(1);

theta=q1*q2*q/((1-p1*t1(i))*(1-p2*t2(j)));
%                  ida(count,1,r)=theta;%用于存贮theta

%a为配件单价，c为配件检测成本，b为装配成本，d为检测成品成本
a1=x1(2);c1=x1(3);
a2=x2(2);c2=x2(3);
b=good(2);d=good(3);
%w为市场售价，h为调换损失，e为拆解费用
w=good(4);
h=n_good(1);
e=n_good(2);

%零件成本
A=a1+a2;
%装配成本
B=b;
%检测零件成本
C=c1*t1(i)+c2*t2(j)+t1(i)*(a1+c1)*p1/(1-p1)+t2(j)*(a2+c2)*p2/(1-p2);
%拆解收益
```

```
D=(A+C-e)*s1(l);
%丢弃收益为0
%成品检测调换费用
E=(1-t0(k))*h;
%成品检测
E1=d*t0(k);
%成品不检测的调换费用
F=(w-A-B-C-w)*(1-t0(k));

%每个策略的收益
result(count)=-A-B-C-E1+(1-theta)*(D-E)+theta*w;
%
    result(r,count)=-a1-a2-c1*t1(i)-c2*t2(j)-b-d*t0(k)-(1-theta)*((e-a1-a2-t1(i)*c1-t2(j)*c2)*s1(l)+(1-t0(
end
end
end
end
%每个情况的最优策略以及对应值
idrr_best=zeros(1,1);
[result_best(m),idrr_best]=max(result,[],2);
idr_best(m,:)=idr(idrr_best,:)-1;%得到每个器件得到的最佳决策
%        result_best___=result_best(m);
%        idr_best___=idr_best(m,:);

end

% P_m
% result_best1=mode(result_best,1)
% idr_best1=mode(idr_best,1)
% P_m_best=mode(P_m,1)
% %
RR=[P_m,result_best,idr_best];
xlswrite('result4_2_4_data.xls',RR)

%第二问真实的策略
idr_right =[1 0 0 1
1 1 0 1
1 0 1 1
1 1 1 1
0 1 0 1
1 0 0 0];
idr4_right=idr_right(4,:);


%找出idr_best中有多少个不同的值
[diff_idr,~,~]=unique(idr_best,'rows')
d_l=size(diff_idr,1);
```

```
% count_right=zeros(1,5);

%count_bad用于计数
count_idr=zeros(d_l,5);%
diff_p=zeros(M,3,d_l);
if d_l>1
for a=1:d_l
if isequal(idr4_right,diff_idr(a,:))
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
else
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
diff_p(idx~=0,:,a)=P_m(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
end
end
else
count_idr=[size(idr_best,1),diff_idr];
end
count_idr
p_count=count_idr(:,1)./M
diff_p;
S1

xlswrite('result4_2_4_p_count.xls',[p_count,count_idr])
```

附录 J　利用第一问的抽样求解第四问中的第二问中的第 5 种情况

```
function [count_idr,S1]=CUMCM_B_4_2_5
%问题二的第5种情况
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.1 4 8];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.2 18 1];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.1 6 2 56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[10 5];

%定义整数规划的变量
t1=[0,1];%配件1是否检测
```

```
t2=[0,1];%配件2是否检测
t0=[0,1];%成品是否检测
s1=[0,1];%不合格是否拆解



%进行蒙特卡洛随机，减少抽样随机性
M=500;
result_best=zeros(M,1);%每个策略的最好的优化函数值
idr_best=zeros(M,4);
P_m=zeros(M,3);
S1=zeros(M,1);
n=390;%之前第一问的结果,抽样样本数
N=1000;%随机的总数

z1=zeros(1,N);
z2=zeros(1,N);
idn1=zeros(1,N*x1(1));
idn2=zeros(1,N*x2(1));
idn1=randperm(N,N*x1(1));%抽取次品的索引
idn2=randperm(N,N*x2(1));%抽取次品的索引
z1(idn1)=1;%把次品的值设成1
z2(idn2)=1;%把次品的值设成1
for m=1:M
% m=1;
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.1 4 8];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.2 18 1];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.1 6 2 56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[10 5];

%提前开数组，提高速度
%    z1=zeros(1,N);
%    z2=zeros(1,N);
%    idn1=zeros(1,N*x1(1));
%    idn2=zeros(1,N*x2(1));
idn0_1=zeros(1,n);
idn0_2=zeros(1,n);
z1_n=zeros(1,n);
z2_n=zeros(1,n);

