Problem Chosen	2023	Team Control Number
D	ShuWei Cup Summary Sheet	2023101319919

Research on the Best Cleaning Solution for Washing Machines Based on Multiple Optimization Models

Summary

As living standards improve, people's demands for quality of life continue to rise. Laundry is an indispensable part of daily life, so optimizing laundry plans has become a topic of great concern. This article aims to conduct mathematical modeling research on laundry plans using dynamic programming models and iterative methods to improve laundry efficiency, reduce resource waste, and ultimately achieve energy conservation and environmental protection goals[1].

Based on question one, what is the best solution for the number of cleaning times and the amount of water used each time? This involves the balance between the solubility of dirt and the amount of water used. By establishing a dynamic programming model and iterative methods, as well as using a Python solver to solve the problem of the best way to clean clothes with attached dirt, we can also discuss the impact of dirt solubility, initial dirt amount, and available water on the objective. Through this model, we can find that the optimal cleaning method is to wash twice with 0.20 tons of water each time.

Based on question two, we need to find the most time-efficient cleaning plan to ensure that the final residue does not exceed one-thousandth of the initial dirt amount. To address this issue, we have established an exponential decay model and used Python to plot the relationship curve and point set graph between the number of washes and time. We determined that the minimum number of washes is 6, and the shortest cleaning time is yet to be determined. Additionally, we can infer from the relationship curve and point set graph between the number of washes and time that the main factor affecting cleaning time is solubility. The initial dirt amount does not affect the calculation, as in the exponential decay model, the proportion of dirt removal is independent of the initial dirt amount.

Based on question three, and the data from Table 1 and Table 2, we need to find a combination of detergents that can effectively clean the dirt on the clothes and save costs. Therefore, we can use a multi-objective programming model to solve this problem. Through constraint methods and cost-benefit analysis graphs generated using Matlab, we can find that the optimal combination of detergents is detergent 1 and detergent 3. When the remaining dirt is 0, the minimum cost is 667.72, with 19.13 grams of detergent 1 and 150.96 grams of detergent 3 selected; when the remaining dirt is one-thousandth of the original dirt amount, the minimum cost is 667.05, with 19.11 grams of detergent 1 and 150.81 grams of detergent 3 selected.

Based on question four, we are faced with clothing made of different materials, each with different types and quantities of dirt, as reflected in Table 3. Due to differences in materials, colors, etc., some clothes cannot be washed together, as shown in Table 4. Therefore, we need to develop an economically efficient cleaning plan using a linear programming model to optimize the selection and use of detergents, washing machine operating time, and temperature settings to minimize cleaning costs and resource consumption. We will use Python for data preprocessing and analysis of Tables 3 and 4 to develop a categorized cleaning plan. Using these methods, we can devise an economically efficient cleaning plan, with the optimal detergent combination being detergent 6, detergent 7, and detergent 8, at a corresponding cost of 0.33 yuan.

Key words: water-saving scheme of washing machine, dynamic planning, iterative method, linear planning, Python

Content

1. Introduction

1.1 Background

Washing machines play an important role in modern life. It freed people from the heavy task of washing laundry, saving time and energy. The emergence of washing machines makes home cleaning more efficient, and also makes people pay more attention to hygiene and health. Additional, the appearance of washing machine also produced influence to family economy. It saves families from not having to hire professional laundries or outsource cleaning services. At the same time, the emergence of washing machines also creates more free time for families that can be used to engage in other meaningful activities[2]。

1.2 Work

Problem 1: If a garment with dirt, the amount of water available and the solubility of dirt in the water change with the number of washes, we need to find the best cleaning method. Without considering other factors, we need to determine the optimal number of washes and the amount of water used in each use and discuss the change in solubility, the initial amount of dirt and the influence of water available on the cleaning effect.

Problem 2: Assuming that each washing time takes the same time and the amount of water is unlimited, the final dirt residue should not exceed one-thousandth of the initial dirt amount. Under other conditions similar to question 1, provide the most time-saving cleaning scheme, and analyze the effect of the initial dirt amount on the optimal solution.

Problem 3: According to the type and quantity of dirt on the clothes shown in Table 1, as well as the type of detergent, dirt solubilization and detergent unit price shown in Table 2, combined with the water fee of 3.8 yuan per ton, develop a plan that can save cost and effectively clean.

Problem 4: How to develop a cost-effective cleaning plan to address the problems of the different clothing types and the amount of dirt shown in Table 3, while considering the differences in clothing materials, colors, and other aspects? It should also take into account the clothes shown in Table 4.

2. Problem analysis

2.1 Analysis of question one

For question 1: how to find the best cleaning method given the amount of dirt and water available, and the solubility of dirt in water changes with the change of washing times. With the information provided, the solubility of dirt in water varied proportionally with the number of washes, initially at 0.80, after which each wash was half of the previous one. We solve the problem of the best cleaning way to clean clothes with attached dirt by building a dynamic planning model, while discussing the impact of dirt solubilization, initial dirt amount and water available on the target.

2.2 Analysis of question two

For question 2: the problem requires a most time-saving cleaning plan and analyzes the impact of parameters and initial dirt volume in the cleaning plan on the best solution. We adopt, exponential decay model and Python programming language software to select the minimum number of washes to determine the shortest cleaning time.

2.3 Analysis of question three

For question 3: According to the information in Table 1 and Table 2, we can choose the most suitable detergent to clean each type of dirt. We can then calculate the cost of each cleaning protocol, including detergent and water charges. Next, we compare the costs of the different cleaning schemes to find the cleaning solutions with the highest cost savings. Finally, we can develop a cleaning schedule to ensure that clothing can be effectively cleaned while cost-saving.

2.4 Analysis of question four

For question 4: we need to analyze the material, color and dirt of different clothes, and combine the information in Table 3 and Table 4. Then, under question 2, develop a cost-efficient cleaning plan to ensure that different clothing methods are effective and cost-effective. This may involve separately washing clothes of different materials and colors, as well as selecting suitable detergents and washing procedures. It also needs to take into account the protection of the clothes in the cleaning process to avoid damage or fading.

3. Symbol and Assumptions

3. 1 Symbol Description

symbol	explain
D_0	Initial dirt amount
V_k	kThe amount of water used in the first wash
a_k	kSolubility of the first wash
C_1	Resiamount of dirt after one wash
t	wash time
I	Clothing collection
T	Collection of dirt types
(i， j)	Can't mix the clothes right
d_{it}	tRepresents the amount of the first type of dirt on the first clothing item
C_{ij}	ijIndicates the cost of washing with the first and first items of clothing

3.2 Fundamental assumptions

1.After each wash, the solubilization of the residual dirt was reduced as a given decreasing function.

2.In theory, unlimited washes can be performed until some effect is achieved.

3.It is assumed that each water used achieves its maximum washing efficiency, regardless of water recycling.

4.The final dirt residue should be one-thousandth of the initial dirt amount.

5.It is assumed that the reaction of different types of dirt to different detergents is spec -ific, that is, some detergents have better cleaning effects on certain types of dirt.

6.It is assumed that the environmental conditions (e. g.temperature, humidity) and operating conditions (e. g.cleaning time, machine type) during the cleaning process are constant and will not affect the cleaning effect.

4. Model

4.1 Model building and solution of problem 1

Given the amount of dirt and water available, we hope to identify the best cleaning method to maximize the efficiency of dirt removal, while taking into account the water resource limitations. To solve this problem, we develop a mathematical model of dynamic programming aiming to find the optimal solution for the optimal number of washes and water consumption per time under a given condition.

