

# A Compound Optimization Greedy Strategy with Reverse Correction Mechanism

Han Shen

School of Computer Science and Engineering  
Xi'an Technological University  
Xi'an, 710021, China  
e-mail: sunny\_shine\_zj@163.com

Zhongsheng Wang

School of Computer Science and Engineering  
Xi'an Technological University  
Xi'an, 710021, China  
e-mail: wzsh1681@163.com

**Abstract**—Greedy strategy is an algorithm thinking with local optimization as the core idea, but only when the problem has no after-effect, the global optimization can be achieved. Therefore, greedy strategy is not the first choice for researchers to solve the problem. Based on the greedy strategy, this paper adds the mechanism of reverse correction thinking, transfers the local optimal solution to the global optimal solution, and puts forward a compound optimal greedy strategy integrating reverse correction thinking. Based on the actual application scenario of blood robot operating costs, the overall "simple greedy strategy model" is constructed and tested based on the greedy strategy as the main modeling basis according to the application needs. On this basis, the interaction relationship between local optimal solutions is deeply analyzed, and the reverse correction mechanism is integrated to optimize the system through the two steps of reverse allocation and reverse merge repair. Gradually improve the model to get the optimized "reverse modified greedy strategy model", the algorithm can effectively reduce the operating cost. On this basis, in order to test the optimization effect, the effectiveness and stability of the reverse correction mechanism were verified by modifying some parameters of the application scene and randomly generating multiple arrays for re-test, etc., and new parameters were selected to re-run the application scene, and satisfactory verification results were obtained. Compared with other modeling ideas of the same topic, this model weakens the expression of the overall function and emphasizes the change relationship and action mechanism between data, and obtains better operation results. Greedy strategy is very conducive to the analysis of the relationship between requirements, constraints and variables. According to the actual application needs, combined with the mathematical

analysis method, the reverse correction mechanism is added to the greedy strategy modeling. In the demand sequence test of 100 groups of simulation, the maximum saving rate can be close to 1.6%, while the lowest saving rate is less than 0.6%, and the average saving rate is 0.9677%. It can save tens of thousands of operating costs for application scenarios.

**Keywords**—*Greedy Strategy; Lowest Operating Cost; Reverse Correction; Fallback Mechanism; Compound Algorithm*

## I. INTRODUCTION

Greedy strategy refers to a solving method that starts from the initial state of the problem and obtains the optimal solution or better solution through several times of greedy selection, also known as greedy strategy, greedy algorithm or greedy algorithm. The most important feature of greedy strategy is the local optimal solution, which is an algorithm strategy that depends on the choices made rather than the choices not made. From a global perspective, the local optimal solution does not mean the global optimal solution. The greedy strategy used in each local optimal solution may fight with each other, and the latter may affect the former, leading to the final global result is not ideal. The after-effect is one of the biggest influencing factors of greedy strategy. Therefore, greedy strategy has great limitations in solving problems such as multi-objective programming and dynamic programming, so it is not the first choice for researchers to solve problems in the past.

From the perspective of mathematical thinking, it is a logical thinking perspective to find the global optimal solution by means of the local optimal solution. However, in the actual application of greedy algorithm, local contradictions and mutual influences are likely to occur, resulting in poor global performance or even "collapse". However, this does not mean that greedy strategy is nothing. It may also work well.

Based on the application scenario of "the lowest operating cost of vascular robot", this paper incorporates a "reverse correction" mechanism into the greedy algorithm. Greedy strategy can meet the operational requirements of the problem set. On this basis, according to the requirement of "lowest operating cost", two key points that can be corrected are further analyzed from the mathematical perspective, and two main correction modules of reverse allocation and reverse merger are established by using "reverse thinking". In the original model, the modified part is searched in reverse, and the greedy strategy of the "local best" principle is still adopted for optimization, and a better optimization result is obtained. At the same time, on the basis of the original application, some application parameters were modified for comprehensive test. In addition, the method of randomly generating array was adopted to generate multiple groups of different data for further test. It was found that the "compound improved greedy strategy integrating reverse correction mechanism" was stable and effective. It can have better operation results and clearer action mechanism.