%    idn1=randperm(N,N*x1(1));%抽取次品的索引
%    idn2=randperm(N,N*x2(1));%抽取次品的索引
%    z1(idn1)=1;%把次品的值设成1
```

```
%     z2(idn2)=1;%把次品的值设成1

idn0_1=randperm(N,n);%抽取随机样本的索引
idn0_2=randperm(N,n);%抽取随机样本的索引
z1_n=z1(idn0_1);%配件1抽取的随机样本
z2_n=z2(idn0_2);%配件2抽取的随机样本
p_1=sum(z1_n)/length(z1_n);%配件1的样品次品率
p_2=sum(z2_n)/length(z2_n);%配件2的样品次品率
z0=zeros(1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));
z0=binornd(1,good(1),1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));%样品生成的成品
p_0=sum(z0)/length(z0);%样本成品的次品率
%替换掉原来的次品率
x1(1)=p_1;
x2(1)=p_2;
good(1)=p_0;
P_m(m,:)=[p_1,p_2,p_0];
%对抽样时成本进行最优决策，选择其是否拆卸，因为t1，t2，t0均为0
A0=x1(2)/p_1+x2(2)/p_2;
C0=x1(3)/p_1+x2(3)/p_2;
E0=n_good(2);
if (A0+C0-E0)>0
S1_0=1;
else
S1_0=0;
end
S1(m)=S1_0;
%     P_mm=P_m(m,:);

result=zeros(1,16);%每个器件的每个策略函数值
idr=zeros(16,4);
count=0;
for i=1:2 %t1的不同情况
for j=1:2 %t2的不同情况
for k=1:2 %t0的不同情况
for l=1:2 %s1的不同情况
%计数
count=count+1;
idr(count,:)=[i,j,k,l];
%p为次品率，q为成品率
p1=x1(1);q1=1-x1(1);
p2=x2(1);q2=1-x2(1);
p=good(1);q=1-good(1);

theta=q1*q2*q/((1-p1*t1(i))*(1-p2*t2(j)));
%                   ida(count,1,r)=theta;%用于存贮theta

%a为配件单价，c为配件检测成本，b为装配成本，d为检测成品成本
```

```
a1=x1(2);c1=x1(3);
a2=x2(2);c2=x2(3);
b=good(2);d=good(3);
%w为市场售价，h为调换损失，e为拆解费用
w=good(4);
h=n_good(1);
e=n_good(2);

%零件成本
A=a1+a2;
%装配成本
B=b;
%检测零件成本
C=c1*t1(i)+c2*t2(j)+t1(i)*(a1+c1)*p1/(1-p1)+t2(j)*(a2+c2)*p2/(1-p2);
%拆解收益
D=(A+C-e)*s1(l);
%丢弃收益为0
%成品检测调换费用
E=(1-t0(k))*h;
%成品检测
E1=d*t0(k);
%成品不检测的调换费用
F=(w-A-B-C-w)*(1-t0(k));

%每个策略的收益
result(count)=-A-B-C-E1+(1-theta)*(D-E)+theta*w;
%
    result(r,count)=-a1-a2-c1*t1(i)-c2*t2(j)-b-d*t0(k)-(1-theta)*((e-a1-a2-t1(i)*c1-t2(j)*c2)*s1(l)+(1-t0(
end
end
end
end
%每个情况的最优策略以及对应值
idrr_best=zeros(1,1);
[result_best(m),idrr_best]=max(result,[],2);
idr_best(m,:)=idr(idrr_best,:)-1;%得到每个器件得到的最佳决策
%        result_best___=result_best(m);
%        idr_best___=idr_best(m,:);

end

% P_m
% result_best1=mode(result_best,1)
% idr_best1=mode(idr_best,1)
% P_m_best=mode(P_m,1)
% %
RR=[P_m,result_best,idr_best];
```

```
xlswrite('result4_2_5_data.xls',RR)

%第二问真实的策略
idr_right =[1 0 0 1
1 1 0 1
1 0 1 1
1 1 1 1
0 1 0 1
1 0 0 0];
idr5_right=idr_right(5,:);


%找出idr_best中有多少个不同的值
[diff_idr,~,~]=unique(idr_best,'rows')
d_l=size(diff_idr,1);
% count_right=zeros(1,5);

