生产过程中的决策优化设计

摘要

本文针对某企业在生产电子产品过程中所面临的次品率、检测及拆解等一系列决策问题展开研究。通过优化生产过程中的抽样、检测和拆解策略，最大化企业利润和生产效率。本文的研究旨在为企业提供一套基于数学建模的决策方案，帮助其在面对不同生产条件时合理选择抽样、检测和拆解策略，提升资源利用率和经济效益。

针对问题一，为通过抽样检测判断次品率是否超过标称值，本文分别针对(1)(2)情形建立了**假设检验**问题，进而得到了是否接收零配件的条件。为确定合适的抽样样本量，本文利用检测花销与假设检验效果(运用假设检验**功效函数**衡量)之间相互制约的关系，建立**决策函数**，得到了**最优样本量**。

针对问题二，为针对六种不同情况做出决策，本研究运用**二项分布和几何分布**的性质，在不同决策方案下，用各零件检测成本、次品率、调换成本等值表示出每生产一个合格成品的利润期望值。再依次将六种情况的具体数值带入到表达式之中，将**利润期望最大化**作为决策标准，得到六种情况下的最优决策。

针对问题三，为解决多工序制造系统中的零件、半成品与成品的次品检测和拆解决策问题，求解目标在于合理设计检测与回收策略及最大化系统的期望利润。首先构建**状态-决策模型**，定义零件、半成品和成品的状态变量与决策变量，将问题转化为多阶段决策问题。模型通过**动态规划算法**有效求解最优决策路径，计算各个阶段的成本和收益，并利用**记忆化存储**提高求解效率。最终求解结果显示，按照最优检测与拆解策略，系统的最优期望利润为**56.7元**。

针对问题四，为描述次品率的不确定性，本问运用**贝叶斯估计**，选用**共轭先验分布**$beta$分布，推导出次品率服从的**后验分布**。完成问题二，计算出在次品率服从的分布下利润的期望，重新进行决策。完成问题三，修改固定的次品率为分布密度后，重新运行程序获得最优决策路径和相应最优利润期望为**54.2元**。经过灵敏度检验，先验分布超参数的选取对利润期望的影响较小。

关键词：**假设检验 状态-决策模型 动态规划算法 贝叶斯估计 共轭先验分布**

中国大学生在线 dxs.moe.gov.cn

一. 问题重述

1.1 问题背景

某企业进行电子产品的装配生产，从零配件供应商处购买所需零配件，但零配件会有一定的次品，次品率会影响该企业成品的合格率；企业装配零配件形成半成品、装配半成品形成成品的过程也并不能达到完全正确，进而也会影响成品合格率。次品会使企业收益降低，但零配件、半成品、成品的检测及拆解也需要企业承担相应成本，因此企业需考虑生产过程中的各步检测及拆解策略，以期达到利润最大化。

1.2 问题重述

企业购买多种零配件进行装配，在生产过程中任一零配件不合格均会导致成品的不合格，此外，在零配件全部合格的情况下会出现装配过程导致的半成品及成品的不合格。企业可以选择对零配件、半成品、成品进行检测，对于不合格的零配件，直接丢弃；对于不合格的半成品及成品，企业可以选择将其进行拆解，拆解过程不会导致零配件的损耗，拆解出的零配件可以重新进行装配使用。企业的生产成本除了包含零配件购买及装配成本外，检测及拆解费用均需企业自行承担，如果出现顾客购买到不合格品的情况，企业给予无条件退换并承担调换损失。

现需根据问题给定的条件对企业购买零配件进行抽样检测的过程以及企业的生产过程进行决策。

问题一 为了检验供应商供应的零配件次品率是否超过供应商给出的标称值，进而决定是否接收这批零配件，企业需要使用抽样检测方法进行验证并自行承担检测费用。现需设计标称值为 10%条件下的抽样检测方案，以期平衡检测结果可靠性尽可能高及检测次数尽可能少两方面，并针对(1)、(2)两种具体情况给出具体结果。

问题二 在仅考虑两种零配件直接装配成为成品的生产流程条件下，企业购买的零配件次品率及成品装配次品率均已知，现需自行确立决策依据和决策指标，进一步建立决策模型来帮助企业判断在生产过程中是否需要检测零配件、成品质量以及对于不合格的成品是否需要拆解，并针对题目给出的零配件次品率、购买成本及检测成本；成品次品率、装配成本、检测成本及市场售价；不合格成品的调换损失及拆解成本均有确定取值的 6 种具体情况给出生产过程决策方案及决策指标值。

问题三 在考虑 8 个零配件装配成 3 个半成品，再由半成品装配成为成品的条件下，建立决策模型来帮助企业判断在生产过程中是否需要检测零配件、半成品、成品质量以及对于不合格的半成品、成品是否需要拆解，并针对题目给出的零配件次品率、购买成本及检测成本；半成品次品率、装配成本、检测成本及拆解费用；成品次品率、装配成本、检测成本及市场售价；成品的调换损失及拆解成本均有确定取值的 1 种具体情况给出生产过程决策方案及决策指标值。

中国大学生在线 dxs.moe.gov.cn

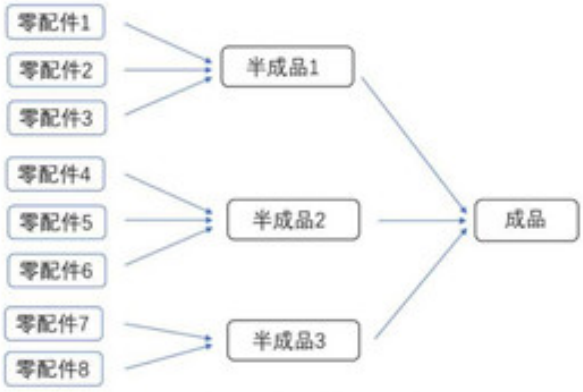

图 1　问题三生产流程示意图

问题四　在问题二、问题三的决策模型基础上，将零配件、半成品、成品的次品率给定取值变化为通过抽样检测方法得到的相应取值，进一步优化决策模型，并给出问题二、问题三具体情况中其余给定取值不变时的决策方案及决策指标值。

二、问题分析

问题一　企业通过抽样检测的方式判断一批零件的次品率是否小于标称值相当于统计学中的假设检验问题。可使用简单随机抽样抽取检测样本，由于企业需要做检测成本与检测效果之间的权衡，需找寻最佳样本量。可利用检测单价与检测样本量的乘积表现检测成本，利用假设检验的功效函数表现假设检验的效果，将二者线性组合为决策函数，寻找使得决策函数取极值的检测样本量作为最佳样本量。

问题二　企业需要解决下述四个决策问题：(1)零配件 1 是否检测；(2)零配件 2 是否检测；(3)成品是否检测；(4)不合格成品是否拆解。故共存在 16 种决策方案。定义最佳方案为利润期望最大的方案，利用二项分布和几何分布的性质，分别严格求出 16 种决策方案每生产出一个合格单品利润的期望。再将六种情型的具体数值代入利润期望公式，得出每种情形的最佳决策方案及决策指标值。

问题三　企业需要解决下述五类决策问题：(1)零配件 1-8 是否检测；(2)半成品是否检测；(3)不合格半成品是否拆解；(4)成品是否进行检测；(5)不合格成品是否拆解。基于题目给出的 8 个零配件和 3 个半成品，问题二使用的对每种决策方案求利润期望的方法对应的决策方案数量过多，且无法推广至m道工序、n个零配件的生产情况，故考虑将该问题转化为一个多阶段决策问题，建立状态-决策模型，并通过动态规划算法来求解最优策略及决策指标值。

问题四　为衡量次品率取值可能存在的“误差”，可利用贝叶斯估计，选用共轭先验分布$Beta$分布，得到参数的后验分布。重新解决问题二，可将第二问中求出的利润期望公式视作次品率的函数，在本问的假定下，次品率服从某一分布，可进一步计算在次品率的后验分布下利润的期望。比较 16 种决策方案的利润期望得到最佳决策方案。重新解决问题三，分析可得唯一的不同在于，在问题三中未检测的零件、半成品(组成零件均合格)、成品(半成品均合格)是否为次品均服从次品率固定的伯努利分布，问题四引入了$Beta$分布来描述次品率的不确定性。通过推导，得出新的次品概率分布，并在此基础上进行最优决策。

中国大学生在线 dxs.moe.gov.cn

三、模型假设

假设一：所有决策为确定性决策，适用于生产线上所有相应节点。
说明：假设生产过程中不涉及概率性问题，每一项决策(如是否检测、是否拆解等)都是确定的。这意味着在任何时候，检测、拆解等操作要么执行，要么不执行，并且这一决策适用于所有相关节点。这样处理降低了流水线决策成本，符合实际生产中工厂流水线的情况。

假设二：退回成品拆解后，若无法确定相应零件或半成品是否为次品，则必须进行检测。
说明：假设在拆解后的零件和半成品存在质量不确定性时(即之前未对该半成品或零件进行检测)，必须进行检测。此假设考虑到企业质量控制的需要，确保每个生产阶段的次品得以及时识别和处理，更重要的是显著提升效率，防止次品在流水线上无限循环，合理地简化了模型。

四、符号说明

符号	说明
n	抽样检测样本量
V	售卖成品的单位利润
P	成品市场售价
C	合格成品的单位成本
S_t	t时刻零件 1-8、半成品 1-3 及成品的状态变量
a_t	t时刻零件 1-8、半成品 1-3 及成品的决策变量
R	成品收益，包括销售收益和相应拆解后半成品或零件的回收收益
α,β	先验分布中的超参数

五、问题一模型建立及求解

在频率学派假设检验的理论体系下，原假设和备择假设的地位存在差异，本问题中企业对于拒收与接收的态度也存在差异，故可以利用假设检验的思想处理该问题[1]。而(1)和(2)两个情形下，不仅仅只有显著性水平的差别，原假设与备择假设互相相反，故在下文中分开对两个情景进行讨论。

5.1 (1)情形

使用简单随机抽样检验次品率，假定共抽取n个样本且每一个样本是否为次品独立同分布，则该问题可运用如下数学语言进行表达：

$$X_1, X_2, \ldots\ldots, X_n i.i.d. \sim B(1,p)$$

$$X_i = \begin{cases} 1, \text{第}i\text{个样品为次品} \\ 0, \text{第}i\text{个样品合格} \end{cases}$$

其中$i.i.d.$表示独立同分布，$B(1,p)$表示二项分布，p为零件的次品率。

根据 10%的标称值与方案要求“在 95%的信度下认定零配件次品率超过标称值，则拒

中国大学生在线 dxs.moe.gov.cn

收这批零件”，可得到假设检验问题：

$$H_0: p \le 0.1$$
$$H_1: p > 0.1$$

该假设检验问题的显著性水平$\alpha = 0.05$。根据假设检验的思想，**次品率不超过10%的标称值但是拒绝了原假设的概率(即犯第一类错误的概率)必须控制在$\alpha = 0.05$之下**，满足以上要求的前提下，**尽量控制次品率超过10%的标称值但是接受原假设的概率**(即犯第二类错误的概率)。

如果检测的次数较大，则企业的花费较大；如果检验的次数较小，则虽然节省了花费，但是导致犯第二类错误的概率较大。可见，**选择合适的检验次数实际上是在检测花销与犯第二类错误的概率之间的权衡。**

由于n的取值较大，可根据如下方式构造检验统计量并利用其渐进正态性质：

$$Z = \frac{\overline{X} - p}{\sqrt{\frac{p(1-p)}{n}}} \to N(0,1)$$

其中：

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

为样本均值。

原假设成立的边界条件为$p = 0.1$，在此条件下：

$$Z = \frac{\overline{X} - 0.1}{\sqrt{\frac{0.09}{n}}} = \frac{\sqrt{n}(\overline{X} - 0.1)}{0.3} \to N(0,1)$$

故$\alpha = 0.05$ 下的拒绝域为：

$$\left\{\overline{X} \middle| \frac{\sqrt{n}(\overline{X} - 0.1)}{0.3} > u_{0.95}\right\}$$

即：

$$\left\{\overline{X} | \overline{X} > \frac{0.3u_{0.95}}{\sqrt{n}} + 0.1\right\}$$

其中$u_{0.95}$表示标准正态分布的上 0.05 分位数

此式为判定方法，若根据样本计算出的$\overline{X}$落入上述集合中，则拒绝原假设，拒收这批零件；反之，若$\overline{X}$落入上述集合的补集中，则接受原假设，接收这批零件。