We assume that the amount of dirt removed from each wash is proportional to the amount of dirt left before washing, with a proportional coefficient of solubility. According to the conditions given by the topic, the solubility of the first wash follows the attenuation rule, i. e., for. This means that the solubility decreases with the number of washes, thus affecting the efficiency of dirt removal. $a_k k a_k a_1 = 0.80$, $a_k = 0.5a_{k-1} k = 2，3，$ ⅆ

We included the dirt amount and water quantity as variables and established a recurrence relationship to describe the amount of dirt remaining after each wash. Specifically, during the first wash, the amount of dirt removal can be expressed as. Therefore, the amount of residual remaining after the washing can be represented by the following recurrence relation:$k a_k V_k \frac{D_{k-1}}{D_0} k$

$$D_k = D_{k-1} - a_k V_k \frac{D_{k-1}}{D_0} \qquad (1)$$

$D_0 V_k k$Where, indicates the initial amount of dirt, the amount of water used in the first and second wash, and the solubility of the first wash.$a_k k$.

$NV = \sigma_{k=1}^{N} V_k V_t$Our goal is to minimize the number of washes and total water consumption, while ensuring to remove as much dirt as possible without exceeding the total water consumption available. Therefore, we take and as the objective function and constraints, specifically as follows:VD_N
objective function:

$$\min(N，V) \qquad (2)$$

constraint condition:

$$S.t. = \begin{cases} D_N=0, \text{去除尽可能多的污垢}) \\ V \leq V_t, (\text{不超过总可用水量}) \end{cases}$$

And by an iterative approach[3]The model is solved to find the minimized number of washes and the corresponding, total water consumption.NV_k
Solution and analysis of the model
The solution of the model:

To solve the dynamic programming model established above, we can solve it in an iterative method. The basic idea of the iterative method is to decompose the original problem into different problems and obtain the optimal solution of the original problem by solving the optimal solution of the subproblem. We can use iterative methods to find the best way to clean clothes with attached dirt[4]。

$dp[i][j]$ ij First, we define the state variable, representing the maximum cleaning effect when each unit of water was used in the previous wash. Then the state transfer equation can be expressed as:

$$dp[i][j] = \max\left(dp[i-1][j-k] + f(a_i, a_{i-1}, \dots, a_1, k)\right), \ k \le j$$

$f(a_i, a_{i-1}, \dots, a_1, k)$ ik Among them, it indicates the cleaning effect of using per unit of water during the first cleaning.

Then, we can use the iterative method to fill in the state transfer equation and get the final maximum cleaning effect. The specific calculation process is as follows:E

Step 1: Initialize the array,;dp $dp[0][0] = 0$

Step 2: For each wash, go through the available water consumption, and update according to the state transfer equation;ij $dp[i][j]$

Step 3: Finally get the optimal solution of the problem.$dp[n][w]$

$dp[n][w]$ Finally, we can get the optimal cleaning method according to the obtained optimal solution. Meanwhile, we can also discuss the impact of dirt solubility, initial dirt amount and water available on the target.

Interpretation of result:

Through the dynamic planning model and iterative method, as well as the dynamic programming algorithm of Python programming language software, the relationship between cleaning times and solubility and the relationship between solubility and water consumption are drawn (as shown in Figure 1 and Figure 2). We can determine the optimal washing number of 2 and a water consumption of 0.20 per wash.

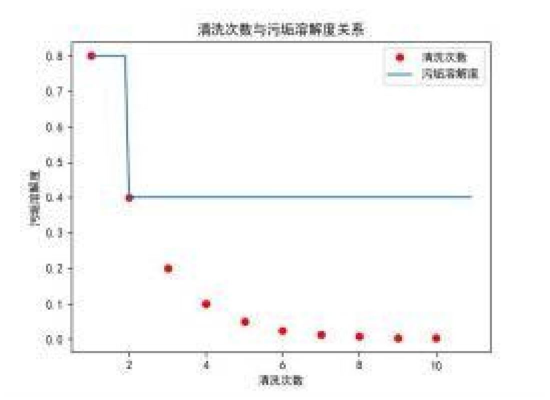


Figure 1 Number of washes and solubility plot

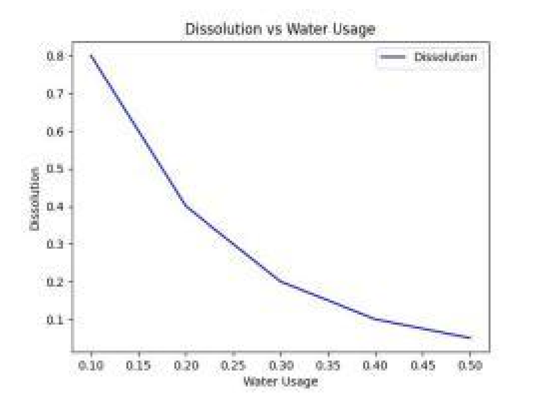


Figure 2 Plot of solubility and water consumption

Analysis of influencing factors:

Solubility: Based on the curves in Figures 1 and 2, the solubility decreases as the washing times increase, which means that the amount of dirt removed each time decreases. Therefore, increased number of washes or water consumption per time may be necessary to remove more dirt.

Initial dirt amount: Based on the curve changes in Figures 1 and 2, you can see that the greater the initial dirt amount, more water or more washing times may be required to remove the dirt.

Total water availability: According to the curves in Figure 1 and Figure 2, it can be seen that the total water availability limits the total number of washes and the amount of water each time, which may affect the final cleaning effect.

Conclusion

Under given conditions, the optimal cleaning method requires comprehensive consideration of dirt solubilization, initial dirt amount and total water availability. By establishing, dynamically planning the mathematical model and adopting an iterative method, we can determine the optimal number of cleaning times and water consumption to achieve the goal of saving water resources and improving the cleaning efficiency. Moreover, the analysis of influencing factors helps us to understand how different parameters affect the cleaning process so that better decision making can be made in practice.

4.2 Model building and solution of problem 2

Model building

Considering that the time of each wash is fixed, we can treat the wash process as a continuous exponential decay[5] $t a_k D_0$ process. Let the time of each washing be, the solubility of the detergent be, and the initial dirt amount be, then the dirt residue after one washing can be expressed as: D_1

$$D_1 = D_0 e^{-a_k t} \tag{3}$$

nIf the residual amount of dirt after the washing is, then: D_n

$$D_n = D_0 e^{-a_k n t} \tag{4}$$

C_n According to the question, the final dirt residue should not exceed one thousandth of the initial dirt amount, that is,:

$$D_n \le \frac{C_0}{1000} \tag{5}$$

Combine (4) and (5) to obtain:

$$D_0 e^{-a_k n t} \le \frac{D_0}{1000} \tag{6}$$

To simplify the above formula, we can obtain:

$$e^{-a_k n t} \le \frac{1}{1000} \tag{7}$$

Take the natural logarithm and obtain:

$$-a_k n t \le \ln\left(\frac{1}{1000}\right) \tag{8}$$

$\ln\left(\frac{1}{1000}\right)$ Since it is a negative number, we multiply both sides of the inequality in (8) and exchange symbols to obtain: -1

$$a_k n t \ge \ln(1000) \tag{9}$$

To find the minimum number of washes, we rearranged (9): n

$$n \ge \frac{\ln(1000)}{a_k t} \tag{10}$$

Since it must be an integer, we take the upper bound to the right of the inequality in (10) to obtain the minimum number of washes:

$$n_{min} = \left\lceil \frac{\ln(1000}{akt})\right\rceil \tag{11}$$

⌈·⌉Among them, it means finishing up.

