

基于模拟退火和贪心思想的企业生产过程决策分析

摘要

在生产过程中适时检测零配件、半成品和成品的次品率，以及对于不合格品的拆解与否等决策问题，对于提高产品质量与利润、降低总成本、提升客户满意度和品牌形象等都具有重要意义。本文建立了利润模型，运用模拟退火、贪心思想、循环遍历法，来研究企业决策问题。

针对问题一：首先将从零配件中抽取的每个样本构建为二项分布模型，当样本数量足够大时，根据中心极限定理，次品数与次品率近似服从正态分布。然后通过假设检验确定情形（1）和情形（2）下的原假设 H_0 与备择假设 H_1 ，构建检验统计量与临界值相比较，根据简单随机抽样的最小样本数公式确定在允许误差 $E = 0.03$ 时，情形（1）企业至少需要抽样检测零配件 271 个；情形（2）企业至少需要抽样检测零配件 165 个。在允许误差 $E = 0.05$ 时，情形（1）企业至少需要抽样检测零配件 98 个；情形（2）企业至少需要抽样检测零配件 60 个。最后进行结果分析与可行性分析。

针对问题二：首先确定决策变量数目，以及影响利润的诸多因素，据此建立利润函数模型，视作 $\mathbb{R}^4 \rightarrow \mathbb{R}$ 的一个函数。对这个函数进行优化即可。由于函数模型并不复制，可以基于循环遍历法并结合贪心思想保留最佳的利润，进而求解。然后通过模拟退火仿真检验贪心思想的结果。最后对两种方法进行比较，讨论操作的合理性。同时，考虑到完全检测不一定达到最优效果，因此引入检测率，用于探究部分检测可能得到的优秀决策方案。

针对问题三：首先通过模块化的方法，将整个 m 道工序与 n 个零配件进行分模块，建立通过次品率传递实现递归的模块化模型，通过探究前后两阶段的次品率关系，以及部分检测导致的次品率变化，完成模型的完整建立。再重复问题二的流程得到最终结果

针对问题四：首先由次品率的分布得到随机抽样检测的次品率观测值，根据波动理论的枢轴量法求得次品率的置信区间，在置信区间内使用蒙特卡洛模拟法，随机产生样本的次品率，并以此替换零配件、半成品与成品的固定次品率。通过重复问题二和问题三，得到新的决策，并与问题二三中的决策进行对比分析，计算误差，阐释误差产生的原因。

关键字： 模拟退火 贪心思想 循环遍历 假设检验 供应链管理 抽样检测

一、 问题背景与重述

1.1 问题背景

虽然现代企业生产规模逐渐扩大，但受限于设备精度，生产技术等因素，不可避免的会出现不合格产品。而检测和拆解又会产生相应的费用，因此在实际生产过程中，找到最优解有助于实现企业盈利最大化。

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

1.2 问题重述

针对上述背景，建立数学模型，解决以下问题

问题一：供应商提供一批零配件（零配件 1 和零配件 2），称其次品率不超过某一标称值。为了来验证情况是否属实，企业决定采用抽样检测的方式验证。为节约检测成本，请设计检测次数尽可能少的检测方案。

如果标称值为 10%，利用上述方案分析如下具体情况，得出结果：

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

问题二：已知两种零配件和成品次品率，需在企业生产过程的各个阶段进行决策：

(1) 决定是否对两种零配件（零配件 1 和零配件 2）进行质量检测。如果选择不检测，不合格的零配件将直接用于装配；如果选择检测，不合格的零配件将被丢弃。

(2) 决定是否对装配好的成品进行质量检测。如果选择不检测，成品将直接进入市场；如果选择检测，只有合格的成品才能进入市场。

(3) 决定是否对检测出的不合格成品进行拆解。如果选择不拆解，不合格成品将被丢弃；如果选择拆解，可以将拆解后的零配件重新用于生产。

(4) 考虑用户购买到不合格品时的调换成本，包括物流成本和企业信誉损失。对于退回的不合格品，需要决定是否重复拆解和重新装配的步骤

问题三：对于一个需要 m 道工序、 n 个零配件的成品，已知所有零配件、半成品和成品的次品率，重新完成问题二，给出决策方案。已知 2 道工序，8 个零配件的具体情况 and 数值，给出具体的决策方案，以及决策的根据及相应指标。

问题四：假设问题二和问题三中均是通过抽样检测方法（例如，在问题一中使用的方法）得到的零配件、半成品和成品的次品率，重新完成问题二和问题三。

二、 问题分析

2.1 问题一

在问题一中，题目要求在供应商提供的标称值 10% 的条件下，针对 **(1) 在 95% 的信度下认定零配件次品率超过标称值，则拒收这批零配件；(2) 在 90% 的信度下认定零配件次品率不超过标称值，则接收这批零配件**这两种情形，为企业设计检测次数尽可能少的抽样检测方案。首先，将从零配件中抽取的每个样本构建为**二项分布**模型，当样本数量足够大时，根据**中心极限定理**，次品数与次品率近似服从**正态分布**，然后通过**假设检验**确定拒绝与接受原假设的临界值，最后根据简单随机抽样的**最小样本数公式**确定两种情形下最少的检测次数。

2.2 问题二

在问题二中我们已知所有零配件的次品率、购买单价、检测成本以及成品的次品率、装配成本、检测成本和售价还要对于不合格成品的调换损失与拆解费用。据此建立利润模型进行优化，求出利润最大的决策方案。可以证明，将决策方案考虑为 0, 1 规划的情况可以求得最优解。首先求得**利润表达式**，然后**遍历**四个决策步骤的 0, 1 规划所有情况，仿照**贪心思想**保留最优解，最后我们采用**模拟退火**算法考虑四维立方体内点对贪心思想求得结果进行**仿真验证**。

2.3 问题三

问题三是问题二的更一般情况。考虑 m 道工序和 n 个零配件的情况，将问题二中的一道工序 2 个零配件的简单情况推广至实际生成过程中的一般情况。并且要求我们详细讨论 $m = 2, n = 8$ 的情况。对于一般的情况，我们首先可以用**模块化的思想**进行分析，即将多个零配件装配成的一个半成品视作问题二中两个零配件合成一个成品。在下一道工序当中，半成品可以视作“零配件”，多个这样的“零配件”再次下一工序中的半成品。这样将问题变成已经完成分析的问题二的情形。然后重新考虑每一模块的利润函数并组合，紧接着模仿问题二进行**遍历**决策步骤进行规划，保留最优解，并采用**模拟退火**算法进行**仿真验证**，最后分析答案，探讨方案可行性。

2.4 问题四

在问题四中，假设零配件、半成品和成品的次品率均是通过**抽样检测方法**得到的，意味着问题二和问题三中的次品率不再是供应商提供的标称值。首先由次品率的分布得到随机抽样检测的次品率观测值，根据**波动理论的枢轴量法**求得次品率的**置信区间**，在置信区间内使用**蒙特卡洛模拟法**，随机产生样本的次品率，并以此替换零配件、半成品与成品的固定次品率。在已知次品率后，通过重复问题二与问题三的过程，得到新的决策。将新决策与问题二与问题三中得到的决策进行对比分析，发现问题二的决策与新决策完全相同，而问题三的决策与新决策基本相同。最后总结规律。

三、 模型假设

1. 独立检测假设：各个检测相互独立，不同零配件，半成品与成品之间的检测结果互不影响。

2. 检测准确性假设：假定每次检测结果都是准确的，不存在漏检或因为操作失误导致的误判。

3. 次品率稳定假设：认为零配件足够多，一定量的出售和拆解并不影响零配件整体的次品率。（注：本文所有对次品率的“矫正”只是经过检测后为了简化计算的处理，不改变涉及零配件作为样品本身的统计性质）

4. 市场稳定假设：在企业整个生产与销售过程中，零配件成本，检测成本以及其他生产费用和成品的市场售价均维持稳定。

5. 次品必定退回假设：用户购买到不合格品时，必定退回并要求调换。

6. 次品率传递假设：不同零配件之间不存在相互影响的情况，成品出现次品的概率由零配件次品率以及合格零配件（或者半成品）装配后的产品次品率传递完成，不存在更多复杂的情况。

7. 拆解还原假设：拆解过程并不会损坏零配件，即合格零配件依然为合格零配件。

四、 符号说明

如表1所示：

五、 问题一模型建立及求解

企业在供应商购买零配件生产电子产品时，供应商会声称该批零配件（零配件1或零配件2）的次品率不会超过某个标称值。其中标称值是根据设计要求或制造标准，为元件、器件、设备或产品在特定运行条件下所规定的量值。企业准备采用抽样检测方法

表 1 符号说明

符号	解释说明	单位
Z	假设检验中不同置信度对应的临界值	°
E	抽样检测允许误差	°
x	对应产品的检测率	°
p	对应产品的次品率	°
b	对应产品的拆解率	°
w	不同情况下每件产品的利润值期望值	元/件

决定是否接收从供应商购买的这批零配件，抽样检测的样本量大小会对检测结果产生一定的影响，样本量越大，抽样误差通常越小，样本能够更全面地反映总体的特性，因此抽样结果的估计越精确。同时样本量的增加会给企业带来更多的检测费用，所以该问要求为企业设计尽可能少的抽样检测方案。

在供应商声称 10% 的标称值下，给出以下两种情形的抽样检测方案：

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

5.1 模型的建立

5.1.1 样本统计模型

假设从一批零配件中抽取了 n 个样本，每个样本有合格和次品两种可能的检测结果。假设每个样本的检测之间相互独立，每个样本检测为次品的概率为 10%，那么 n 个样本中的次品数 X 服从二项分布，即

$$X \sim B(n, p) \quad (1)$$

其中该批零配件的次品率为 p ，那么样本中有 k 个次品的概率为

$$P\{X = k\} = \binom{n}{k} p^k (1-p)^{n-k} \quad (2)$$

该问目标是通过一定数量的抽样检测数据，判断在一定的置信度下次品率 p 是否超过标称值 $p_0 = 0.10$ ，并做出接收或拒收的决策。

5.1.2 正态分布近似

根据中心极限定理，在样本量较大时（例如 $np \geq 10$ 或 $np(1-p) \geq 10$ ），可以用正态分布来近似二项分布

$$X \sim N(np, np(1-p)) \quad (3)$$

其中样本中次品数量服从均值为 $u = np$ ，方差为 $\sigma^2 = np(1-p)$ 的正态分布。

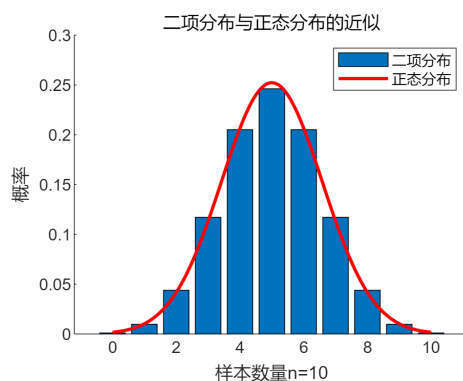


图1 正态分布近似，n=10

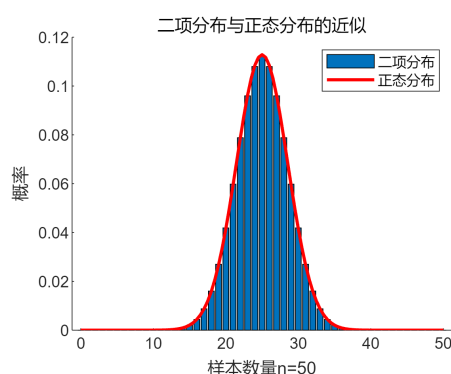


图2 正态分布近似，n=50

5.1.3 假设检验

假设检验又称统计假设检验，是一种统计推断方法，用于判断样本数据是否支持某个关于总体特征的假设。这种方法在科学研究、医学实验、市场调查等领域有着广泛的应用。通过假设检验，研究者可以基于样本数据对总体特征做出推断，从而支持或反驳某个假设。

