ORDERING AND BIOLOGICAL LEARNING OF VASCULAR ROBOTICS

Bo Wu
Beijing International Studies University
2021221257@stu.bisu.edu.cn

ABSTRACT

Vascular robots have two main features: (1) four manipulators installed around the vessel tank; (2) the need for learning. Newly purchased manipulators require a week of biological learning (training) before they can work. This training takes place in a specific environment, where already-trained manipulators ("skilled workers") "guide" a number of new manipulators ("novices") in a simulated blood vessel until these novices reach the skill level of the skilled workers. The vascular robot can operate in a patient's blood vessel for one week and must then be removed. Once removed, the manipulators must be detached and undergo one week of maintenance before they can be put back to work; if they are not scheduled for further work, they must remain under maintenance.

A newly purchased vessel tank must undergo one week of inspection and debugging before it can be put into operation. After use, the tank does not necessarily require maintenance and can be used continuously; however, if it is not used, it still needs maintenance.

We assume that newly purchased vessel tanks and manipulators arrive at the beginning of each week and are immediately scheduled for inspection/debugging and training. By using looping statements, the fixed pricing problem (questions 1–3 in the original text) is solved, and by changing the parameters a–h, the model can be repeatedly solved.

Keywords: **Vascular Robot, Cost, Damage Rate**

https://github.com/sxjs1st

1 PROBLEM ANALYSIS

A thorough reading of the problem statement allows us to organize the following relationships:

- A manipulator used in week i must undergo maintenance in week $i + 1$ and can only be used again in week $i + 2$.
- A manipulator purchased in week i undergoes training for one week, so it can be used in week $i + 1$.
- A vessel tank used in week i can still be used in week $i + 1$.
- A vessel tank purchased in week i requires one week of inspection and debugging, so it can be used in week $i + 1$.

Symbol	Description	Unit
a	Purchase cost of container boat	yuan/unit
b	Purchase cost of operator	yuan/unit
c	Maintenance cost of container boat	yuan/unit/week
d	Maintenance cost of operator	yuan/unit/week
e	Training cost for operator (including "skilled worker")	yuan/unit
f	Number of new operators a skilled operator can train	unit
g	Damage rate of vascular robot	%
h	Number of vascular robots used	unit
x_{i+1}	Number of container boats maintained in week i	unit
y_{i+1}	Number of container boats used in week i	unit
z_{i+1}	Number of container boats purchased in week i	unit
r_{i+1}	Number of operators maintained in week i	unit
s_{i+1}	Number of operators used in week i	unit
t_{i+1}	Number of operators purchased in week i	unit
CN$[x_{i+1}, y_{i+1}, z_{i+1}]$	Number of container boats in week i	unit
MN$[r_{i+1}, s_{i+1}, t_{i+1}]$	Number of operators in week i	unit

Table 1: Parameter Definitions

1.1 Model Assumptions

Because each skilled manipulator can guide 10 or 20 new manipulators (a number significantly larger than the number of newly purchased manipulators required in week i), the number of “skilled workers” participating in training in week i can be assumed to be t_{i+1}/f (a sufficiently large constant to train the new batch). The symbols of the model are shown in **Table 1**.

2 Model Construction and Solution

For the first three questions (with no discount policy), we need to purchase a certain number of vessel tanks and manipulators so that treatment demands are met while keeping operating costs at a minimum. The solution can be divided into two parts:

2.1 Using the Manipulators

First, check whether the number of manipulators under maintenance in week i can meet the usage needs in week $i+1$. If not, purchase additional manipulators in week i and start their training immediately in week i:

1. If the manipulators under maintenance in week i are fewer than the manipulators needed in week $i+1$, then:
   - Number of new manipulators purchased in week i = (manipulators needed in week $i+1$) $-$ (manipulators under maintenance in week i).
   - Number of manipulators to be maintained in week $i+1$ = (manipulators under maintenance in week i) + (manipulators used in week i) + (manipulators purchased in week i) $-$ (manipulators damaged in week i) $-$ (manipulators used in week $i+1$).
   - Number of manipulators needed in week $i+1$ = (number of vascular robots used in week $i+1$) $\times 4$.
2. Otherwise:
   - Number of new manipulators purchased in week $i = 0$.
   - Number of manipulators to be maintained in week $i+1$ = (manipulators under maintenance in week i) + (manipulators used in week i) + (manipulators purchased in week i) $-$ (manipulators damaged in week i) $-$ (manipulators used in week $i+1$).
   - Number of manipulators needed in week $i+1$ = (number of vascular robots used in week $i+1$) $\times 4$.

Afterward, continue checking whether the number of manipulators under maintenance in week $i + 1$ can meet the needs in week $i + 2$. If not, purchase the shortfall in week $i + 1$ and arrange for training during week $i + 1$.

Question	a	b	c	d	e	f	g
Question 1	200	100	10	5	10	10	0
Question 2	200	100	10	5	10	10	0.2
Question 3	200	100	10	5	10	20	0.1

Table 2: Parameter Settings for Questions 1–3

2.2 Using the Vessel Tanks

Next, determine whether the total number of vessel tanks is sufficient for the following week. If not, purchase additional tanks in week i and begin inspection in week i:

1. If (vessel tanks under maintenance in week i + vessel tanks used in week i − vessel tanks damaged in week i) is less than the number of tanks required in week $i + 1$, then:
   - Number of new vessel tanks purchased in week i = (tanks needed in week $i + 1$) − (vessel tanks under maintenance in week i) − (vessel tanks used in week i) + (tanks damaged in week i).
   - Number of vessel tanks to be maintained in week $i + 1$ = (vessel tanks under maintenance in week i) + (vessel tanks used in week i) + (vessel tanks purchased in week i) − (tanks damaged in week i) − (tanks used in week $i + 1$).
   - Number of vessel tanks required in week $i + 1$ = (number of vascular robots used in week $i + 1$).
2. Otherwise:
   - Number of new vessel tanks purchased in week $i = 0$.
   - Number of vessel tanks to be maintained in week $i + 1$ = (vessel tanks under maintenance in week i) + (vessel tanks used in week i) + (vessel tanks purchased in week i) − (tanks damaged in week i) − (tanks used in week $i + 1$).
   - Number of vessel tanks required in week $i + 1$ = (number of vascular robots used in week $i + 1$).

Then, continue checking whether the tanks under maintenance in week $i + 1$ can meet the needs in week $i + 2$. If not, purchase the shortfall in week $i + 1$ and arrange inspection and debugging in the same week.