## II. APPLICAIOTN SCENARIOS

A vascular robot used by a hospital consists of a Container Boat and four operators. A Container Boat (similar to a submarine) that is powered and swims in the blood; The vessel is equipped with a human-like operator hand around the Container Boat, with a biological brain and a mechanical arm. The biological brain controls the mechanical arm to work. The operator can remove, install and replace from the Container Boats. It is necessary to combine the actual demand with the cost, operation, maintenance, damage and other requirements of the vascular robot, and calculate the number and cost of parts purchase,

maintenance and training of this type of vascular robot on a weekly basis. On the basis of meeting the treatment needs, the operating cost can be minimized. The requirements for weeks 1 to 104 are shown in Table 1.

TABLE I. NUMBER OF VASCULAR ROBOTS USED IN WEEKS 1-104

Week 1-8	11	5	4	7	16	6	5	7
Week 9-16	13	6	5	7	12	5	4	6
Week 17-24	9	5	5	11	29	21	17	20
Week 25-32	27	13	9	10	16	6	5	7
Week 33-40	11	5	5	6	12	7	7	10
Week 41-48	15	10	9	11	15	10	10	16
Week 49-56	26	21	23	36	50	45	45	49
Week 57-64	57	43	40	44	52	43	42	45
Week 65-72	52	41	39	41	48	35	34	35
Week 73-80	42	34	36	43	55	48	54	65
Week 81-88	80	70	74	85	101	89	88	90
Week 89-96	100	87	88	89	104	89	89	90
Week 97-104	106	96	94	99	109	99	96	102

The specific requirements for the composition, purchase, operation and maintenance of the vascular robot are as follows:

a) A vascular robot is composed of a Container Boat and four operators. Each vascular robot works for a week.

b) There were 13 usable Container Boats and 50 skilled operators before the start.

c) The newly purchased Container Boat needs to undergo a week of inspection and commissioning before it can be put into use. The commissioning of a Container Boat requires no additional Container Boat "guidance" and no additional cost.

d) Container boats do not need maintenance after use, can be used continuously, but not used container boats need maintenance. The maintenance fee for container boats is 10 yuan per week.

e) The newly purchased operators (new workers) need to learn the good operators (skilled workers) under the "guidance" (a certain proportion, not one to one, each skilled operator can "guide" the number of new operators is not more than 20), for one week of biological learning (training) can be put into use. For the purpose of differentiation, the newly purchased "novice" operators who need training are called trainees, and the "skilled" operators who "guide" trainees to

train are called coaches. The "trainer" must be a working operator, that is, the one who has just been removed from the patient's blood Container Boats must enter the maintenance process and cannot serve as the "trainer". The "coach" cannot work in the patient's blood Container Boats during the week, but does not need maintenance. He can work directly the next week and continue to serve as the "guidance" work. The unit price of training is not divided into "coach" and "trainee", and the unified price is 10 yuan per week.

f) The operator needs a week of maintenance to start work again, if there is no arrangement, it needs to be maintained all the time. Therefore, there are two reasons for the maintenance of the operating hand, that is, just removed from the patient's blood Container Boats and idle do not work. In order to facilitate the distinction, the idle do not work operator is called maintenance, and the maintenance of the operating hand just removed from the blood Container Boats is called maintenance. Maintenance and maintenance of the operator is priced at 5 yuan per week.

g) Purchased Container Boats and operators will arrive at the beginning of each week and immediately arrange for inspection, debugging and biology learning (training).

h) Every week, 10 percent of vascular robots are destroyed (rounded off), and for each vascular robot, a Container Boats and four operators are destroyed. The number of damaged vascular robots is calculated by multiplying the number of vascular robots that actually entered the patient's blood Container Boats in that week by a specific proportion, while other vascular robots in that week are not included in the damaged base.

i) The unit price of container boats is 200 yuan per Container Boat when the purchase quantity is not more than 5; If the one-time purchase quantity of container boats is more than 5 but not more than 10, the unit price of the part of more than 5 boats is 180 yuan per boat; When the one-time purchase quantity of container boats is more than 10, the unit price of the part over 10 Container Boats is 160 yuan/piece.

j) Manipulators preferential policy for the one-time purchase of not more than 20 when the

unit price of 100 yuan/a; When the operator purchases more than 20 but not more than 40 pieces at a time, the unit price of the part exceeding 20 pieces is 90 yuan/piece; When the operator purchases more than 40 units at a time, the unit price of that part exceeding 40 units is 80 yuan/unit.

### III. GREEDY STRATE MODELING

This application scenario studies the problem of the lowest cost, and the problem setting conditions are very rich, and the direction of the modeling results is very clear -- the lowest cost to meet the requirements. According to the requirements of the problem, the composition of the cost mainly lies in the purchase, maintenance and training, so according to the local optimal idea of greedy strategy, on a weekly basis, in principle, buy less rather than more, buy near rather than far. "Buy when you need it" is the core initial design concept of this model.