为衡量假设检验的好坏，下计算该假设检验的功效函数$K(n,p)$：

$$K(n,p) = P(\overline{X} > \frac{0.3u_{0.95}}{\sqrt{n}} + 0.1 | p)$$
$$= P(\frac{\overline{X} - p}{\sqrt{\frac{p(1-p)}{n}}} > \frac{\frac{0.3u_{0.95}}{\sqrt{n}} + 0.1 - p}{\sqrt{\frac{p(1-p)}{n}}} | p)$$

中国大学生在线 dxs.moe.gov.cn

$$= 1 - \Phi\left(\frac{\frac{0.3u_{0.95}}{\sqrt{n}} + 0.1 - p}{\sqrt{\frac{p(1-p)}{n}}}\right)$$

其中$\Phi(\cdot)$表示标准正态分布的分布函数。

5.2 (2)情形

(2)情形下，方案要求为“在90%的信度下认为零配件次品率不超过标称值，则接收这批零配件”，可见，与(1)情形的原假设和备择假设互相颠倒，如下：

$$H_0: p > 0.1$$

$$H_1: p \leq 0.1$$

由于信度为90%，故显著性水平$\alpha = 0.1$，在原假设成立的边界条件$p = 0.1$下，$Z = \frac{\overline{X}-0.1}{\sqrt{\frac{0.09}{n}}} = \frac{\sqrt{n}(\overline{X}-0.1)}{0.3} \to N(0,1)$仍然成立。与(1)情形不同的在于拒绝域的构造，如下：

$$\left\{\overline{X}\left|\frac{\sqrt{n}(\overline{X}-0.1)}{0.3} < u_{0.1}\right.\right\}$$

即：

$$\left\{\overline{X}|\overline{X} < \frac{0.3u_{0.1}}{\sqrt{n}} + 0.1\right\}$$

其中$u_{0.1}$表示标准正态分布的下0.1分位数。

若根据样本计算出的$\overline{X}$落入上述集合中，则拒绝原假设，拒收这批零件；反之，若$\overline{X}$落入上述集合的补集中，则接受原假设，接收这批零件。

下面推导(2)情形下的功效函数$K(n,p)$：

$$K(n,p) = P(\overline{X} < \frac{0.3u_{0.1}}{\sqrt{n}} + 0.1|p)$$

$$= P(\frac{\overline{X}-p}{\sqrt{\frac{p(1-p)}{n}}} < \frac{\frac{0.3u_{0.1}}{\sqrt{n}} + 0.1 - p}{\sqrt{\frac{p(1-p)}{n}}}|p)$$

$$= \Phi\left(\frac{\frac{0.3u_{0.1}}{\sqrt{n}} + 0.1 - p}{\sqrt{\frac{p(1-p)}{n}}}\right)$$

5.3 (1)(2)情形下样本量n的选取

上文提到，**n的选取实际上就是检测花销与假设检验效果之间的权衡**，为衡量假设检验的效果，要用到上文计算过的功效函数$K(n,p)$。显然，对于给定的p，n越大则$K(n,p)$越大，$K(n,p)$越大则犯第二类错误的概率越低，假设检验的效果越好。下面计算$K(n,p)$的期望$q(n)$

中国大学生在线 dxs.moe.gov.cn

作为样本量为n时假设检验效果的衡量因素：

$$q(n) = E(K(n,p)|n) = \int_{p\epsilon\Theta_1} f(p)\,K(n,p)dp$$

其中，Θ_1表示备择假设的参数空间，$f(p)$为p的先验分布密度函数，在本文中可假定p的先验分布为均匀分布$U(0,1)$。

另一方面，n越大则花销越大。假定花销正比于样本量n，可列出如下函数$g(n)$作为抉择n取值的决策函数：

$$g(n) = w_1 cn - w_2 q(n)$$

$$= w_1 cn - w_2 \int_{p\epsilon\Theta_1} f(p)\,K(n,p)dp$$

其中w_1, w_2为给定的权重，满足$w_1 + w_2 = 1, w_1 > 0, w_2 > 0$，由企业认为的检测花销和检测效果的重要性所决定，$c$为每检测一件样本的花销，。如果企业认为检测效果为上，则w_2应当较大；如果企业认为减少检测花销为上，则w_1应当较大。

如：若假定$w_1 = 0.001, w_2 = 0.999$，则通过R语言计算，得到n的最优取值为 43。

5.4 (1)(2)结果总结

综合上述内容，(1)(2)的具体结果总结如下：

表 1　问题一结果总结表

	(1)	(2)
假定	样本$X_1, X_2, \ldots\ldots, X_n i.i.d. \sim B(1,p)$	
原假设	$H_0: p \le 0.1$	$H_0: p > 0.1$
备择假设	$H_1: p > 0.1$	$H_1: p \le 0.1$
显著性水平	0.05	0.1
检验统计量	$\overline{X}$	
拒绝域	$\left\{\overline{X}\middle\vert \overline{X} > \frac{0.3u_{0.95}}{\sqrt{n}} + 0.1\right\}$	$\left\{\overline{X}\middle\vert \overline{X} < \frac{0.3u_{0.1}}{\sqrt{n}} + 0.1\right\}$
功效函数$K(n,p)$	$1 - \Phi\left(\frac{\frac{0.3u_{0.95}}{\sqrt{n}} + 0.1 - p}{\sqrt{\frac{p(1-p)}{n}}}\right)$	$\Phi\left(\frac{\frac{0.3u_{0.1}}{\sqrt{n}} + 0.1 - p}{\sqrt{\frac{p(1-p)}{n}}}\right)$
n的确定	最大化决策函数$g(n)$	
决策函数$g(n)$	$w_1 cn - w_2 \int_{0.1}^{1} f(p)\,K(n,p)dp$	$w_1 cn - w_2 \int_{0}^{0.1} f(p)\,K(n,p)dp$

六、问题二模型建立及求解

决策的本质为最大化收益的期望。实际情况下，生产厂家往往会有某个预期产量n，企业需要最大化生产每个合格成品收益的期望，由于销售价格的恒定，上述问题等价于**最小化生产每个合格成品花销的期望**。

中国大学生在线 dxs.moe.gov.cn

可供企业考虑的决策共下述 16 种情况：

表 2　16 种决策方案及对应编号表

不合格成品是否拆解	是否检测零配件 1	是否检测零配件 2	是否检测成品	编号
是	是	是	是	001
			否	002
		否	是	003
			否	004
	否	是	是	005
			否	006
		否	是	007
			否	008
否	是	是	是	009
			否	010
		否	是	011
			否	012
	否	是	是	013
			否	014
		否	是	015
			否	016

由于**当对不合格的成品进行拆解时，如果始终未对零配件 1 和零配件 2 进行检测，则零配件的次品会永远处于流水线中永远不会被丢弃**，故做出如下假设：**当企业的决策为对不合格的成品进行拆解时，若开始进行(1)时未对零配件 1 或零配件 2 进行检测，则拆解后重复(1)时必然对其进行检测。**

下文将计算 16 种不同决策下生产每个合格成品花销的期望与各产品次品率、购买单价、检测成本、装配成本以及调换损失、拆解费用的关系。由于上述假设的存在，是否对不合格的成品拆解会对表达式的求解产生较大影响，**丢弃不合格成品的 8 种决策情况可以给出统一的表达式**，拆解不合格成品的 8 种决策情况较为复杂，由于篇幅原因，只详细说明编号为 008 的表达式推导过程，其余编号直接给出表达式。

6.1 丢弃不合格的成品下表达式的求解

在丢弃不合格的成品的前提条件下，根据是否检测零配件 1、是否检测零配件 2、是否检测成品建立三个示性变量I_1, I_2, I_3, 其定义如下：

$$I_1 = \begin{cases} 1, \text{不检验零配件 1} \\ 0, \text{检验零配件 1} \end{cases}$$

$$I_2 = \begin{cases} 1, \text{不检验零配件 2} \\ 0, \text{检验零配件 2} \end{cases}$$

中国大学生在线 dxs.moe.gov.cn

$$I_3=\begin{cases}1,\text{不检验成品}\\0,\text{检验成品}\end{cases}$$

根据是否检验零配件分为如下两种情况①②：

①若对零配件i进行检验，则生产出能够进入下一流程的零配件(对于检验零配件而言，需要保证零配件合格才能进入下一流程)所需要的生产次数ξ_i服从如下分布：

$$\xi_i\sim G(1-p_i)(i=1,2)$$

其中G表示几何分布，p_i为零配件i的次品率。故：

$$E(\xi_i)=\frac{1}{1-p_i}$$

②若不对零配件进行检验，则生产出能够进入下一流程的零配件所需的生产次数$\xi_i\equiv 1$。

- 综合①②，可得出：

$$E(\xi_i)=\frac{1}{1-(1-I_i)p_i}$$

考虑成品合格的概率，不难发现：

$$\begin{aligned}P(\text{成品合格})&=P(\text{零配件 1 合格})P(\text{零配件 2 合格})P(\text{成品装配正确})\\&=(1-I_1p_1)(1-I_1p_2)(1-p_3)\end{aligned}$$

其中p_1,p_2分别表示零配件 1 和零配件 2 的次品率，p_3为成品的次品率。

显然，成品出现第一件合格品时制造出的成品的个数λ服从几何分布：

$$\lambda\sim G((1-I_1p_1)(1-I_1p_2)(1-p_3))$$

每制造出一件成品所需要的花费C_0的均值$E(C)$为：

$$\begin{aligned}E(C_0)&=(a_1+(1-I_1)b_1)E(\xi_1)+(a_2+(1-I_2)b_2)E(\xi_2)+v\\&=\frac{a_1+(1-I_1)b_1}{1-(1-I_1)p_1}+\frac{a_2+(1-I_2)b_2}{1-(1-I_2)p_2}+v+(1-I_3)b_3\end{aligned}$$

其中，a_1,a_2分别表示零配件 1 和零配件 2 的购买单价，b_1,b_2,b_3分别表示零配件 1、零配件 2 和成品的检测成本，v表示成品的装配成本。

故制造出一件合格成品所需花费C的均值$E(C)$表达式如下：

$$\begin{aligned}E(C)&=E(C_0)E(\lambda)+eI_3(E(\lambda)-1)\\&=\frac{\frac{a_1+(1-I_1)b_1}{1-(1-I_1)p_1}+\frac{a_2+(1-I_2)b_2}{1-(1-I_2)p_2}+v+(1-I_3)b_3}{(1-I_1p_1)(1-I_1p_2)(1-p_3)}+\\&\quad eI_3\left(\frac{1}{(1-I_1p_1)(1-I_1p_2)(1-p_3)}-1\right)\end{aligned}$$

其中e表示调换损失。

6.2 拆解不合格成品下表达式的求解——以决策编号 008 为例

基于步骤(1)中不检测零配件 1 和零配件 2 的质量直接投入成品装配使用，且不对成品进行检测，顾客收到的成品有可能为合格品，也有可能为次品。对于顾客反映为次品的成品进行退回拆解，并检测零配件 1 和零配件 2。假设对被检测为次品的零配件进行更换时，新更

换的零配件必须经过检测合格后方能再次投入成品装配步骤使用。以下分五类考虑顾客收到的成品情况，分别为：顾客收到合格成品、零配件 1 和零配件 2 均为合格品但装配不合格致使顾客收到次品、零配件 1 合格但零配件 2 不合格致使顾客收到次品、零配件 2 合格但零配件 1 不合格致使顾客收到次品、零配件 1 和零配件 2 均不合格致使顾客收到次品。计算五类情况出现的概率并分析在相应检测结果条件下更换零配件、装配、调换至顾客收到合格成品需要的总花费。

(1)顾客收到合格成品

此类事件发生的条件为零配件 1 合格、零配件 2 合格、成品第一次装配正确，三个分事件互不相关，故顾客收到合格成品的概率P_1为：

$$\begin{aligned} P_1 &= P(\text{零配件 1 合格})P(\text{零配件 2 合格})P(\text{成品第一次装配正确}) \\ &= (1-p_1)(1-p_2)(1-p_3) \end{aligned}$$

顾客收到合格成品的总花费C_1为：

$$C_1 = a_1 + a_2 + v$$

(2)零配件 1 和零配件 2 均为合格品但装配不合格致使顾客收到次品

此类事件发生的条件为零配件 1 合格、零配件 2 合格、成品第一次装配不正确，三个分事件互不相关，故此类事件发生的概率P_2为：

$$\begin{aligned} P_2 &= P(\text{零配件 1 合格})P(\text{零配件 2 合格})P(\text{成品第一次装配不正确}) \\ &= (1-p_1)(1-p_2)p_3 \end{aligned}$$