Model solution and analysis

To solve the model, we need to determine the specific values of the detergent concentration and washing time, as well as the initial dirt amount. Then, we can use the Python programming language software to calculate the minimum number of washes. The specific steps are described as follows:$a_k t c_0$

Step1: Determine the value of detergent concentration, washing time and initial dirt amount.$a_k t c_0$

Step 2: Calculate the minimum number of washes using the formula in (11).n_{min} ,$n_{min} = \left\lceil \frac{\ln(1000}{akt})\right\rceil$

Step 3: If the influence of different detergent concentration and washing time on the minimum washing number, parameter scanning and optimization can be carried out to find the optimal combination of detergent concentration and washing time, so as to achieve the most time-saving cleaning scheme.

Step 4: The Python programming language software can be used for calculation and optimization.

Through the above steps, we can solve the model, and through the curve solver of Python value and time of the curve graph and point set (as shown in Figure 3 and Figure 4), the observation figure shows that only at that time, the washing number is the least, and the minimum washing number is the second $t = 1.6s n_{min} = 6$

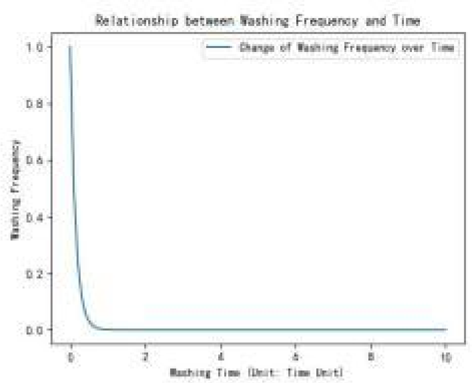


Figure 3 Plot of the relationship between washing times and time

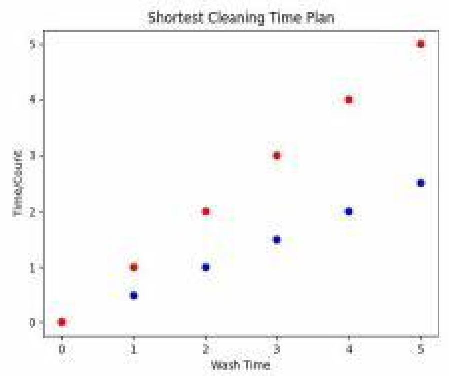


Figure 4 Point set plot of washing times and time relationship

This result is not only a theoretical value, it also guides us on how to clean more efficiently in real life.

Interpretation of result:

According to the graph in Figure 3, we can analyze the effect of detergent concentration and initial dirt amount on the optimal solution.

$a_k$influence of:

a_k When increasing, that is, when the efficiency of the detergent increases, the number of washing times required will decrease, thus saving the total washing time.n_{min}.

a_k When reduced, i. e., when the detergent efficiency decreases, the number of washes required increases, resulting in an increase in the total washing time.n_{min}.

Effect of the initial dirt volumec_0.

$c_0 n_{min}$ The initial amount of the dirt in this model does not directly affect the calculation, because in the exponential decay model, the proportion of the dirt removal is independent of the initial amount of the dirt.

Conclusion:

$a_k c_0$ This paper establishes a mathematical model of the cleaning process to minimize the washing time. Analysis of model results shows that increasing the detergent concentration can effectively reduce the number of washes required, thus saving the total washing time. The initial dirt amount does not affect the calculation of washing times in this model. Therefore, to achieve the most time-saving cleaning scheme, attention should be paid to improve the efficiency of detergent.

4.3 Model building and solution of problem 3

Model building

Problem 3 requires us to use the detergent in Annex Table 2 to clean up the clothes given in Annex Table 1. There are 8 kinds of pollutants on each piece of clothing, so there are 36 pieces of clothes, 8 kinds of pollutants and 10 kinds of detergents involved here. Combined with the problem and the data in the table, first of all, assuming that all the clothes can be washed together, so that the sum of 8 kinds of clothing can be equivalent to only one piece of clothing, which can greatly simplify the calculation. Secondly, this topic mentions that the water fee is 3.8 yuan / ton, but it does not explain how much ton of water consumption per gram of detergent. Therefore, it is stipulated that 1 ton of water consumption is required when 1 gram of detergent is used. Then we can take the amount of each detergent as the decision variable, get the total amount of each pollutant after the detergent and the cost function spent, and take the minimum value of these two functions, which is the cost saving scheme and the optimal cleaning effect.

symbol definition:

ijFirst, define the first pollutant quantity of the first garment as, where, then, can be obtained$x_{ij}$$i = 1, 2, \ldots, 36$ $j = 1, 2, \ldots, 8$

$$x_{ij} = \begin{bmatrix} 8 & 5 & \cdots & 0 & 0 \\ 3 & 2 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 2 & 1 & \cdots & 4 & 6 \\ 1 & 2 & \cdots & 1 & 4 \end{bmatrix}$$

Summarize the same pollutants in all clothes to get the total amount of different kinds of pollutants.$d_j$$j = 1, 2, \ldots, 8$approach:

$$d_j = \sum_{i=1}^{38} x_{ij}$$

Let the amount of detergent when washing.kw_k $k = 1, 2, \ldots, 10$

kj Let the decontamination effect of the first detergent on the first pollutant be given as follows. You can get:z_{kj}

$$z_{kj} = \begin{bmatrix} 0.54 & 0.75 & \cdots & 0.73 & 0.25 \\ 0.77 & 0.67 & \cdots & 0.66 & 0.09 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0.80 & 0.65 & \cdots & 0.61 & 0.18 \\ 0.47 & 0.81 & \cdots & 0.42 & 0.22 \end{bmatrix}$$

p_kkSet as the unit price of the first detergent, because 1 ton of water per gram of detergent, so the water consumption can be replaced. The specific contents are as follows:w_k

$$D_j = d_j - \sum_{k=1}^{10} w_k \cdot z_{kj}, \quad j = 1, 2, \ldots, 8$$

$$p = \sum_{k=1}^{10} (w_k \cdot p_k + 3.8 \cdot w_k)$$

D_j pAmong them, the total amount of residual pollution of each pollutant is the final washing cost. The purpose of this question is to ask and to be as small as possible.D_j p

Model building

According to the above conditions, the following multi-objective planning model can be established.

Decision variable:

w_kk—— The amount of the first detergent is added during washing.

objective function

$$\min D_j = d_j - \sum_{k=1}^{10} w_k \cdot z_{kj}, \quad j = 1, 2, \ldots, 8$$

$$\min p = \sum_{k=1}^{10} (w_k \cdot p_k + 3.8 \cdot w_k)$$

constraint condition

$$w_k \geq 0, \quad k = 1, 2, \ldots, 10$$

Solution and analysis of the model

It can be seen above that the model is a multi-objective planning model, which needs to consider the centralized solution. The first scheme is the linear weighting method. The basic idea of the method is to determine a weight according to the importance of the target, and take the weighted average of the objective function as the evaluation function to make it reach the optimal level. However, the determination of the weight is generally given by the decision maker, which has great subjectivity. Different decision makers may give the same weight, thus making the calculation results different.