By following the **Table 2** for weeks 1–104 and storing the results in an Excel file, we can answer the posed questions.**Code A**

Table 3: Results for Problem 1 from Week 1 to Week 8

Week	Pigs Bought	Insects Bought	Insects Raised	Pigs Raised	Training Participants	Total Training	Total Cost (Yuan)
Week 1	0	0	14	6	2	16	1610
Week 2	0	0	44	0	8	0	330
Week 3	0	0	48	0	0	0	330
Week 4	3	28	36	6	6	31	3950
Week 5	0	2	30	0	0	0	140
Week 6	0	0	44	0	4	0	440
Week 7	0	0	72	11	0	0	470
Week 8	0	0	64	9	0	0	410
Total (1–8)	3	42	366	55	47		7650

3 Results and Analysis

If we only consider weeks 1–8, assuming a 20% weekly damage rate for vascular robots, the required number of manipulators almost doubles compared to a zero-damage scenario, in order to both meet

Table 4: Results for Problem 2 from Week 1 to Week 8

Week	Pigs Bought	Insects Bought	Insects Raised	Pigs Raised	Training Participants	Total Training	Total Cost (Yuan)
Week 1	0	0	14	6	2	16	1610
Week 2	0	0	35	0	6	0	235
Week 3	0	0	35	0	0	0	235
Week 4	8	44	20	2	2	49	6610
Week 5	0	2	22	0	0	0	340
Week 6	0	0	51	0	3	0	335
Week 7	0	0	50	7	0	0	320
Week 8	3	14	38	14	16	16	2390
Total (1–8)	11	74	257	34	29		12085

Table 5: Results for Problem 2

Week	Containers Bought	Operators Bought	Operators Maintained	Containers Maintained	Operators Trained (Skilled + Novice)	Total Cost (Yuan)
Week 12	5	19	29	1	21	3265
Week 26	0	0	98	9	0	580
Week 52	21	126	74	0	139	18560
Week 78	16	40	176	0	44	8520
Week 101	12	79	317	0	87	12755
Week 102	17	35	349	0	39	9035
Week 103	25	91	317	0	101	16695
Week 104	0	0	307	0	0	1535
Total (Weeks 1–104)	879	3838	13458	131	4253	670730

Table 6: Results for Problem 3

Week	Containers Bought	Operators Bought	Operators Maintained	Containers Maintained	Operators Trained (Skilled + Novice)	Total Cost (Yuan)
Week 12	3	8	40	3	9	1720
Week 26	0	0	117	11	0	695
Week 52	18	117	83	0	123	16945
Week 78	10	18	198	1	19	4990
Week 101	1	40	356	0	42	6400
Week 102	7	0	392	0	0	3360
Week 103	16	44	364	0	47	9890
Week 104	0	0	346	0	0	1730
Total (Weeks 1–104)	498	2290	16115	281	2427	436255

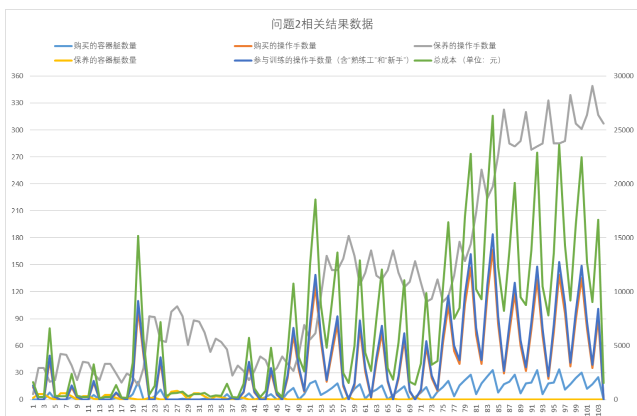

Figure 1: **Question 2 Related Results Data**

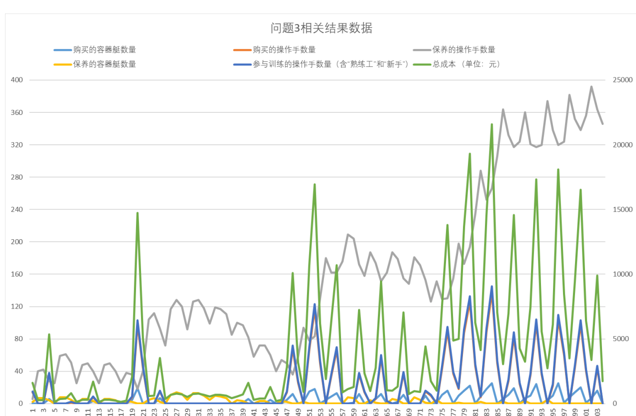

Figure 2: **Question 3 Related Results Data**

treatment needs and minimize operating costs. Over weeks 1–8, the total purchase of vessel tanks increases by 2.7 times, and the total purchase of manipulators increases by 76.2%. This indicates that robot damage primarily affects the number of vessel tanks required, as shown in the **Table 3** and **Table 4**.

With technological advances—where each skilled manipulator can guide 20 novices instead of 10, and the weekly damage rate of vascular robots decreases from 20% to 10%—the total number of vessel tanks and manipulators needed for weeks 1–104 decreases by about 70%. This still meets treatment requirements while minimizing operating costs. Specifically, the total purchase of vessel tanks for weeks 1–104 decreases by 76.5%, and the total purchase of manipulators decreases by 67%, indicating that damage to vascular robots synchronously affects the reduction in both tanks and manipulators, as shown in the **Table 5** and **Table 6**.

Observing weeks 1–104, we find a cyclical upward trend in the number of manipulators under maintenance at weeks 20, 40, and 80; the number of purchased vessel tanks and manipulators shows the same monotonic tendency. Moreover, the total weekly cost also exhibits a roughly cyclical upward trend, with similar peak values achieved in the last seven weeks, most of which occur after week 75, as shown in the **Figure 1** and **Figure 2**.

3.1 Model Evaluation and Extension

We recommend that vascular robot manufacturers increase the number of novices that each skilled manipulator can guide and reduce the weekly damage rate of vascular robots, thus lowering the total number of vessel tanks and manipulators that need to be purchased. In this way, both treatment needs can be met and operating costs can be minimized.Wang & Zou (2008)

This model for the fixed pricing problem can be repeatedly solved by altering the parameters a–h. However, it does not apply to segmented pricing problems (e.g., questions 4–5 in the original text).Jiang (2005)

4 Question 4

4.1 Overview of the Discount Policy

In this question, we introduce a preferential policy for purchasing both container boats and manipulators:

- **Container boats:**
  1. When the one-time purchase quantity is not more than 5, the unit price is 200 yuan/boat.
  2. When the one-time purchase quantity is more than 5 but not more than 10, the portion that exceeds 5 is priced at 180 yuan/boat.
  3. When the one-time purchase quantity is more than 10, the portion that exceeds 10 is priced at 160 yuan/boat.
- **Manipulators:**
  1. When the one-time purchase quantity is not more than 20, the unit price is 100 yuan/unit.
  2. When the one-time purchase quantity is more than 20 but not more than 40, the portion that exceeds 20 is priced at 90 yuan/unit.
  3. When the one-time purchase quantity is more than 40, the portion that exceeds 40 is priced at 80 yuan/unit.