在此类事件发生的条件下顾客最终得到 1 个合格品只需检测零配件 1 和零配件 2 各 1 次。

与 6.1 同理，在零配件均为合格品的条件下，成品出现第一件合格品时制造出的成品的个数λ服从几何分布：

$$\lambda \sim G(1-p_3)$$

又因成品不经检测即出售，故在检测零配件之后顾客收到第一件合格品时企业经历了$(E(\lambda)-1)$次产品调换和拆解，加上零配件检测前的 1 次产品调换和拆解，共需$E(\lambda)$次产品调换和拆解。故此类情况的条件下，顾客最终得到 1 个合格成品的企业总花费C_2为：

$$\begin{aligned} C_2 &= a_1 + a_2 + v + b_1 + b_2 + (v+e+s)E(\lambda) \\ &= a_1 + a_2 + v + b_1 + b_2 + \frac{v+e+s}{1-p_3} \end{aligned}$$

其中，e表示调换损失，s表示拆解费用。

(3)零配件 1 合格但零配件 2 不合格致使顾客收到次品

此类事件发生的条件为零配件 1 合格、零配件 2 不合格，两个分事件互不相关，故此类事件发生的概率P_3为：

$$\begin{aligned} P_3 &= P(\text{零配件 1 合格})P(\text{零配件 2 不合格}) \\ &= (1-p_1)p_2 \end{aligned}$$

在此类事件发生的条件下顾客最终得到 1 个合格品只需检测 1 次零配件 1，且在零配件 1 为合格品的条件下得到一个合格成品还需 1 个经检测为正品的零配件和正确的装配过程。

与 2.1 同理，生产出一个合格零配件 2 所需要的生产次数μ满足如下分布：

$$\mu \sim G(1-p_2)$$

中国大学生在线 dxs.moe.gov.cn

故更换一个合格的零配件 2 共需进行$E(\mu)$次零配件 2 生产与检测。

与(2)同理，顾客最终收到一个合格成品时企业共需进行$E(\lambda)$次产品调换和拆解。故此类情况的条件下，顾客最终得到 1 个合格成品的企业总花费C_3为：

$$C_3 = a_1 + a_2 + v + b_1 + b_2 + (a_2 + b_2)E(\mu) + (v + e + s)E(\lambda)$$
$$= a_1 + a_2 + v + b_1 + b_2 + \frac{a_2 + b_2}{1 - p_2} + \frac{v + e + s}{1 - p_3}$$

(4)零配件 2 合格但零配件 1 不合格致使顾客收到次品

此类情况与(3)同理，故此类事件发生的概率P_4与在此类情况的条件下顾客最终得到 1 件合格成品的企业总花费C_4为：

$$P_4 = P(\text{零配件 1 不合格})P(\text{零配件 2 合格})$$
$$= p_1(1 - p_2)$$
$$C_4 = a_1 + a_2 + v + b_1 + b_2 + (a_1 + b_1)E(\xi) + (v + e + s)E(\lambda)$$
$$= a_1 + a_2 + v + b_1 + b_2 + \frac{a_1 + b_1}{1 - p_1} + \frac{v + e + s}{1 - p_3}$$

(5)零配件 1 和零配件 2 均不合格致使顾客收到次品

此类事件发生的条件为零配件 1 不合格、零配件 2 不合格，两个分事件互不相关，故此类事件发生的概率P_5为：

$$P_5 = P(\text{零配件 1 不合格})P(\text{零配件 2 不合格})$$
$$= p_1 p_2$$

此类情况的条件下顾客最终得到 1 个合格成品的企业总花费C_5为(3)与(4)的结合，故有：

$$C_5 = a_1 + a_2 + v + b_1 + b_2 + (a_1 + b_1)E(\xi) + (a_2 + b_2)E(\mu) + (v + e + s)E(\lambda)$$
$$= a_1 + a_2 + v + b_1 + b_2 + \frac{a_1 + b_1}{1 - p_1} + \frac{a_2 + b_2}{1 - p_2} + \frac{v + e + s}{1 - p_3}$$

- 由期望计算公式可知在决策编号 008 的条件下成品第一次合格时生产所需**总花费$E(C)$**为：

$$E(C) = C_1P_1 + C_2P_2 + C_3P_3 + C_4P_4 + C_5P_5$$
$$= a_1 + a_2 + v + \left(b_1 + b_2 + \frac{v + e + s}{1 - p_3}\right)[1 - (1 - p_1)(1 - p_2)(1 - p_3)]$$
$$+ \frac{a_1 + b_1}{1 - p_1}p_1 + \frac{a_2 + b_2}{1 - p_2}p_2$$

6.3 决策编号 001-016 生产每个合格成品花销的期望表达式求解结果

对于编号 001-008 对应的拆解不合格成品分析及分类思路与 2.2 同理；对于编号 009-016 对应的情况，我们在 2.1 中已求解出表达通式，故决策编号 001-016 生产每个合格成品花销的期望表达式求解结果可表示如下表：

表 3　16 种决策方案生成单件合格成品总花销期望表达式

编号	生产 1 个合格成品总花销的期望表达式
001	$E(C) = \frac{a_1 + b_1}{1 - p_1} + \frac{a_2 + b_2}{1 - p_2} + \frac{v + b_3}{1 - p_3} + \frac{sp_3}{1 - p_3}$

中国大学生在线 dxs.moe.gov.cn

002	$E(C)=\frac{a_1+b_1}{1-p_1}+\frac{a_2+b_2}{1-p_2}+\frac{v}{1-p_3}+\frac{(e+s)p_3}{1-p_3}$
003	$E(C)=\frac{a_1+b_1}{1-p_1}+a_2+v+b_3+(p_2+p_3-p_2p_3)\left(b_2+\frac{v+b_3+s}{1-p_3}\right)+p_2\frac{a_2+b_2}{1-p_2}$
004	$E(C)=\frac{a_1+b_1}{1-p_1}+a_2+v+(p_2+p_3-p_2p_3)\left(b_2+\frac{v+e+s}{1-p_3}\right)+p_2\frac{a_2+b_2}{1-p_2}$
005	$E(C)=\frac{a_2+b_2}{1-p_2}+a_1+v+b_3+(p_1+p_3-p_1p_3)\left(b_1+\frac{v+b_3+s}{1-p_3}\right)+p_1\frac{a_1+b_1}{1-p_1}$
006	$E(C)=\frac{a_2+b_2}{1-p_2}+a_1+v+(p_1+p_3-p_1p_3)\left(b_1+\frac{v+e+s}{1-p_3}\right)+p_1\frac{a_1+b_1}{1-p_1}$
007	$E(C)=a_1+a_2+v+b_3+\left(b_1+b_2+\frac{v+b_3+s}{1-p_3}\right)(1-(1-p_1)(1-p_2)(1-p_3))+p_1\frac{a_1+b_1}{1-p_1}+p_2\frac{a_2+b_2}{1-p_2}$
008	$E(C)=a_1+a_2+v+\left(b_1+b_2+\frac{v+e+s}{1-p_3}\right)(1-(1-p_1)(1-p_2)(1-p_3))+p_1\frac{a_1+b_1}{1-p_1}+p_2\frac{a_2+b_2}{1-p_2}$
009	$E(C)=\frac{\frac{a_1+b_1}{1-p_1}+\frac{a_2+b_2}{1-p_2}+v+b_3}{1-p_3}$
010	$E(C)=\frac{\frac{a_1+b_1}{1-p_1}+\frac{a_2+b_2}{1-p_2}+v}{1-p_3}+\frac{ep_3}{1-p_3}$
011	$E(C)=\frac{\frac{a_1+b_1}{1-p_1}+a_2+v+b_3}{1-(p_2+(1-p_2)p_3)}$
012	$E(C)=\frac{\frac{a_1+b_1}{1-p_1}+a_2+v}{1-(p_2+(1-p_2)p_3)}+\frac{e(p_2+(1-p_2)p_3)}{1-(p_2+(1-p_2)p_3)}$
013	$E(C)=\frac{\frac{a_2+b_2}{1-p_2}+a_1+v+b_3}{1-(p_1+(1-p_1)p_3)}$
014	$E(C)=\frac{\frac{a_2+b_2}{1-p_2}+a_1+v}{1-(p_1+(1-p_1)p_3)}+\frac{e(p_1+(1-p_1)p_3)}{1-(p_1+(1-p_1)p_3)}$
015	$E(C)=\frac{a_1+a_2+v+b_3}{(1-p_1)(1-p_2)(1-p_3)}$
016	$E(C)=\frac{a_1+a_2+v}{(1-p_1)(1-p_2)(1-p_3)}+e(\frac{1}{(1-p_1)(1-p_2)(1-p_3)}-1)$

6.4 具体情况对应决策方案的求解

在成品市场售价P为定值的情况下，以利润V最大化作为决策依据，又有利润=售价-成本，故有：

$$max\,E(V)=max\,E(P-C)=max\,(P-E(C))$$

由于P及$E(C)$中各次品率、购买单价、装配成本、检测成本、调换损失、拆解费用在6种情况下的取值均已知，所以可在R软件中分别对6种情况遍历16种决策方案对应的$E(V)$表达式，筛选出16种决策方案对应结果中的最大值，以每种情况下$E(V)$最大值相应决策方

案作为该情况的最终决策方案。$E(V)$遍历结果如下表：

表 4　6 种情况 16 种决策方案对应的利润期望值表

决策编号	情况 1	情况 2	情况 3	情况 4	情况 5	情况 6
001	15.44	9.75	15.44	14.75	9.47	16.00
002	18.11	12.00	15.44	9.75	10.58	18.63
003	16.47	8.87	16.47	12.79	7.59	16.26
004	18.84	10.52	13.77	2.19	7.10	18.54
005	15.66	8.23	15.66	12.79	14.65	15.36
006	18.03	9.88	12.96	2.19	14.96	17.64
007	16.43	7.27	16.43	11.09	11.66	15.51
008	18.53	8.44	11.30	-3.99	10.54	17.46
009	12.67	2.56	12.67	8.50	5.91	16.61
010	15.33	4.81	12.67	3.50	7.02	19.24
011	14.44	2.09	14.44	5.61	1.37	19.09
012	16.73	3.40	11.10	-8.14	0.26	21.33
013	11.14	-5.33	11.14	0.14	11.86	17.10
014	13.44	-4.02	7.81	-13.61	11.99	19.35
015	13.48	-4.55	13.48	-2.60	9.70	19.84
016	15.36	-4.41	6.44	-27.28	7.36	21.68

表中标黄部分表示每种情况下单件合格成品利润期望的最大值，其对应的决策编号即为该情况下的最终决策方案。

以**单件合格成品利润期望最大化**为**决策依据**，**单件合格成品利润期望值**为**决策指标**，可整理每种情况最终**决策方案**如下表：

表 5　6 种情况最终决策方案及决策指标值表

情况	方案	决策指标终值
1	检测零配件 1，不检测零配件 2，不检测成品，拆解不合格成品	18.84
2	检测零配件 1，检测零配件 2，不检测成品，拆解不合格成品	12.00
3	检测零配件 1，不检测零配件 2，检测成品，拆解不合格成品	16.47
4	检测零配件 1，检测零配件 2，检测成品，拆解不合格成品	14.75
5	不检测零配件 1，检测零配件 2，不检测成品，拆解不合格成品	14.96
6	不检测零配件 1，不检测零配件 2，不检测成品，不拆解不合格成品	21.68

表中决策指标终值列对应的数据单位均为元/件。

七、问题三模型建立及求解

本问题涉及的多工序制造系统包括零件采购与检测、半成品装配与检测、成品装配与检

中国大学生在线 dxs.moe.gov.cn

测以及成品及半成品的拆解决策四个主要阶段。问题目标是通过合理设计检测与回收策略，最大化系统的期望利润。期望利润的需要考虑成品销售收益、零件回收收益，以及系统在检测、装配、拆解中的相关成本。为此，我们将问题转化为一个多阶段决策问题，建立**状态-决策模型**，并通过动态规划算法来求解最优策略。