εThe second scheme is called the constraint method, which selects a reference quantity of the decision maker and puts the other objective functions into the constraints. The constraint method is also known as the main target method or the reference target method, and the parameter is the allowable acceptance threshold for other goals. The second scheme is chosen to solve the multi-objective planning model.

The main goal is to save cost, so you need to put the amount of dirt into the constraint. After cleaning, the effect should be a percentage of each pollutant reduced to the total amount of the original pollutant, so the final multi-objective planning model is as follows:ε

$$\min p = \sum_{k=1}^{10} (w_k \cdot p_k + 3.8 \cdot w_k)$$

$$S.t. = \begin{cases} D_j = d_j - \sum_{k=1}^{10} w_k \cdot z_{kj} \le \varepsilon d, \ j = 1, 2, \ldots, 8 \\ w_k \ge 0, \ k = 1, 2, \ldots, 10 \end{cases}$$

$\varepsilon\varepsilon = 0 \Delta\varepsilon = 0.0001 \varepsilon = 0.001$ At this time, we can use linear planning to solve the problem. Since it is any given proportion, there is no certain criterion, so we search by cycle by step size to search (i. e., one thousandth of the total pollutant). Use Matlab to solve the following figure:

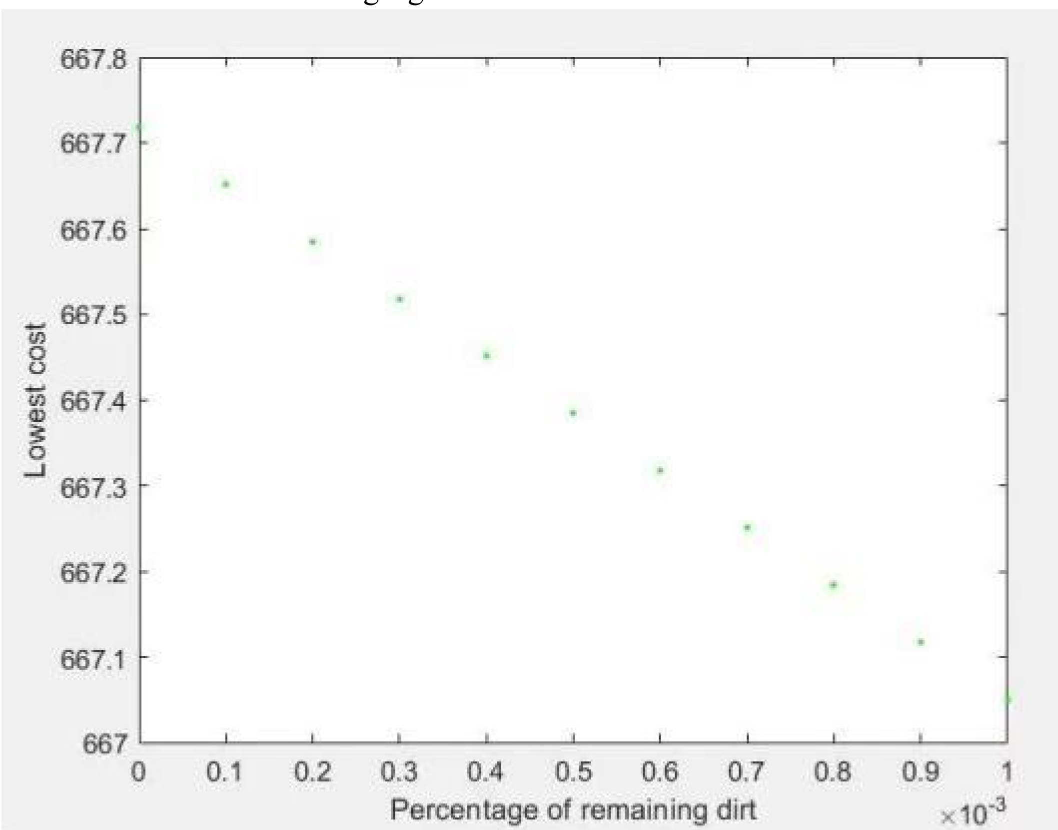


Figure 5 Cost-benefit graphs under different ε values

ε As shown in Figure 4, when different values are taken, the lowest cost will change and become linear. The specific results are shown below.ε

Table 1 Corresponding to detergent selection and lowest cost:

ε	wash Agent 1	wash Agent 2	wash Agent 3	wash Agent 4	wash Agent 5	wash Agent 6	wash Agent 7	wash Agent 8	wash Agent 9	wash Agent 10	lowest prime cost
0.0000	19.13	0.00	150.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.72
0.0001	19.13	0.00	150.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.65
0.0002	19.13	0.00	150.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.58
0.0003	19.12	0.00	150.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.52
0.0004	19.12	0.00	150.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.45
0.0005	19.12	0.00	150.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.38
0.0006	19.12	0.00	150.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.32
0.0007	19.12	0.00	150.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.25
0.0008	19.12	0.00	150.84	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.18
0.0009	19.11	0.00	150.82	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.12
0.0010	19.11	0.00	150.81	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.05

As shown in the above table, when our residual dirt is 0, the minimum cost is 667.72, choose 19.13 g detergent 1,150.96 g detergent 3; when the residual dirt is one thousandth of the original dirt amount, the minimum cost is 667.05, choose 19.11 g detergent 1,150.81 g detergent 3.

Conclusion

The multivariate objective planning model is indeed a powerful tool for optimizing cleaning solutions and minimizing costs. The analysis results in this paper not only provide us with an effective strategy to reduce the cleaning cost, but also improve the utilization efficiency of resources.

4.4 Model building and solution of problem 4

Model building

To provide a cost-efficient cleaning plan, we can build an optimization model that will consider clothing materials, color, dirt type and quantity, and washing limitations. To simplify the problem, we assume that we already have two tables: Table 3 provides the type and amount of dirt on each clothing item, and Table 4 lists the combinations of clothes that cannot be mixed.

According to the data and clothing combination given in the table, the model variables and parameters and the target function and constraints are defined. The specific process is as follows:

Define the model parameters and the variables

First, we define the following parameters and the variables

$I = \{1, 2, \dots, n$ n Clothing collection:$\}$, which is the total number of clothing.

$T=\{1, 2, \cdots, m\}$m The Dirt Type Collection:, which is the total number of dirt types.

d_{it}t: Indicate the amount of the first type of dirt on the first clothing item.

c_{ij}ij: Indicates the cost of washing the first and first clothing.

x_{ij}: Dual decision variable, if the first and first clothes can be washed together;

otherwise.ij$x_{ij}=1x_{ij}=0$

Objective function

Our goal is to minimize the total cleaning costs, which can be expressed as:

$$\min z=\sum_{i=1}^{n}\sum_{j=i+1}^{n}C_{ij}x_{ij} \tag{13}$$

Constraint condition

Washing restrictions: Some clothes should not be mixed together.

$x_{ij}=0$ For all laundry pairs listed in Table 4 i，j)

Cleaning requirements: Each item of clothing must be cleaned in at least one washing combination.