These discounts encourage batch purchases. However, stocking up devices too early incurs additional maintenance costs (5 yuan/unit/week for manipulators and 10 yuan/boat/week for container boats). The core optimization challenge is to balance the benefits of discounted bulk purchases against the rising maintenance costs from holding surplus resources in earlier weeks. All other conditions from Question 3 (e.g., 10% of vascular robotics are damaged each week, each skilled operator can guide up to 20 new manipulators, etc.) still apply.

4.2 Combining Results from Question 3

4.2.1 Importing Question 3 Data

First, we import:

- The weekly purchase plan and usage plan obtained from Question 3, which already satisfies treatment requirements for weeks 1–104.
- The cost parameters and maintenance/training policies.
- The new discount rules (described above).

Although the existing purchasing plan can meet demand, it does not exploit the newly added discounts. We therefore aim to modify the timing of purchases to achieve possible bulk savings, while minimizing extra maintenance costs.

4.2.2 Recomputing Baseline Costs

With the original (unadjusted) purchase schedule from Question 3, we replace the constant unit prices (200 yuan per container boat, 100 yuan per manipulator) by the new tiered pricing scheme. We sum up how much cost the original plan would now incur if purchased exactly according to the original week-by-week amounts. This yields a baseline cost figure under the discount policy, without trying to bundle purchases across weeks.

4.3 Advanced Purchase Strategy

4.3.1 Motivation

To leverage discounts, we bundle demands from several consecutive weeks into a single (earlier) purchase, thus potentially lowering the overall unit price. If a hospital decides to pull forward, for instance, the next n weeks' worth of container boats into a single large purchase this week, then:

- The unit price decreases (due to a large one-time purchase).
- However, the surplus container boats must be stored until used, incurring weekly maintenance costs.

The same logic applies to manipulators, which have their own discount thresholds and maintenance fees.

4.3.2 Implementation of Bundling

We define n as the maximum number of weeks we consider to consolidate into an advanced bulk purchase. For each possible n (e.g., 1 to 10 weeks in advance), we:

1. Sum all container boats originally planned to be purchased in the next n weeks.
2. Compute the new (discounted) cost of that bulk purchase.
3. Compare it to the sum of the (already discounted) week-by-week purchases in those n weeks.
4. Calculate the additional maintenance cost for holding these items for n weeks (less in earlier weeks, more in later ones).
5. Compute the net "savings or waste" (bulk-purchase savings minus added maintenance).

A similar calculation is applied to manipulators.

Cost Efficiency Metric. To systematically decide which bundling strategy yields the greatest benefit, a per-week cost efficiency (i.e., savings divided by the advance period n) is used. The hospital prioritizes the bundling options that yield the highest positive net savings per week of advance purchase, subject to the constraint that no single week's demand can be "covered" by multiple bundling decisions.

4.4 Optimizing and Adjusting the Purchase Plan

4.4.1 Selecting the Best Bundling Intervals

By iterating from the highest efficiency bundling choice to the lowest, we greedily decide whether to apply that bundling interval. Once we adopt a bundling interval for certain weeks, we mark those weeks as "covered," so that they are not used again in another bundle. This ensures each week can only be advanced once, preventing double-counting or overlapping intervals. As soon as an interval is chosen, we update the purchase plan to reflect the combined order in the earliest of those weeks, and zero out the separate purchases in the succeeding weeks within that bundle.

4.4.2 Final Computation of Costs and Quantities

After determining all possible beneficial bundling intervals, we obtain a revised schedule of purchases and an updated maintenance plan for weeks 1–104. We then compute:

- The number of container boats purchased each week after bundling.
- The number of manipulators purchased each week after bundling.
- The updated maintenance cost of maintaining both container boats and manipulators.
- The overall training cost (still 10 yuan per manipulator, including skilled and novice).
- The total weekly cost as the sum of (maintenance + purchase + training).

These results are then consolidated into **Table 3**, including specific week checkpoints (Week 12, 26, 52, 78, 101, 102, 103, 104) and the grand total from Week 1–104.

Summary of Outcomes in Table 3. In the final step, the relevant data for each week is saved in the variable `table3`. The main items include:

- Number of container boats purchased.
- Number of manipulators purchased.
- Number of manipulators under maintenance.
- Number of container boats under maintenance.
- Number of trained manipulators (skilled workers plus novices).
- The total weekly cost, itemized as purchase cost, maintenance cost, and training cost.

Specific weeks of interest (12, 26, 52, 78, 101, 102, 103, 104) as well as the sum over weeks 1–104 are excerpted into `table3_1` to answer the question prompt. All relevant variables can then be saved for subsequent analysis, as requested in the problem statement.**Code** B

5 Question 5

5.1 Problem Description

Following the conditions set out in Question 4, we now extend the planning horizon to weeks 105–112. The goal is to investigate how hospitals should meet the additional demands for vascular robotics in these extra weeks while minimizing operating costs. Specifically, we compare two different strategies:

- **Solution 1**: Continue using the optimal purchasing and training scheme for weeks 1–104 from Question 4, and then, at the beginning of week 105, purchase container boats at 300 yuan each and manipulators at 150 yuan each, both of which can be used immediately. For the subsequent weeks (106–112), purchases follow the same discount policy as in Question 4.
- **Solution 2**: Integrate the demand for weeks 105–112 directly into the overall plan for weeks 1–112, so that all purchases from the very beginning (week 1) take into account the total requirement from weeks 1–112.

Our objective is to quantify the difference in total operating costs between these two solutions over the entire 1–112 weeks.

5.2 Prediction of Demand for Weeks 105–112

5.2.1 Data Import

We first import the weekly data for vascular robotic usage from weeks 1–104, as prepared in earlier questions. Using these data, we employ a forecasting approach to predict the number of vascular robotics required for each of weeks 105–112. In essence, the historical usage is extrapolated by an appropriate model to generate estimates for the eight additional weeks.

5.2.2 Forecasting Procedure

From a time-series perspective (or any suitable prediction method), we calculate the predicted usage for each of the next eight weeks. After obtaining the forecast values, they are rounded up (ceiling) to ensure that hospitals prepare for the maximum possible whole number of vascular robotics each week.

5.3 Solution 1: Week 105 Direct Purchase and Continued Discounts

5.3.1 Transition from Week 104 to Week 105

In this scenario, the hospital follows precisely the optimal plan from Question 4 for weeks 1–104. When week 105 begins, two special high-cost purchasing options become available to ensure immediate usage:

- Each container boat purchased at 300 yuan.
- Each manipulator purchased at 150 yuan (and fully skilled, requiring no training delay).

Thus, the hospital may opt to fill any additional need at week 105 in one lump-sum purchase, so that from the outset of week 105 all required container boats and manipulators are instantly ready. For weeks 106–112, the same discount policy from Question 4 remains in effect.