7.1 模型建立

7.1.1 状态变量与决策变量定义

为了构建状态-决策模型，首先需要明确该问题背景下不同阶段的状态变量与决策变量定义。

(1)状态变量定义

系统的状态 S 描述了当前零件、半成品和成品的质量状况及其检测结果。具体而言，状态变量包括：

①零件状态

对于每个零件i，其状态由以下两个维度构成

- 检测状态d_i：零件i是否已检测，

$$d_i = \begin{cases} 0，未检测 \\ 1，检测 \end{cases}$$

- 质量状态q_i：零件i是否为次品，0 表示合格，1 表示次品。未检测时，零件状态通过次品率p_i=0.1 进行描述，检测后其状态变为确定的质量状态。

$$q_i = \begin{cases} 0，合格 \\ 1，次品 \end{cases}$$

最终有零件状态变量S_{part_i}=(d_i, q_i)

②半成品和成品状态

半成品和成品的状态同样由检测状态和质量状态构成。与零件类似，未检测的半成品或成品状态由其次品率表示，检测后状态转变为确定值。

半成品状态:$S_{semi_j} = \left(d_{semi_j}, q_{semi_j}, r_{semi_j}\right)$：对于每个半成品$j$，若$d_{semi_j} = 0$，且组成零件均合格，次品率为$p_{semi} = 0.1$；若$d_{semi_i} = 1$，次品状态$q_{semi_i}$确定为 0(合格)或 1(次品);$r_{semi_j} \in \{0,1\}$，其中，1 表示拆解操作，0 表示未拆解。

成品状态：$S_{final} = \left(d_{final}, q_{final}, r_{final}\right)$：对于每个成品，若$d_{final} = 0$，且组成半成品均合格，次品率为$p_{final} = 0.1$；若$d_{final} = 1$，次品状态$q_{final}$确定为 0(合格)或 1(次品)；$r_{final} \in \{0,1\}$，其中，1 表示拆解操作，0 表示未拆解。

③状态变量总结

$$S = \left(S_{part_1}, \ldots, S_{part_8}, S_{semi_1}, S_{semi_2}, S_{semi_3}, S_{final}\right)$$

(2)决策变量定义

决策变量用于描述企业在每个阶段的操作选择，包括是否检测零件、半成品和成品，以及是否拆解不合格的半成品或成品。

①检测决策变量

- 零件检测决策变量x_i：是否对零件i进行检测，

中国大学生在线 dxs.moe.gov.cn

$$x_i = \begin{cases} 0，对零件i跳过检测 \\ 1，进行检测 \end{cases}$$

检测后更新状态(d_i, q_i)。

- 半成品检测决策变量x_{semi_j}：

$$x_{semi_j} = \begin{cases} 0，对半成品j跳过检测 \\ 1，决定检测 \end{cases}$$

- 成品检测变量x_{final}：

$$x_{final} = \begin{cases} 0，对成品跳过检测 \\ 1，决定检测 \end{cases}$$

②拆解决策变量

对于不合格的半成品和成品，企业可以选择拆解。值得注意的是，当成品拆解时，如果其中的半成品合格，合格的半成品将不再被拆解，因为合格的半成品比其组成零件的回收收益更高(相较于对应零件增加半成品检测成本和装配成本)。

- 半成品拆解决策变量 y_{semi_j}：1 表示拆解次品半成品，0 表示不拆解。

$$y_{semi_j} = \begin{cases} 0，不拆解次品半成品j \\ 1，拆解 \end{cases}$$

- 成品拆解决策变量 y_{final}：1 表示拆解次品成品，0 表示不拆解。

$$y_{final} = \begin{cases} 0，不拆解次品成品 \\ 1，拆解 \end{cases}$$

7.2 成本函数

成本函数$C(a_t)$包括零件采购成本、检测成本、装配成本和拆解成本。分别定义如下：

7.2.1 采购成本$C_{purchase}$

假设零件i的采购成本为 p_i,则总采购成本为：

$$C_{purchase} = \sum_{i=1}^{8} n_i \times p_i$$

其中n_i表示采购零件i的数量。

7.2.2 检测成本$C_{inspect}$

零件、半成品和成品的检测成本分别为：

$$C_{inspect} = \sum_{i=1}^{8} n_i \times c_{inspect_part_i} \times x_i + \sum_{j=1}^{3} n_{semi_j} \times c_{inspect_semi_i} \times x_{semi_j}$$

$$+ n_{final} \times c_{inspect_final} \times x_{final}$$

中国大学生在线 dxs.moe.gov.cn

其中，$c_{inspect_part_i}$、$c_{inspect_part_j}$、$c_{inspect_final}$分别指代零件i、半成品j和成品的单独检测成本；n_i、n_{semi_j}、n_{final}分别表示零件i、半成品j和成品的数量；x_i、x_{semi_j}、x_{final}分别表示零件i、半成品j和成品是否检测的决策变量。

7.2.3 装配成本$C_{assemble}$

零件装配为半成品、半成品装配为成品的成本分别为：

$$C_{assemble_semi} = \sum_{j=1}^{3} n_{semi_j} \times c_{a_j}$$

$$C_{assemble_final} = n_{final} \times c_{a_final}$$

其中，c_{a_j}、c_{a_final}分别表示零件装配为半成品j、半成品装配为成品的单个装配成本。

7.2.4 拆解成本$C_{disassemble}$

拆解半成品和成品的成本为：

$$C_{disassemble_semi} = n_{semi_j} \times c_{d_j} \times r_{semi_j}$$

$$C_{disassemble_final} = n_{final} \times c_{d_final} \times r_{final}$$

其中，c_{d_j}、c_{d_final}分别表示半成品j和成品的单个拆解成本。

7.2.5 成本函数总结

总成本$C(a_t)$的表达式为：

$$C(a_t) = C_{purchase}+C_{inspect}+C_{assemble_semi}+C_{assemble_final}+C_{disassemble_semi}+C_{disassemble_final}$$

7.3 收益函数

收益函数$R(S_t, a_t)$包括成品销售收益和回收收益。

7.3.1 成品销售收入 R_{final}

$$R_{final} = n_{final} \times P \times q_{final}$$

其中，P表示成品售价，$q_{final} = 1$ 表示仅合格成品能带来销售收益。

7.3.2 回收收益$R_{recycle}$

拆解回收零件和半成品的收益为：

$$R_{recycle_part} = \sum_{i=1}^{8} n_i \times (p_i + c_{inspect_part_i}) \times x_i \times (1 - q_i)$$

$$R_{recycl_semi} = \sum_{j=1}^{3} n_{semi_j} \times \Big(\sum_{i \in semi_j} p_i + c_{inspect_semi_j} + c_{a_j} \Big) \times x_{semi_j} \times (1 - q_{semi_j})$$

7.3.3 收益函数总结

$$R(S_t, a_t) = R_{final} + R_{recycle_part} + R_{recycl_semi}$$

7.4 模型总结

综上所述，系统的目标是找到使得总收益最大化的决策路径a_t。即，我们希望最大化如

中国大学生在线 dxs.moe.edu.cn

下目标函数，即：

$$max\ V(S_t, a_t) = \max_{a_t} \left[R(S_t, a_t) - C(a_t) + E[V(S_{t+1}, a_{t+1})]\right]$$

其中：

- $V(S_t, a_t)$为状态S_t的最优利润：$V(S_t, a_t) = R(S_t, a_t) - C(a_t)$;
- $R(S_t, a_t)$为当前决策a_t产生的收益；
- $C(a_t)$为当前决策的成本；
- $E[V(S_{t+1})]$为未来状态S_{t+1}的期望利润。

7.5 算法求解

7.5.1 算法介绍

动态规划是一种常用于解决多阶段决策问题的优化算法。其核心思想是将原问题分解为一系列更小的子问题，并计算这些子问题的最优解，然后根据子问题的最优解构建整个问题的解。动态规划特别适合具有重叠子问题和最优子结构性质的问题[2]。

在本问题中，动态规划算法的任务是通过求解每个状态下的最优决策和最大期望收益。在每个状态下，系统可以做出若干决策，每个决策会导致状态的变化，并产生相应的收益与成本。算法通过计算各个状态的最优值，并利用记忆化技术储存中间结果，以提高求解效率。

7.5.2 状态转移与决策

每个决策a_t会引起状态的转移，即系统从当前状态S_t转移到下一状态S_{t+1}。状态转移过程遵循以下规则：

(1)**零件检测**：检测零件i后，系统更新零件的状态，则是否为次品确定；若不对零件进行检测，即

$$d_i = 1, q_i = B(1, p)$$

其中，p=0.1。

此时，系统转移到新的状态S_{t+1},该状态反映零件的检测结果。

(2)**半成品检测**：检测半成品j后，如果所有零件均为合格品，则半成品有10%的次品概率：$q_{semi_i} \sim B(1, p)$,当组成零件确定均合格；如果任何零件为次品，则半成品直接被判定为次品，系统状态随之转移。

(3)**成品检测**：成品由三个半成品组成，即使所有半成品合格，成品仍有10%的次品概率：$q_{semi_i} \sim B(1, p)$,当组成半成品确定均合格. 该决策会更新成品的状态，决定成品是否可以销售。

(4)**拆解决策**：如果检测出次品，系统可以选择对不合格品进行拆解，回收合格半成品或者零件，拆解后，系统转移到新的状态S_{t+1},记录拆解操作的收益和成本。

- 通过状态转移函数$T(S_t, a_t)$,系统能够根据当前的决策a_t更新状态，并进入下一个决策周期。

7.5.3 求解步骤

可以通过以下步骤求解每个状态下的最优决策路径与最大期望收益：

(1)**初始化状态**

中国大学生在线 dxs.moe.gov.cn

初始化系统的状态S_0，此时所有零件、半成品和成品均未检测，且初始收益为0。

(2)递归求解

从初始状态S_0开始，递归地计算每个状态下的最大期望收益。具体步骤如下：

①遍历每个可能的决策a_t：针对当前状态,系统可以选择检测某个零件、检测半成品、检测成品，或者拆解次品。

②状态转移：根据当前的决策a_t,通过状态转移函数$S_{t+1} = T(S_t, a_t)$进入下一个状态S_{t+1}。

③计算收益与成本：在状态S_t下，根据决策a_t产生的收益$R(S_t, a_t)$和成本$C(a_t)$。

④调用递归函数：计算在新状态S_{t+1}下的最优期望收益$E[V(S_{t+1})]$。

⑤求最大收益：通过遍历所有可能的决策a_t,选择使得$V(S_t) = max_{a_t}[R(S_t, a_t) - C(a_t) + E[V(S_{t+1})]]$最大的决策，并保存该决策路径。

(3)递归终止条件

系统达到最终状态，即所有零件、半成品和成品均已检测，或达到预设的递归深度，在终止条件下，返回计算结果。

7.5.4 求解结果

修改未检测的零件、半成品(组成零件均合格)、成品(半成品均合格)是否为次品的概率分布后，再次运行第三问中的求解程序，最终求解得到高稳定性最优决策如下表及文字所示：

表6　最终决策方案表

零配件1	零配件2	零配件3	零配件4	零配件5	零配件6
检测	检测	检测	检测	检测	检测
零配件7	零配件8	半成品1	半成品2	半成品3	成品
检测	检测	检测	检测	检测	检测

针对成品的拆解决策和半成品拆解决策，对于检测后不合格成品，选择拆解成为半成品。对于检测后的半成品，若合格，则不再进行拆解；若为次品，则进一步拆解成为零件。

在此最优决策下，**所得的利润期望为 56.7 元**。

八、问题四模型建立及求解

8.1 基于贝叶斯统计的估计

在问题四的假设下，为衡量次品率真值的不确定性，本文运用**贝叶斯统计**[3]的思想，估计出次品率服从的后验分布，再计算出各策略下生产一件合格成品花费的均值，为企业计算出最佳策略。

假定通过简单随机抽样抽取的n个样本均独立同分布于$B(1,p)$，p的先验分布$\pi(p)$的选取利用共轭先验分布策略，选为二项分布的共轭先验分布$Beta$分布$Beta(\alpha,\beta)$,本文中选定均匀分布作为先验分布，即$\alpha = \beta = 1$,p的后验分布$\pi(p|x)$满足:

$$\pi(p|x) \propto \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}p^{\beta-1}\binom{n}{k}p^k(1-p)^{n-k} \propto p^{\alpha+k-1}(1-p)^{\beta+n-k-1}$$