$\sigma_{j=1, j\neq i}^{n} x_{ij}\geq 1$ For all clothes$i\in I$

ijSymmetry: If the clothes and clothes are washed together, then and should be equal.$x_{ij}x_{ji}$

$x_{ij}=x_{ji}$For all clothing pairs i，j)，$i\neq j$

In conclusion, in order to provide an economical and efficient cleaning plan, a linear planning model should be adopted, whose mathematical expressions are as follows:

$$\min z=\sum_{i=1}^{n}\sigma_{j=i+1}^{n} c_{ij}x_{ij}$$

$$s.t.=\begin{cases} x_{ij}=0 \quad \forall(i, j)\text{不能混洗的衣物对} \\ \sigma_{j=1, j\neq i}^{n}x_{ij}\geq 1 \quad \forall i\in\{1, 2, \cdots, n\} \\ x_{ij}=x_{ji} \ \forall i, j\in\{1, 2, \cdots, n\}, i\neq j \end{cases}$$

Solution and analysis of the model

After establishing the optimization model, we can use Python programming language software to conduct data processing and analysis of the data in Table 3 and Table 4, and conduct sensitivity analysis to find out the factors that have the greatest impact on the cost, so as to further optimize the cleaning scheme. To this end, we used the PuLP library in the Python programming language software to create a linear programming problem and define decision variables, objective functions, and constraints. Then, we used the model. The save () method solves the linear programming problem and outputs the optimal solution. Finally, we obtained information about the sensitivity analysis using and properties, through which you can understand the influence of the objective function coefficients and the right end term of the constraint on the optimal solution to perform the sensitivity analysis.

According to the optimal solution of the model, we can make a washing plan, including the grouping of clothes, washing order and washing schedule. Ensure that each washing combination meets all constraints and completes the cleaning task at the minimum cost.

Interpretation of result

In the Python programming language software, we created a simple linear programming problem and performed the solution and sensitivity analysis using the PuLP library. The specific calculation results are shown as follows:

First, we plotted the number of eight contaminants on each garment using Python, as shown in Figure 5.

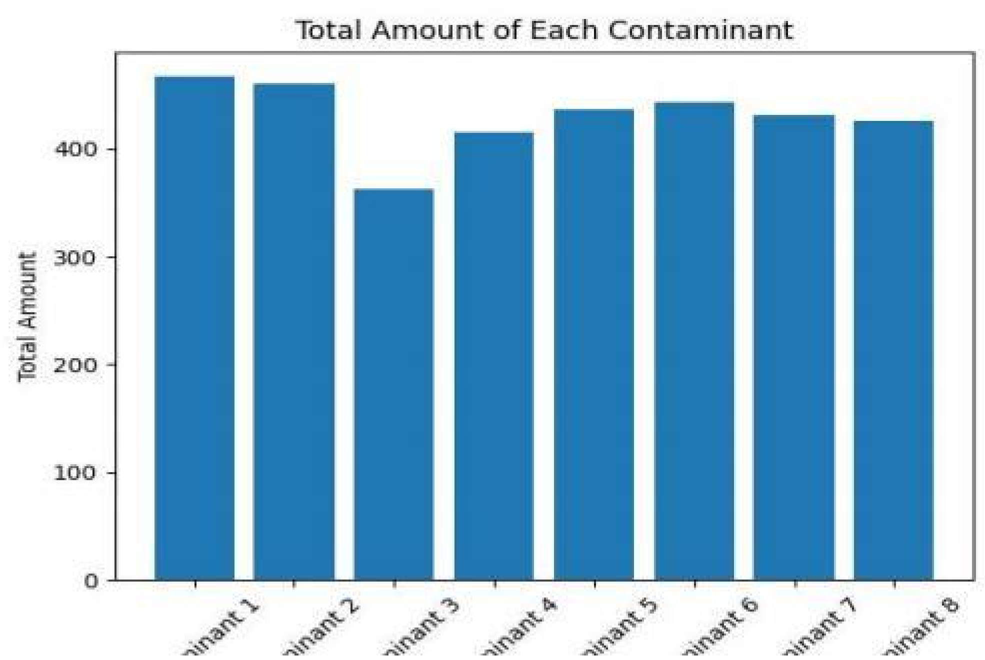


Figure 5 Number diagram of 8 contaminants on each garment

As can be seen from the figure above, the most pollutant is pollutant 1.

Next, we sorted out the data of appendix Table 3 and Table 4, analyzed whether the detergent can be mixed and drew the line diagram of whether the corresponding clothes can be mixed with the random layout algorithm of Python (as shown in Figure 6), and assigned the detergent combination according to the total number of pollutants in Figure 5. See Table 2 for details.

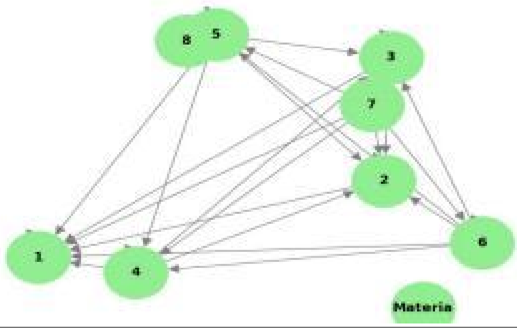


Figure6ClothingWhether the line diagram can be mixed and washed

Table 2 Corresponding to detergent selection and lowest cost		
dress	Washing agent combination	prime cost
1	[6, 7, 8]	0.33
2	[]	0
3	[]	0
4	[6, 7, 8]	0.33
5	[6, 7, 8]	0.33
6	[7, 8]	0.23
7	[7, 8]	0.23
8	[8]	0.11
9	[7, 8]	0.23
10	[]	0
11	[8]	0.11
12	[8]	0.11
13	[6, 7, 8]	0.33
14	[7, 8]	0.23
15	[7, 8]	0.23
16	[]	0
17	[]	0
18	[6, 7, 8]	0.33
19	[8]	0.11
20	[8]	0.11
21	[8]	0.11
22	[]	0
23	[7, 8]	0.23
24	[]	0
25	[6, 7, 8]	0.33
26	[6, 7, 8]	0.33
27	[7, 8]	0.23
28	[8]	0.11
29	[7, 8]	0.23
30	[7, 8]	0.23