5.4 Solution 2: 1–112 Weeks in One Plan

5.4.1 Purchasing and Maintenance Strategy

As per Question 4, the hospital can purchase container boats and manipulators at discounted prices depending on the size of each purchase order:

- Container boats:
  - Up to 5 per week: 200 yuan per boat
  - From 6 to 10 per week: 200 yuan for the first 5, 180 yuan for each unit above 5
  - Above 10 per week: 200 yuan for the first 5, 180 yuan for the next 5, 160 yuan for each unit above 10
- Manipulators:
  - Up to 20 per week: 100 yuan per manipulator
  - From 21 to 40 per week: 100 yuan for the first 20, 90 yuan for each unit above 20
  - Above 40 per week: 100 yuan for the first 20, 90 yuan for the next 20, 80 yuan for each unit above 40

We assume:

- A 10% damage rate per week for the vascular robotics currently in use (rounded to the nearest whole number).
- Each skilled manipulator can train up to 20 new manipulators in a single week.
- A container boat, once bought, needs one week of checking before use. Manipulators, once bought, need one week of training (under skilled guidance) before they become fully qualified.

- Maintenance costs apply to any container boat or manipulator that is not currently in use (or is waiting to be used again after a week's operation).

By formulating the weekly purchase, usage, damage, and carry-over maintenance constraints from week 1 to week 112 (with the forecasted demands included for weeks 105–112), one can compute the minimal total cost over the 112-week horizon. This integrated approach is referred to as *Solution 2*.

5.5 Cost Comparison and Final Result

Basic Computation To compare total costs:

- **Solution 1** cost =
  - The sum of all costs from the existing 1–104 strategy in Question 4.
  - Plus the cost of directly purchasing the required container boats and manipulators in week 105 at the higher prices (300 yuan and 150 yuan, respectively).
  - Plus the cost of any subsequent weekly purchases (weeks 106–112) under the usual discount rates.
- **Solution 2** cost =
  - The integrated minimal cost when the entire horizon of 1–112 weeks is planned at once, obeying the discount policy throughout, with newly purchased manipulators still requiring training.

The difference in operating cost (Solution 1 minus Solution 2, or vice versa) is then computed. This difference essentially captures the trade-off between direct purchase (high price but immediate availability) and the more gradual approach that uses discount purchases and training.

Interpretation of the Cost Differential The primary driver of the differential is the extra cost paid in Solution 1 at week 105 for immediate usage, instead of training newly purchased manipulators or checking newly purchased boats in week 104. However, Solution 1 might also reduce some training and maintenance costs if properly timed. Therefore, the net difference must be calculated by summing all related cost components.

Summary After determining the forecast for the additional weeks (105–112), the two solutions are compared. The resulting difference in total costs provides insight into the financial trade-off of late lump-sum "immediate-usage" purchasing versus early integrated planning. In practice, a hospital might choose between these two strategies based on available budgets, urgency of deployment, or constraints on the timing of training and maintenance.**Code C**

All specific purchase and maintenance details, along with the resulting total costs, are preserved in the calculations (see the appended code for full implementation and the resulting numeric values). The final outcome includes:

- The number of container boats and manipulators bought each week under each solution.
- The overall minimum cost for weeks 1–112.
- The difference in minimum total operating costs between Solution 1 and Solution 2.

References

Qiyuan Jiang. *University Mathematical Experiments*. Tsinghua University Press Co., Ltd., 2005.

Xiaoyin Wang and Tingrong Zou. Mathematical software and mathematical experiments. *Beijing: Science Press*, 2008.

Part I

Appendix

Table of Contents

A QUESTION 1-3

```matlab
clc; clear;

% Prompt user to input various costs and parameters
a = input('Please enter the purchase cost of a container vessel: ');
b = input('Please enter the purchase cost of an operator: ');
c = input('Please enter the maintenance cost of a container vessel: ');
d = input('Please enter the maintenance cost of an operator: ');
e = input('Please enter the training cost for an operator (including experienced ones): ');
f = input('Please enter the number of new operators one experienced operator can train: ');
g = input('Please enter the damage rate of vascular robots (as a decimal): ');

% Read weekly demand data from Excel
h = xlsread('Appendix 1 2.xlsx','Sheet1','A1:A104');

% Ask user how many weeks to consider (<= 104)
v = input('Please enter the number of weeks to consider (<=104): ');
w = v + 1;

% Initialize CN (container vessels) matrix: [existing, assigned, newly purchased]
CN = zeros(w, 3);
CN_0 = [13, 0, 0];  % Initial state
CN(1, :) = CN_0;

% Initialize MN (operators) matrix: [existing, assigned, newly hired]
MN = zeros(w, 3);
MN_0 = [50, 0, 0];  % Initial state
MN(1, :) = MN_0;

% Main loop over weeks
for i = 1:v
    if MN(i,1) < h(i)*4
        MN(i,3) = h(i)*4 - MN(i,1);  % Number of new operators needed
        MN(i+1,1) = sum(MN(i,1:3)) - round(MN(i,2)*g) - h(i)*4;
        MN(i+1,2) = h(i)*4;
    else
        MN(i,3) = 0;
        MN(i+1,1) = sum(MN(i,1:3)) - round(MN(i,2)*g) - h(i)*4;
        MN(i+1,2) = h(i)*4;
    end

    if sum(CN(i,1:2)) - round(CN(i,2)*g) < h(i)
        CN(i,3) = h(i) - sum(CN(i,1:2)) + round(CN(i,2)*g);  % New container vessels needed
        CN(i+1,1) = sum(CN(i,1:3)) - round(CN(i,2)*g) - h(i);
        CN(i+1,2) = h(i);
    else
        CN(i,3) = 0;
        CN(i+1,1) = sum(CN(i,1:3)) - round(CN(i,2)*g) - h(i);
        CN(i+1,2) = h(i);
    end
end

% Extract relevant data
k = CN(2:end,3);    % New container vessels per week
l = MN(2:end,3);    % New operators per week
m = MN(2:end,1);    % Remaining operators
n = CN(2:end,1);    % Remaining container vessels
```

```
o = ceil(MN(2:end,3)./f) + MN(2:end,3);  % Total number of trainees
    including trainers

% Cost calculations
p1 = k .* a;  % Cost of container vessel purchase
p2 = l .* b;  % Cost of operator purchase
p3 = n .* c;  % Maintenance cost for vessels
p4 = m .* d;  % Maintenance cost for operators
p5 = o .* e;  % Training cost
p = p1 + p2 + p3 + p4 + p5;

% Create summary table
table = [k, l, m, n, o, p];
table(end+1, :) = sum(table);  % Add total at the bottom

% Save result
save t.mat table
```