可见，p的后验分布服从$Beta(\alpha+k,\beta+n-k)$。$\alpha=\beta=1$时，为$Beta(1+k,1+n-k)$。

假定问题二和问题三中的次品率通过贝叶斯估计中的最大后验估计得到，即有：

中国大学生在线 dxs.moe.gov.cn

$$\hat{p}=\frac{\alpha+k-1}{\alpha+\beta+n-2}=\frac{k}{n}$$

故p的后验分布为$Beta(1+n\hat{p},1+n-n\hat{p})$。

8.2 求解问题二

回忆第二问中每生产一件合格单品利润期望的表达式，不难发现在给定策略之下，利润的期望仅与零配件 1、零配件 2、成品的次品率p_1,p_2,p_3有关，故可将策略i的利润期望写成p_1,p_2,p_3的函数$h_i(p_1,p_2,p_3)$。假定p_1,p_2,p_3的密度函数分别为$\varphi_1(p_1),\varphi_2(p_2),\varphi_3(p_3)$，则在上述条件之下利润期望$E(h_i(p_1,p_2,p_3))$满足：

$$E\big(h_i(p_1,p_2,p_3)\big)=\int_0^1\int_0^1\int_0^1 h_i(p_1,p_2,p_3)\,\varphi_1(p_1)\varphi_2(p_2)\varphi_3(p_3)dp_1dp_2dp_3$$

由于不同零件检测单价的不同，不同零件的检测样本量n并不完全相同。在第一问中，得到了样本量n的决策函数$g(n)$，即：

$$n_i=argming(n_i)=\underset{n}{\operatorname{argmin}}\,(w_1c_in_i-w_2\int_{0.1}^{1}\pi(p)\,K(n_i,p)dp)\ (i=1,2,3)$$

其中n_1,n_2,n_3分别表示零配件 1、零配件 2、成品的检验样本量，决定p_1,p_2,p_3的后验分布；c_1,c_2,c_3分别表示零配件 1、零配件 2、成品的检测成本；w_1,w_2为实现给定的权重，衡量企业对于检测费用和检测效果的看重程度；$K(n_i,p)$为问题一中假设检验的功效函数，$\pi(p)$为p的先验分布，已假定为均匀分布。

综合上述信息，得到$E\big(h_i(p_1,p_2,p_3)\big)$的计算公式如下：

$$\begin{cases}E\big(h_i(p_1,p_2,p_3)\big)=\int_0^1\int_0^1\int_0^1 h_i(p_1,p_2,p_3)\,\varphi_1(p_1)\varphi_2(p_2)\varphi_3(p_3)dp_1dp_2dp_3\\ \varphi_i(p_i)=\dfrac{\Gamma(2+n_i)}{\Gamma(1+n_i\hat{p}_i)\Gamma(1+n_i-n_i\hat{p}_i)}p_i^{n_i\hat{p}_i}(1-p_i)^{n_i-n_i\hat{p}_i}\\ n_i=\underset{n}{\operatorname{argmin}}\,(w_1c_in_i-w_2\int_{0.1}^{1}\pi(p)\,K(n_i,p)dp)\ (i=1,2,3)\end{cases}$$

运用R语言计算积分，得到在问题四假设下，各种决策策略生成一个合格成品的平均利润如下表所示：

表 7　问题二 6 种情况 16 种决策方案对应的利润期望值表

决策编号	情况 1	情况 2	情况 3	情况 4	情况 5	情况 6
001	14.99	9.24	14.99	14.29	8.96	15.20
002	17.62	11.45	14.69	8.94	9.98	17.76
003	15.87	8.25	15.87	12.24	6.99	14.98
004	18.18	9.84	12.65	1.13	6.36	17.12
005	15.07	7.62	15.07	12.24	13.92	14.09
006	17.38	9.21	11.86	1.13	14.08	16.24
007	15.68	6.57	15.68	10.49	10.86	13.77
008	17.70	7.67	9.87	-5.21	9.55	15.53
009	11.90	1.60	11.90	7.65	4.99	15.91

中国大学生在线 dxs.moe.gov.cn

010	14.53	3.81	11.60	2.30	6.01	18.47
011	13.52	0.89	13.52	4.50	0.08	18.27
012	15.74	2.11	9.51	-10.13	-1.25	20.36
013	9.95	-6.98	9.95	-1.36	10.68	16.07
014	12.17	-5.76	5.95	-15.99	10.60	18.16
015	12.19	-6.38	12.19	-4.37	8.30	18.75
016	13.95	-6.42	4.03	-30.71	5.58	20.34

表中涂色部分表示每种情况的平均利润最大值，黄色表示平均利润最大值出现的决策与第二问中一致，绿色表示平均利润最大值出现的决策与第二问中不一致。

对应最佳决策方案如下表所示：

表 8　问题二 6 种情况最终决策方案及决策指标值表

情况	方案	决策指标终值
1	检测零配件 1，不检测零配件 2，不检测成品，拆解不合格成品	18.18
2	检测零配件 1，检测零配件 2，不检测成品，拆解不合格成品	11.45
3	检测零配件 1，不检测零配件 2，检测成品，拆解不合格成品	15.87
4	检测零配件 1，检测零配件 2，检测成品，拆解不合格成品	14.29
5	不检测零配件 1，检测零配件 2，不检测成品，拆解不合格成品	14.08
6	检测零配件 1，不检测零配件 2，不检测成品，不拆解不合格成品	20.36

表中决策指标终值列对应的数据单位均为元/件。

可知，第四问中采用贝叶斯估计求出次品率的后验分布得到的利润均值与第二问中通过点估计次品率得到的利润均值差距不大，对几乎最佳决策几乎没有影响，仅在情况 6 出现了决策的不同，且问题二中的最佳决策 016 在本问中的表现与本问中的最佳决策差距较小。

8.3 求解问题三

在生产流程中，零件、半成品和成品的次品率具有不确定性，问题三中，条件给定为未检测的零件、半成品(组成零件均合格)、成品(半成品均合格)的次品率均为定值。然而，在问题四的背景下次品率不再是定值，而是服从$Beta$分布，$p\sim Beta(1+n\hat{p},1+n-n\hat{p})$。记未检测的零件、半成品(组成零件均合格)、成品(半成品均合格)是否为次品服从随机变量X，由于p是不确定的，不能直接使用单一的伯努利分布来描述X的分布，只能通过对p的不确定性(即$Beta$分布)进行积分来获得X的分布。

进一步推导如下，

已知，

$$\begin{cases} X\mid p\sim B(1,p) \\ p\sim Beta(1+n\hat{p},1+n-n\hat{p}) \end{cases}$$

由于p是随机变量，零件X的分布需要将p的不确定性（$Beta$分布)整合进来。因此，X的分布是通过对p的所有可能取值进行积分得到的：

中国大学生在线 dxs.moe.gov.cn

$$P(X=1)=\int_0^1 P(X=1|p)\cdot\frac{p^{(1+n\hat{p})-1}(1-p)^{(1+n-n\hat{p})-1}\Gamma(2+n)}{\Gamma(1+n\hat{p})\Gamma(1+n-n\hat{p})}dp$$

其中，$P(X=1|p)$就是伯努利分布的概率p，

因此有：

$$P(X=1)=\int_0^1 p\cdot\frac{p^{n\hat{p}}(1-p)^{n-n\hat{p}}\Gamma(2+n)}{\Gamma(1+n\hat{p})\Gamma(1+n-n\hat{p})}dp\triangleq p_0$$

同理，对于$X=0$的情况：

$$P(X=0)=\int_0^1 (1-p)\cdot\frac{p^{n\hat{p}}(1-p)^{n-n\hat{p}}\Gamma(2+n)}{\Gamma(1+n\hat{p})\Gamma(1+n-n\hat{p})}dp=1-p_0$$

综上可得，

$$X\sim \mathrm{B}(1,p_0)$$

其中，n的选择同 8.2，$\hat{p}=\frac{k}{n}$，在本问题下，$\hat{p}$=0.1。

修改未检测的零件、半成品(组成零件均合格)、成品(半成品均合格)是否为次品的概率分布后，再次运行第三问中的求解程序，最终求解得到高稳定性最优决策如下表及文字所示。

表 9　问题三最终决策方案表

零配件 1	零配件 2	零配件 3	零配件 4	零配件 5	零配件 6
检测	检测	检测	检测	检测	检测
零配件 7	零配件 8	半成品 1	半成品 2	半成品 3	成品
检测	检测	检测	检测	检测	检测

针对成品的拆解决策和半成品拆解决策，对于检测后不合格成品，选择拆解成为半成品。对于检测后的半成品，若合格，则不再进行拆解；若为次品，则进一步拆解成为零件。

在此最优决策下，**所得的利润期望为 54.2 元**。

九、模型灵敏度分析

第四问运用了贝叶斯估计的方法，贝叶斯估计中的先验分布存在一定的主观成分。第四问假定先验分布$beta$分布$Beta(\alpha,\beta)$中的超参数α和β均等于 1，即使得$beta$分布退化为了均匀分布。应让α和β在一定范围内变动，观察超参数的改变对期望值的影响。

9.1 α变动对收益期望的影响

以第二问情况 1 中的最优决策为例，令$\beta\equiv 1,\alpha=1,2,\ldots\ldots,10$，考察期望的变动，如下图所示：

中国大学生在线 dxs.moe.gov.cn

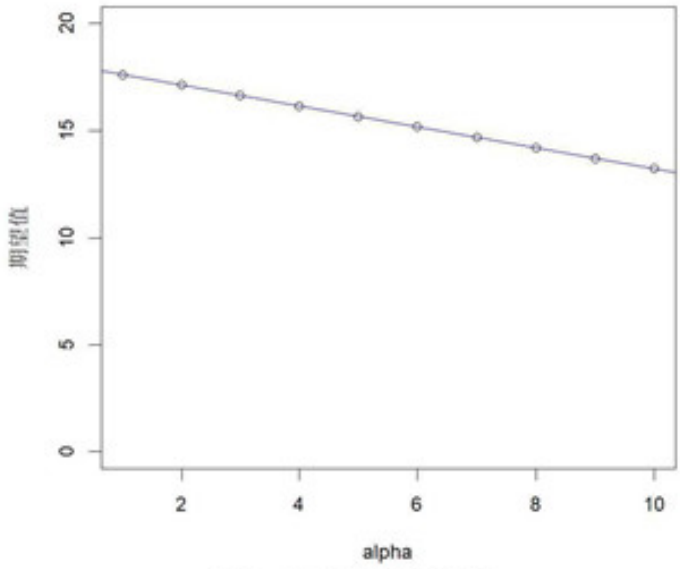

图 2　α变动对期望影响表

9.2 β变动对收益期望的影响

以第二问情况 1 中的最优决策为例，令$\alpha \equiv 1, \beta = 1,2, \ldots\ldots, 10$, 考察期望的变动，如下图所示：

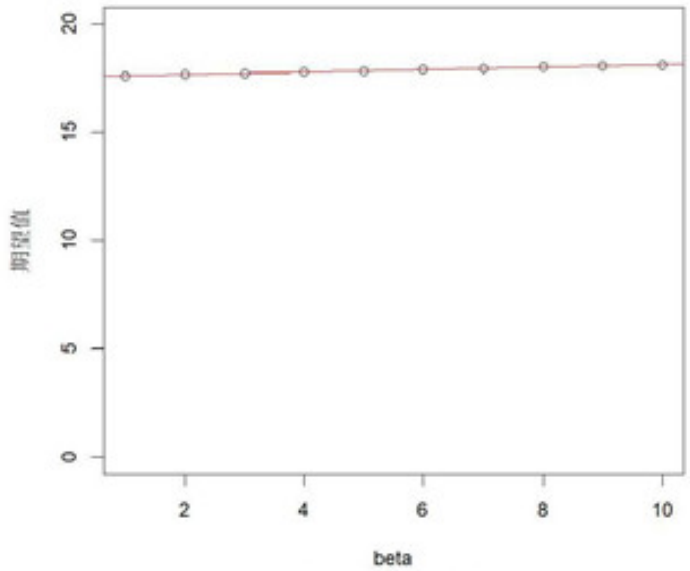

图 3　β变动对期望影响表

综合上文，不难从图像中看出，超参数的选取对收益期望的影响不大，故虽使用贝叶斯估计，但受主观因素影响较小。

十、模型评价

问题一模型评价

优点：综合考虑了假设检验效果与检测花费，得到的最佳样本量较为合理。

缺点：最佳样本量受到决策函数的参数的影响较大，具有一定的主观性。

中国大学生在线 dxs.moe.gov.cn

问题二模型评价

优点：精准计算利润期望表达式，不含有任何估计与不确定性成分，计算结果准确率高。

缺点：如果将此方法迁移到较为复杂的加工流程，则精准计算的难度增加较大。且流程的细微变化也会使得模型产生较大变化，模型不易推广。

问题三模型评价

优点：状态-决策模型结构合理，决策路径清晰。通过多阶段划分（如零件采购、半成品装配、成品检测及拆解决策），动态规划算法求解最优策略，提高了资源利用效率，确保了模型的逻辑性和适用性。

缺点：尽管动态规划提高了求解效率，但随着系统规模增大，状态空间的指数增长可能导致计算负担加重，尤其在大规模生产中，计算复杂性凸显。如何对现有算法进行改进是未来方向之一。

十一、参考文献

[1] 陈希孺,倪国熙.数理统计学教程[M].中国科学技术大学出版社,2009.