Step1 设定假设

情形一：在 95 % 的信度下认定零配件次品率超过标称值，则拒收这批零配件

原假设 H_0 : 零配件次品率 $p \leq 0.1$

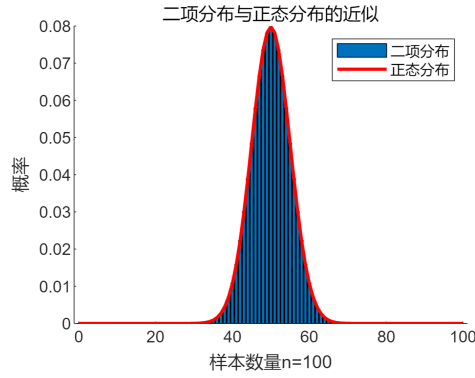


图3 正态分布近似，n=100

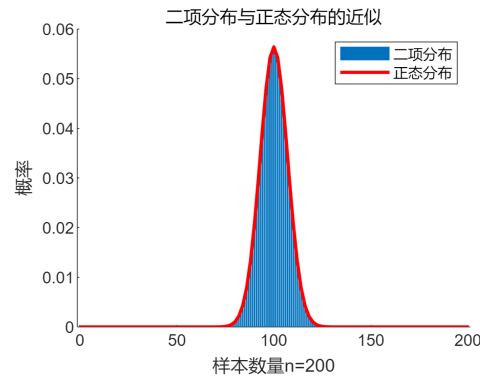


图4 正态分布近似，n=200

备择假设 H_1 : 零配件次品率 $p > 0.1$

情形二：在 90% 的信度下认定零配件次品率不超过标称值，则接收这批零配件

原假设 H_0 : 零配件次品率 $p > 0.1$

备择假设 H_1 : 零配件次品率 $p \leq 0.1$

Step2 选定检验统计量

由于次品率 $p = \frac{X}{n}$ ，统计量 p 满足分布

$$p \sim N(p_0, \frac{p_0(1-p_0)}{n}) \quad (4)$$

该问为单个正态总体参数的单边假设检验，在方差已知的情况下检验均值，于是构造一个检验统计量

$$Z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \sim N(0, 1) \quad (5)$$

其中 $\hat{p} = \frac{x}{n}$ ， x 表示某次抽样检测中次品数的观测值，那么 \hat{p} 表示抽样检测中次品率的观测值。

Step3 确定显著水平与临界值

显著水平为研究者愿意承担的风险，即拒绝原假设但实际上原假设是正确的概率。根据题目中的信度要求可知，情形一中显著水平 $\alpha_1 = 1 - 95\% = 0.05$ ，情形二中显著水平 $\alpha_2 = 1 - 90\% = 0.1$ 。

根据显著水平 $\alpha_i (i = 1, 2)$ 可以求出标准正态分布下的临界值 $Z_{1-\alpha_i} (i = 1, 2)$ ，查表可知 $Z_{0.95} = 1.645$ ， $Z_{0.9} = 1.282$ 。

Step4 计算检验统计量实际值与决策

情形一中，在 95% 信度下，拒绝原假设 H_0 即拒收零配件的条件是： $Z > Z_{0.95}$ ，即 $\hat{p} > p_0 + Z_{0.95}\sigma_p$ ；

情形二中，在 90% 信度下，拒绝原假设 H_0 即接受零配件的条件是： $Z < Z_{0.9}$ ，即 $\hat{p} \leq p_0 + Z_{0.9}\sigma_p$ 。

5.1.4 确定样本量

根据简单随机抽样的最小样本数公式可知，样本量 n 的可以确定为

$$n = \frac{Z_{1-\alpha_i}^2 \sigma^2}{E^2} = \frac{Z_{1-\alpha_i}^2 p_0(1-p_0)}{E^2} (i = 1, 2) \quad (6)$$

从式(6)可知抽样数量的大小与需要达到的误差范围 E 与置信水平对应的临界值 $Z_{1-\alpha_i}$ 有关，其中 $p_0 = 0.1$ 。

5.2 模型的求解

5.2.1 假设容许的抽样误差 $E = 0.03$

根据式(6)可以计算，在信度为 95% 时， $Z_{1-\alpha_1} = 1.645$ ， $p_0 = 0.1$

$$n_1 = \frac{(1.645)^2 0.1(1-0.1)}{0.03^2} = 270.6$$

因此在信度为 95% 时，企业至少需要抽取约 271 个样本检测来得出对次品率是否超过标称值的判断。

在信度为 90% 时， $Z_{1-\alpha_2} = 1.282$ ， $p_0 = 0.1$

$$n_2 = \frac{(1.282)^2 0.1(1-0.1)}{0.03^2} = 164.3$$

因此在信度为 90% 时，企业至少需要抽取约 165 个样本检测来得出对次品率是否超过标称值的判断。

5.2.2 假设容许的抽样误差 $E = 0.05$

计算方法同抽样误差 $E = 0.03$ 的情况，结果为在信度为 95% 时，企业至少需要抽取约 98 个样本检测来得出对次品率是否超过标称值的判断；在信度为 90% 时，企业至少需要抽取约 60 个样本检测来得出对次品率是否超过标称值的判断。

5.2.3 不同抽样误差下的最小样本量趋势图

计算出不同抽样误差下的最小样本量，绘制成如下趋势图

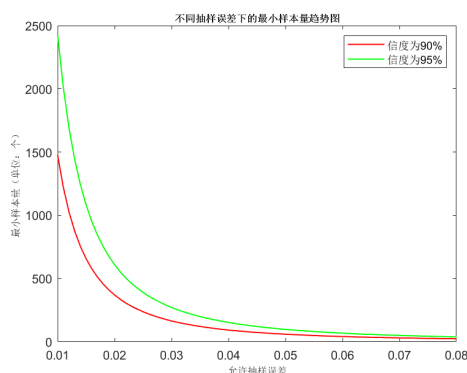


图 5 不同抽样误差下的最小样本量趋势图

从图中可以看出随着允许误差的减小，所需抽样检测的样本量迅速增加，而适当的减弱对允许误差的要求，能够显著降低企业所需的检测成本。

综上所述，在允许误差 $E=0.03$ 时，企业选择的抽样检测为：

情形（1）（在 95% 的信度下认定零配件次品率超过标称值，则拒收这批零配件）抽样检测 **271** 个零配件；

情形（2）（在 90% 的信度下认定零配件次品率不超过标称值，则接收这批零配件）抽样检测 **165** 个零配件。

在允许误差 $E=0.05$ 时，企业选择的抽样检测为：

情形（1）（在 95% 的信度下认定零配件次品率超过标称值，则拒收这批零配件）抽样检测 **98** 个零配件；

情形（2）（在 90% 的信度下认定零配件次品率不超过标称值，则接收这批零配件）抽样检测 **60** 个零配件。

六、问题二模型的建立与求解

6.1 模型建立

如图6所示，该问中只需考虑两个零配件合成一个成品的情况。

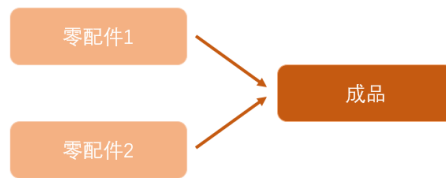


图 6 问题二合成流程

首先画出其流程图图7。通过图7，总结出四次**关键决策**：（1）决定零配件 1 是否检测（2）决定零配件 2 是否检测（3）决定成品是否检测（4）决定不合格成品是否进行拆解。

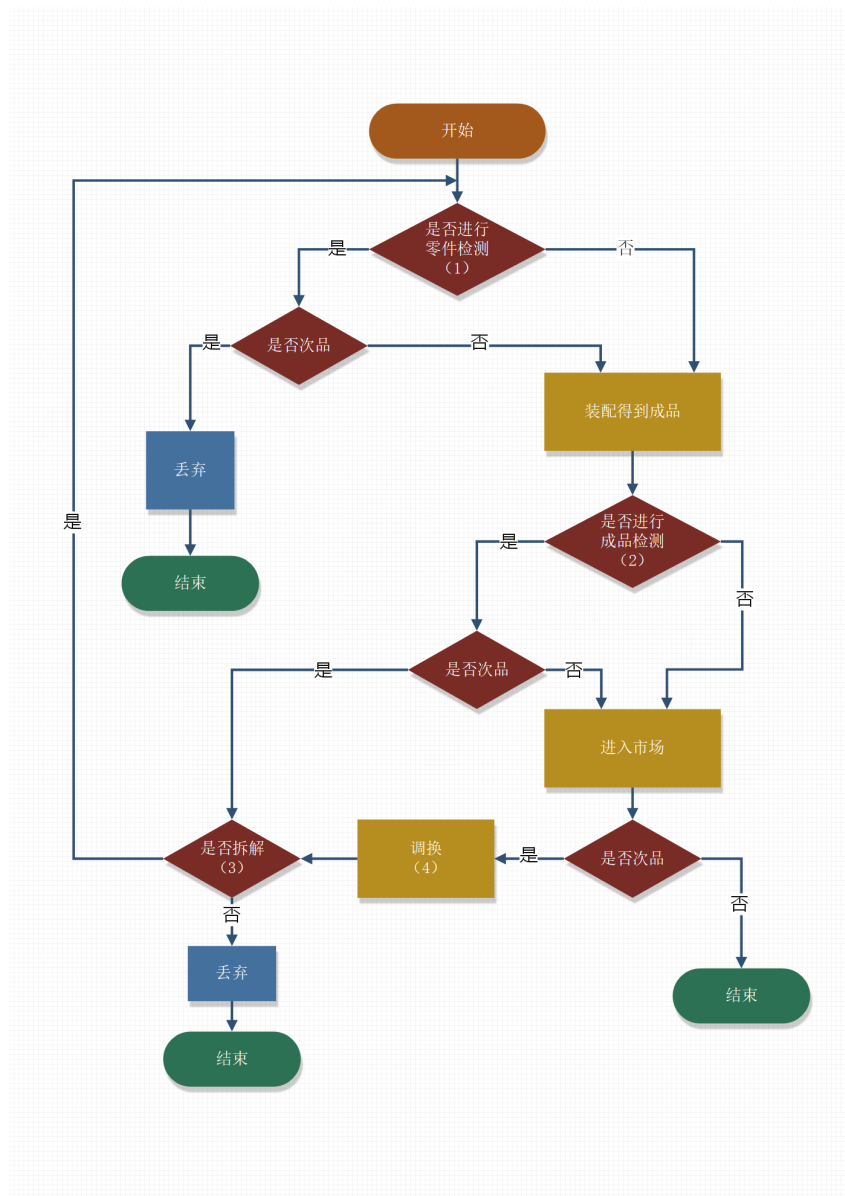


图 7 问题二决策流程

不难注意到，无论决策（3）的结果如何，决策（4）依然影响利润求解。若决策（3）

检测了成品，立刻决定不合格成品是否拆解；若决策（3）没检测成品而直接卖出，则会在顾客退回时决定该不合格品是否拆解。因此，决策（3）的结果不改变决策（4）的必要性。

对于上述四个决策，我们用一个四维向量 $\vec{x} = [x_1, x_2, x_3, x_4]$ 表示上述决策结果。其中， $x_i = 1$ 表示全部进行检测或者第四决策为全部拆解； $x_i = 0$ 表示全部不检测或者第四决策为全部丢弃； $0 < x_i < 1$ 表示所有零配件或者成品中的百分之 $100x_i\%$ 检测，其余不检测，或者所有不合格品中的 $100x_i\%$ 拆解，其余丢弃。

如下图所示，我们假设流程中的临时变量 a_k 和 $n_l, k \in \{1, 2\}, l \in \{1, 2, 3, 4, 5, 6\}, 0 < a_k, n_l < 1$ 。临时变量的定义如图8所示，均是在流程中产生的零配件或成品的比例，关系式也如图8所示。