%count_bad用于计数
count_idr=zeros(d_l,5);%
diff_p=zeros(M,3,d_l);
if d_l>1
for a=1:d_l
if isequal(idr5_right,diff_idr(a,:))
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
else
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
diff_p(idx~=0,:,a)=P_m(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
end
end
else
count_idr=[size(idr_best,1),diff_idr];
end
count_idr
p_count=count_idr(:,1)./M
diff_p;
S1

xlswrite('result4_2_5_p_count.xls',[p_count,count_idr])
```

附录 K　利用第一问的抽样求解第四问中的第二问中的第 6 种情况

```
function [count_idr,S1]=CUMCM_B_4_2_6
%问题二的第6种情况
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.05 4 2];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.05 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.05 6 3  56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[10 40];

%定义整数规划的变量
t1=[0,1];%配件1是否检测
t2=[0,1];%配件2是否检测
t0=[0,1];%成品是否检测
s1=[0,1];%不合格是否拆解




%进行蒙特卡洛随机，减少抽样随机性
M=500;
result_best=zeros(M,1);%每个策略的最好的优化函数值
idr_best=zeros(M,4);
P_m=zeros(M,3);
S1=zeros(M,1);
n=390;%之前第一问的结果,抽样样本数
N=1000;%随机的总数

z1=zeros(1,N);
z2=zeros(1,N);
idn1=zeros(1,N*x1(1));
idn2=zeros(1,N*x2(1));
idn1=randperm(N,N*x1(1));%抽取次品的索引
idn2=randperm(N,N*x2(1));%抽取次品的索引
z1(idn1)=1;%把次品的值设成1
z2(idn2)=1;%把次品的值设成1
for m=1:M
% m=1;
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.05 4 2];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.05 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.05 6 3  56];
```

```
%不合格成品的基本条件：调换损失，拆解费用
n_good=[10 40];

%提前开数组，提高速度
%     z1=zeros(1,N);
%     z2=zeros(1,N);
%     idn1=zeros(1,N*x1(1));
%     idn2=zeros(1,N*x2(1));
idn0_1=zeros(1,n);
idn0_2=zeros(1,n);
z1_n=zeros(1,n);
z2_n=zeros(1,n);

%     idn1=randperm(N,N*x1(1));%抽取次品的索引
%     idn2=randperm(N,N*x2(1));%抽取次品的索引
%     z1(idn1)=1;%把次品的值设成1
%     z2(idn2)=1;%把次品的值设成1

idn0_1=randperm(N,n);%抽取随机样本的索引
idn0_2=randperm(N,n);%抽取随机样本的索引
z1_n=z1(idn0_1);%配件1抽取的随机样本
z2_n=z2(idn0_2);%配件2抽取的随机样本
p_1=sum(z1_n)/length(z1_n);%配件1的样品次品率
p_2=sum(z2_n)/length(z2_n);%配件2的样品次品率
z0=zeros(1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));
z0=binornd(1,good(1),1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n)]));%样品生成的成品
p_0=sum(z0)/length(z0);%样本成品的次品率
%替换掉原来的次品率
x1(1)=p_1;
x2(1)=p_2;
good(1)=p_0;
P_m(m,:)=[p_1,p_2,p_0];
%对抽样时成本进行最优决策，选择其是否拆卸，因为t1，t2，t0均为0
A0=x1(2)/p_1+x2(2)/p_2;
C0=x1(3)/p_1+x2(3)/p_2;
E0=n_good(2);
if (A0+C0-E0)>0
S1_0=1;
else
S1_0=0;
end
S1(m)=S1_0;
%     P_mm=P_m(m,:);

result=zeros(1,16);%每个器件的每个策略函数值
idr=zeros(16,4);
count=0;
```

```
for i=1:2 %t1的不同情况
for j=1:2 %t2的不同情况
for k=1:2 %t0的不同情况
for l=1:2 %s1的不同情况
%计数
count=count+1;
idr(count,:)=[i,j,k,l];
%p为次品率，q为成品率
p1=x1(1);q1=1-x1(1);
p2=x2(1);q2=1-x2(1);
p=good(1);q=1-good(1);

theta=q1*q2*q/((1-p1*t1(i))*(1-p2*t2(j)));
%                   ida(count,1,r)=theta;%用于存贮theta

%a为配件单价，c为配件检测成本，b为装配成本，d为检测成品成本
a1=x1(2);c1=x1(3);
a2=x2(2);c2=x2(3);
b=good(2);d=good(3);
%w为市场售价，h为调换损失，e为拆解费用
w=good(4);
h=n_good(1);
e=n_good(2);