B QUESTION 4

```
%% Question 4
% If there is a discount policy for purchasing manipulators and
    container boats:
%  1) For container boats:
%      - When the one-time purchase quantity is no more than 5, the
    unit price is 200 yuan/unit.
%      - When the one-time purchase quantity exceeds 5 but is not more
    than 10, the portion above 5 has a unit price of 180 yuan/unit.
%      - When the one-time purchase quantity exceeds 10, the portion
    above 10 has a unit price of 160 yuan/unit.
%
%  2) For manipulators:
%      - When the one-time purchase quantity is no more than 20, the
    unit price is 100 yuan/unit.
%      - When the one-time purchase quantity exceeds 20 but is not more
     than 40, the portion above 20 has a unit price of 90 yuan/unit.
%      - When the one-time purchase quantity exceeds 40, the portion
    above 40 has a unit price of 80 yuan/unit.
%
% Other conditions follow Question 3. How should the total purchase of
     container boats and manipulators be adjusted over Weeks 1-104?

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Step 1: Import the results from Question 3 and other necessary data
% Load the result data from Question 3
load('ques3_result.mat');

% Costs:
RobotShipCost = 200;            % Cost of one container boat (baseline
     unit price)
RobotArmCost = 100;             % Cost of one manipulator (baseline
    unit price)
RobotShipMaintainCost = 10;     % Maintenance cost for one container
    boat per week
RobotArmMaintainCost = 5;       % Maintenance cost for one manipulator
     per week
RobotArmTrainingCost = 10;      % Training cost per manipulator (both
    new and skilled)

OneCanTrainNumber = 20;         % One skilled manipulator can train up
     to 20 new manipulators
DestroyedRate = 0.1;            % 10% of vascular robotics are
    destroyed per week
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Step 2: Calculate the purchasing cost of the original plan (
    Question 3) under the new discount policy
Cost = [];
for i = 1:size(table2,1)
    % Cost of purchasing container boats in the original plan
    % Discount policy for container boats:
    %  <=5: 200 yuan/unit
    %  >5 & <=10: portion above 5 costs 180 yuan/unit
    %  >10: portion above 10 costs 160 yuan/unit

    if table2(i,1) <= 5
        Cost1 = table2(i,1) * 200;
    elseif table2(i,1) <= 10
        Cost1 = 200*5 + (table2(i,1)-5)*180;
    else
        Cost1 = 200*5 + 180*5 + (table2(i,1)-10)*160;
    end
    Cost(i,1) = Cost1;

    % Cost of purchasing manipulators in the original plan
    % Discount policy for manipulators:
    %  <=20: 100 yuan/unit
    %  >20 & <=40: portion above 20 costs 90 yuan/unit
    %  >40: portion above 40 costs 80 yuan/unit

    if table2(i,2) <= 20
        Cost2 = table2(i,2) * 100;
    elseif table2(i,2) <= 40
        Cost2 = 100*20 + (table2(i,2)-20)*90;
    else
        Cost2 = 100*20 + 90*20 + (table2(i,2)-40)*80;
    end
    Cost(i,2) = Cost2;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Step 3: Compute the cost savings (or additional costs) of shifting
    purchases 1~n weeks earlier
% Up to n weeks in advance (here "in advance" means consolidating the
    total demand of several weeks into one earlier week).
% Example: Suppose from Week 1 to Week 5 the purchase demands of
    container boats are 2,5,7,3,1 respectively.
%   - Bringing Week 5's purchase 4 weeks earlier implies that in Week
    1, we purchase 2+5+7+3+1 = 18 boats at once.
%   - In that case, Weeks 1-5 are "covered" by that single large
    purchase.
%   - The same week cannot be covered by two different "advance
    intervals."

n = 10; % Maximum number of weeks to attempt advancing purchases

SaveOrWaste = cell(n,1);
for i = 1:n
    SaveOrWaste{i} = [];
    for j = 1:104
        if j <= i
            SaveOrWaste{i}(j,1:6) = [0,0,0,0,0,0];
        else
            % --- Container boat consolidation ---
            BundleOfRobotShip = sum(table2(j-i:j,1));
            % Calculate the purchasing cost for the consolidated
                container boats
            if BundleOfRobotShip <= 5
```

```
                CostMoneyofRobotShip = BundleOfRobotShip * 200;
            elseif BundleOfRobotShip <= 10
                CostMoneyofRobotShip = 200*5 + (BundleOfRobotShip-5)*180;
            else
                CostMoneyofRobotShip = 200*5 + 180*5 + (BundleOfRobotShip-10)*160;
            end
            % Savings from container-boat bulk purchase
            SaveMoneyofRobotShip = sum(Cost(j-i:j,1)) - CostMoneyofRobotShip;
            % Maintenance cost from holding these container boats earlier
            WasteMoneyofRobotShipMaintain = sum(table2(j-i+1:j,1) .* (1:i)') * RobotShipMaintainCost;

            % --- Manipulator consolidation ---
            BundleOfRobotArm = sum(table2(j-i:j,2));
            % Calculate the purchasing cost for the consolidated manipulators
            if BundleOfRobotArm <= 20
                CostMoneyofRobotArm = BundleOfRobotArm * 100;
            elseif BundleOfRobotArm <= 40
                CostMoneyofRobotArm = 100*20 + (BundleOfRobotArm-20)*90;
            else
                CostMoneyofRobotArm = 100*20 + 90*20 + (BundleOfRobotArm-40)*80;
            end
            % Savings from manipulator bulk purchase
            SaveMoneyofRobotArm = sum(Cost(j-i:j,2)) - CostMoneyofRobotArm;
            % Maintenance cost for holding these manipulators earlier
            WasteMoneyofRobotArmMaintain = sum(table2(j-i+1:j,2) .* (1:i)') * RobotArmMaintainCost;

            % Net difference for container boats (savings minus maintenance)
            a = SaveMoneyofRobotShip - WasteMoneyofRobotShipMaintain;
            % Net difference for manipulators (savings minus maintenance)
            b = SaveMoneyofRobotArm - WasteMoneyofRobotArmMaintain;

            % Store the results:
            % SaveOrWaste{i}(j,:) = [SaveShip, MaintainShip, NetShip, SaveArm, MaintainArm, NetArm]
            SaveOrWaste{i}(j,1:6) = [SaveMoneyofRobotShip, WasteMoneyofRobotShipMaintain, a, ...
                                     SaveMoneyofRobotArm, WasteMoneyofRobotArmMaintain, b];
        end
    end
end

% Extract container-boat savings across different lead times
SaveOrWasteOfRobotShip = [];
RateOfRobotShip = [];  % Cost-saving efficiency = (Saving) / (Weeks in advance)

% Extract manipulator savings across different lead times
SaveOrWasteOfRobotArm = [];
RateOfRobotArm = [];

for i = 1:n
```

```
    SaveOrWasteOfRobotShip = [SaveOrWasteOfRobotShip, SaveOrWaste{i
        }(:,3)];
    RateOfRobotShip = [RateOfRobotShip, SaveOrWaste{i}(:,3) ./ i];

    SaveOrWasteOfRobotArm = [SaveOrWasteOfRobotArm, SaveOrWaste{i
        }(:,6)];
    RateOfRobotArm = [RateOfRobotArm, SaveOrWaste{i}(:,6) ./ i];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Step 4: Based on cost-saving efficiency, determine the optimal
    advance weeks for each purchase
% A given week cannot be covered by two overlapping "advance intervals
    ."
% Here we use a record array: 0 means not covered yet, 1 means it has
    been covered (and cannot be covered again).

recordRobotShip = zeros(104,1);
recordRobotArm = zeros(104,1);

% We'll make a copy of the original plan:
% RobotShipNumber and RobotArmNumber are from the loaded data (
    ques3_result.mat).

% Eliminate negative values in the efficiency tables (only positive
    net savings are relevant)
RateOfRobotShip(RateOfRobotShip < 0) = 0;
RateOfRobotArm(RateOfRobotArm < 0) = 0;

% Greedy selection: repeatedly find the maximum cost-saving efficiency
    , implement that advance if valid
MaxRateOfRobotShip = max(max(RateOfRobotShip));
while MaxRateOfRobotShip > 0
    [x, y] = find(RateOfRobotShip == MaxRateOfRobotShip, 1); % x =
        week index, y = lead time
    if sum(recordRobotShip(x-y : x,1)) == 0
        % Apply this consolidation to container boats

        % Adjust the maintenance plan: the covered weeks' "added
            maintenance" is replaced by the big purchase
        RobotShipNumber(x-y+2 : x+1,1) = sum(RobotShipNumber(x-y+2 : x
            +1,3));

        % Update the original purchase plan
        RobotShipNumber(x-y+1 : x+1,3) = sum(RobotShipNumber(x-y+1 : x
            +1,3));
        RobotShipNumber(x-y+2 : x+1,3) = 0;

        % Mark as covered
        recordRobotShip(x-y : x,1) = 1;
    end

    % Remove the used entry and find the next best
    RateOfRobotShip(x,y) = 0;
    MaxRateOfRobotShip = max(max(RateOfRobotShip));
end

MaxRateOfRobotArm = max(max(RateOfRobotArm));
while MaxRateOfRobotArm > 0
    [x, y] = find(RateOfRobotArm == MaxRateOfRobotArm, 1); % x = week
        index, y = lead time
    if sum(recordRobotArm(x-y : x,1)) == 0
        % Apply this consolidation to manipulators

        % Adjust the maintenance plan
```

```matlab
        RobotArmNumber(x-y+2 : x+1,1) = sum(RobotArmNumber(x-y+2 : x+1,3));

        % Update the original purchase plan
        RobotArmNumber(x-y+1 : x+1,3) = sum(RobotArmNumber(x-y+1 : x+1,3));
        RobotArmNumber(x-y+2 : x+1,3) = 0;

        % Mark as covered
        recordRobotArm(x-y : x,1) = 1;
    end

    % Remove the used entry and move on
    RateOfRobotArm(x,y) = 0;
    MaxRateOfRobotArm = max(max(RateOfRobotArm));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Step 5: Fill in the updated plan into table3 and compute final costs
disp('Number of container boats purchased:')
a = RobotShipNumber(2:end, 3)

disp('Number of manipulators purchased:')
b = RobotArmNumber(2:end, 3)

disp('Number of manipulators under maintenance:')
c = RobotArmNumber(2:end, 1)

disp('Number of container boats under maintenance:')
d = RobotShipNumber(2:end, 1)

disp('Number of manipulators participating in training (including both "skilled workers" and "newcomers"):')
e = ceil(RobotArmNumber(2:end, 3) ./ OneCanTrainNumber) + RobotArmNumber(2:end, 3)

disp('Weekly costs (unit: yuan):')
f1 = RobotShipNumber(2:end,1) .* RobotShipMaintainCost; % Maintenance cost for container boats
f2 = []; % Purchase cost for container boats (with discount)
f3 = RobotArmNumber(2:end,1) .* RobotArmMaintainCost;  % Maintenance cost for manipulators
f4 = []; % Purchase cost for manipulators (with discount)

for i = 1:104
    % Calculate container boat purchase cost under discount
    if RobotShipNumber(i+1, 3) <= 5
        Cost1 = RobotShipNumber(i+1, 3) * 200;
    elseif RobotShipNumber(i+1, 3) <= 10
        Cost1 = 200*5 + (RobotShipNumber(i+1, 3)-5)*180;
    else
        Cost1 = 200*5 + 180*5 + (RobotShipNumber(i+1, 3)-10)*160;
    end
    f2 = [f2; Cost1];

    % Calculate manipulator purchase cost under discount
    if RobotArmNumber(i+1, 3) <= 20
        Cost2 = RobotArmNumber(i+1, 3) * 100;
    elseif RobotArmNumber(i+1, 3) <= 40
        Cost2 = 100*20 + (RobotArmNumber(i+1, 3)-20)*90;
    else
        Cost2 = 100*20 + 90*20 + (RobotArmNumber(i+1, 3)-40)*80;
    end
    f4 = [f4; Cost2];
```

```
end

% Training cost
f5 = (ceil(RobotArmNumber(2:end,3) ./ OneCanTrainNumber) +
    RobotArmNumber(2:end,3)) * 10;

f = f1 + f2 + f3 + f4 + f5

table3 = [a, b, c, d, e, f];
table3(end+1,:) = sum(table3);

disp('All data for Table 3 is stored in the variable "table3". The
    subset "table3_1" contains the required weeks:')
table3_1 = table3([12,26,52,78,101,102,103,104,end], :)

%% Save the data
save('ques4_result.mat', 'table3', 'table3_1', 'RobotShipNumber', '
    RobotArmNumber');
```