[2] 陈荣军,刘永财,黄河,等.单机供应链排序问题动态规划算法[J/OL].运筹学学报(中英文),1-9[2024-09-08].http://kns.cnki.net/kcms/detail/31.1732.O1.20240702.1235.004.html.

[3] 茆诗松,汤银才.贝叶斯统计[M].中国统计出版社:201209.330.

中国大学生在线 dxs.moe.gov.cn

附录

表 1　支撑文件目录

支撑文件名称	文件内容
问题一代码.R	问题一求解代码
问题二代码.R	问题二求解代码
问题三代码.py	问题三求解代码
问题四求解问题二代码.R	问题四求解问题二代码
问题四求解问题三代码.py	问题四求解问题三代码
灵敏度检验代码.R	灵敏度检验代码

表 2　附录目录

附录	名称
附录 1	问题一求解代码
附录 2	问题二求解代码
附录 3	问题三求解代码
附录 4	问题四求解问题二代码
附录 5	问题四求解问题三代码
附录 6	灵敏度检验代码

中国大学生在线 dxs.moe.gov.cn

附录 1　问题一求解代码

```r
# 定义 calculate_expression 函数
calculate_expression <- function(n, p) {
  u_0.95 <- qnorm(0.95)
  term1 <- 0.3 * u_0.95 / sqrt(n)
  term2 <- 0.1 - p
  term3 <- sqrt(p * (1 - p) / n)

  z_value <- (term1 + term2) / term3
  result <- 1 - pnorm(z_value)

  return(result)
}

# 初始化结果向量
x <- numeric(1000)

# 循环计算积分
for(n in 1:1000) {
  # 定义被积函数，将 n 固定到当前循环的值
  integrand <- function(p) {
    return(calculate_expression(n, p))
  }

  # 计算从 0.1 到 1 对 p 的积分
  result <- integrate(integrand, lower = 0.1, upper = 1)

  # 计算最终结果并存储
  x[n] <- -result$value * 0.999 + 0.001 * n
}
min_value <- min(x)
min_position <- which.min(x)

cat("最小值:", min_value, "\n")
cat("最小值出现的位置:", min_position, "\n")
```

附录 2　问题二求解代码

```
## 第一组数据
a=0.1
b=4
c=2
d=0.1
e=18
f=3
g=0.1
h=6
i=3
j=56
k=6
l=5
jjjc=j-(b+c)/(1-a)-(e+f)/(1-d) -(h+i)/(1-g)-l*g/(1-g)
jjbc=j-(b+c)/(1-a)-(e+f)/(1-d)-h/(1-g)-(k+l)*g/(1-g)
jjjb=j-((b+c)/(1-a)+(e+f)/(1-d)+h+i)/(1-g)
jjbb=j-((b+c)/(1-a)+(e+f)/(1-d)+h)/(1-g)-k*g/(1-g)
jbjb=j-((b+c)/(1-a)+e+h+i)/(1-(d+(1-d)*g))
jbbb=j-((b+c)/(1-a)+e+h)/(1-(d+(1-d)*g))-k*(d+(1-d)*g)/(1-(d+(1-d)*g))
bjjb=j-(b+(e+f)/(1-d)+h+i)/(1-(a+(1-a)*g))
bjbb=j-(b+(e+f)/(1-d)+h)/(1-(a+(1-a)*g))-k*(a+(1-a)*g)/(1-(a+(1-a)*g))
bbjb=j-(b+e+h+i)/(1-a)/(1-d)/(1-g)
bbbb=j-(b+e+h)/(1-a)/(1-d)/(1-g)-k*(1/(1-a)/(1-d)/(1-g)-1)
bjjc=j-(b+(e+f)/(1-d)+h+i+(a+g-a*g)*(c+(h+i+l)/(1-g))+a*(b+c)/(1-a))
jbjc=j-(e+(b+c)/(1-a)+h+i+(d+g-d*g)*(f+(h+i+l)/(1-g))+d*(e+f)/(1-d))
bbjc=j-((b+e+h+i)+((c+f)+(h+i+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
bjbc=j-(b+(e+f)/(1-d)+h+(c+(h+k+l)/(1-g))*(a+g-a*g)+(b+c)*a/(1-a))
jbbc=j-(e+(b+c)/(1-a)+h+(f+(h+k+l)/(1-g))*(d+g-d*g)+(e+f)*d/(1-d))
bbbc=j-((b+e+h)+((c+f)+(h+k+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
jjjc
jjbc
jjjb
jjbb
jbjb
jbbb
bjjb
bjbb
bbjb
bbbb
bjjc
jbjc
bbjc
```

中国大学生在线 dxs.moe.gov.cn

```
bjbc
jbbc
bbbc

## 第二组数据
a=0.2
b=4
c=2
d=0.2
e=18
f=3
g=0.2
h=6
i=3
j=56
k=6
l=5
jjjc=j-(b+c)/(1-a)-(e+f)/(1-d) -(h+i)/(1-g)-l*g/(1-g)
jjbc=j-(b+c)/(1-a)-(e+f)/(1-d)-h/(1-g)-(k+l)*g/(1-g)
jjjb=j-((b+c)/(1-a)+(e+f)/(1-d)+h+i)/(1-g)
jjbb=j-((b+c)/(1-a)+(e+f)/(1-d)+h)/(1-g)-k*g/(1-g)
jbjb=j-((b+c)/(1-a)+e+h+i)/(1-(d+(1-d)*g))
jbbb=j-((b+c)/(1-a)+e+h)/(1-(d+(1-d)*g))-k*(d+(1-d)*g)/(1-(d+(1-d)*g))
bjjb=j-(b+(e+f)/(1-d)+h+i)/(1-(a+(1-a)*g))
bjbb=j-(b+(e+f)/(1-d)+h)/(1-(a+(1-a)*g))-k*(a+(1-a)*g)/(1-(a+(1-a)*g))
bbjb=j-(b+e+h+i)/(1-a)/(1-d)/(1-g)
bbbb=j-(b+e+h)/(1-a)/(1-d)/(1-g)-k*(1/(1-a)/(1-d)/(1-g)-1)
bjjc=j-(b+(e+f)/(1-d)+h+i+(a+g-a*g)*(c+(h+i+l)/(1-g))+a*(b+c)/(1-a))
jbjc=j-(e+(b+c)/(1-a)+h+i+(d+g-d*g)*(f+(h+i+l)/(1-g))+d*(e+f)/(1-d))
bbjc=j-((b+e+h+i)+((c+f)+(h+i+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
bjbc=j-(b+(e+f)/(1-d)+h+(c+(h+k+l)/(1-g))*(a+g-a*g)+(b+c)*a/(1-a))
jbbc=j-(e+(b+c)/(1-a)+h+(f+(h+k+l)/(1-g))*(d+g-d*g)+(e+f)*d/(1-d))
bbbc=j-((b+e+h)+((c+f)+(h+k+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
jjjc
jjbc
jjjb
jjbb
jbjb
jbbb
bjjb
bjbb
bbjb
bbbb
bjjc
jbjc
bbjc
bjbc
jbbc
bbbc

## 第三组数据
a=0.1
b=4
c=2
d=0.1
e=18
f=3
g=0.1
h=6
i=3
j=56
k=30
l=5
jjjc=j-(b+c)/(1-a)-(e+f)/(1-d) -(h+i)/(1-g)-l*g/(1-g)
jjbc=j-(b+c)/(1-a)-(e+f)/(1-d)-h/(1-g)-(k+l)*g/(1-g)
jjjb=j-((b+c)/(1-a)+(e+f)/(1-d)+h+i)/(1-g)
jjbb=j-((b+c)/(1-a)+(e+f)/(1-d)+h)/(1-g)-k*g/(1-g)
jbjb=j-((b+c)/(1-a)+e+h+i)/(1-(d+(1-d)*g))
jbbb=j-((b+c)/(1-a)+e+h)/(1-(d+(1-d)*g))-k*(d+(1-d)*g)/(1-(d+(1-d)*g))
bjjb=j-(b+(e+f)/(1-d)+h+i)/(1-(a+(1-a)*g))
bjbb=j-(b+(e+f)/(1-d)+h)/(1-(a+(1-a)*g))-k*(a+(1-a)*g)/(1-(a+(1-a)*g))
bbjb=j-(b+e+h+i)/(1-a)/(1-d)/(1-g)
bbbb=j-(b+e+h)/(1-a)/(1-d)/(1-g)-k*(1/(1-a)/(1-d)/(1-g)-1)
bjjc=j-(b+(e+f)/(1-d)+h+i+(a+g-a*g)*(c+(h+i+l)/(1-g))+a*(b+c)/(1-a))
jbjc=j-(e+(b+c)/(1-a)+h+i+(d+g-d*g)*(f+(h+i+l)/(1-g))+d*(e+f)/(1-d))
bbjc=j-((b+e+h+i)+((c+f)+(h+i+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
bjbc=j-(b+(e+f)/(1-d)+h+(c+(h+k+l)/(1-g))*(a+g-a*g)+(b+c)*a/(1-a))
jbbc=j-(e+(b+c)/(1-a)+h+(f+(h+k+l)/(1-g))*(d+g-d*g)+(e+f)*d/(1-d))
bbbc=j-((b+e+h)+((c+f)+(h+k+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
jjjc
jjbc
jjjb
jjbb
jbjb
```

中国大学生在线 dxs.moe.gov.cn

```
jbbb
bjjb
bjbb
bbjb
bbbb
bjjc
jbjc
bbjc
bjbc
jbbc
bbbc

## 第四组数据
a=0.2
b=4
c=1
d=0.2
e=18
f=1
g=0.2
h=6
i=2
j=56
k=30
l=5
jjjc=j-(b+c)/(1-a)-(e+f)/(1-d) -(h+i)/(1-g)-l*g/(1-g)
jjbc=j-(b+c)/(1-a)-(e+f)/(1-d)-h/(1-g)-(k+l)*g/(1-g)
jjjb=j-((b+c)/(1-a)+(e+f)/(1-d)+h+i)/(1-g)
jjbb=j-((b+c)/(1-a)+(e+f)/(1-d)+h)/(1-g)-k*g/(1-g)
jbjb=j-((b+c)/(1-a)+e+h+i)/(1-(d+(1-d)*g))
jbbb=j-((b+c)/(1-a)+e+h)/(1-(d+(1-d)*g))-k*(d+(1-d)*g)/(1-(d+(1-d)*g))
bjjb=j-(b+(e+f)/(1-d)+h+i)/(1-(a+(1-a)*g))
bjbb=j-(b+(e+f)/(1-d)+h)/(1-(a+(1-a)*g))-k*(a+(1-a)*g)/(1-(a+(1-a)*g))
bbjb=j-(b+e+h+i)/(1-a)/(1-d)/(1-g)
bbbb=j-(b+e+h)/(1-a)/(1-d)/(1-g)-k*(1/(1-a)/(1-d)/(1-g)-1)
bjjc=j-(b+(e+f)/(1-d)+h+i+(a+g-a*g)*(c+(h+i+l)/(1-g))+a*(b+c)/(1-a))
jbjc=j-(e+(b+c)/(1-a)+h+i+(d+g-d*g)*(f+(h+i+l)/(1-g))+d*(e+f)/(1-d))
bbjc=j-((b+e+h+i)+((c+f)+(h+i+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
bjbc=j-(b+(e+f)/(1-d)+h+(c+(h+k+l)/(1-g))*(a+g-a*g)+(b+c)*a/(1-a))
jbbc=j-(e+(b+c)/(1-a)+h+(f+(h+k+l)/(1-g))*(d+g-d*g)+(e+f)*d/(1-d))
bbbc=j-((b+e+h)+((c+f)+(h+k+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
jjjc
jjbc
jjjb
jjbb
jbjb
jbbb
bjjb
bjbb
bbjb
bbbb
bjjc
jbjc
bbjc
bjbc
jbbc
bbbc

## 第五组数据
a=0.1
b=4
c=8
d=0.2
e=18
f=1
g=0.1
h=6
i=2
j=56
k=10
l=5
jjjc=j-(b+c)/(1-a)-(e+f)/(1-d) -(h+i)/(1-g)-l*g/(1-g)
jjbc=j-(b+c)/(1-a)-(e+f)/(1-d)-h/(1-g)-(k+l)*g/(1-g)
jjjb=j-((b+c)/(1-a)+(e+f)/(1-d)+h+i)/(1-g)
jjbb=j-((b+c)/(1-a)+(e+f)/(1-d)+h)/(1-g)-k*g/(1-g)
jbjb=j-((b+c)/(1-a)+e+h+i)/(1-(d+(1-d)*g))
jbbb=j-((b+c)/(1-a)+e+h)/(1-(d+(1-d)*g))-k*(d+(1-d)*g)/(1-(d+(1-d)*g))
bjjb=j-(b+(e+f)/(1-d)+h+i)/(1-(a+(1-a)*g))
bjbb=j-(b+(e+f)/(1-d)+h)/(1-(a+(1-a)*g))-k*(a+(1-a)*g)/(1-(a+(1-a)*g))
bbjb=j-(b+e+h+i)/(1-a)/(1-d)/(1-g)
bbbb=j-(b+e+h)/(1-a)/(1-d)/(1-g)-k*(1/(1-a)/(1-d)/(1-g)-1)
bjjc=j-(b+(e+f)/(1-d)+h+i+(a+g-a*g)*(c+(h+i+l)/(1-g))+a*(b+c)/(1-a))
jbjc=j-(e+(b+c)/(1-a)+h+i+(d+g-d*g)*(f+(h+i+l)/(1-g))+d*(e+f)/(1-d))
bbjc=j-((b+e+h+i)+((c+f)+(h+i+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
```