%零件成本
A=a1+a2;
%装配成本
B=b;
%检测零件成本
C=c1*t1(i)+c2*t2(j)+t1(i)*(a1+c1)*p1/(1-p1)+t2(j)*(a2+c2)*p2/(1-p2);
%拆解收益
D=(A+C-e)*s1(l);
%丢弃收益为0
%成品检测调换费用
E=(1-t0(k))*h;
%成品检测
E1=d*t0(k);
%成品不检测的调换费用
F=(w-A-B-C-w)*(1-t0(k));

%每个策略的收益
result(count)=-A-B-C-E1+(1-theta)*(D-E)+theta*w;
%
    result(r,count)=-a1-a2-c1*t1(i)-c2*t2(j)-b-d*t0(k)-(1-theta)*((e-a1-a2-t1(i)*c1-t2(j)*c2)*s1(l)+(1-t0(
end
end
end
```

```
end
%每个情况的最优策略以及对应值
idrr_best=zeros(1,1);
[result_best(m),idrr_best]=max(result,[],2);
idr_best(m,:)=idr(idrr_best,:)-1;%得到每个器件得到的最佳决策
%        result_best___=result_best(m);
%        idr_best___=idr_best(m,:);

end

% P_m
% result_best1=mode(result_best,1)
% idr_best1=mode(idr_best,1)
% P_m_best=mode(P_m,1)
% %
RR=[P_m,result_best,idr_best];
xlswrite('result4_2_6_data.xls',RR)

%第二问真实的策略
idr_right =[1 0 0 1
1 1 0 1
1 0 1 1
1 1 1 1
0 1 0 1
1 0 0 0];
idr6_right=idr_right(6,:);


%找出idr_best中有多少个不同的值
[diff_idr,~,~]=unique(idr_best,'rows')
d_l=size(diff_idr,1);
% count_right=zeros(1,5);

%count_bad用于计数
count_idr=zeros(d_l,5);%
diff_p=zeros(M,3,d_l);
if d_l>1
for a=1:d_l
if isequal(idr6_right,diff_idr(a,:))
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
else
[~,idx]=ismember(idr_best,diff_idr(a,:),'rows');
same_rows=idr_best(idx~=0,:);
diff_p(idx~=0,:,a)=P_m(idx~=0,:);
```

```
count_idr(a,1)=size(same_rows,1);
count_idr(a,2:end)=diff_idr(a,:);
end
end
else
count_idr=[size(idr_best,1),diff_idr];
end
count_idr
p_count=count_idr(:,1)./M
diff_p;
S1

xlswrite('result4_2_6_p_count.xls',[p_count,count_idr])
```

附录 L　利用第一问抽样求解第四问的第三问（遍历算法）

```
function [result_best,idr_best,S1]=CUMCM_B_4_3_2_bianli
%8个零配件的基本信息
x=[.10 2 1
.10 8 1
.10 12 2
.10 2 1
.10 8 1
.10 12 2
.10 8 1
.10 12 2];
%3个半成品的基本信息
y=[.10 8 4 6
.10 8 4 6
.10 8 4 6];
%最终成品的基本信息
z=[.10 8 6 10 200 40];

%配件是否检测
t1=[0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1];
%半成品是否检测
t2=[0,1
0,1
```

```
0,1];
%半成品是否拆卸
s2=[0,1
0,1
0,1];
%成品是否检测
t0=[0,1];
%成品是否拆卸
s0=[0,1];



%蒙特卡洛模拟的次数
M=50;
N=1000;%随机的样本总数
result_best=zeros(M,1);%每个策略的最好的优化函数值
idr_best=zeros(M,16);
P_m=zeros(M,12);
S1=zeros(M,1);
n=390;%之前第一问的结果,抽样样本数

%8个配件的随机
z1=zeros(1,N);
z2=zeros(1,N);
z3=zeros(1,N);
z4=zeros(1,N);
z5=zeros(1,N);
z6=zeros(1,N);
z7=zeros(1,N);
z8=zeros(1,N);
%8个配件次品的随机
idn1=zeros(1,N*x(1,1));
idn2=zeros(1,N*x(2,1));
idn3=zeros(1,N*x(3,1));
idn4=zeros(1,N*x(4,1));
idn5=zeros(1,N*x(5,1));
idn6=zeros(1,N*x(6,1));
idn7=zeros(1,N*x(7,1));
idn8=zeros(1,N*x(8,1));
idn1=randperm(N,N*x(1,1));%抽取次品的索引
idn2=randperm(N,N*x(2,1));%抽取次品的索引
idn3=randperm(N,N*x(3,1));%抽取次品的索引
idn4=randperm(N,N*x(4,1));%抽取次品的索引
idn5=randperm(N,N*x(5,1));%抽取次品的索引
idn6=randperm(N,N*x(6,1));%抽取次品的索引
idn7=randperm(N,N*x(7,1));%抽取次品的索引
idn8=randperm(N,N*x(8,1));%抽取次品的索引
```