C QUESTION 5

```
%% Problem 5
% Predict the usage demand of vascular robotics for weeks 105 - 112.
% In order to study the cost of using vascular robotics in weeks
    105-112,
% following the conditions from Question 4, two schemes can be
    considered.
%
% Scheme 1:
% On the basis of the optimal result from weeks 1-104, the hospital
    may need
% at the beginning of week 105 to purchase container boats that can be
     used
% immediately at 300 yuan each, and skilled manipulators at 150 yuan
    each.
% In subsequent weeks, continue with the discount policy for
    purchasing new
% container boats and manipulators in order to meet the demand from
    weeks
% 105-112.
%
% Scheme 2:
% Consider the demand for vascular robotics for weeks 1-112 in a
    single plan.
% Compare the difference in minimum operating costs for weeks 1-112
    between
% the two schemes.

%% Step 1: Import the necessary data
% Load the processed data
load('data_of_202251A.mat');

%% Step 2: Predict the usage demand of vascular robotics for weeks
    105-112
for i=1:8
    yhatpredict = ceil(predict(RobotsUseNumber, 8));
    RobotsUseNumber = [RobotsUseNumber, yhatpredict];
end
plot(1:length(RobotsUseNumber), RobotsUseNumber)

%% Step 3: Calculate the demand for weeks 1-112 under Scheme 2
% (i.e., reuse the algorithm from Questions 3 and 4)