中国大学生在线 dxs.moe.gov.cn

```
bjbc=j-(b+(e+f)/(1-d)+h+(c+(h+k+l)/(1-g))*(a+g-a*g)+(b+c)*a/(1-a))
jbbc=j-(e+(b+c)/(1-a)+h+(f+(h+k+l)/(1-g))*(d+g-d*g)+(e+f)*d/(1-d))
bbbc=j-((b+e+h)+((c+f)+(h+k+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
jjjc
jjbc
jjjb
jjbb
jbjb
jbbb
bjjb
bjbb
bbjb
bbbb
bjjc
jbjc
bbjc
bjbc
jbbc
bbbc

## 第六组数据
a=0.05
b=4
c=2
d=0.05
e=18
f=3
g=0.05
h=6
i=3
j=56
k=10
l=40
jjjc=j-(b+c)/(1-a)-(e+f)/(1-d) -(h+i)/(1-g)-l*g/(1-g)
jjbc=j-(b+c)/(1-a)-(e+f)/(1-d)-h/(1-g)-(k+l)*g/(1-g)
jjjb=j-((b+c)/(1-a)+(e+f)/(1-d)+h+i)/(1-g)
jjbb=j-((b+c)/(1-a)+(e+f)/(1-d)+h)/(1-g)-k*g/(1-g)
jbjb=j-((b+c)/(1-a)+e+h+i)/(1-(d+(1-d)*g))
jbbb=j-((b+c)/(1-a)+e+h)/(1-(d+(1-d)*g))-k*(d+(1-d)*g)/(1-(d+(1-d)*g))
bjjb=j-(b+(e+f)/(1-d)+h+i)/(1-(a+(1-a)*g))
bjbb=j-(b+(e+f)/(1-d)+h)/(1-(a+(1-a)*g))-k*(a+(1-a)*g)/(1-(a+(1-a)*g))
bbjb=j-(b+e+h+i)/(1-a)/(1-d)/(1-g)
bbbb=j-(b+e+h)/(1-a)/(1-d)/(1-g)-k*(1/(1-a)/(1-d)/(1-g)-1)
bjjc=j-(b+(e+f)/(1-d)+h+i+(a+g-a*g)*(c+(h+i+l)/(1-g))+a*(b+c)/(1-a))
jbjc=j-(e+(b+c)/(1-a)+h+i+(d+g-d*g)*(f+(h+i+l)/(1-g))+d*(e+f)/(1-d))
bbjc=j-((b+e+h+i)+((c+f)+(h+i+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
bjbc=j-(b+(e+f)/(1-d)+h+(c+(h+k+l)/(1-g))*(a+g-a*g)+(b+c)*a/(1-a))
jbbc=j-(e+(b+c)/(1-a)+h+(f+(h+k+l)/(1-g))*(d+g-d*g)+(e+f)*d/(1-d))
bbbc=j-((b+e+h)+((c+f)+(h+k+l)/(1-g))*(1-(1-a)*(1-d)*(1-g))+(e+f)*d/(1-d)+(b+c)*a/(1-a))
jjjc
jjbc
jjjb
jjbb
jbjb
jbbb
bjjb
bjbb
bbjb
bbbb
bjjc
jbjc
bbjc
bjbc
jbbc
bbbc
```

附录 3　问题三求解代码

```python
# 定义零件及其属性
parts = [
    {'name': 'part1', 'defect_rate': 0.1, 'purchase_price': 2, 'inspection_cost': 1, 'assembly_cost': 0, 'disassembly_cost': 0},
    {'name': 'part2', 'defect_rate': 0.1, 'purchase_price': 8, 'inspection_cost': 1, 'assembly_cost': 0, 'disassembly_cost': 0},
    {'name': 'part3', 'defect_rate': 0.1, 'purchase_price': 12, 'inspection_cost': 2, 'assembly_cost': 8, 'disassembly_cost': 6},
    {'name': 'part4', 'defect_rate': 0.1, 'purchase_price': 2, 'inspection_cost': 1, 'assembly_cost': 0, 'disassembly_cost': 0},
    {'name': 'part5', 'defect_rate': 0.1, 'purchase_price': 8, 'inspection_cost': 1, 'assembly_cost': 0, 'disassembly_cost': 0},
    {'name': 'part6', 'defect_rate': 0.1, 'purchase_price': 12, 'inspection_cost': 2, 'assembly_cost': 8, 'disassembly_cost': 6},
    {'name': 'part7', 'defect_rate': 0.1, 'purchase_price': 8, 'inspection_cost': 1, 'assembly_cost': 0, 'disassembly_cost': 0},
    {'name': 'part8', 'defect_rate': 0.1, 'purchase_price': 12, 'inspection_cost': 2, 'assembly_cost': 8, 'disassembly_cost': 6}
]
```

中国大学生在线 dxs.moe.gov.cn

```python
# 定义市场售价与调换损失
product_price = 200
replacement_loss = 40

# 使用缓存保存已计算的状态
memo = {}
decision_trace = {}

# 状态转移函数
def state_transition(state, decision, parts):
    new_state = state.copy()
    if decision == 'inspect_part':
        for part in parts:
            part['detected'] = True
            part['is_defective'] = (part['defect_rate'] >= 0.1)
        new_state['all_parts_inspected'] = True
    elif decision == 'assemble_to_semi':
        new_state['semi_product_pass'] = all(not part['is_defective'] for part in parts)
    elif decision == 'assemble_to_final':
        new_state['final_product_pass'] = new_state.get('semi_product_pass', False)
    elif decision == 'disassemble_final':
        new_state['recycled'] = True
    elif decision == 'disassemble_semi':
        if not new_state.get('semi_product_pass', False):
            new_state['recycled'] = True
    return new_state

# 计算销售利润
def calculate_sales_profit(state):
    return product_price if state.get('final_product_pass', False) else 0

# 计算回收收益
def calculate_recycle_profit(state, parts):
    recycle_profit = 0
    if state.get('recycled', False):
        for part in parts:
            if part['detected'] and not part['is_defective']:
                recycle_profit += part['purchase_price'] + part['inspection_cost']
    if state.get('semi_product_pass', False):
        recycle_profit += sum([part['purchase_price'] + part['inspection_cost'] for part in parts])
    return recycle_profit

# 计算总成本
def calculate_cost(decision, parts):
    if decision == 'inspect_part':
        return sum([part['inspection_cost'] for part in parts])
    elif decision == 'assemble_to_semi':
        return sum([part['assembly_cost'] for part in parts])
    elif decision == 'disassemble_final' or decision == 'disassemble_semi':
        return sum([part['disassembly_cost'] for part in parts])
    return 0

# 动态规划函数
def dynamic_programming(state, parts, depth=0):
    if depth > 20:  # 限制递归深度，防止无限递归
        return 0

    # 使用元组表示状态作为缓存键
    state_key = tuple(sorted(state.items()))

    # 如果状态已经计算过，直接返回
    if state_key in memo:
        return memo[state_key]
```

中国大学生在线 dxs.moe.gov.cn

```
    # 终止条件：如果达到终态，计算收益
    if state.get('final_product_pass', False) or state.get('recycled', False):
        sales_profit = calculate_sales_profit(state)
        recycle_profit = calculate_recycle_profit(state, parts)
        total_profit = sales_profit + recycle_profit
        memo[state_key] = total_profit  # 记忆化保存
        return total_profit

    # 初始化最大利润
    max_profit = -float('inf')
    best_decision = None

    # 遍历所有可能的决策
    for decision in ['inspect_part', 'assemble_to_semi', 'assemble_to_final', 'disassemble_semi', 'disassemble_final']:
        # 剪枝：某些无效决策直接跳过
        if decision == 'assemble_to_final' and not state.get('semi_product_pass', False):
            continue
        if decision == 'disassemble_semi' and state.get('semi_product_pass', True):
            continue

        next_state = state_transition(state, decision, parts)
        cost = calculate_cost(decision, parts)
        future_profit = dynamic_programming(next_state, parts, depth + 1)
        total_profit = future_profit - cost

        # 输出调试信息
        print(f"决策: {decision}, 当前利润: {total_profit}, 成本: {cost}, 状态: {state}")

        # 更新最大利润和最优决策
        if total_profit > max_profit:
            max_profit = total_profit
            best_decision = decision
    # 保存当前状态的最优解和决策
    memo[state_key] = max_profit
    if best_decision is not None:
        decision_trace[state_key] = best_decision
    return max_profit


# 输出最优决策路径
def get_best_decision_path(state, parts):
    state_key = tuple(sorted(state.items()))
    decision_path = []

    while state_key in decision_trace:
        decision = decision_trace[state_key]
        decision_path.append(decision)
        state = state_transition(state, decision, parts)
        state_key = tuple(sorted(state.items()))

    return decision_path

best_path = get_best_decision_path(initial_state, parts)
print(f"最大期望利润: {max_profit}")
print("最优决策路径:", best_path)
```

附录 4　问题四求解问题二代码

```
library(cubature)

# 定义 Beta 分布的密度函数
beta_density <- function(x, alpha, beta) {
  return(dbeta(x, shape1 = alpha, shape2 = beta))
}

# 定义目标函数
jjjc <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; i <- param$i; l <- param$l
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (b + c) / (1 - p1) - (e + f) / (1 - p2) - (h + i) / (1 - p3) -
           l * p3 / (1 - p3))
}
```

中国大学生在线 dxs.moe.gov.cn

```r
jjbc <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; k <- param$k; l <- param$l
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (b + c) / (1 - p1) - (e + f) / (1 - p2) - h / (1 - p3) -
           (k + l) * p3 / (1 - p3))
}

jjjb <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; i <- param$i
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - ((b + c) / (1 - p1) + (e + f) / (1 - p2) + h + i) / (1 - p3))
}

jjbb <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; k <- param$k
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - ((b + c) / (1 - p1) + (e + f) / (1 - p2) + h) / (1 - p3) -
           k * p3 / (1 - p3))
}

jbjb <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; h <- param$h;
  i <- param$i
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - ((b + c) / (1 - p1) + e + h + i) / (1 - (p2 + (1 - p2) * p3)))
}
```

```r
jbbb <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; h <- param$h;
  k <- param$k
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - ((b + c) / (1 - p1) + e + h) / (1 - (p2 + (1 - p2) * p3)) -
           k * (p2 + (1 - p2) * p3) / (1 - (p2 + (1 - p2) * p3)))
}

bjjb <- function(p, param) {
  j <- param$j; b <- param$b; e <- param$e; f <- param$f; h <- param$h;
  i <- param$i
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (b + (e + f) / (1 - p2) + h + i) / (1 - (p1 + (1 - p1) * p3)))
}

bjbb <- function(p, param) {
  j <- param$j; b <- param$b; e <- param$e; f <- param$f; h <- param$h;
  k <- param$k
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (b + (e + f) / (1 - p2) + h) / (1 - (p1 + (1 - p1) * p3)) -
           k * (p1 + (1 - p1) * p3) / (1 - (p1 + (1 - p1) * p3)))
}

bbjb <- function(p, param) {
  j <- param$j; b <- param$b; e <- param$e; h <- param$h; i <- param$i
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (b + e + h + i) / (1 - p1) / (1 - p2) / (1 - p3))
}
```