```
z1(idn1)=1;%把次品的值设成1
z2(idn2)=1;%把次品的值设成1
z3(idn3)=1;%把次品的值设成1
z4(idn4)=1;%把次品的值设成1
z5(idn5)=1;%把次品的值设成1
z6(idn6)=1;%把次品的值设成1
z7(idn7)=1;%把次品的值设成1
z8(idn8)=1;%把次品的值设成1

for m=1:M

%8个零配件的基本信息
x=[.10 2 1
.10 8 1
.10 12 2
.10 2 1
.10 8 1
.10 12 2
.10 8 1
.10 12 2];
%3个半成品的基本信息
y=[.10 8 4 6
.10 8 4 6
.10 8 4 6];
%最终成品的基本信息
z=[.10 8 6 10 200 40];

%配件是否检测
t1=[0,1
0,1
0,1
0,1
0,1
0,1
0,1
0,1];
%半成品是否检测
t2=[0,1
0,1
0,1];
%半成品是否拆卸
s2=[0,1
0,1
0,1];
%成品是否检测
t0=[0,1];
%成品是否拆卸
```

```
s0=[0,1];


idn0_1=zeros(1,n);z1_n=zeros(1,n);
idn0_2=zeros(1,n);z2_n=zeros(1,n);
idn0_3=zeros(1,n);z3_n=zeros(1,n);
idn0_4=zeros(1,n);z4_n=zeros(1,n);
idn0_5=zeros(1,n);z5_n=zeros(1,n);
idn0_6=zeros(1,n);z6_n=zeros(1,n);
idn0_7=zeros(1,n);z7_n=zeros(1,n);
idn0_8=zeros(1,n);z8_n=zeros(1,n);



idn0_1=randperm(N,n);%抽取随机样本的索引
idn0_2=randperm(N,n);%抽取随机样本的索引
idn0_3=randperm(N,n);%抽取随机样本的索引
idn0_4=randperm(N,n);%抽取随机样本的索引
idn0_5=randperm(N,n);%抽取随机样本的索引
idn0_6=randperm(N,n);%抽取随机样本的索引
idn0_7=randperm(N,n);%抽取随机样本的索引
idn0_8=randperm(N,n);%抽取随机样本的索引
z1_n=z1(idn0_1);%配件1抽取的随机样本
z2_n=z2(idn0_2);%配件2抽取的随机样本
z3_n=z3(idn0_3);%配件2抽取的随机样本
z4_n=z4(idn0_4);%配件2抽取的随机样本
z5_n=z5(idn0_5);%配件2抽取的随机样本
z6_n=z6(idn0_6);%配件2抽取的随机样本
z7_n=z7(idn0_7);%配件2抽取的随机样本
z8_n=z8(idn0_8);%配件2抽取的随机样本
p_1=sum(z1_n)/length(z1_n);%配件1的样品次品率
p_2=sum(z2_n)/length(z2_n);%配件2的样品次品率
p_3=sum(z3_n)/length(z3_n);%配件2的样品次品率
p_4=sum(z4_n)/length(z4_n);%配件2的样品次品率
p_5=sum(z5_n)/length(z5_n);%配件2的样品次品率
p_6=sum(z6_n)/length(z6_n);%配件2的样品次品率
p_7=sum(z7_n)/length(z7_n);%配件2的样品次品率
p_8=sum(z8_n)/length(z8_n);%配件2的样品次品率
%替换掉原来产品的次品率
x(:,1)=[p_1;p_2;p_3;p_4;p_5;p_6;p_7;p_8];
P_m(m,1:8)=[p_1;p_2;p_3;p_4;p_5;p_6;p_7;p_8];

%计算半成品的次品率
y0=zeros(1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n),length(z3_n)-sum(z3_n)]));
y1=zeros(1,min([length(z4_n)-sum(z4_n),length(z5_n)-sum(z5_n),length(z6_n)-sum(z6_n)]));
y2=zeros(1,min([length(z7_n)-sum(z7_n),length(z8_n)-sum(z8_n)]));
y0=binornd(1,y(1,1),1,min([length(z1_n)-sum(z1_n),length(z2_n)-sum(z2_n),length(z3_n)-sum(z3_n)]));%样品生
```