```

```
% One skilled manipulator can train up to 20 new manipulators per week
OneCanTrainNumber = 20;
% 10% of vascular robotics in use are destroyed each week
DestroyedRate = 0.1;

% Compute how many container boats and manipulators are purchased each week
% Initialize the container boats and manipulators status per week.
% Each row represents a week. The first row is the initial data.
% The first column is the number of units under maintenance (unused),
% the second column is the number in use, and the third column is the number purchased.

RobotShipNumber = zeros(112, 3);
RobotShipNumber(1,:) = RobotShipNumber_0;  % Preset initial data
RobotArmNumber = zeros(112, 3);
RobotArmNumber(1,:) = RobotArmNumber_0;

for i = 1:112
    % Manipulators used this week will need maintenance next week
    % and can only be used the week after. Purchased this week -> train for 1 week -> available next week.
    % Container boats used this week can be reused next week.
    % Purchased this week -> check for 1 week -> available next week.

    % 1) Calculate the necessary manipulators
    % Check if the manipulators under maintenance (from last week) are enough for this week's usage
    if RobotArmNumber(i,1) < RobotsUseNumber(i)*4
        % If not enough, purchase new manipulators this week
        RobotArmNumber(i,3) = RobotsUseNumber(i)*4 - RobotArmNumber(i,1);
        % Next week's maintenance = sum of all manipulators we had - destroyed ones - used ones
        RobotArmNumber(i+1,1) = sum(RobotArmNumber(i,1:3)) - round(RobotArmNumber(i,2)*DestroyedRate) - RobotsUseNumber(i)*4;
        RobotArmNumber(i+1,2) = RobotsUseNumber(i)*4;  % manipulators in use
    else
        % Otherwise, no new manipulators need to be purchased
        RobotArmNumber(i,3) = 0;
        RobotArmNumber(i+1,1) = sum(RobotArmNumber(i,1:3)) - round(RobotArmNumber(i,2)*DestroyedRate) - RobotsUseNumber(i)*4;
        RobotArmNumber(i+1,2) = RobotsUseNumber(i)*4;
    end

    % 2) Calculate the necessary container boats
    % Check if all container boats (in maintenance + in use) minus destroyed are enough for this week
    if sum(RobotShipNumber(i,1:2)) - round(RobotShipNumber(i,2)*DestroyedRate) < RobotsUseNumber(i)
        RobotShipNumber(i,3) = RobotsUseNumber(i) - sum(RobotShipNumber(i,1:2)) + round(RobotShipNumber(i,2)*DestroyedRate);
        RobotShipNumber(i+1,1) = sum(RobotShipNumber(i,1:3)) - round(RobotShipNumber(i,2)*DestroyedRate) - RobotsUseNumber(i);
        RobotShipNumber(i+1,2) = RobotsUseNumber(i);  % boats in use
    else
        RobotShipNumber(i,3) = 0;
        RobotShipNumber(i+1,1) = sum(RobotShipNumber(i,1:3)) - round(RobotShipNumber(i,2)*DestroyedRate) - RobotsUseNumber(i);
        RobotShipNumber(i+1,2) = RobotsUseNumber(i);
    end
end
```

```
% Compute purchase costs for container boats and manipulators according to the new discount policy
Cost = [];
for i = 1:size(RobotShipNumber,1)-1
    % 1) Container boat purchase cost
    %   Up to 5 boats: 200 yuan each
    %   6-10 boats: first 5 cost 200 each, above 5 cost 180 each
    %   above 10 boats: first 5 cost 200 each, next 5 cost 180 each, above 10 cost 160 each
    if RobotShipNumber(i+1,3) <= 5
        Cost1 = RobotShipNumber(i+1,3)*200;
    elseif RobotShipNumber(i+1,3) <= 10
        Cost1 = 200*5 + (RobotShipNumber(i+1,3)-5)*180;
    else
        Cost1 = 200*5 + 180*5 + (RobotShipNumber(i+1,3)-10)*160;
    end
    Cost(i,1) = Cost1;

    % 2) Manipulator purchase cost
    %   Up to 20 units: 100 yuan each
    %   21-40 units: first 20 cost 100 each, above 20 cost 90 each
    %   above 40 units: first 20 cost 100 each, next 20 cost 90 each, above 40 cost 80 each
    if RobotArmNumber(i+1,3) <= 20
        Cost2 = RobotArmNumber(i+1,3)*100;
    elseif RobotArmNumber(i+1,3) <= 40
        Cost2 = 100*20 + (RobotArmNumber(i+1,3)-20)*90;
    else
        Cost2 = 100*20 + 90*20 + (RobotArmNumber(i+1,3)-40)*80;
    end
    Cost(i,2) = Cost2;
end

% Compute the savings from possibly advancing purchases some number of weeks
% in order to reduce the total cost, minus the increased maintenance cost incurred.

n = 10;   % maximum number of weeks we can advance

for i = 1:n
    SaveOrWaste{i} = [];
    for j = 1:112
        if j <= i
            SaveOrWaste{i}(j,1:6) = [0,0,0,0,0,0];
        else
            % Sum of container boats purchased from week (j-i) to j
            BundleOfRobotShip = sum(RobotShipNumber(j-i:j,3));
            % Calculate container boat cost if purchased together
            if BundleOfRobotShip <= 5
                CostMoneyofRobotShip = BundleOfRobotShip*200;
            elseif BundleOfRobotShip <= 10
                CostMoneyofRobotShip = 200*5 + (BundleOfRobotShip-5)*180;
            else
                CostMoneyofRobotShip = 200*5 + 180*5 + (BundleOfRobotShip-10)*160;
            end
            % Savings in container boats = original cost - bundled cost
            SaveMoneyofRobotShip = sum(Cost(j-i:j,1)) - CostMoneyofRobotShip;
            % Maintenance cost for boats purchased in advance
            WasteMoneyofRobotShipMaintain = sum(RobotShipNumber(j-i+1:j,3) .* (1:i)') * RobotShipMaintainCost;
```

```

            % Sum of manipulators purchased from week (j-i) to j
            BundleOfRobotArm = sum(RobotArmNumber(j-i:j,3));
            % Calculate manipulator cost if purchased together
            if BundleOfRobotArm <= 20