中国大学生在线 dxs.moe.gov.cn

```
bbbb <- function(p, param) {
  j <- param$j; b <- param$b; e <- param$e; h <- param$h; k <- param$k
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (b + e + h) / (1 - p1) / (1 - p2) / (1 - p3) -
           k * (1 / (1 - p1) / (1 - p2) / (1 - p3) - 1))
}

bjjc <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; i <- param$i; l <- param$l
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (b + (e + f) / (1 - p2) + h + i + (p1 + p3 - p1 * p3) *
                (c + (h + i + l) / (1 - p3)) + p1 * (b + c) / (1 - p1)))
}

jbjc <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; i <- param$i; l <- param$l
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (e + (b + c) / (1 - p1) + h + i + (p2 + p3 - p2 * p3) *
                (f + (h + i + l) / (1 - p3)) + p2 * (e + f) / (1 - p2)))
}

bbjc <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; i <- param$i; l <- param$l
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - ((b + e + h + i) + ((c + f) + (h + i + l) / (1 - p3)) *
                (1 - (1 - p1) * (1 - p2) * (1 - p3)) + (e + f) * p2 / (1 - p2) +
                (b + c) * p1 / (1 - p1)))
}
```

```
bjbc <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; i <- param$i; k <- param$k; l <- param$l
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (b + (e + f) / (1 - p2) + h + (c + (h + k + l) / (1 - p3)) *
                (p1 + p3 - p1 * p3) + (b + c) * p1 / (1 - p1)))
}

jbbc <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; i <- param$i; k <- param$k; l <- param$l
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - (e + (b + c) / (1 - p1) + h + (f + (h + k + l) / (1 - p3)) *
                (p2 + p3 - p2 * p3) + (e + f) * p2 / (1 - p2)))
}

bbbc <- function(p, param) {
  j <- param$j; b <- param$b; c <- param$c; e <- param$e; f <- param$f;
  h <- param$h; i <- param$i; k <- param$k; l <- param$l
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  return(j - ((b + e + h) + ((c + f) + (h + k + l) / (1 - p3)) *
                (1 - (1 - p1) * (1 - p2) * (1 - p3)) + (e + f) * p2 / (1 - p2) +
                (b + c) * p1 / (1 - p1)))
}
```

中国大学生在线 dxs.moe.gov.cn

```r
# 定义联合密度函数
joint_density <- function(p, alpha1, beta1, alpha2, beta2, alpha3, beta3) {
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]

  return(beta_density(p1, alpha1, beta1) *
           beta_density(p2, alpha2, beta2) *
           beta_density(p3, alpha3, beta3))
}

# 定义被积函数，考虑联合密度
integrand <- function(p, func, alpha1, beta1, alpha2, beta2, alpha3, beta3, param) {
  return(func(p, param) * joint_density(p, alpha1, beta1, alpha2, beta2, alpha3, beta3))
}

# 设置积分范围：每个变量的范围是 [0, 1]
lower_limit <- c(0, 0, 0)
upper_limit <- c(1, 1, 1)

# 定义参数组
params_list <- list(
  list(n = 100, a = 0.1, b = 4, c = 2, d = 0.1, e = 18, f = 3, g = 0.1, h = 6,
       i = 3, j = 56, k = 6, l = 5),
  list(n = 100, a = 0.2, b = 4, c = 2, d = 0.2, e = 18, f = 3, g = 0.2, h = 6,
       i = 3, j = 56, k = 6, l = 5),
  list(n = 100, a = 0.1, b = 4, c = 2, d = 0.1, e = 18, f = 3, g = 0.1, h = 6,
       i = 3, j = 56, k = 30, l = 5),
  list(n = 100, a = 0.2, b = 4, c = 1, d = 0.2, e = 18, f = 1, g = 0.2, h = 6,
       i = 2, j = 56, k = 30, l = 5),
  list(n = 100, a = 0.1, b = 4, c = 8, d = 0.2, e = 18, f = 1, g = 0.1, h = 6,
       i = 2, j = 56, k = 10, l = 5),
  list(n = 100, a = 0.05, b = 4, c = 2, d = 0.05, e = 18, f = 3, g = 0.05, h = 6,
       i = 3, j = 56, k = 10, l = 40)
)
```

```r
# 函数列表
functions <- list(jjjc, jjbc, jjjb, jjbb, jbjb, jbbb, bjjb, bjbb, bbjb, bbbb,
                  bjjc, jbjc, bbjc, bjbc, jbbc, bbbc)

# 初始化结果矩阵
results_matrix <- matrix(0, nrow = 16, ncol = 6)
# 计算每组参数下的积分
for (col in 1:6) {
  params <- params_list[[col]]

  # 计算 Beta 分布参数
  alpha1 <- 1 + params$n * params$a
  beta1 <- 1 + params$n - params$n * params$a
  alpha2 <- 1 + params$n * params$d
  beta2 <- 1 + params$n - params$n * params$d
  alpha3 <- 1 + params$n * params$g
  beta3 <- 1 + params$n - params$n * params$g
  # 计算每个函数的积分
  for (row in 1:16) {
    func <- functions[[row]]
    result <- adaptIntegrate(function(p) integrand(p, func, alpha1, beta1,
                                                   alpha2, beta2, alpha3, beta3, params),
                             lowerLimit = lower_limit, upperLimit = upper_limit)
    results_matrix[row, col] <- result$integral
  }
}
# 输出结果矩阵
print(results_matrix)
# 安装并加载 necessary packages
library(openxlsx)
# 定义 Excel 文件路径
file_path <- "results_matrix.xlsx"
# 创建一个工作簿
wb <- createWorkbook()
# 添加一个工作表
addWorksheet(wb, "Results")
# 写入矩阵到工作表
writeData(wb, "Results", results_matrix)
```

中国大学生在线 dxs.moe.gov.cn

附录 5　问题四求解问题三代码

```python
import numpy as np
from scipy.stats import beta

# 模拟基于Beta分布的次品率概率的函数
def defect_probability(n, p):
    # Beta分布参数：np1（成功次数），n - np1（失败次数）
    a = 1 + n * p
    b = 1 + n - n * p
    return beta(a, b)

# 计算零件、半成品和成品的次品分布的函数
def simulate_defect(n, p, trials=10000):

    beta_distribution = defect_probability(n, p)
    defect_rate_samples = beta_distribution.rvs(size=trials)

    part_defects = np.random.binomial(1, defect_rate_samples, size=trials)
    semi_product_defects = np.random.binomial(1, defect_rate_samples, size=trials)
    final_product_defects = np.random.binomial(1, defect_rate_samples, size=trials)

    return part_defects, semi_product_defects, final_product_defects

n = 100
p = 0.1
trials = 10000

# 运行模拟
part_defects, semi_product_defects, final_product_defects = simulate_defect(n, p, trials)

# 计算次品率
part_defect_prob = np.mean(part_defects)
semi_product_defect_prob = np.mean(semi_product_defects)
final_product_defect_prob = np.mean(final_product_defects)
```

附录 6　灵敏度检验代码

```r
library(cubature)
# 定义常数参数
n <- 100
a <- 0.1
d <- 0.1
g <- 0.1
j <- 56
b <- 4
c <- 2
e <- 18
f <- 3
h <- 6
i <- 3
l <- 5
beta0 <- 1

# 定义 Beta 分布的密度函数
beta_density <- function(x, alpha, beta) {
  return(dbeta(x, shape1 = alpha, shape2 = beta))
}

# 定义目标函数 h_func(p1, p2, p3)
h_func <- function(p) {
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]

  result <- j - (b + c) / (1 - p1) - (e + f) / (1 - p2) - h / (1 - p3) -
    (k + l) * p3 / (1 - p3)
  return(result)
}
```

中国大学生在线 dxs.moe.gov.cn

```r
# 定义联合密度函数
joint_density <- function(p, alpha1, beta1, alpha2, beta2, alpha3, beta3) {
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]

  return(beta_density(p1, alpha1, beta1) *
           beta_density(p2, alpha2, beta2) *
           beta_density(p3, alpha3, beta3))
}

# 定义被积函数，考虑联合密度
integrand <- function(p, alpha1, beta1, alpha2, beta2, alpha3, beta3) {
  return(h_func(p) * joint_density(p, alpha1, beta1, alpha2, beta2, alpha3, beta3))
}

# 设置积分范围：每个变量的范围是 [0, 1]
lower_limit <- c(0, 0, 0)
upper_limit <- c(1, 1, 1)

# 初始化结果向量和 alpha0 向量
results <- numeric(10)
alpha0_values <- 1:10

# 计算每个 alpha0 对应的期望值
for (alpha0 in alpha0_values) {
  alpha1 <- alpha0 + n * a
  beta1 <- beta0 + n - n * a

  alpha2 <- alpha0 + n * d
  beta2 <- beta0 + n - n * d

  alpha3 <- alpha0 + n * g
  beta3 <- beta0 + n - n * g

  result <- adaptIntegrate(function(p) integrand(p, alpha1, beta1, alpha2, beta2,
                                                 alpha3, beta3),
                           lowerLimit = lower_limit, upperLimit = upper_limit)

  results[alpha0] <- result$integral
}
```

```r
# 输出结果
cat("期望值向量：", results, "\n")

# 将结果绘制成散点图
plot(alpha0_values, results,
     type = "p",
     xlab = "alpha",
     ylab = "期望值",
     ylim = c(0, 20))

# 进行线性拟合
fit <- lm(results ~ alpha0_values)

# 添加线性拟合线
abline(fit, col = "blue")

# 输出线性拟合结果
cat("线性拟合结果：\n")
print(summary(fit))

n <- 100
a <- 0.1
d <- 0.1
g <- 0.1
j <- 56
b <- 4
c <- 2
e <- 18
f <- 3
h <- 6
i <- 3
l <- 5
alpha0 <- 1  # alpha0 不变

# 定义 Beta 分布的密度函数
beta_density <- function(x, alpha, beta) {
  return(dbeta(x, shape1 = alpha, shape2 = beta))
}
```

中国大学生在线 dxs.moe.gov.cn

```r
# 定义目标函数 h_func(p1, p2, p3)
h_func <- function(p) {
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]

  result <- j - (b + c) / (1 - p1) - (e + f) / (1 - p2) - h / (1 - p3) -
    (k + l) * p3 / (1 - p3)
  return(result)
}

# 定义联合密度函数
joint_density <- function(p, alpha1, beta1, alpha2, beta2, alpha3, beta3) {
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]

  return(beta_density(p1, alpha1, beta1) *
           beta_density(p2, alpha2, beta2) *
           beta_density(p3, alpha3, beta3))
}

# 定义被积函数，考虑联合密度
integrand <- function(p, alpha1, beta1, alpha2, beta2, alpha3, beta3) {
  return(h_func(p) * joint_density(p, alpha1, beta1, alpha2, beta2, alpha3, beta3))
}

# 设置积分范围：每个变量的范围是 [0, 1]
lower_limit <- c(0, 0, 0)
upper_limit <- c(1, 1, 1)

# 初始化结果向量和 beta0 向量
results <- numeric(10)
beta0_values <- 1:10
```

```r
# 计算每个 beta0 对应的期望值
for (beta0 in beta0_values) {
  alpha1 <- alpha0 + n * a
  beta1 <- beta0 + n - n * a

  alpha2 <- alpha0 + n * d
  beta2 <- beta0 + n - n * d

  alpha3 <- alpha0 + n * g
  beta3 <- beta0 + n - n * g

  result <- adaptIntegrate(function(p) integrand(p, alpha1, beta1, alpha2, beta2,
                                                 alpha3, beta3),
                           lowerLimit = lower_limit, upperLimit = upper_limit)

  results[beta0] <- result$integral
}

# 输出结果
cat("期望值向量：", results, "\n")

# 将结果绘制成散点图
plot(beta0_values, results,
     type = "p",
     xlab = "beta",
     ylab = "期望值",
     ylim = c(0, 20))

# 进行线性拟合
fit <- lm(results ~ beta0_values)

# 添加线性拟合线
abline(fit, col = "red")

# 输出线性拟合结果
cat("线性拟合结果：\n")
print(summary(fit))
```

中国大学生在线 dxs.moe.gov.cn

中国大学生在线
dxs.moe.gov.cn