```
y1=binornd(1,y(2,1),1,min([length(z4_n)-sum(z4_n),length(z5_n)-sum(z5_n),length(z6_n)-sum(z6_n)]));%样品生
y2=binornd(1,y(3,1),1,min([length(z7_n)-sum(z7_n),length(z8_n)-sum(z8_n)]));%样品生成的半成品
p_y0=sum(y0)/length(y0);%样本半成品的次品率
p_y1=sum(y1)/length(y1);%样本半成品的次品率
p_y2=sum(y2)/length(y2);%样本半成品的次品率
%替换掉原来的半成品次品率
y(:,1)=[p_y0;p_y1;p_y2];
P_m(m,9:11)=[p_y0;p_y1;p_y2];

%计算现在成品的次品率
x0=zeros(1,min([length(y0)-sum(y0),length(y1)-sum(y1),length(y2)-sum(y2)]));
x0=binornd(1,z(1),1,min([length(y0)-sum(y0),length(y1)-sum(y1),length(y2)-sum(y2)]));
p_x0=sum(x0)/length(x0);%样本成品的次品率
%替换掉原来的半成品次品率
z(1)=[p_x0];
P_m(m,end)=[p_x0];

%对抽样时成本进行最优决策，选择其是否拆卸，因为t1，t2，t0均为0
A0=sum(x(:,2)./(1-x(:,1)));
C0=sum(x(:,3)./(1-x(:,1)));
E0=z(4);
if (A0+C0-E0)>0
S1_0=1;
else
S1_0=0;
end
S1(m)=S1_0;
% result_best(m,2)=S1_0;



%次品率
p1=x(1,1);p2=x(2,1);p3=x(3,1);p4=x(4,1);
p5=x(5,1);p6=x(6,1);p7=x(7,1);p8=x(8,1);
p9=y(1,1);p10=y(2,1);p11=y(3,1);p=z(1,1);
pp1=[p1,p2,p3,p4,p5,p6,p7,p8];
%成功率
q1=1-x(1,1);q2=1-x(2,1);q3=1-x(3,1);q4=1-x(4,1);
q5=1-x(5,1);q6=1-x(6,1);q7=1-x(7,1);q8=1-x(8,1);
q9=1-y(1,1);q10=1-y(2,1);q11=1-y(3,1);q=1-z(1,1);

A=sum(x(:,2));%A为配件需要的总成本
B=sum(y(:,2))+z(2);%第一道工序半成品与成品的装配和
e=z(4);%成品拆解费用
h=z(6);%成品调换损失
w=z(5);%成品市场售价

%储存结果的数组
```

```
result=zeros(2^16,1);
idr=zeros(2^16,16);
idx=zeros(1,8);
idy=zeros(1,3);
idz=zeros(1,2);
pp1=zeros(1,8);
count=0;

%i代表检测，j代表拆卸
for i0=1:length(t0)
for j0=1:length(s0)
for i1=1:2
for i2=1:2
for i3=1:2
for i4=1:2
for i5=1:2
for i6=1:2
for i7=1:2
for i8=1:2
for i9=1:2
for i10=1:2
for i11=1:2
for j1=1:2
for j2=1:2
for j3=1:2


%储存结果的数组
% result=zeros(2^16,1);
% idr=zeros(1,16);
idx=zeros(1,8);
idy=zeros(1,3);
idz=zeros(1,2);
pp1=zeros(1,8);
% count=1;

count=count+1;
i0=i0-1;i1=i1-1;i2=i2-1;i3=i3-1;
i4=i4-1;i5=i5-1;i6=i6-1;i7=i7-1;
i8=i8-1;i9=i9-1;i10=i10-1;i11=i11-1;
j0=j0-1;j1=j1-1;j2=j2-1;j3=j3-1;
idr(count,:)=[i0,j0,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,j1,j2,j3];
idx=idr(count,3:10);
idy=idr(count,11:end);
idz=idr(count,1:2);
idx=idx';
idy=idy';
```

```
idz=idz';

%贝叶斯概率
theta1=q1*q2*q3*q9/((1-p1*i1)*(1-p2*i2)*(1-p3*i3));
theta2=q4*q5*q6*q10/((1-p4*i4)*(1-p5*i5)*(1-p6*i6));
theta3=q7*q8*q11/((1-p7*i7)*(1-p8*i8));
theta=q1*q2*q3*q4*q5*q6*q7*q8*q9*q10*q11*q/((1-p1*i1)*(1-p2*i2)*(1-p3*i3)