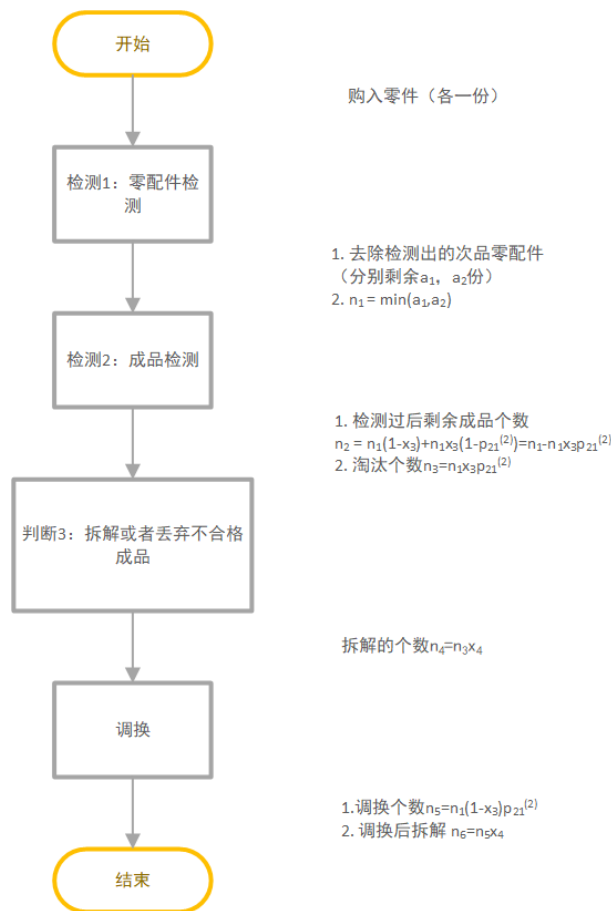


图8 问题二变量假设

注意到问题二中的多种情况中，零配件与成品次品率均为 10%，而次品将直接生成不合格成品，故而成品的 10% 次品率应该是当零配件都是合格品的情况下生成的不合格品次品率。在计算过程中，由于第一步不知是否进行零配件的检测，成品的 10% 次品率使用并不方便。对此我们进行修正。记次品率为 $p_{i,j}^{(m)}$ ，其中 i 表示这是零配件的次品率（ $i = 1$ ）或者是成品的次品率（ $i = 2$ ）， j 表示这是零配件 j 的次品率， m 表示这是修正前（ $m = 1$ ）后（ $m = 2$ ）的次品率（对于不需要修正的次品率，略去上标）。

通过上述分析，可以写出如下关系式联系上述所有假设变量：

$$a_1 = 1 - x_{11}p_{11}^{(1)} \quad (7)$$

$$a_2 = 1 - x_{12}p_{12}^{(1)} \quad (8)$$

$$p_{11}^{(3)} = \frac{(1 - x_{11})p_{11}^{(1)}}{1 - x_{11}p_{11}^{(1)}} \quad (9)$$

$$p_{12}^{(3)} = \frac{(1 - x_{12})p_{12}^{(1)}}{1 - x_{12}p_{12}^{(1)}} \quad (10)$$

$$p_{21}^{(2)} = 1 - (1 - p_{11}^{(3)})(1 - p_{12}^{(3)})(1 - p_{21}^{(1)}) \quad (11)$$

$$n_1 = \min \{a_1, a_2\} \quad (12)$$

$$n_2 = n_1(1 - x_3) + n_1x_3(1 - x_3) = n_1 - n_1x_3p_{21}^{(2)} \quad (13)$$

$$n_3 = n_1x_3p_{21}^{(2)} \quad (14)$$

$$n_4 = n_3x_4 \quad (15)$$

$$n_5 = n_1(1 - x_3)p_{21}^{(2)} \quad (16)$$

$$n_6 = n_5x_4 \quad (17)$$

据此可以写出利润表达式

$$\begin{aligned} \text{利润期望} w = & \text{销售额} - \text{购买零配件成本} - \sum \text{检测成本} \times \text{检测率} \\ & - \text{装配成本} \times \text{装配率} - \text{拆卸成本} \times \text{拆卸率} \\ & - \text{调换成本} \times \text{调换率} + \text{拆解所得零配件成本} \times \text{拆卸率} \end{aligned} \quad (18)$$

其中，销售额 = 售价 × 销量，销量 = $n_2 - n_5$ ；零配件一和零配件二的检测率分别为 x_1, x_2 ；成品的检测率为 $n_1 \times x_3$ ；组装率 = n_1 ；拆卸率 = $n_4 + n_6$ ；调换率 = n_5 。将上面所有式子和公式(7)到公式(17)全部带入得到利润模型表达式。

6.2 模型求解

6.2.1 基于贪心思想和决策代价求解

定理 1 模型最优解不是内点，必定是在四维立方体的十六个顶点上。

证明 1 逆流程图考虑每个决策的代价。对于决策 (4) 的两种情况拆卸或者丢弃：拆卸产生的净收入是：-拆解费用 + 两个零配件成本；丢弃产生的净收入是：0。当拆解产生的净收入大于丢弃产生的净收入时，无论前面流程如何，此处必然会选择拆卸，反之则选择丢弃。例如问题二当中前五种情况都必然拆卸（即 $x_4 = 1$ ），而最后一种情况则直接丢弃（即 $x_4 = 0$ ）。否则只有有一部分选择了最佳解法的对立面，总利润都会亏损。

进一步考虑决策 (3)，该步决策的两个结果产生的净收入由决策 (4) 支配。而决策 (4) 结果根据上述推导已经可以得出，故决策 (3) 也是 0 或者 1。

由数学归纳法，每一步决策的净收入将由后续流程以及后续决策得到，而最后一步决策结果无外乎为 0 或者 1，导致每一步的决策结果也都无外乎为 0 或者 1。总决策数是一个有限的正整数，我们完成了命题的证明

基于定理1，我们编写 *Matlab* 程序对问题二中的每个情况遍历 $2^4 = 16$ 个可能的解，即可得到最优解。但为了检验定理的正确性，我们将 [0,1] 区间 20 等分，以步长 0.05 遍历整个区间，用贪心思想保存结果，得到最优解：

情况 1 的最佳期望利润为 15.805，最优结果是：[0,0,0,1]

情况 2 的最佳期望利润为 6.040，最优结果是：[0,0,0,1]

情况 3 的最佳期望利润为 14.431，最优结果是：[0,0,1,1]

情况 4 的最佳期望利润为 8.160，最优结果是：[1,1,1,1]

情况 5 的最佳期望利润为 12.272，最优结果是：[0,0,1,1]

情况 6 的最佳期望利润为 18.587，最优结果是：[0,0,0,0]

上述结果符合预期，但是 0.05 的步长对于定理1只能初步检验，下面再通过模拟退火算法进行双重验证。

6.2.2 基于模拟退火算法进行仿真检验

为验证模型的合理性，现用模拟退火算法对上述结果进行仿真检验。**模拟退火算法** (Simulated Annealing, SA) 是一种基于蒙特卡洛迭代求解策略的**随机寻优算法**，其灵感来源于物理中的固体退火过程。该算法通过模拟固体物质在加热后缓慢冷却的过程，以寻找问题的全局最优解。在搜索过程中，模拟退火算法不仅接受使目标函数值更优的解，还以一定概率接受一个比当前解差的解，这种机制使得算法能够跳出局部最优解，从而增加找到全局最优解的可能性。

模拟退火流程：

1. **初始化参数：**退火温度为 $T = 1^\circ C$ ，结束温度为 $T_{end} = 0^\circ C$ ，最大退火次数为 $L = 100000$ ，退火系数为 $at = 0.9999$
2. **随机较优解：**基于蒙特卡洛方法循环随机模拟结果，求出当前较优解，记最优利润 $w_{current}$ ，最优路径是 $\overrightarrow{x_{current}}$ 。
3. **产生新解：**当前解为 \vec{x} ，随机一个位置 c ，有一半概率增加一半概率减小，并且变化值的大小不超过当前温度。记新解为 $w_{try} \overrightarrow{x_{try}}$ 这样保证退火开始时能够在可行域内任意点寻找最优解，在退火过程后期只在最优解领域附近移动。符合模拟退火算法逻辑。

4. **是否接受新解**: $df = w_{current} - w_{try}$, $df > 0$ 表示新解更差, 反之新解更好, 对于更好解我们必定接受, 对于差解我们检查 **Metropolis 接受准则**, 即 df 满足:

$$e^{-\frac{df}{T}} > rand, rand \text{ 是一个随机数} \quad (19)$$

时, 可以更新当前解。为了保证不错过最优解, 记录一个历史最佳解 w_{best}, \vec{x}_{best} 不进入循环更新, 当当前解优于历史最佳解时候, 将历史最佳解更新为当前解。

$$\begin{cases} \text{若 } df < 0 \text{ 必定接受新解} \\ \text{若 } df \geq 0 \text{ 且满足 Metropolis 接受准则, 则接受新解} \\ \text{若 } df \geq 0 \text{ 但不满足 Metropolis 接受准则, 则不接受新解} \end{cases} \quad (20)$$

5. **判断是否满足循环条件**: 若当不满足跳出循环条件, 即当前温度依然大于结束温度且退火次数小于最大退火系数时, 更新温度 $T = T \cdot at$, 然后重复步骤 3-5。反之结束, 输出结果。

通过模拟退火我们可以求出保留三位小数的精度下, 我们无法分辨其与贪心思想求得解的区别:

情况 1 的最佳期望利润为 15.805, 最优结果是: $[0, 0, 0, 1]$

情况 2 的最佳期望利润为 6.040, 最优结果是: $[0, 0, 0, 1]$

情况 3 的最佳期望利润为 14.431, 最优结果是: $[0, 0, 1, 1]$

情况 4 的最佳期望利润为 8.160, 最优结果是: $[1, 1, 1, 1]$

情况 5 的最佳期望利润为 12.272, 最优结果是: $[0, 0, 1, 1]$

情况 6 的最佳期望利润为 18.587, 最优结果是: $[0, 0, 0, 0]$

6.2.3 模拟退火与贪心思想的检验

比较模拟退火仿真与贪心思想所得结果。对于所求得解, 是一个四维向量, 我们计算他们的欧氏距离, 除以贪心思想解的模长得到误差率; 对于求得的最优利润, 我们直接做差除以贪心思想求得的利润, 可以得到误差率。结果如表2所示:

用 *Matlab* 进行检验发现两种方法误差率极小, 可以忽略。其中注意到情况六最优解误差率给出的答案是 NaN, 产生这个结果的原因是情况六贪心思想的最优解是 $[0, 0, 0, 0]$, 模长为 0。重新验证情况六, 将模拟退火得到的结果作为分母, 得到的答案依然是 NaN, 说明模拟退火的答案也是模长为 0, 二者相等。

综上所述, 经过上述的模型求解与仿真检验, 得到如下表3的决策结果与依据。

表 2 模拟退火与贪心思想的误差率分析

情况	最优解欧氏距离误差率	最优利润误差率
1	0%	0%
2	0%	0%
3	0%	0%
4	0%	0%
5	0%	0%
6	NaN	0%

表 3 问题二决策结果与依据

	零配件 1 检测	零配件 2 检测	成品检测	次品拆解	期望利润（元/件）
情况 1	不检测	不检测	不检测	拆解	15.805
情况 2	不检测	不检测	不检测	拆解	6.040
情况 3	不检测	不检测	检测	拆解	14.431
情况 4	检测	检测	检测	拆解	8.160
情况 5	不检测	不检测	检测	拆解	12.272
情况 6	不检测	不检测	不检测	不拆解	18.587

七、问题三模型的建立与求解

7.1 模型建立

首先分析最一般的情况，即当整个流程有 m 道工序， n 个零配件的情况。我们采用**模块化的**思路，将这个大问题的根据工序拆解 m 个子问题。

如第一步由所有零配件合成了多个半成品，再将这若干半成品视为零配件，继续重复上述步骤，到最终合成成品。二对于零配件合成半成品这个子问题，它相较于问题二中的情况只是增加了零配件的个数，和成品的个数，减少了售卖和退回的过程。对于利润表达式(18)依然成立。可以由类似问题二的公式(7)到公式(17)的方式不断的递推表示得到。具体分析如下：

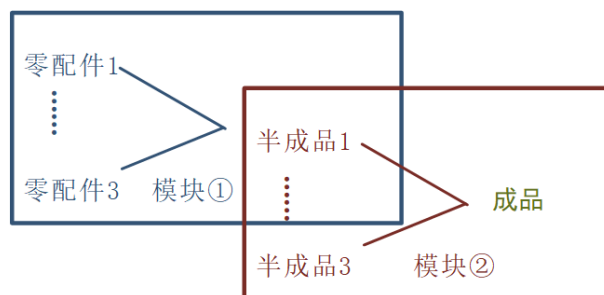


图9 工序模块化示意图

7.1.1 利润期望的计算

对于任意阶段的产品，均存在相对应的检测成本以及次品率。同时，对于所有半成品和成品，均存在相应的装配成本和拆解成本。因此，有如下的利润计算公式：

利润期望 w = 销售额

- 零配件购买成本 — 零配件检测成本
- 半成品装配成本 — 半成品检测成本 — 半成品拆解成本 + 拆解所得零配件成本
- 成品装配成本 — 成品检测成本 — 成品拆解成本 + 拆解所获零配件成本
- 调换成本 + 拆解所得零配件成本

(21)