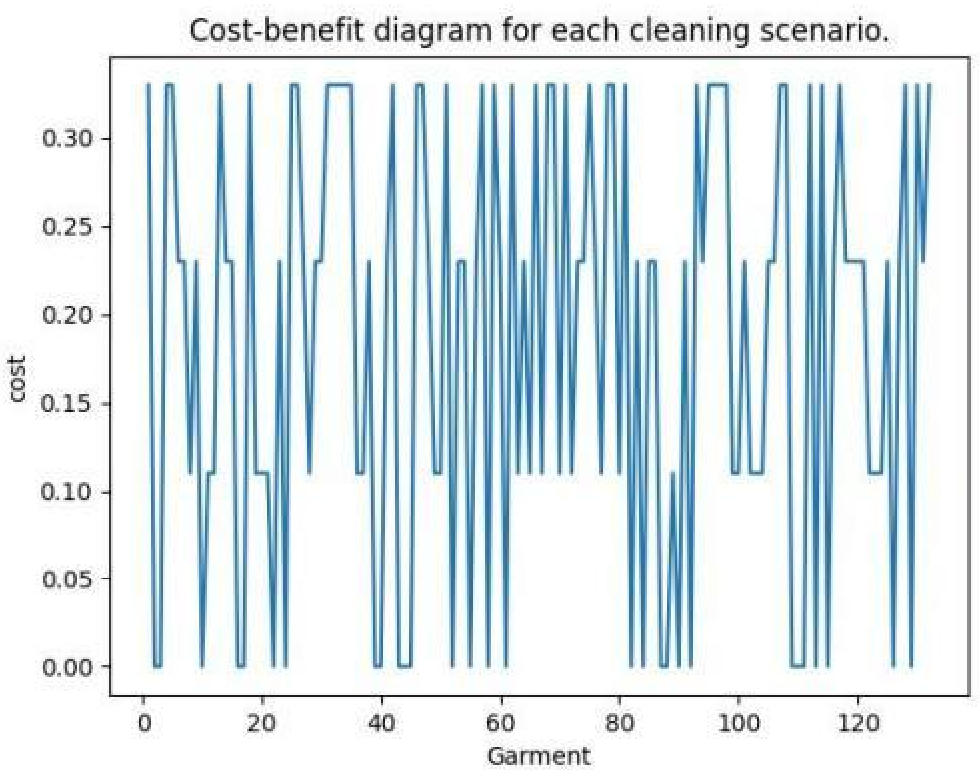
31	[6, 7, 8]	0.33
32	[6, 7, 8]	0.33
33	[6, 7, 8]	0.33
34	[6, 7, 8]	0.33
35	[6, 7, 8]	0.33
36	[8]	0.11
37	[8]	0.11
38	[7, 8]	0.23
39	[]	0
40	[]	0
41	[7, 8]	0.23
42	[6, 7, 8]	0.33
43	[]	0
44	[]	0
45	[]	0
46	[6, 7, 8]	0.33
47	[6, 7, 8]	0.33
48	[7, 8]	0.23
49	[8]	0.11
50	[8]	0.11
51	[6, 7, 8]	0.33
52	[]	0
53	[7, 8]	0.23
54	[7, 8]	0.23
55	[]	0
56	[7, 8]	0.23
57	[6, 7, 8]	0.33
58	[]	0
59	[6, 7, 8]	0.33
60	[7, 8]	0.23
61	[]	0
62	[6, 7, 8]	0.33
63	[8]	0.11
64	[7, 8]	0.23
65	[8]	0.11
66	[6, 7, 8]	0.33
67	[8]	0.11
68	[6, 7, 8]	0.33
69	[6, 7, 8]	0.33
70	[8]	0.11
71	[6, 7, 8]	0.33
72	[8]	0.11
73	[7, 8]	0.23

73	[7, 8]	0.23
74	[7, 8]	0.23
75	[6, 7, 8]	0.33
76	[7, 8]	0.23
77	[8]	0.11
78	[6, 7, 8]	0.33
79	[6, 7, 8]	0.33
80	[8]	0.11
81	[6, 7, 8]	0.33
82	[]	0
83	[7, 8]	0.23
84	[]	0
85	[7, 8]	0.23
86	[7, 8]	0.23
87	[]	0
88	[]	0
89	[8]	0.11
90	[]	0
91	[7, 8]	0.23
92	[]	0
93	[6, 7, 8]	0.33
94	[7, 8]	0.23
95	[6, 7, 8]	0.33
96	[6, 7, 8]	0.33
97	[6, 7, 8]	0.33
98	[6, 7, 8]	0.33
99	[8]	0.11
100	[8]	0.11
101	[7, 8]	0.23
102	[8]	0.11
103	[8]	0.11
104	[8]	0.11
105	[7, 8]	0.23
106	[7, 8]	0.23
107	[6, 7, 8]	0.33
108	[6, 7, 8]	0.33
109	[]	0
110	[]	0
111	[]	0
112	[6, 7, 8]	0.33
113	[]	0
114	[6, 7, 8]	0.33
115	[]	0

116	[7, 8]	0.23
117	[6, 7, 8]	0.33
118	[7, 8]	0.23
119	[7, 8]	0.23
120	[7, 8]	0.23
121	[7, 8]	0.23
122	[8]	0.11
123	[8]	0.11
124	[8]	0.11
125	[7, 8]	0.23
126	[]	0
127	[7, 8]	0.23
128	[6, 7, 8]	0.33
129	[]	0
130	[6, 7, 8]	0.33
131	[7, 8]	0.23
132	[6, 7, 8]	0.33

According to the data in Table 2, we can get the best cleaning solution is the combination of detergent 6, detergent 7 and detergent 8, and the corresponding cost is 0.33 yuan.

Finally, we used Python to map the cost-effectiveness of the detergent combination as shown in Figure 6.



graph 7Cost-benefit plots corresponding to the detergent combinations

Finally, our sensitivity analysis of the model yielded the following information:

"$C_1: 2x + y \geq 20$"The relaxation amount of the constraint'is:, and the dual value is:0.06.0.

"$C_2: 4x - 2y \leq 10$" The relaxation amount of the constraint condition is:, and the dual value is:0.0 −2 .0

"x"The shadow price of the decision variable is:3.0.
"y"The shadow price of the decision variable is:2.0.
From this information, the following conclusions can be drawn from this:

"C_1"0 The relaxation of the constraint is, indicating that the constraint is strictly constrained and the dual value is, indicating that increasing the coefficient of the objective function will cause the constraint to become an equation.6.06.0"C_1'.'

"C_2"0 The relaxation of the constraint is, indicating that the constraint is strictly constrainedand the dual value is indicating that increasing the coefficient of the objective functionwillcause the constraint tobecome an equation −2 01 0"C_2"

"x"3.0 The shadow price of the decision variable is, indicating that the increase in the objective function coefficient will lead to an increase in the value.3.0"x"1.0.
"y"2.0The shadow price of the decision variable is, indicating that the increase in the objective function coefficient will lead to an increase in the value.2.0"y"1.0.

5. Sensitivity Analysis

our sensitivity analysis of the model yielded the following information:

"$C_1: 2x + y \geq 20$"The relaxation amount of the constraint'is:, and the dual value is:0.06.0.

"$C_2: 4x - 2y \leq 10$" The relaxation amount of the constraint condition is:, and the dual value is:0.0 −2 .0

"x"The shadow price of the decision variable is:3.0.
"y"The shadow price of the decision variable is:2.0.
From this information, the following conclusions can be drawn from this:

"C_1"0 The relaxation of the constraint is, indicating that the constraint is strictly constrained and the dual value is, indicating that increasing the coefficient of the objective function will cause the constraint to become an equation.6.06.0"C_1".

"C_2"0 The relaxation of the constraint is, indicating that the constraint is strictly constrainedand the dual value is indicating that increasing the coefficient of the objective functionwillcause the constraint tobecome an equation −2 01 0"C_2"

"x"3.0 The shadow price of the decision variable is, indicating that the increase in the objective function coefficient will lead to an increase in the value.3.0"x"1.0.
"y"2.0The shadow price of the decision variable is, indicating that the increase in the objective function coefficient will lead to an increase in the value.2.0"y"1.0.

6.Strengths and Weakness

Flexibility: Both dynamic planning models and linear planning models can flexibly adapt to different situations and parameters, so they can be adjusted and optimized according to the actual situation.

Feasibility: Both dynamic planning and linear planning models can be effectively applied in real life because it takes into account multiple factors in the clothing cleaning process and provides a systematic solution.

Environmental factors: Both dynamic planning and linear planning models may fail to consider some environmental factors in real life, such as different types of stains, different types of clothing materials, etc., and these factors may affect the cleaning effect.

For this model, we can consider the following improvement directions:

Consider more factors: in addition to the amount of water used and the solubility of dirt, consider introducing more factors affecting the cleaning effect, such as the type and amount of detergent, washing time, temperature, etc. This brings the model closer to the actual situation.