*(1-p4*i4)*(1-p5*i5)*(1-p6*i6)*(1-p7*i7)*(1-p8*i8)*(1-p9*i9)*(1-p10*i10)*(1-p11*i11));

%设立一个存放数组
pp1=pp1./(1-pp1);%存放检测的概率
pp1=pp1';
%8配件检测的总成本
C=sum(x(:,3).*idx+idx.*pp1.*(x(:,2)+x(:,3)));
%半成品拆解成本
D1=sum(y(1,2)+y(1,4))*i9*j1;%i代表检测, j代表拆卸
D2=sum(y(2,2)+y(2,4))*i10*j2;
D3=sum(y(3,2)+y(3,4))*i11*j3;
%半成品丢弃成本
G1=(sum(x(1:3,2)+x(1:3,3).*idx(1:3)+idx(1:3).*pp1(1:3).*(x(1:3,2)+x(1:3,3)))+y(1,2))*i9*(1-j1);
G2=(sum(x(4:6,2)+x(4:6,3).*idx(4:6)+idx(4:6).*pp1(4:6).*(x(4:6,2)+x(4:6,3)))+y(2,2))*i10*(1-j2);
G3=(sum(x(7:8,2)+x(7:8,3).*idx(7:8)+idx(7:8).*pp1(7:8).*(x(7:8,2)+x(7:8,3)))+y(3,2))*i11*(1-j3);
%成品拆解收益
D0=(A+C-z(4))*j0;
%成品调换费用
E=(1-i0)*z(6);
%成品检验费用
E1=sum(y(:,3).*idy(1:3))+z(3)*idz(1);

result(count)=-A-B-C-E1-(1-theta1)*(D1+G1)/theta1-(1-theta2)*(D2+G2)/theta2-(1-theta3)*(D3+G3)/theta3+(1-t
i0=i0+1;i1=i1+1;i2=i2+1;i3=i3+1;
i4=i4+1;i5=i5+1;i6=i6+1;i7=i7+1;
i8=i8+1;i9=i9+1;i10=i10+1;i11=i11+1;
j0=j0+1;j1=j1+1;j2=j2+1;j3=j3+1;



end
end
end
end
end
end
end
end
end
end
```

```
end
end
end
end
end
end
[result_best(m),idrr_best]=max(result);
idr_best(m,:)=idr(idrr_best,:);
end
result_best
idr_best
S1
[result_best_best,idr_best_best]=max(result_best);
result_best_best
P_m_beat=P_m(idr_best_best,:)
```

附录 M　灵敏性分析以第二问第二种情况为例

```
function idr_best1=CUMCM_B_lingminxing_2
%零配件1的基本条件：次品率、购买单价、检测成本
x1=[0.1 4 2
0.2  4 2
0.1  4 2
0.2  4 1
0.1  4 8
0.05 4 2];
%零配件2的基本条件：次品率、购买单价、检测成本
x2=[0.1 18 3
0.2  18 3
0.1  18 3
0.2  18 1
0.2  18 1
0.05 18 3];
%成品的基本条件:次品率、装配成本、检测成本、市场售价
good=[0.1 6 3 56
0.2  6 3 56
0.1  6 3 56
0.2  6 2 56
0.1  6 2 56
0.05 6 3 56];
%不合格成品的基本条件：调换损失，拆解费用
n_good=[6 5
6 5
30 5
30 5
```

```
10 5
10 40];

%定义整数规划的变量
t1=[0,1];%配件1是否检测
t2=[0,1];%配件2是否检测
t0=[0,1];%成品是否检测
s1=[0,1];%不合格是否拆解
idr=zeros(16,4,6);
ida=zeros(16,1,6);

l_g=7;
result=zeros(6,16);%每个策略的优化函数值
result_lingmin=zeros(l_g,16);%每个策略的优化函数值
idr_lingmin=zeros(16,4,l_g);
% for r=1:6 %六种不同情况
r=2;


for g=1:l_g
x1(2,1)=0.19+g/100;
count=0;
for i=1:2 %t1的不同情况
for j=1:2 %t2的不同情况
for k=1:2 %t0的不同情况
for l=1:2 %s1的不同情况
%计数
count=count+1;
idr(count,:,r)=[i,j,k,l];
idr(count,:,r)=idr(count,:,r)-1;
idr_lingmin(count,:,g)=idr(count,:,r);
%p为次品率，q为成品率
p1=x1(r,1);q1=1-x1(r,1);
p2=x2(r,1);q2=1-x2(r,1);
p=good(r,1);q=1-good(r,1);

theta=q1*q2*q/((1-p1*t1(i))*(1-p2*t2(j)));
ida(count,1,r)=theta;%用于存贮theta

%a为配件单价，c为配件检测成本，b为装配成本，d为检测成品成本
a1=x1(r,2);c1=x1(r,3);
a2=x2(r,2);c2=x2(r,3);
b=good(r,2);d=good(r,3);
%w为市场售价，h为调换损失，e为拆解费用
w=good(r,4);
h=n_good(r,1);
e=n_good(r,2);
```