在实际操作过程中，仅需要不断重复半成品的相关过程，即可得到完整的利润公式。

7.1.2 次品率的传递

次品率是唯一一项不断向后传递的量，因此，理清次品率的前后相关关系对于完善递归过程有重要意义。

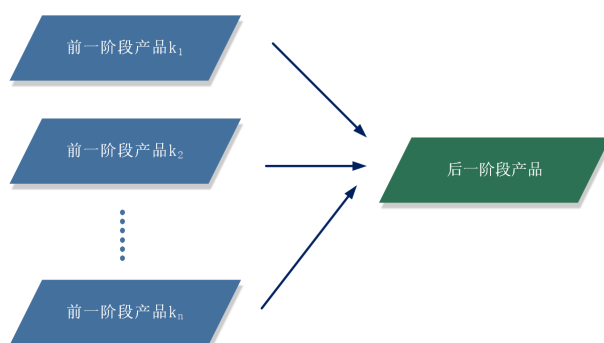


图10 装配关系示意图

对于上述装配关系示意图，后一阶段的产品次品率与前一阶段的产品次品率和前一

阶段产品全部正品得到后一阶段产品的次品率相关。

做如下角标规定：对于 $p_{ij}^{(k)}$

(1) i 为产品所在阶段，即 $i = 1$ 为零配件， $i \in [2, m]$ 为半成品， $i = m + 1$ 为成品。

(2) j 为产品在所在阶段内的编号。

(3) $k = 1$ 为前一阶段产品全为正品产生的次品率（此为已知量）， $k = 2$ 为考虑前一阶段产品存在次品导致后一阶段产品的次品率， $k = 3$ 为部分检测过后，去除检测出的次品后得到的次品率。

$$p_{is}^{(2)} = 1 - \prod_{j=k_1}^{k_n} (1 - p_{i-1j}^{(3)})(1 - p_{is}^{(1)}) \quad (22)$$

$$p_{is}^{(3)} = \frac{(1 - x_{is})p_{is}^{(2)}}{1 - x_{is}p_{is}^{(2)}} \quad (23)$$

其中 s 为第 i 阶段中的第 s 个零配件或者半成品。

由(22)和(23)，通过递推，可以得到全部各个阶段产品的次品率。

7.2 模型检验：以 $m = 2, n = 8$ 为例

图11展示了一种两道工序、8个零配件的组装情况：

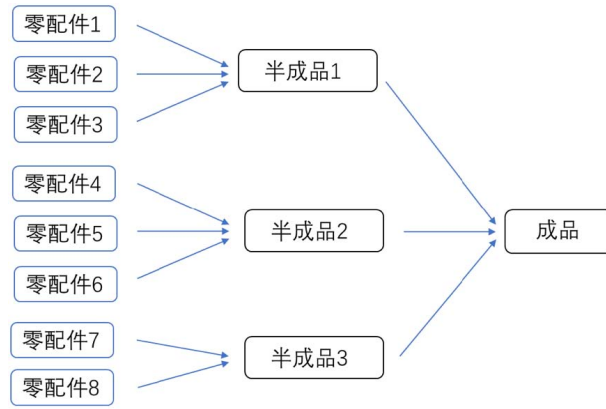


图 11 两道工序、8 个零配件的组装情况

第一步工序通过零配件 1-3 合成半成品 1，零配件 4-6 合成半成品 2，零配件 7-8 合成半成品 3；第二步工序再将半成品 1-3 合成成品。第一个工序中可以拆成三个模块，以零配件 1-3 合成半成品 1 为例子：

已知零配件 1-3 各自的次品率为 0.1，半成品 1 次品率为 0.1，即 $p_{11}^{(1)}, p_{12}^{(1)}, p_{13}^{(1)}, p_{21}^{(1)}$ ，接着我们实际求解 $p_{21}^{(2)}$ 和 $p_{21}^{(3)}$ ：

通过零配件 1 检测的比例：

$$a_1 = 1 - x_{11}p_{11}^{(1)} \quad (24)$$

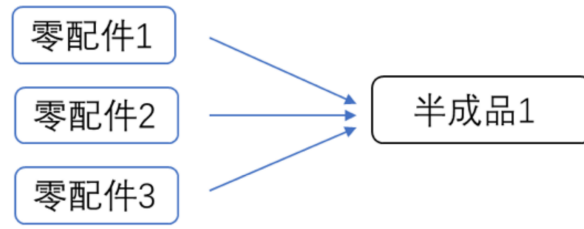


图 12 零配件 1-3 合成半成品 1

这包含了经过检测的合格零配件和没经过检测的全部零配件。同理可以得到 a_2 和 a_3 ，合成半成品 1 的部分仅有 $n_1 = \min(a_1, a_2, a_3)$ 。检测完成的零配件成品和次品混在一起，无法区分，这部分的次品占总体的比例就是 $p_{11}^{(3)}$, $p_{12}^{(3)}$, $p_{13}^{(3)}$

$$p_{11}^{(3)} = \frac{(1 - x_{11})p_{11}^{(2)}}{1 - x_{11}p_{11}^{(2)}} \quad (25)$$

其余两个同理。

根据公式(22)可以得到，组装完成后的 n_1 总体中，次品率 p_{21}^2 的表达式：

$$p_{21}^2 = 1 - (p_{11}^{(1)})(p_{12}^{(1)})(p_{13}^{(1)})(p_{21}^{(1)}) \quad (26)$$

进而得到，通过半成品 1 的检验后的总体次品率 $p_{21}^{(3)}$ 为：

$$p_{21}^3 = \frac{(1 - x_{21})p_{21}^{(2)}}{1 - x_{21}p_{21}^{(2)}} \quad (27)$$

以此类推，可以求得通过每一个过程的总体比例，进而求出各项行为包括购买零配件、组装、检测、售卖、调换带来的利润与损失。进而求得总利润

7.3 模型求解

7.3.1 基于贪心思想的决策求解

类似于问题二，我们考虑遍历整个可行域，求出最优解，但是由于有 16 个决策，计算机的算力不支持完成遍历，不得已将其视作 0-1 规划求解。结果如下：

根据题目条件，最佳的利润为：37.680000

流程 1 的第 1 个零配件全部检测

流程 1 的第 2 个零配件全部检测

流程 1 的第 3 个零配件全部检测

流程 1 的第 4 个零配件全部检测

流程 1 的第 5 个零配件全部检测

流程 1 的第 6 个零配件全部检测

流程 1 的第 7 个零配件全部检测
流程 1 的第 8 个零配件全部检测
流程 2 的第 1 个半成品全部检测
流程 2 的第 2 个半成品全部检测
流程 2 的第 3 个半成品全部检测
流程 3 的第 1 个成品不检测
流程 2 的第 1 个半成品全部拆解
流程 2 的第 2 个半成品全部拆解
流程 2 的第 3 个半成品全部拆解
流程 3 的第 1 个成品全部拆解

7.4 基于模拟退火算法的求解

类似问题二我们利用模拟退火算法进行仿真模拟。模拟退火可以求解到整个解空间中的最优解，而不会遇到遍历所产生的算例困境。更新思路上，每次迭代 8 个零配件检测中的一个决策、3 个半成品检测中的一个决策、成品检测决策、3 个半成品拆解中的一个决策和成品拆解决策。

最终结果如下所示：

根据题目条件，最佳的利润为：37.680000

流程 1 的第 1 个零配件全部检测
流程 1 的第 2 个零配件全部检测
流程 1 的第 3 个零配件全部检测
流程 1 的第 4 个零配件全部检测
流程 1 的第 5 个零配件全部检测
流程 1 的第 6 个零配件全部检测
流程 1 的第 7 个零配件全部检测
流程 1 的第 8 个零配件全部检测
流程 2 的第 1 个半成品全部检测
流程 2 的第 2 个半成品全部检测
流程 2 的第 3 个半成品全部检测
流程 3 的第 1 个成品不检测
流程 2 的第 1 个半成品全部拆解
流程 2 的第 2 个半成品全部拆解
流程 2 的第 3 个半成品全部拆解
流程 3 的第 1 个成品全部拆解

可以发现，二者结果完全相同。说明定理1依然成立。

八、问题四模型的建立与求解

在生产流程中，企业面临诸多决策问题，例如是否对零配件、半成品和成品进行检测，是否对不合格成品进行拆解等，这些决策将会直接影响企业的利润。其中零配件、半成品和成品的次品率是决策中非常重要的一环，**问题二和问题三中次品率采用的是供应商提供的标称值，而问题四中的次品率均是通过抽样检测方法得到的。**

8.1 模型的建立

8.1.1 抽样检测估计

首先由次品率的分布（式(4)）得到随机抽样检测的次品率观测值 \hat{p} ，再根据**波动理论的枢轴量法**求得次品率的置信区间，用于描述该次品率在抽样检测中存在一定的波动性。置信区间是统计学中的一个重要概念，用于估计总体参数（如均值、比例、方差等）的一个区间范围，该区间以一定的概率（即置信水平）包含真正的总体参数值。

在总体方差已知的情况下，次品率 p 的双侧置信区间为

$$(\hat{p} - \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} Z_{1-\frac{\alpha}{2}}, \hat{p} + \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} Z_{1-\frac{\alpha}{2}}) \quad (28)$$

- \hat{p} 表示通过抽样检测得到的次品率
- n 表示通过简单随机抽样的最小样本数公式（式(6)）计算得到的样本量，其中允许误差设置为 $E = 0.01$

- α 表示显著水平， $1 - \alpha$ 表示次品率 p 的置信水平，该问中取 $\alpha = 0.01$

置信区间反映了次品率的波动范围，该区间能够更真实地反映抽样检测过程中次品率地不确定性。

8.1.2 蒙特卡洛模拟

针对**问题二和问题三**中的每种情况，在置信区间内使用蒙特卡洛模拟方法，随机产生样本的次品率，以此替换零配件、半成品与成品的固定次品率，并重新计算 N 种情况下的生产决策与期望收益。

$$E(w) = \frac{1}{N} \sum_{i=1}^N w_i \quad (29)$$

将次品率围绕标称值波动的范围内的期望收益值绘制成变化趋势图，与原决策与原收益比较，得出抽样检测次品率的合理性。

$$\forall p_{ij}^t \in (\underline{\hat{p}}, \bar{\hat{p}}) \quad (30)$$

其中 p_{ij}^t 表示在第 t 钟情况下, 第 i 阶段第 j 个产品的抽样检测次品率, $\underline{\hat{p}}$ 表示置信区间下界, $\bar{\hat{p}}$ 表示置信区间上界。

8.1.3 问题二的生产决策与收益分析

在各置信区间已知的情况下, 针对每一个零配件及成品的次品率随机更新为相应置信区间的一个值, 重复问题二的遍历和模拟退火过程, 得到新的最优策略以及最佳利润。将得到的策略与问题二中的策略进行比较, 观察策略改变情况, 分析产生变化的原因, 比较利润变化, 分析利润变化原因。

8.1.4 问题三的生产决策与收益分析

同理, 对于问题三, 再次改变每一个次品率, 然后再重新利用**模拟退火算法**进行计算最佳的决策方案。比较决策方案的变化, 分析产生变化的原因。同样由于一共有十六个决策变量, 不可能将区间分割得足够细致来遍历求解。我们选择遍历时将此题视作 0-1 规划问题, 试着求解并于模拟退火的结果进行比较。

8.2 模型的求解

8.2.1 问题二重新求解

大量重复实验表明, 对于情况 1、2、3、5、6, 次品率的波动对于检测的策略不会产生影响, 而只影响利润。但是对于情况 4 而言, 最优策略中检测零配件 1 和检测零配件 2 的比例与问题二中的结果有波动, 且波动的大小会随着不同次品率改变而改变。但是无论如何定理1似乎不再成立。最终结果如下:

遍历结果寻找最佳:

情况 1 的最佳期望利润为 15.373446, 最优结果是: $[0,0,0,1]$

情况 2 的最佳期望利润为 6.357039, 最优结果是: $[0,0,0,1]$

情况 3 的最佳期望利润为 14.374709, 最优结果是: $[0,0,1,1]$

情况 4 的最佳期望利润为 8.265174, 最优结果是: $[1.0000\ 0.9200\ 1.0000\ 1.0000]$

情况 5 的最佳期望利润为 12.226724, 最优结果是: $[0,0,1,1]$

情况 6 的最佳期望利润为 17.034863, 最优结果是: $[0,0,0,0]$

模拟退火方法:

情况 1 的最佳期望利润为 15.373446, 最优结果是: $[0,0,0,1]$

情况 2 的最佳期望利润为 6.357039, 最优结果是: $[0,0,0,1]$

情况 3 的最佳期望利润为 14.374709, 最优结果是: $[0,0,1,1]$

情况 4 的最佳期望利润为 8.128315, 最优结果是: $[1.0000\ 0.8944\ 1.0000\ 1.0000]$

情况 5 的最佳期望利润为 12.226724, 最优结果是: $[0,0,1,1]$

情况 6 的最佳期望利润为 17.034863, 最优结果是: $[0,0,0,0]$

计算两种方法的误差:

情况 1 的最优解欧氏距离误差率是 0, 最优利润误差率是 0

情况 2 的最优解欧氏距离误差率是 0, 最优利润误差率是 0

情况 3 的最优解欧氏距离误差率是 0, 最优利润误差率是 0

情况 4 的最优解欧氏距离误差率是 0.013039, 最优利润误差率是 0.016558

情况 5 的最优解欧氏距离误差率是 0, 最优利润误差率是 0

情况 6 的最优解欧氏距离误差率是 NaN, 最优利润误差率是 0

比较问题二结果与上述结果, 发现两者决策基本相同。具体而言, 正对于情况 1、2、3、5、6, 我们计算出相应波动区间的利润区间, 并比较利润的期望与问题二中的利润。结果如下

对于情况 1 在次品率 10% 波动在区间 $[0.087100, 0.107100]$ 的时候, 利润波动区间为 $[15.034724, 17.235930]$

此时利润期望为: 16.127283 元

当次品率为 10% 时, 利润为 15.805000 元

对于情况 2 在次品率 20% 波动在区间 $[0.194800, 0.214800]$ 的时候, 利润波动区间为 $[4.880152, 6.492207]$

此时利润期望为: 5.652026 元

当次品率为 20% 时, 利润为 6.040000 元

对于情况 3 在次品率 10% 波动在区间 $[0.087100, 0.107100]$ 的时候, 利润波动区间为 $[13.763427, 15.671140]$

此时利润期望为: 14.710312 元

当次品率为 10% 时, 利润为 14.431000 元

对于情况 5 在次品率 10% 波动在区间 $[0.087100, 0.107100]$, 次品率 20% 波动在区间 $[0.194800, 0.195002]$ 的时候, 利润波动区间为 $[11.414652, 13.170667]$

此时利润期望为: 12.285965 元

当次品率不波动时, 利润为 12.272000 元

对于情况 6 在次品率 5% 波动在区间 $[0.040800, 0.060800]$ 的时候, 利润波动区间为 $[16.678701, 20.246716]$

此时利润期望为: 18.450306 元

当次品率为 5% 时, 利润为 18.586750 元

对于情况 4, 经过实验发现零配件 1 次品率与零配件 2 次品率相差不大时候, 利润随着零配件 1 和零配件 2 的检测率增加而增加。而当零配件 1 次品率远小于 (相对于置信区间两端来说) 零配件 2 次品率时, 随着零配件 2 检测率增加利润会先上升再下降, 在约 90% 到 100% 中间产生一个拐点; 当零配件 2 次品率远小于 (相对于置信区

间两端来说) 零配件 1 次品率时, 随着零配件 1 检测率增加利润会先上升再下降, 在约 90% 到 100% 中间产生一个拐点。以零配件 1 的次品率为 0.188000, 零配件 2 的次品率为 0.208000, 成品的次品率为 0.188896 为例, 结果如图13所示:

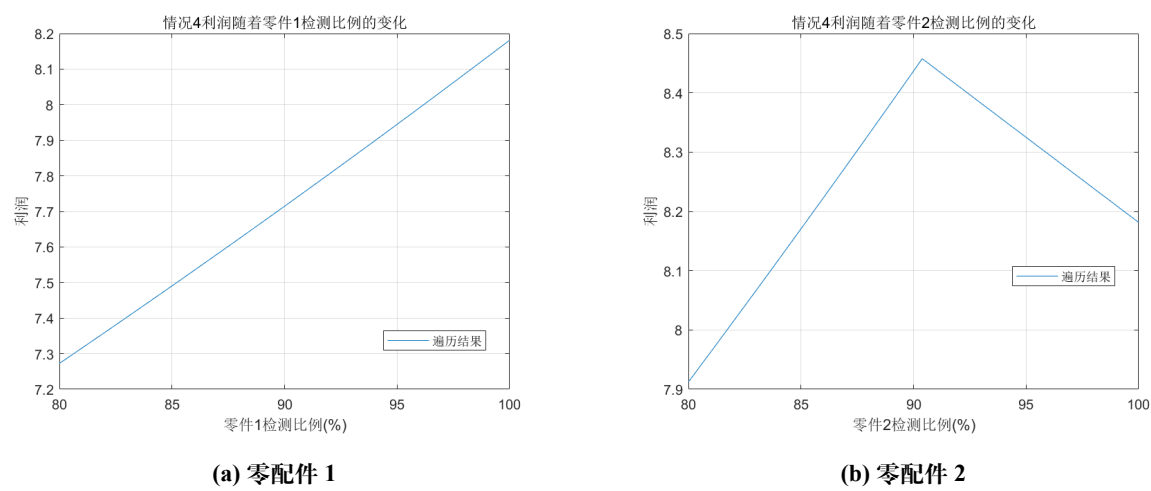


图 13 情况 4 利润随着零配件检测比例的变化

8.2.2 问题三重新求解

首先更新每一个零配件、半成品和成品的次品率, 然后利用模拟退火进行实验, 求出每一个零配件的最佳检测率, 结果如下:

- 最佳的利润为: 53.929766 元。
- 流程 1 的第 1 个零配件应当有 91.456520% 检测
- 流程 1 的第 2 个零配件应当有 69.327321% 检测
- 流程 1 的第 3 个零配件应当有 66.550733% 检测
- 流程 1 的第 4 个零配件应当有 98.976913% 检测
- 流程 1 的第 5 个零配件应当有 88.706974% 检测
- 流程 1 的第 6 个零配件应当有 99.153347% 检测
- 流程 1 的第 7 个零配件应当有 93.600751% 检测
- 流程 1 的第 8 个零配件应当有 71.962191% 检测
- 流程 2 的第 1 个半成品全部检测
- 流程 2 的第 2 个半成品全部检测
- 流程 2 的第 3 个半成品应当有 99.999751% 检测
- 流程 3 的第 1 个成品不检测
- 流程 2 的第 1 个半成品/成品全部拆解
- 流程 2 的第 2 个半成品/成品应当有全部拆解
- 流程 2 的第 3 个半成品/成品应当有全部拆解

流程 3 的第 1 个半成品/成品全部拆解

紧接着我们在将该问题视为 0-1 规划，重新遍历求解最佳策略，结果如下：

根据题目条件，最佳的利润为：50.448853

流程 1 的第 1 个零配件全部检测

流程 1 的第 2 个零配件全部检测

流程 1 的第 3 个零配件全部检测

流程 1 的第 4 个零配件全部检测

流程 1 的第 5 个零配件全部检测

流程 1 的第 6 个零配件全部检测

流程 1 的第 7 个零配件全部检测

流程 1 的第 8 个零配件全部检测

流程 2 的第 1 个半成品全部检测

流程 2 的第 2 个半成品全部检测

流程 2 的第 3 个半成品全部检测

流程 3 的第 1 个成品不检测

流程 2 的第 1 个半成品/成品全部拆解

流程 2 的第 2 个半成品/成品全部拆解

流程 2 的第 3 个半成品/成品全部拆解

流程 3 的第 1 个半成品/成品全部拆解

比较问题三结果与上述结果，发现两者决策**完全相同**。

比较 0-1 规划遍历与模拟退火求解结果，我们不难发现，模拟退火算法求得的最优解并不在全部检测或者全都不检测零配件、半成品和成品时候取到，并且模拟退火求得的解比遍历得到的利润高。说明最佳策略应当对大部分的零配件、半成品和成品进行检测，但是过多的检测会降低最终所得利润，企业应当根据实际次品率做出相应调整。对于实际次品率未知的情况，企业可以考虑按照问题三中的决策所得利润与模拟退火得出的最佳利润相差不到 5%。

九、模型评价与改进

9.1 模型的优点

1. 全文的建模始终基于数学理性思维，利用概率论与数理统计知识、目标决策、数学证明、优化方程等数学理论建立模型，并且进行了结论分析、原因探究、仿真检验等，模型建立、结果分析全面严谨。

2. 问题三中模块化处理，使得拓展模型变成可能，通过分模块将每道工序不断递归，可以解决对于任意的 m 道工序以及 n 个零配件的情况。

3. 在解决整个问题的过程中,使用了模拟退火算法对求解结果仿真验证,并不断优化,可以在较短时间内得到较为精确的结果,并通过仿真检验证明了模型求解的正确性与可靠性。

9.2 模型的缺点

1. 在问题一中,最小样本数的估计采用的是简单随机抽样公式,并未采用分层抽样检测或动态抽样检测方法来验证结果的准确性。

2. 在问题二和问题三中,采用了遍历和模拟退火算法,其中,遍历只能进行 01 变量的遍历,无法模拟部分检测的情况。

9.3 模型的改进

对于问题二与问题三的模型,并未考虑用来给客户调换的正品的相关成本与损失,直接认为出售失败并且承担调换损失,属于对模型的简化。事实上,用于调换的合格成品也需要经过装配,检测等工序。因此,我们可以将用于调换的合格成品也考虑进模型中,使模型更加与实际情况吻合。

十、参考文献

[1] 葛祥龙,史成东. 次品率对闭环供应链的影响分析和优化决策研究 [J]. 物流科技, 2020, 43 (10): 137-143. DOI:10.13714/j.cnki.1002-3100.2020.10.034.

[2] 李秀喜,钱宇. 一种新的优化算法——进化-模拟退火算法 (ESA)[J]. 吉首大学学报 (自然科学版),1997,(03):13-17.

十一、附件

11.1 question1_main.m

```
1 clear;
2 clc;
3
4 %设置次品率和误差精度
5 p0 = 0.1;
6 E = 0.01:0.001:0.08;
7 alpha_95 = 0.05;
8 alpha_90 = 0.1;
9
10 %计算临界值
11 z_alpha_95 = norminv(1-alpha_95,0,1);
```

```

12 z_alpha_90 = norminv(1-alpha_90,0,1);
13
14 %计算最小样本量
15 n_95 = z_alpha_95^2*p0*(1-p0)./E.^2;
16 n_90 = z_alpha_90^2*p0*(1-p0)./E.^2;
17 %取整
18 n_95_integer = round(n_95);
19 n_90_integer = round(n_90);
20
21 %绘制样本量与允许误差的趋势图
22 plot(E, n_90_integer, 'r-', 'LineWidth', 1, 'DisplayName', '信度为90%');
23 hold on
24 plot(E, n_95_integer, 'g-', 'LineWidth', 1, 'DisplayName', '信度为95%');
25 % 设置图例、标题和坐标轴标签
26 legend show;
27 title('不同抽样误差下的最小样本量趋势图','FontSize',8,'FontWeight','bold');
28 xlabel('允许抽样误差','FontSize',8);
29 ylabel('最小样本量 (单位: 个) ','FontSize',8);

```

11.2 question2_main.m

```

1 clear,clc;
2 A = [0.1 4 2 0.1 18 3 0.1 6 3 56 6 5
3       0.2 4 2 0.2 18 3 0.2 6 3 56 6 5
4       0.1 4 2 0.1 18 3 0.1 6 3 56 30 5
5       0.2 4 1 0.2 18 1 0.2 6 2 56 30 5
6       0.1 4 8 0.2 18 1 0.1 6 2 56 10 5
7       0.05 4 2 0.05 18 3 0.05 6 3 56 10 40];
8
9 c_cir = cell(6,2);
10 disp('遍历结果寻找最佳: ')
11 for check = 1:size(A,1)
12     w = -inf;
13     p = -ones(1,4);
14     lingpeijian = [A(check,1:3);A(check,4:6)];
15     chengpin = A(check,7:10);
16     buhege = A(check,11:12);
17     for x1 = 0:0.01:1
18         for x2 = 0:0.01:1
19             for x3 = 0:0.01:1
20                 for x4 = 0:0.01:1
21
22                     w_temp = question2_profit([x1,x2,x3,x4],lingpeijian,chengpin,buhege);
23                     if w_temp>w
24                         w = w_temp;