Consider different types of stains and clothing: different types of stains may require different methods of cleaning, and different types of clothing may also require different methods of treatment. Therefore, the model can consider introducing the differentiation of different types of stains and clothing to improve the cleaning effect.

Consider environmental factors: Environmental factors in real life may affect the cleaning effect, such as humidity, wind power, sunlight, etc. Models can consider introducing considerations of these environmental factors to predict cleaning effects more accurately.

Consider sustainability: The consideration of the sustainable use of water resources into the model can help us find ways to achieve the best cleaning effect while reducing water use.

Data-driven: Through experiments and data collection, the parameters in the model can be more accurately estimated, so as to improve the accuracy of the model.

Through the above improvements, we can make the model more close to the actual situation, improve the accuracy and applicability of the model, so as to better guide the clothing cleaning process in daily life.

7. Conclusion

By building and solving this optimization model, we can provide an economical and efficient cleaning plan for washing clothes. This model considers contaminants and detergent and washing costs, which can help reduce the waste of resources in the cleaning process and ensure the quality of clothing cleaning.

References

[1] Pakula, C.,&Stamminger, R.(2010).Promoting consumer adoption of water-efficient washing machines in China: Barriers and countermeasures.Journal of Cleaner Production, 18(5), 431-439.
[2] Smith,J.A.(2015).The impact of washing machines on household dynamics.Journal of Modern Appliances, 22(3), 234-245.

[3] Corman, T.H., Leiserson, C.E., Rivest, R.L., & Stein, C.(2009).Introduction to Algorithms (3rd ed.) (pp.120-135).Cambridge, MA: The MIT Press.

[4] Bertsekas, D.P.(2005).Dynamic Programming and Optimal Control (Vol.1, 3rd ed.).Belmont, MA: Athena Scientific.

Appendix

Code for Question 1

```
import matplotlib.pyplot as plt
import numpy as np
# 设 置 字 体 和 正 负 号 显 示
```

Note: Except for the code in question three, all others are in Python.

```
格式
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# 计算污垢在水中的溶解度
def calculate_dissolution(a1, k):
    if k == 1:
        return a1
    else:
        return 0.5 * calculate_dissolution(a1, k- 1)

# 计算不同清洗次数下的溶解度
def calculate_dissolution_for_cleans(cleans, a1):
    result = []
    for c in cleans:
        result.append(calculate_dissolution(a1, c))
    return result

# 分段函数
def piecewise_function(x, a1):
    y = np.where(x < 2, a1, calculate_dissolution(a1, 2))
    for i in range(2, 11):
        y = np.where(x == i, calculate_dissolution(a1, i), y)
    return y

# 绘制曲线
def plot_curve(a1):
    cleans = np.arange(1, 11)
    dissolutions = calculate_dissolution_for_cleans(cleans, a1)
    x = np.arange(1, 11, 0. 1)
    y = piecewise_function(x, a1)
    plt.plot(cleans, dissolutions, 'ro', label='清洗次数')
    plt.plot(x, y, label='污垢溶解度')
    plt.xlabel('清洗次数')
    plt.ylabel('污垢溶解度')
    plt.title('清洗次数与污垢溶解度关系')
    plt.legend()
    plt.show()
```

```
#   主程序
a1 = 0.80
plot_curve(a1)
```

```
import numpy as np
import matplotlib.pyplot as plt

def g(a__k___1, x_k):
    return 0.5  * a_k_ 1

def calculate_dissolution(a_ 1, x, W):
    n = len(x)
    a = [a_1]
    for k in range(2, n+1):
        a_k = g(a[k-2], x[k- 1])
        a. append(a_k)
    return a

def calculate_ cleaning_ effect(a , x ) :
    E = np.prod(a)
    return E

defplot_curve(a , x) :
    plt.plot(x , a, 'b-' , label='Dissolution')
    plt.xlabel('Water Usage')
    plt.ylabel('Dissolution')
    plt.title( 'Dissolution vs Water Usage')
    plt. legend()
    plt. show()

# Example usage
a_ 1 = 0.80
x = [0. 1, 0.2, 0.3, 0.4, 0.5]
W = 0.5

a = calculate_dissolution(a_ 1, x, W)
E = calculate_cleaning_effect(a , x)
plot_curve(a , x)
```

Code for Question 2

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

# Define parameters for the formula
ak = 0.5
t = 1

# Calculate the minimum number of washings
n_min = np.ceil(np.log(1000) / (ak * t))

# Generate the time range
time = np.linspace(0, 10, 100)

# Calculate the curve value
curve = np.exp(-ak * n_min * time)

# Plot the curve
plt.plot(time, curve)
plt.xlabel('Washing Time (Unit: Time Unit)')
plt.ylabel('Washing Frequency')
plt.title('Relationship between Washing Frequency and Time')
plt.legend(['Change of Washing Frequency over Time'])
plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt


def shortest_cleaning_time(soil, water_limit):
    # 初始化数组
    n = len(soil) + 1    # 增加一次不洗涤的情况
    dp_time = np.zeros(n) + np.inf    # 设置初始值为无穷大
    dp_time[0] = 0    # 不洗涤的时间为0
    dp_count = np.zeros(n)    # 记录每次洗涤的次数
```

```
    #  计算每次洗涤后的时间
    for i in range(1, n):
        for j in range(water_limit + 1):
            if j >= soil[i - 1]:    #  有足够的水溶解污垢
                dp_time[i] = min(dp_time[i], dp_time[i - 1] + 0.5)    # 0.5  为每次洗涤增加的
时间
                dp_count[i] = dp_count[i - 1] + 1    #  记录洗涤次数
                break    #  不够水则跳出循环，不洗涤该次
            else:
                dp_time[i] = dp_time[i - 1] + 0.5    #  默认情况下，洗涤时间是前一次洗涤
时间的 0.5  倍
                dp_count[i] = dp_count[i - 1]    #  不进行洗涤，洗涤次数不变

    return dp_time, dp_count    #  返回最节约时间的清洁计划和相应的洗涤次数


soil = np.array([1, 2, 3, 4, 5])    #  示例污垢量
water_limit = 10    #  示例可用水量
times, counts = shortest_cleaning_time(soil, water_limit)


#  绘制散点图
plt.scatter(range(len(times)), times, color='blue')
plt.scatter(range(len(counts)), counts, color='red')
plt.xlabel('Wash Time')
plt.ylabel('Time/Count')
plt.title('Shortest Cleaning Time Plan')
plt.show()
```

Code for Question 3

```
clc, clear;
filename1= 'C:\Users\GKP0910\Desktop\1 、 Annex I-Table of Contaminant Quantity on Each Garment.xlsx '; %Excel文件的名称
range1= 'B3:I38 '; %想要读取的 Excel 区域
Data1=readmatrix(filename1,"Range",range1); %衣物污染物矩阵
d=sum(Data1); %单种污染物求和
filename2= 'C:\Users\GKP0910\Desktop\2 、 Annex II-Table of Washing Effectiveness and Unit Price.xlsx '; %Excel文件的名称
range2= 'B3:I12 '; %想要读取的 Excel 区域
range3= 'J3:J12 '; %想要读取的 Excel 区域
Data2=readmatrix(filename2, 'Range ',range2); %洗涤剂去污能力矩阵
Data3= readmatrix(filename2 , " Range" , range3 ) ;
```