```
%零件成本
A=a1+a2;
%装配成本
B=b;
%检测零件成本
C=c1*t1(i)+c2*t2(j)+t1(i)*(a1+c1)*p1/(1-p1)+t2(j)*(a2+c2)*p2/(1-p2);
%拆解收益
D=(A+C-e)*s1(l);
%丢弃收益为0
%成品检测调换费用
E=(1-t0(k))*h;
%成品检测
E1=d*t0(k);
%成品不检测的调换费用
F=(w-A-B-C-w)*(1-t0(k));

%每个策略的收益
result(r,count)=-A-B-C-E1+(1-theta)*(D-E)+theta*w;
result_lingmin(g,count)=result(r,count);
end
end
end
end
end

%每个情况的最优策略以及对应值
result_best=zeros(l_g,1);
idr_best=zeros(l_g,1);
result_lingmin
idr_lingmin
[result_best,idr_best]=max(result_lingmin,[],2)
for m=1:l_g
idr_best1(m,:)=idr_lingmin(idr_best(m),:,m);
end
idr_best1

% xlswrite('result2_data',result)
```

附录 N 稳定性分析以第四问的第二问情况为例

```
function u=CUMCM_B_wendingxing_4_2

p_count1 =[
0.3700   0.6300
```

```
0.3600   0.6400
0.3650   0.6350
0.3648   0.6352
0.3600   0.6400
0.3600   0.6400
0.3655   0.6345];

x=500:250:2000;
plot(x,p_count1(:,1),'DisplayName','1001');
hold on
plot(x,p_count1(:,2),'DisplayName','1101');
hold on

ylim([0 1])

xlabel('蒙特卡洛次数')
ylabel('概率')
title('决策结果的稳定性分析')
```

附录 O　第二问 6 种情况结果折线图

```
function r=CUMCM_2_draw_6
c1=[11.198 15.805 9.824 14.431 10.88666667 15.13 9.026666667 13.27 13.55333333 17.29
    11.69333333 15.43 13.8 16.3 11.4 13.9];
c0=1:1:16;
figure(1);
plot(c0,c1);
axis([1 16 9 18]);xlabel('方案序号');
ylabel('收益/元');
set(gca,'xtick',1:1:16);
c2=[-2.256 6.04 -2.328 5.968 -2.57 6.52 -3.41 5.68 2.18 9.56 1.34 8.72 3.85 9.6 2.05 7.8];
figure(2);
plot(c0,c2);
axis([1 16 -4 10]);xlabel('方案序号');
ylabel('收益/元');
set(gca,'xtick',1:1:16);
c3=[4.694 9.301 9.824 14.431 6.326666667 10.57 9.026666667 13.27 8.993333333 12.73
    11.69333333 15.43 11.4 13.9 11.4 13.9];
figure(3);
plot(c0,c3);
axis([1 16 4 16]);xlabel('方案序号');
ylabel('收益/元');
set(gca,'xtick',1:1:16);
c4=[-13.968 -5.672 -1.328 6.968 -8.71 -0.52 0.09 8.28 -5.21 1.72 3.59 10.52 2.8 7.8 6.8
    11.8];
```

```
figure(4);
plot(c0,c4);
axis([1 16 -14 12]);xlabel('方案序号');
ylabel('收益/元');
set(gca,'xtick',1:1:16);
c5=[4.768 10.752  6.288 12.272 9.71 14.0325 9.61 13.9325 0.186666667 7.56 0.986666667
    8.36 6.316666667 9.525 5.316666667 8.525];
figure(5);
plot(c0,c5);
axis([1 16 0 15]);xlabel('方案序号');
ylabel('收益/元');
set(gca,'xtick',1:1:16);
c6=[18.58675 16.0195 17.013 14.44575 17.45973684 16.105 15.43473684 14.08 19.24921053
    17.72 17.22421053 15.695 18.27894737 17.7 15.77894737 15.2];
figure(6);
plot(c0,c6);
axis([1 16 14 20]);xlabel('方案序号');
ylabel('收益/元');
set(gca,'xtick',1:1:16);
fb
```