```

```

25         p = [x1,x2,x3,x4];
26     end
27 end
28 end
29 end
30 end
31 fprintf('情况%f 的最佳期望利润为%f ,最优结果是: ',check,w);
32 disp(p);
33 c_cir{check,1} = w;
34 c_cir{check,2} = p;
35 end
36
37 c_SA = cell(6,2);
38 disp('模拟退火方法: ')
39 for check = 1:size(A,1)
40     % check = 1;
41     lingpeijian = [A(check,1:3);A(check,4:6)];
42     chengpin = A(check,7:10);
43     buhege = A(check,11:12);
44     [w,p]=question2_SA(lingpeijian,chengpin,buhege);
45     fprintf('情况%f 的最佳期望利润为%f ,最优结果是: ',check,w);
46     disp(p);
47     c_SA{check,1} = w;
48     c_SA{check,2} = p;
49 end
50
51
52 d = zeros(1,6);
53 format longE
54 for i = 1:6
55     err_q2 = pdist([c_SA{i,2};c_cir{i,2}], "euclidean");
56     fprintf('情况%f 的最优解欧氏距离误差率是%f , 最优利润误差率是%f
57         \n',i,err_q2/norm(c_cir{i,2}),abs(c_cir{i,1}-c_SA{i,1})/c_cir{i,1});
58 end
59 format short

```

11.3 question3_main.m

```

1 clear,clc
2 lingpeijian = [
3     0.1 2 1
4     0.1 8 1
5     0.1 12 2
6     0.1 2 1
7     0.1 8 1

```

```

8     0.1 12 2
9     0.1 8 1
10    0.1 12 2
11    ];
12    banchengpin = [
13        0.1 8 4 6
14        0.1 8 4 6
15        0.1 8 4 6
16    ];
17    chengpin = [0.1 8 6 4];
18    sell_change = [200 40];
19
20    parameter = {lingpeijian;banchengpin;chengpin;sell_change};
21    hechenglujing = {[3 3 2];3};
22    plan = cell(2,1);
23
24    [w, sol] = question3_SA(parameter,hechenglujing);
25    x_sol = sol{1}
26    b_sol = sol{2}
27    fprintf('根据题目条件, 最佳的利润为: %f\n',w);
28    for i = 1:3
29        x_temp = x_sol{i};
30        for j = 1:length(x_temp)
31            if x_temp(j) == 0
32                fprintf('流程%f 的第 %f 个零配件不检测\n',i,j);
33            elseif x_temp(j) == 1
34                fprintf('流程%f 的第 %f 个零配件全部检测\n',i,j);
35            else
36                fprintf('流程%f 的第 %f 个零配件应当有%f %% 检测\n',i,j,...
37                    100*x_temp(j));
38            end
39        end
40    end
41    for i = 1:2
42        b_temp = b_sol{i};
43        for j = 1:length(b_temp)
44            if b_temp(j) == 0
45                fprintf('流程%f 的第 %f 个半成品/成品不拆解\n',i+1,j);
46            elseif b_temp(j) == 1
47                fprintf('流程%f 的第 %f 个半成品/成品全部拆解\n',i+1,j);
48            else
49                fprintf('流程%f 的第 %f 个零配件应当有%f %% 拆解\n',i+1,j,...
50                    100*b_temp(j));
51            end
52        end
53    end
54

```

```

55 % 遍历检验
56 w_best = -inf;
57 sol=cell(2,1);
58 for x11 = 0:1
59 for x12 = 0:1
60 for x13 = 0:1
61 for x14 = 0:1
62 for x15 = 0:1
63 for x16 = 0:1
64 for x17 = 0:1
65 for x18 = 0:1
66 x_sol1 = [x11 x12 x13 x14 x15 x16 x17 x18];
67 for x21 = 0:1
68 for x22 = 0:1
69 for x23 = 0:1
70 x_sol2 = [x21 x22 x23];
71 for x31 = 0:1
72 x_sol_try = {x_sol1;x_sol2;x31};
73 for b21 = 0:1
74 for b22 = 0:1
75 for b23 = 0:1
76 b_sol1 = [b21,b22,b23];
77 for b31 = 0:1
78 b_sol_try = {b_sol1;b31};
79 w_try = question3_profit(2,8,x_sol_try,b_sol_try,parameter,hechenglujing);
80 if w_try>w_best
81 w_best = w_try;
82 sol_best = {x_sol_try;b_sol_try};
83 end
84
85 end
86 end
87 end
88 end
89 end
90 end
91 end
92 end
93 end
94 end
95 end
96 end
97 end
98 end
99 end
100 end
101

```

```

102
103 w_best_bianli = w_best;
104 sol_best_bianli = sol_best;
105 x_sol_bianli = sol_best_bianli{1};
106 b_sol_bianli = sol_best_bianli{2};
107 fprintf('根据题目条件, 最佳的利润为: %f\n',w_best_bianli);
108 for i = 1:3
109     x_temp = x_sol_bianli{i};
110     for j = 1:length(x_temp)
111         if x_temp(j) == 0
112             fprintf('流程%f 的第 %f 个零配件不检测\n',i,j);
113         elseif x_temp(j) == 1
114             fprintf('流程%f 的第 %f 个零配件全部检测\n',i,j);
115         else
116             fprintf('流程%f 的第 %f 个零配件应当有%f %% 检测\n',i,j,...
117                 100*x_temp(j));
118         end
119     end
120 end
121 for i = 1:2
122     b_temp = b_sol_bianli{i};
123     for j = 1:length(b_temp)
124         if b_temp(j) == 0
125             fprintf('流程%f 的第 %f 个半成品/成品不拆解\n',i+1,j);
126         elseif b_temp(j) == 1
127             fprintf('流程%f 的第 %f 个半成品/成品全部拆解\n',i+1,j);
128         else
129             fprintf('流程%f 的第 %f 个零配件应当有%f %% 拆解\n',...
130                 i+1,j,100*b_temp(j));
131         end
132     end
133 end
134
135 err_q3 = pdist([x_sol{1},x_sol{2},x_sol{3},b_sol{1},b_sol{2};...
136     x_sol_bianli{1},x_sol_bianli{2},x_sol_bianli{3},b_sol_bianli{1},...
137     b_sol_bianli{2}], "euclidean");
138
139 fprintf('最优解欧氏距离误差率是%f %, 最优利润误差率是%f % \n',...
140     100*err_q3/norm([x_sol_bianli{1},x_sol_bianli{2},x_sol_bianli{3},...
141     b_sol_bianli{1},b_sol_bianli{2}]), 100*abs(w_best_bianli-w)...
142     /w_best_bianli);

```

11.4 question4_main.m

```

1 clear;clc

```

```

2 range = question4_range;
3
4 A = [0.1 4 2 0.1 18 3 0.1 6 3 56 6 5
5      0.2 4 2 0.2 18 3 0.2 6 3 56 6 5
6      0.1 4 2 0.1 18 3 0.1 6 3 56 30 5
7      0.2 4 1 0.2 18 1 0.2 6 2 56 30 5
8      0.1 4 8 0.2 18 1 0.1 6 2 56 10 5
9      0.05 4 2 0.05 18 3 0.05 6 3 56 10 40];
10
11 [n,m] = size(A);
12 for i = 1:n
13     for j = 1:m
14         if A(i,j) == 0.05
15             A(i,j) = range(1,1) + rand*(range(1,2)-range(1,1));
16         elseif A(i,j) == 0.1
17             A(i,j) = range(2,1) + rand*(range(2,2)-range(2,1));
18         elseif A(i,j) == 0.2
19             A(i,j) = range(3,1) + rand*(range(3,2)-range(3,1));
20         end
21     end
22 end
23 disp('将所有次品率替换为一个随机的参数，重新给出所有参数：')
24 disp(A);
25
26 c_cir = cell(6,2);
27 disp('遍历结果寻找最佳：')
28 for check = 1:size(A,1)
29     w = -inf;
30     p = -ones(1,4);
31     lingpeijian = [A(check,1:3);A(check,4:6)];
32     chengpin = A(check,7:10);
33     buhege = A(check,11:12);
34     for x1 = 0:0.02:1
35         for x2 = 0:0.02:1
36             for x3 = 0:0.02:1
37                 for x4 = 0:0.02:1
38                     w_temp = question2_profit([x1,x2,x3,x4],...
39                                                lingpeijian,chengpin,buhege);
40                     if w_temp>w
41                         w = w_temp;
42                         p = [x1,x2,x3,x4];
43                     end
44                 end
45             end
46         end
47     end
48 fprintf('情况%f 的最佳期望利润为%f ,最优结果是： ',check,w);

```

```

49     disp(p);
50     c_cir{check,1} = w;
51     c_cir{check,2} = p;
52 end
53
54 c_SA = cell(6,2);
55 disp('模拟退火方法: ')
56 for check = 1:size(A,1)
57     % check = 1;
58     lingpeijian = [A(check,1:3);A(check,4:6)];
59     chengpin = A(check,7:10);
60     buhege = A(check,11:12);
61     [w,p]=question2_SA(lingpeijian,chengpin,buhege);
62     fprintf('情况%f 的最佳期望利润为%f ,最优结果是: ',check,w);
63     disp(p);
64     c_SA{check,1} = w;
65     c_SA{check,2} = p;
66 end
67
68 d = zeros(1,6);
69 format longE
70 for i = 1:6
71     err1 = pdist([c_SA{i,2};c_cir{i,2}], "euclidean");
72     fprintf(['情况%f 的最优解欧氏距离误差率是%f , 最优利润误差率' ...
73         '%f \n'],i,err1/norm(c_cir{i,2}),...
74         abs(c_cir{i,1}-c_SA{i,1})/c_cir{i,1});
75 end
76 format short
77
78 clear;
79 range = question4_range;
80 y_biaozhun = [15.805 6.040 14.431 8.160 12.272 18.586750];
81
82 A = [0.1 4 2 0.1 18 3 0.1 6 3 56 6 5
83     0.2 4 2 0.2 18 3 0.2 6 3 56 6 5
84     0.1 4 2 0.1 18 3 0.1 6 3 56 30 5
85     0.2 4 1 0.2 18 1 0.2 6 2 56 30 5
86     0.1 4 8 0.2 18 1 0.1 6 2 56 10 5
87     0.05 4 2 0.05 18 3 0.05 6 3 56 10 40];
88 x1 = linspace(range(1,1),range(1,2),100);
89 x2 = linspace(range(2,1),range(2,2),100);
90 x3 = linspace(range(3,1),range(3,2));
91
92 % 情况1
93 check = 1;
94 lingpeijian = [A(check,1:3);A(check,4:6)];
95 chengpin = A(check,7:10);

```



```

96 buhege = A(check,11:12);
97 w = zeros(1,100);
98
99 for i = 1:100
100     lingpeijian(1,1) = x2(i); % 原始是10%, 修改为置信区间的值
101     lingpeijian(2,1) = x2(i);
102     chengpin(1) = x2(i);
103     w(i) = question2_SA(lingpeijian,chengpin,buhege);
104 end
105 fprintf(['对于情况1在次品率10%% 波动在区间[%f ,%f ]的时候, ' ...
106     '利润波动区间为[%f ,%f ]'],x2(1),x2(100),min(w),max(w));
107 fprintf('此时利润期望为: %f元',sum(w)/length(w));
108 fprintf('当次品率为10%%时, 利润为%f元',y_biaozhun(check));
109
110 % 情况2
111 check = 2;
112 lingpeijian = [A(check,1:3);A(check,4:6)];
113 chengpin = A(check,7:10);
114 buhege = A(check,11:12);
115 w = zeros(1,100);
116
117 for i = 1:100
118     lingpeijian(1,1) = x3(i); % 原始是20%, 修改为置信区间的值
119     lingpeijian(2,1) = x3(i);
120     chengpin(1) = x3(i);
121     w(i) = question2_SA(lingpeijian,chengpin,buhege);
122 end
123 fprintf(['对于情况2在次品率20%% 波动在区间[%f ,%f ]的时候, ' ...
124     '利润波动区间为[%f ,%f ]元'],x3(1),x3(100),min(w),max(w));
125 fprintf('此时利润期望为: %f元',sum(w)/length(w));
126 fprintf('当次品率为20%%时, 利润为%f元',y_biaozhun(check));
127
128 % 情况3
129 check = 3;
130 lingpeijian = [A(check,1:3);A(check,4:6)];
131 chengpin = A(check,7:10);
132 buhege = A(check,11:12);
133 w = zeros(1,100);
134
135 for i = 1:100
136     lingpeijian(1,1) = x2(i); % 原始是10%, 修改为置信区间的值
137     lingpeijian(2,1) = x2(i);
138     chengpin(1) = x2(i);
139     w(i) = question2_SA(lingpeijian,chengpin,buhege);
140 end
141 fprintf(['对于情况3在次品率10%% 波动在区间[%f ,%f ]的时候, ' ...
142     '利润波动区间为[%f ,%f ]'],x2(1),x2(100),min(w),max(w));