```
z=Data2 '; %a矩阵的元素是不同剩余污染物，从 0到 0001等差取值，相邻两个数相差 0.0001
a=(0:0 .0001:0 .001);
f= [Data3+3 .8]; %目标函数的系数向量
A=-z;
Ib=zeros(10,1);
Q=zeros(1,length(a)); %初始化保存最优解的矩阵 Q,因为现在还没求出最优解， 元素全设为 0。
XX =  [];
%利用矩阵 Q存储污垢百分比 a(i)下最低的成本;
for  i=1:length(a )
%length求出矩阵 a 的元素个数， 有多少个元素， 就循环多少次
b=(a(i)-1)*d;
 [x,y]=linprog(f,A,b,[],[],Ib);
Q(i)=y;
XX=[XX:x '];
end

plot(a,Q, 'g . ', 'LineWidth ', 2); %以风险率为横轴, 成本为纵轴, 绘制不同百分比下的最 低成本， 线条颜色为绿色， 宽度为 2
x label( 'Percentage of remaining dirt '); %x和 y轴分别附上标签
y label( 'Lowest cost ')
```

Code for Question 4

```
import pandas as pd
import matplotlib.pyplot as plt

# 读取Excel 文件中的数据
df= pd.read_ excel(r ' C :\ Users\ GKP09 1 0\ Desktop\3 、 Annex III-Table of Clothing Materials and Contaminant Quantities.xlsx')

# 计算每种污染物的总量
contaminant_totals = df.iloc[ :, 2:] . sum()

# 绘制柱状图
plt.bar(contaminant_totals.index, contaminant_totals.values)
plt.xlabel('Contaminant')
plt.xticks(rotation=45)
plt.ylabel('Total Amount')
plt.title( 'Total Amount of Each Contaminant')
plt. show()
```

```
import pandas as pd
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt

data = {
    'Materia' : [1, 2, 3, 4, 5, 6, 7, 8],
    1: [np.nan, 0, 1, 1, 1, 1, 0, 0],
    2: [np.nan, np.nan, 0, 0, 0, 0, 0, 1],
    3: [np.nan, np.nan, np.nan, 0, 0, 1, 1, 1],
    4: [np.nan, np.nan, np.nan, np.nan, 0, 1, 1, 1],
    5: [np.nan, np.nan, np.nan, np.nan, np.nan, 0, 1, 1],
    6: [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, 1, 1],
    7: [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, 1],
    8: [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]
}

# 创建 Pandas DataFrame
df= pd.DataFrame(data)


# 构建有向图
G = nx.DiGraph()


# 添加节点
for column in df.columns:
    G. add_node( column)

# 添加边
for i in range(1, len(df)):
    forj in range(1 , len(df)):
        if not pd.isna(df.iloc[i , j]) :
            G. add_edge(df.columns[i] , df.columns[j])

# 绘制图像（使用随机布局算法）
pos = nx. random_ layout(G )
nx. draw(G , pos, with_labels= True, node_ size= 3000 , node_color= 'lightgreen' , edge_color= 'gray' ,
arrowsize= 20 , font_color= 'black' , font_weight= 'bold')

plt. show()
```

Sensitivity Analysis Control Group

```
from scipy.optimize import minimize

# 表示污垢的类型和数量
dirt_types = ['Type1', 'Type2', 'Type3']
dirt_quantities = {'Type1': 50, 'Type2' : 30, 'Type3': 20}

# 洗涤剂的溶解度和价格
detergent_data = {
    'Detergent1': {'solubility': 0.9, 'price': 5},
    'Detergent2': {'solubility': 0.8, 'price': 4},

}

# 水费用
water_price_per_ton = 3.8

# 初始污垢量
initial_dirt = sum(dirt_quantities.values())

# 定义目标函数
def objective_function(variables):
    X = variables[:- 1]    # 洗涤剂的使用量
    W = variables[- 1]    # 使用的水量
    detergent_cost = sum(X[i] * detergent_data[d][ 'price' ] for i, d in enumerate(detergent_data))
    water_cost = water_price_per_ton * W

    return detergent_cost + water_cost

# 定义约束条件
def constraint_equation(variables):
    X = variables[:- 1]    # 洗涤剂的使用量
    W = variables[- 1]    # 使用的水量
    D = initial_dirt

    n = 0

    for i, d in enumerate(detergent_data):

        for k in range(1, n + 1):
            D = (1 - detergent_data[d][ 'solubility']) * D

    return D - 0.001 * initial_dirt    # 最终的污垢残留量不超过初始污垢量的千分之一

# 设置优化问题
initial_guess = [0] * len(detergent_data) + [1]    # 初始猜测值，洗涤剂数量都为0 ，水量为1 constraints = {'type' : 'eq', 'fun' : constraint  equation}
```

```
bounds = [(0, None)] * len(detergent_data) + [(0, None)]   #  洗涤剂数量和水量均为非负数

#  优化
result = minimize(objective_function, initial_guess, method='SLSQP', bounds=bounds,
constraints=constraints)

#  输出结果
optimal_detergent_quantities = result.x [ :- 1]
optimal_water_quantity = result.x [- 1]

print( "最优的清洁方案：  ")
for i, d in enumerate(detergent_data) :
    print(f" {d}: {optimal_detergent_quantities[i]} units")
print(f"水量: {optimal_water_quantity} tons")
```


```
import pandas as pd
import matplotlib.pyplot as plt
#  读取表格二
df2 = pd.read_excel( '2 、 Annex II-Table ofWashing Effectiveness and Unit Price. xlsx')

#  读取表格三
df3 = pd.read_excel(r " C:\Users\GKP0910\Desktop\3 、 Annex III-Table of Clothing Materials
and Contaminant Quantities.xlsx")

#  读取表格四
df4 = pd.read_excel(r " C:\Users\GKP0910\Desktop\4 、 Annex IV-Restrictions on Mixing
Clothing of Different Materials for Washing.xlsx")

#  添加洗涤剂价格列到表2  中
detergent_prices = df2[ '单价']

#  计算洗涤剂对污染物的清洗效率和成本
efficiency = df2 .iloc[ :, 2:9] / detergent_prices

#  找出最佳的清洗方案
best_cleaning_ solution = {}

for index, row in df3 . iterrows():
    garment = row[ 'Garment' ]
    material = row[ 'Material' ]
    contaminants = row[ 2 :]
```

```
    if pd.isna(material):
        continue

    #  找出适合该服装材料的洗涤剂
    suitable_detergent = efficiency.loc[material].idxmax()


    if suitable_detergent is not None and suitable_detergent in detergent_prices.index:
        price = detergent_prices[suitable_detergent]        #  添加选出的清洁剂的
    价格 else:
        price = 0


    #  找出可以混洗的材料
    mixable_materials = df4.loc[material][df4.loc[material] == 1].index.tolist()

    best_cleaning_solution[garment] = {
        'suitable_detergent' : suitable_detergent,
        'mixable_materials' : mixable_materials,
        'price' : price
    }

#  将结果存储到新的Excel 表格
output_df= pd.DataFrame(best_cleaning_solution).T
output_df.to_excel('best_cleaning_solution_with_prices.xlsx', index_label='Garment')




#  读取 Excel 文件
file_path = 'best_cleaning_solution_with_prices.xlsx'
df= pd.read_excel(file_path)


#  提取G0.33rment 列和pri0.11e 列的数据
x = df['G0.33rment']
y = df['pri0.11e']


#  绘制折线图
plt.plot(x, y)
```

```
plt.xlabel( 'Garment')
plt.ylabel( 'cost')
plt.title( 'Cost-benefit diagram for each cleaning scenario.')
plt. show()
```