                CostMoneyofRobotArm = BundleOfRobotArm*100;
            elseif BundleOfRobotArm <= 40
                CostMoneyofRobotArm = 100*20 + (BundleOfRobotArm-20)*90;
            else
                CostMoneyofRobotArm = 100*20 + 90*20 + (BundleOfRobotArm-40)*80;
            end
            % Savings in manipulators = original cost - bundled cost
            SaveMoneyofRobotArm = sum(Cost(j-i:j,2)) - CostMoneyofRobotArm;
            % Maintenance cost for manipulators purchased in advance
            WasteMoneyofRobotArmMaintain = sum(RobotArmNumber(j-i+1:j,3) .* (1:i)') * RobotArmMaintainCost;

            % Net savings (container boats)
            a = SaveMoneyofRobotShip - WasteMoneyofRobotShipMaintain;
            % Net savings (manipulators)
            b = SaveMoneyofRobotArm - WasteMoneyofRobotArmMaintain;

            % Store results in a cell array, each cell for 1~n weeks in advance
            SaveOrWaste{i}(j,1:6) = [SaveMoneyofRobotShip, WasteMoneyofRobotShipMaintain, a, SaveMoneyofRobotArm, WasteMoneyofRobotArmMaintain, b];
        end
    end
end

% Summarize container boat and manipulator savings
SaveOrWasteOfRobotShip = [];
SaveOrWasteOfRobotArm = [];
RateOfRobotShip = [];
RateOfRobotArm = [];

for i = 1:n
    SaveOrWasteOfRobotShip = [SaveOrWasteOfRobotShip, SaveOrWaste{i}(:,3)];
    RateOfRobotShip        = [RateOfRobotShip, SaveOrWaste{i}(:,3)./i];
    SaveOrWasteOfRobotArm  = [SaveOrWasteOfRobotArm, SaveOrWaste{i}(:,6)];
    RateOfRobotArm         = [RateOfRobotArm, SaveOrWaste{i}(:,6)./i];
end

% Choose the best "advance" weeks strategy based on the highest per-week saving rate
RateOfRobotShip(RateOfRobotShip<0) = 0;
RateOfRobotArm(RateOfRobotArm<0)   = 0;

recordRobotShip = zeros(112,1);
recordRobotArm  = zeros(112,1);

% For container boats
MaxRateOfRobotShip = max(max(RateOfRobotShip));
while MaxRateOfRobotShip > 0
    [x,y] = find(RateOfRobotShip==MaxRateOfRobotShip,1);
    if sum(recordRobotShip(x-y:x,1))==0
        RobotShipNumber(x-y+2:x+1,1) = sum(RobotShipNumber(x-y+2:x+1,3));
```

```
        RobotShipNumber(x-y+1:x+1,3) = sum(RobotShipNumber(x-y+1:x+1,3));
        RobotShipNumber(x-y+2:x+1,3) = 0;
        recordRobotShip(x-y:x,1) = 1;
    end
    RateOfRobotShip(x,y) = 0;
    MaxRateOfRobotShip = max(max(RateOfRobotShip));
end

% For manipulators
MaxRateOfRobotArm = max(max(RateOfRobotArm));
while MaxRateOfRobotArm > 0
    [x,y] = find(RateOfRobotArm==MaxRateOfRobotArm,1);
    if sum(recordRobotArm(x-y:x,1))==0
        RobotArmNumber(x-y+2:x+1,1) = sum(RobotArmNumber(x-y+2:x+1,3));
        RobotArmNumber(x-y+1:x+1,3) = sum(RobotArmNumber(x-y+1:x+1,3));
        RobotArmNumber(x-y+2:x+1,3) = 0;
        recordRobotArm(x-y:x,1) = 1;
    end
    RateOfRobotArm(x,y) = 0;
    MaxRateOfRobotArm = max(max(RateOfRobotArm));
end

% Compile results in table4
disp('Number of container boats purchased')
a = RobotShipNumber(2:end,3)
disp('Number of manipulators purchased')
b = RobotArmNumber(2:end,3)
disp('Number of manipulators under maintenance')
c = RobotArmNumber(2:end,1)
disp('Number of container boats under maintenance')
d = RobotShipNumber(2:end,1)
disp('Number of manipulators in training (including "skilled" and "novices")')
e = ceil(RobotArmNumber(2:end,3)./OneCanTrainNumber) + RobotArmNumber(2:end,3)
disp('Weekly cost (unit: yuan)')
f1 = RobotShipNumber(2:end,1)*RobotShipMaintainCost;  % maintenance cost for container boats
f2 = RobotShipNumber(2:end,3)*RobotShipCost;          % purchase cost for container boats
f3 = RobotArmNumber(2:end,1)*RobotArmMaintainCost;    % maintenance cost for manipulators
f4 = RobotArmNumber(2:end,3)*RobotArmCost;            % purchase cost for manipulators
f5 = (ceil(RobotArmNumber(2:end,3)./OneCanTrainNumber) + RobotArmNumber(2:end,3))*10;  % training cost
f  = f1 + f2 + f3 + f4 + f5

table4 = [a, b, c, d, e, f];
table4(end+1,:) = sum(table4);
disp('All Scheme 2 data for weeks 1--112 is stored in table4. table4_1 only contains data from weeks 105--112 as required by the problem.')
table4_1 = table4(105:end,:)

%% Step 4: Compare the difference in minimum operating costs (weeks 1--112) between the two schemes
% The difference in costs lies in the expenses incurred by purchasing and training
% manipulators and container boats in week 104 under Scheme 2 versus
% directly purchasing them in week 105 under Scheme 1 at higher prices.
```

```
% Number of container boats/manipulators to be purchased in week 104 (
    Scheme 2)
RobotShip_104 = RobotShipNumber(105,3);
RobotArm_104  = RobotArmNumber(105,3);

% Original purchase + training cost in week 104
% Container boat cost
if RobotShip_104 <= 5
    Cost1 = RobotShip_104*200;
elseif RobotShip_104 <= 10
    Cost1 = 200*5 + RobotShip_104*180;
else
    Cost1 = 200*5 + 180*5 + (RobotShip_104-10)*160;
end

% Manipulator cost
if RobotArm_104 <= 20
    Cost2 = RobotArm_104*100;
elseif RobotArm_104 <= 40
    Cost2 = 100*20 + (RobotArm_104-20)*90;
else
    Cost2 = 100*20 + 90*20 + (RobotArm_104-40)*80;
end

% Training cost
Cost3 = (ceil(RobotArm_104 / OneCanTrainNumber) + RobotArm_104)*10;

% Direct purchase cost if done in week 105 (Scheme 1)
DrectBuyCost = RobotShip_104*300 + RobotArm_104*150;

% Cost difference = (original purchase + training) - direct purchase
    in week 105
disp('Scheme 1 cost - Scheme 2 cost =')
CostDifferential = Cost1 + Cost2 + Cost3 - DrectBuyCost

%% Save data
save ques5_result.mat table4 table4_1 CostDifferential
```