```

```

143 fprintf('此时利润期望为: %f元',sum(w)/length(w));
144 fprintf('当次品率为10%%时, 利润为%f元',y_biaozhun(check));
145
146 % 情况5
147 check = 5;
148 lingpeijian = [A(check,1:3);A(check,4:6)];
149 chengpin = A(check,7:10);
150 buhege = A(check,11:12);
151 w = zeros(1,100);
152
153 for i = 1:100
154     lingpeijian(1,1) = x2(i); % 原始是10%, 修改为置信区间的值
155     lingpeijian(2,1) = x3(i);
156     chengpin(1) = x2(i);
157     w(i) = question2_SA(lingpeijian,chengpin,buhege);
158 end
159 fprintf(['对于情况5在次品率10%% 波动在区间[%f ,%f ], 次品率20%% ' ...
160         '波动在区间[%f ,%f ]的时候, 利润波动区间为[%f ,%f ]'],...
161         x2(1),x2(100),x3(1),x3(2),min(w),max(w));
162 fprintf('此时利润期望为: %f元',sum(w)/length(w));
163 fprintf('当次品率不波动时, 利润为%f元',y_biaozhun(check));
164
165 % 情况6
166 check = 6;
167 lingpeijian = [A(check,1:3);A(check,4:6)];
168 chengpin = A(check,7:10);
169 buhege = A(check,11:12);
170 w = zeros(1,100);
171
172 for i = 1:100
173     lingpeijian(1,1) = x1(i); % 原始是10%, 修改为置信区间的值
174     lingpeijian(2,1) = x1(i);
175     chengpin(1) = x1(i);
176     w(i) = question2_SA(lingpeijian,chengpin,buhege);
177 end
178 fprintf(['对于情况6在次品率5%% 波动在区间[%f ,%f ]的时候, ' ...
179         '利润波动区间为[%f ,%f ]'],x1(1),x1(100),min(w),max(w));
180 fprintf('此时利润期望为: %f元',sum(w)/length(w));
181 fprintf('当次品率为5%%时, 利润为%f元',y_biaozhun(check));
182
183 % 单独分析情况4
184 check = 4;
185 lingpeijian = [A(check,1:3);A(check,4:6)];
186 chengpin = A(check,7:10);
187 buhege = A(check,11:12);
188 w = zeros(1,100);
189 % p_rand1 = range(3,1) + rand*(range(3,2)-range(3,1));

```

```

190 p_rand1 = range(3,1);
191 % p_rand2 = range(3,1) + rand*(range(3,2)-range(3,1));
192 p_rand2 = range(3,2);
193 p_rand3 = range(3,1) + rand*(range(3,2)-range(3,1));
194 lingpeijian(1,1)=p_rand1;
195 lingpeijian(2,1)=p_rand2;
196 chengpin(chengpin==0.2)=p_rand3;
197 fprintf(['当零配件1的次品率更改为%f, 零配件2的次品率更改为%f, ' ...
198         '成品的次品率更改为%f时: ',p_rand1,p_rand2,p_rand3);
199
200 % 改变零配件1检测率的情况:
201 x_plot = linspace(0.8,1,1000);
202 x_sol = [1 1 1 1];
203 w_plot = zeros(1,length(x_plot));
204 for i = 1:length(x_plot)
205     x_sol(1) = x_plot(i);
206     w_plot(i) = question2_profit(x_sol,lingpeijian,chengpin,buhege);
207 end
208 figure;
209 plot(100*x_plot,w_plot,DisplayName='遍历结果');
210 title('情况4利润随着零件1检测比例的变化')
211 xlabel('零件1检测比例(%)')
212 ylabel('利润')
213 grid on
214 legend show
215 legend Location best
216 % 改变零配件2检测率的情况:
217 x_plot = linspace(0.8,1,1000);
218 x_sol = [1 1 1 1];
219 w_plot = zeros(1,length(x_plot));
220 for i = 1:length(x_plot)
221     x_sol(2) = x_plot(i);
222     w_plot(i) = question2_profit(x_sol,lingpeijian,chengpin,buhege);
223 end
224 figure;
225 plot(100*x_plot,w_plot,DisplayName='遍历结果');
226 title('情况4利润随着零件2检测比例的变化')
227 xlabel('零件2检测比例(%)')
228 ylabel('利润')
229 grid on
230 legend show
231 legend Location best
232
233 % 重做问题三
234 clear;
235 range = question4_range;
236 p = range(2,:);

```

```

237 lingpeijian = [
238     0.1 2 1
239     0.1 8 1
240     0.1 12 2
241     0.1 2 1
242     0.1 8 1
243     0.1 12 2
244     0.1 8 1
245     0.1 12 2
246 ];
247 banchengpin = [
248     0.1 8 4 6
249     0.1 8 4 6
250     0.1 8 4 6
251 ];
252 chengpin = [0.1 8 6 4];
253 sell_change = [200 40];
254 for i = 1:8
255     lingpeijian(i,1) = range(1) + rand*(range(2)-range(1));
256 end
257 for i = 1:3
258     banchengpin(i,1) = range(1) + rand*(range(2)-range(1));
259 end
260 chengpin(1) = range(1) + rand*(range(2)-range(1));
261 parameter = {lingpeijian;banchengpin;chengpin;sell_change};
262 hechenglujing = {[3 3 2];3};
263 plan = cell(2,1);
264
265 [w, sol] = question3_SA(parameter,hechenglujing);
266 x_sol = sol{1};
267 b_sol = sol{2};
268 fprintf('根据题目条件，最佳的利润为： %f\n',w);
269 for i = 1:3
270     x_temp = x_sol{i};
271     for j = 1:length(x_temp)
272         if x_temp(j) == 0
273             fprintf('流程%f 的第 %f 个零配件/半成品/成品不检测\n',i,j);
274         elseif x_temp(j) == 1
275             fprintf('流程%f 的第 %f 个零配件/半成品/成品全部检测\n',i,j);
276         else
277             fprintf(['流程%f 的第 %f 个零配件/半成品/成品应当有' ...
278                 '%f %% 检测\n'],i,j,100*x_temp(j));
279         end
280     end
281 end
282 for i = 1:2
283     b_temp = b_sol{i};

```

```

284     for j = 1:length(b_temp)
285         if b_temp(j) == 0
286             fprintf('流程%f 的第 %f 个半成品/成品不拆解\n',i+1,j);
287         elseif b_temp(j) == 1
288             fprintf('流程%f 的第 %f 个半成品/成品全部拆解\n',i+1,j);
289         else
290             fprintf('流程%f 的第 %f 个半成品/成品应当有%f %% 拆解\n',...
291                 i+1,j,100*b_temp(j));
292         end
293     end
294 end
295
296 % 遍历检验
297 w_best = -inf;
298 sol=cell(2,1);
299 for x11 = 0:1
300     for x12 = 0:1
301         for x13 = 0:1
302             for x14 = 0:1
303                 for x15 = 0:1
304                     for x16 = 0:1
305                         for x17 = 0:1
306                             for x18 = 0:1
307                                 x_sol1 = [x11 x12 x13 x14 x15 x16 x17 x18];
308                                 for x21 = 0:1
309                                     for x22 = 0:1
310                                         for x23 = 0:1
311                                             x_sol2 = [x21 x22 x23];
312                                             for x31 = 0:1
313                                                 x_sol_try = {x_sol1;x_sol2;x31};
314                                                 for b21 = 0:1
315                                                     for b22 = 0:1
316                                                         for b23 = 0:1
317                                                             b_sol1 = [b21,b22,b23];
318                                                             for b31 = 0:1
319                                                                 b_sol_try = {b_sol1;b31};
320                                                                 w_try = question3_profit(2,8,x_sol_try,b_sol_try,...
321                                                                     parameter,hechenglujing);
322                                                                 if w_try>w_best
323                                                                     w_best = w_try;
324                                                                     sol_best = {x_sol_try;b_sol_try};
325                                                                 end
326                                                             end
327                                                         end
328                                                     end
329                                                 end
330                                             end

```

```

331 end
332 end
333 end
334 end
335 end
336 end
337 end
338 end
339 end
340 end
341 end
342 end
343
344
345 w_best_bianli = w_best;
346 sol_best_bianli = sol_best;
347 x_sol_bianli = sol_best_bianli{1};
348 b_sol_bianli = sol_best_bianli{2};
349 fprintf('根据题目条件, 最佳的利润为: %f\n', w_best_bianli);
350 for i = 1:3
351     x_temp = x_sol_bianli{i};
352     for j = 1:length(x_temp)
353         if x_temp(j) == 0
354             fprintf('流程%f 的第 %f 个零配件/半成品/成品不检测\n', i, j);
355         elseif x_temp(j) == 1
356             fprintf('流程%f 的第 %f 个零配件/半成品/成品全部检测\n', i, j);
357         else
358             fprintf(['流程%f 的第 %f 个零配件/半成品/成品应当有' ...
359                 '%f %% 检测\n'], i, j, 100*x_temp(j));
360         end
361     end
362 end
363 for i = 1:2
364     b_temp = b_sol_bianli{i};
365     for j = 1:length(b_temp)
366         if b_temp(j) == 0
367             fprintf('流程%f 的第 %f 个半成品/成品不拆解\n', i+1, j);
368         elseif b_temp(j) == 1
369             fprintf('流程%f 的第 %f 个半成品/成品全部拆解\n', i+1, j);
370         else
371             fprintf(['流程%f 的第 %f 个半成品/成品应当有' ...
372                 '%f %% 拆解\n'], i+1, j, 100*b_temp(j));
373         end
374     end
375 end

```

11.5 question1_plot.m

```
1 % 定义成功概率p和样本量n的数组
2 p = 0.5;
3 n_values = [200] % 不同的样本量
4
5 % 初始化图形
6 figure;
7 hold on;
8
9 % 遍历不同的样本量
10 for n = n_values
11     % 计算二项分布的参数
12     x = 0:n; % 成功的次数
13     y_binom = binopdf(x, n, p); % 二项分布的概率质量函数
14
15     % 计算对应正态分布的参数（均值和标准差）
16     mu = n * p;
17     sigma = sqrt(n * p * (1 - p));
18     x_normal = min(x):(max(x)-min(x))/100:max(x); % 为正态分布创建更平滑的x轴
19     y_normal = normpdf(x_normal, mu, sigma); % 正态分布的概率密度函数
20
21     % 绘制二项分布
22     bar(x, y_binom, 'DisplayName', sprintf('二项分布'));
23
24     % 绘制正态分布
25     plot(x_normal, y_normal, 'r-', 'LineWidth', 2, 'DisplayName', sprintf('正态分布'));
26 end
27
28 % 设置图例、标题和坐标轴标签
29 legend show;
30 % axis square
31 title('二项分布与正态分布的近似');
32 xlabel('样本数量n=200');
33 ylabel('概率');
34
35
36
37 hold off;
```

11.6 question2_profit.m

```
1 function w = question2_profit(x,lingpeijian,chengpin,buhege)
2 %QUESTION2_PROFIT 输入相关参数，返回利润
3 % 此处显示详细说明
```

```

4      c1 = lingpeijian(1,1); % 零配件1次品率
5      c2 = lingpeijian(2,1); % 零配件2次品率
6      p1 = x(1)*(1-c1)+(1-x(1))*(1-c1);
7      p2 = (1-x(1))*c1;
8      p3 = x(2)*(1-c2)+(1-x(2))*(1-c2);
9      p4 = (1-x(2))*c2;
10     % p1_temp = p1/(p1+p2);
11     % p2_temp = p2/(p1+p2);
12     % p3_temp = p3/(p3+p4);
13     % p4_temp = p4/(p3+p4);
14     % p1 = p1_temp;
15     % p2 = p2_temp;
16     % p3 = p3_temp;
17     % p4 = p4_temp;
18     % if p1+p2~=1
19     %     error('p1+p2=1');
20     % end
21
22     c3 = (p1*p3*chengpin(1)+p1*p4+p2*p3+p2*p4)/(p1*p3+p1*p4+p2*p3+p2*p4); % 成品修正次品率
23     % c3 = (p1_temp*p3_temp*chengpin(1)+p1_temp*p4_temp+p2_temp*p3_temp+p2_temp*p4_temp)...
24     %     /(p1_temp*p3_temp+p1_temp*p4_temp+p2_temp*p3_temp+p2_temp*p4_temp);
25     % if abs(c3-c3_temp)>0.0001
26     %     disp('c3~=c3_check');
27     %     disp(x);
28     % end
29
30     % if c3~=c3_check
31     %     error(c3~=c3_check);
32     % end
33
34     a1 = x(1)*(1-c1) + (1-x(1));
35     a2 = x(2)*(1-c2) + (1-x(2));
36
37     n1 = min(a1,a2); % 进行零件检测后装配成的成品
38     n2 = n1 - n1*x(3)*c3; % 进行成品检测后剩余产品（包括没有检测部分）
39     n3 = n1*x(3)*c3; % 成品检测不合格品
40     n4 = n3*x(4); %
41     n5 = n1*(1-x(3))*c3;
42     n6 = n5*x(4);
43
44     w = chengpin(4)*(n2-n5)... % 销售额
45         - sum(lingpeijian(:,2))... % 零件成本
46         - x(1)*lingpeijian(1,3) - x(2)*lingpeijian(2,3)... % 检测零件成本
47         - n1*chengpin(2)... % 成品组装成本
48         - x(3)*n1*chengpin(3)... % 成品检测成本
49         - buhege(2)*(n4+n6)... % 成品不合格拆卸
50         - buhege(1)*n5... % 调换损失

```



```

51     + sum(lingpeijian(:,2))*(n4+n6) ... % 调换产生收益
52     ;
53 end

```

11.7 question2_SA.m

```

1 function [w_best,plan_best] = question2_SA(lingpeijian,chengpin,buhege)
2 %QUESTION2_SA 此函数用于完成模拟退火过程
3 % 此处显示详细说明
4 w = -inf;
5
6 for j=1:100 %求较好的初始解
7     plan0= rand(1,4);
8     w_temp = question2_profit(plan0,lingpeijian,chengpin,buhege) ;
9     if w_temp>w
10         plan = plan0; w = w_temp;
11     end
12 end
13 plan_best = plan;
14 w_best = w;
15
16
17 % fprintf('模拟退火当前最优解是: %f \n',w);
18 % disp('此时路径为: ')
19 % disp(plan);
20 % 开始模拟退火过程
21 e=0.1^100;L=100000;at=0.9999;T=1;
22 for k=1:L %退火过程
23     c=randi(4); %产生新解
24     if rand>=0.5
25         plan_try = plan;
26         plan_try(c) = rand*T+plan_try(c);
27         if plan_try(c)>1
28             plan_try(c)=1;
29         end
30     else
31         plan_try = plan;
32         plan_try(c) = rand*T-plan_try(c);
33         if plan_try(c)<0
34             plan_try(c)=0;
35         end
36     end
37     if plan_try(c)>1 || plan_try(c)<0
38         disp(plan_try(c));
39         error('不在四维单位立方体中');

```

```

40     end
41
42     w_temp = question2_profit(plan_try,lingpeijian,chengpin,buhege) ;
43     df = w-w_temp;
44     if df<0 %接受准则
45         plan=plan_try; w=w_temp; % 如果更好，一定接受
46     elseif exp(-df/T)>=rand
47         plan=plan_try; w=w_temp; % 如果更差，可能接受
48     end
49     if w>w_best
50         w_best = w;
51         plan_best = plan;
52     end
53     T=T*at;
54     if T<e
55         break;
56     end
57 end
58 end

```

11.8 question3_profit.m

```

1 function w = question3_profit(m,n,x,b,canshu,path) % path 表示合成路径
2 %QUESTION3_PROFIT 输入相关参数，返回利润
3 % m表示工序，n表示零配件数量
4 % x表示每一步的检测
5 % b表示每一步的拆解率
6 m_check = size(canshu,1);
7 if m_check ~= m+2
8     error('工序和参数数量不对应，请检查每一步的零配件半成品成品参数是否正确输入');
9 end
10 if sum(path{1}) ~= n
11     disp(canshu{1});
12     disp(sum(canshu{1}));
13     error('零件数量和合成步骤不对应，请重新检查');
14 end
15 % 对cell进行拆解
16 x1 = x{1}; % 零配件检测
17 x2 = x{2}; % 半成品检测
18 x3 = x{3}; % 成品检测
19 b2 = b{1}; % 半成品拆解
20 b3 = b{2}; % 成品拆解
21 lingpeijian = canshu{1};
22 banchengping = canshu{2};
23 chengpin = canshu{3};

```

```

24     sell_change = canshu{4};
25
26     p11 = lingpeijian(1,1);
27     p12 = lingpeijian(2,1);
28     p13 = lingpeijian(3,1);
29     p14 = lingpeijian(4,1);
30     p15 = lingpeijian(5,1);
31     p16 = lingpeijian(6,1);
32     p17 = lingpeijian(7,1);
33     p18 = lingpeijian(8,1);
34
35     % 开始计算中间变量
36     p11_3 = (1-x1(1))*p11;
37     p12_3 = (1-x1(2))*p12;
38     p13_3 = (1-x1(3))*p13;
39     p14_3 = (1-x1(4))*p14;
40     p15_3 = (1-x1(5))*p15;
41     p16_3 = (1-x1(6))*p16;
42     p17_3 = (1-x1(7))*p17;
43     p18_3 = (1-x1(8))*p18;
44
45     a1 = 1-x1(1)*p11;
46     a2 = 1-x1(2)*p12;
47     a3 = 1-x1(3)*p13;
48     a4 = 1-x1(4)*p14;
49     a5 = 1-x1(5)*p15;
50     a6 = 1-x1(6)*p16;
51     a7 = 1-x1(7)*p17;
52     a8 = 1-x1(8)*p18;
53
54     p21_2 = 1-(1-p11_3)*(1-p12_3)*(1-p13_3)*(1-banchengping(1,1));
55     p22_2 = 1-(1-p14_3)*(1-p15_3)*(1-p16_3)*(1-banchengping(2,1));
56     p23_2 = 1-(1-p17_3)*(1-p18_3)*(1-banchengping(3,1));
57
58     p21_3 = (1- x2(1))*p21_2/(1-x2(1)*p21_2);
59     p22_3 = (1- x2(2))*p22_2/(1-x2(2)*p22_2);
60     p23_3 = (1- x2(3))*p23_2/(1-x2(3)*p23_2);
61
62     n1 = min([a1,a2,a3]);
63     n2 = min([a4,a5,a6]);
64     n3 = min([a7,a8]);
65
66     a9 = n1-n1*x2(1)*p21_2;
67     a10 = n2-n2*x2(2)*p22_2;
68     a11 = n3-n3*x2(3)*p23_2;
69
70     n4 = min([a9,a10,a11]);

```

```

71 n5 = n1*x2(1)*p21_2*b2(1);
72 n6 = n1*x2(2)*p22_2*b2(2);
73 n7 = n1*x2(3)*p23_2*b2(3);
74
75 p31_2 = 1-(1-p21_3)*(1-p22_3)*(1-p23_3)*(1-chengpin(1));
76 p31_3 = ((1-x3)*p31_2)/(1-x3*p31_2);
77
78 a12 = n4-n4*p31_2*x3;
79
80 n8 = n4*p31_2*x3*b3;
81 n9 = a12*(1-p31_3);
82 n10 = a12*p31_3;
83 n11 = a12*p31_3*b3;
84
85 % k = whos
86
87
88 w = sell_change(1)*n9 ... % 销售额
89     - sum(lingpeijian(:,2))... % 零件成本
90     - x1*(lingpeijian(:,3))... % 检测零件成本，矩阵乘法
91     - [n1,n2,n3] * banchengping(:,2) ... % 半成品组装成本，矩阵乘法
92     - ([n1,n2,n3].*x2) * banchengping(:,3) ... % 半成品检测成本
93     - [n5,n6,n7] * banchengping(:,4) ... % 不合格半成品拆卸
94     + [n5,n6,n7] * [sum(lingpeijian(1:3,2)); sum(lingpeijian(4:6,2));
95         sum(lingpeijian(7:8,2)))]... % 不合格半成品拆卸收益
96     - n4 * chengpin(2)... % 成品组装成本
97     - n4 * x3 * chengpin(3) ... % 成品检测成本
98     - (n8+n11) * chengpin(4) ... % 成品不合格拆卸
99     - n10 * sell_change(2)... % 调换损失
100     + sum(lingpeijian(:,2))*(n8+n11) ... % 不合格成品拆卸产生收益
101     ;
102 end

```

11.9 question3_SA.m

```

1 % 2024-9-8-2:00
2 function [w_best,sol_best] = question3_SA(canshu,path)
3 %QUESTION3_SA 此函数用于完成模拟退火过程
4 % 此处显示详细说明
5 w = -inf;
6
7 for j=1:100 %求较好的初始解
8     x01 = rand(1,8);
9     x02 = rand(1,3);

```

```

10     x03 = rand;
11     b01 = rand(1,3);
12     b02 = rand;
13     x0 = {x01;x02;x03};
14     b0 = {b01;b02};
15     sol0= {x0;b0};
16     w_temp = question3_profit(2,8,x0,b0,canshu,path);
17     if w_temp>w
18         plan = sol0; w = w_temp;
19     end
20 end
21 sol_best = plan;
22 w_best = w;
23
24
25 % fprintf('模拟退火当前最优解是: %f \n',w);
26 % disp('此时路径为: ')
27 % disp(plan);
28 % 开始模拟退火过程
29 e=0.1^100;L=100000;at=0.9999;T=1;
30 for k=1:L %退火过程
31     c = [randi(8) randi(3) rand randi(3) rand]; %产生新解
32     x_try = plan{1};
33     b_try = plan{2};
34
35     x1 = x_try{1};
36     x2 = x_try{2};
37     x3 = x_try{3};
38
39     x1(c(1)) = x1(c(1)) + (2*rand-1)*T;
40     if x1(c(1))>1
41         x1(c(1)) = 1;
42     elseif x1(c(1))<0
43         x1(c(1)) = 0;
44     end
45
46     x2(c(2)) = x2(c(2)) + (2*rand-1)*T;
47     if x2(c(2))>1
48         x2(c(2)) = 1;
49     elseif x2(c(2))<0
50         x2(c(2)) = 0;
51     end
52
53     x3 = x3 + (2*c(3)-1)*T;
54     if x3>1
55         x3 = 1;
56     elseif x3<0

```

```

57     x3 = 0;
58 end
59 x_try = {x1;x2;x3};
60
61 b1 = b_try{1};
62 b2 = b_try{2};
63
64 b1(c(4)) = b1(c(4)) + (2*rand-1)*T;
65 if b1(c(4))>1
66     b1(c(4)) = 1;
67 elseif b1(c(4))<0
68     b1(c(4)) = 0;
69 end
70
71 b2 = b2 + (2*c(5)-1)*T;
72 if b2>1
73     b2 = 1;
74 elseif b2<0
75     b2 = 0;
76 end
77 b_try = {b1;b2};
78 w_temp = question3_profit(2,8,x_try,b_try,canshu,path);
79 df = w-w_temp;
80 if df<0 %接受准则
81     plan={x_try;b_try}; w=w_temp; % 如果更好，一定接受
82 elseif exp(-df/T)>=rand
83     plan={x_try;b_try}; w=w_temp; % 如果更差，可能接受
84 end
85 if w>w_best
86     w_best = w;
87     sol_best = plan;
88 end
89 T=T*at;
90 if T<e
91     break;
92 end
93 end
94 end

```

11.10 question4_range.m

```

1 function range = question4_range()
2 %QUESTION4_RANGE 返回问题四置信区间
3 % 此处显示详细说明
4 n0 = 10000;

```

```

5  p0 = [0.05,0.1,0.2];
6  E = 0.01;
7  Z = norminv(0.99,0,1);
8
9  %计算抽样检测的次品率
10 x = binornd(n0,p0);
11 p_hat = x./n0;
12 n = Z^2.*p_hat.*(1-p_hat)/E^2;
13
14 %置信区间上下边界值
15 lower_bound = p_hat - sqrt(p_hat.*(1-p_hat)/n)*Z;
16 upper_bound = p_hat + sqrt(p_hat.*(1-p_hat)/n)*Z;
17
18 %从上到下为5%, 10%, 20%的置信区间
19 range = [lower_bound',upper_bound'];
20 end

```