In addition, it should be noted that either the Container Boats or the operator cannot be directly put into use in the week of purchase, which requires a week of debugging or training. The training of the trainee operator also requires the "guidance" of the trainer who can work normally, and the trainer in the training stage cannot participate in the normal work of the vascular robot. Therefore, the requirements of the following week need to be prepared one week in advance. In case of reverse data update, the global data will be updated in the way of restarting data for each forward update.

#### A. Analytical Variable Relationship

Although the manipulators and Container Boats of the vascular robot in question work at the same time, the actual data of the two do not interfere with each other, so the two can be considered separately in modeling and calculation. Obviously, compared with the analysis of the operator who needs a "coach" to "guide" the "student", the variable analysis of the container boat is relatively simple, which only needs "debugging" rather than "guidance". Therefore, the variable of the container boat is analyzed first.

With week as the minimum time unit, Container Boats variables include: total number of the week, the number of the week purchased, the number of the week debugging, the number of the week able to work, the number of the week needed to work, the free amount of the week, the maintenance amount of the week.

With week as the minimum time unit, operator variables include: the total number of this week, the number of this week's purchases, the number of this week's trainees, the number of this week's coaches, the number of this week's work, the number of this week's work, the number of this week's spare maintenance, the amount of this week's maintenance.

Since the Container Boats and operator do not interfere with each other but their internal relations are complicated, their internal variable relations are analyzed separately.

### B. The Variable Relationship of Container Boats

Data changes and updates of container boats are carried out on a weekly basis. Take a certain week (not the first or last week) as an example, and its process diagram is shown in Figure 1.

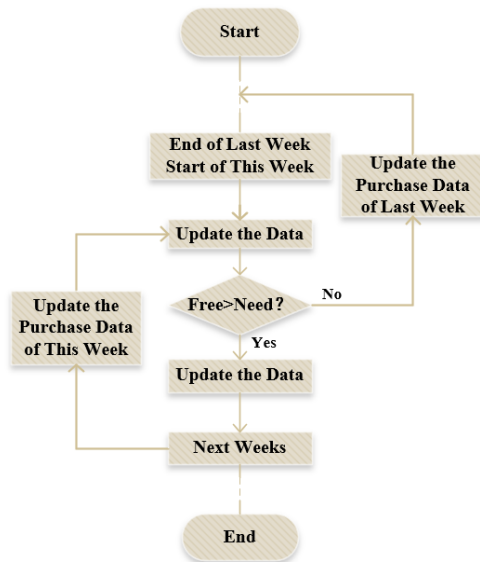


Figure. 1. Data update flow chart of Container Boats

a) The new container boat purchased this week is  $R_{buy-1}$ . Note that the purchase quantity is updated by the following weekly. According to hypothesis 1, debugging is needed in the week of purchase,

so the number of debugging in this week  $R_{study-1}$  is equal to the number of purchases in this week, as shown below:

$$R_{study-1} = R_{buy-1} \quad (1)$$

In practice, the number of debugging this week is always equal to the number of purchase this week, and this rule is always maintained in the subsequent data update. Therefore, the variable of debugging number can be removed when programming the model, but in order to conform to the process of the problem and the integrity of the analysis, the variable is kept during the analysis.

b) The total amount of container boats this week  $R_{total-1}$  is variable, which includes the total container boats of last week  $R_{total-0}$  and the newly purchased container boats of this week  $R_{buy-1}$ . At the same time, 10% of the loss in last week's work  $R_{use-0}$  should be deducted, namely:

$$R_{total-1} = R_{total-0} + R_{buy-1} - 10\% \times R_{use-0} \quad (2)$$

c) The Container Boats needed for this week  $R_{use-1}$  are obtained according to the needs of this week  $N_1$ . Each vessel robot carries one Container Boat, namely:

$$R_{use-1} = N_1 \quad (3)$$

d) Due to the commissioning of the newly purchased Container Boats, the Container Boats actually available for use this week  $R_{can-1}$  cannot include the newly purchased part, that is, only on the basis of the total amount of last week's Container Boats  $R_{total-0}$ , the work loss of last week should be subtracted. The Container Boats that can be used this week during the work of this week, include the actual work and need warranty due to spare time, namely, the following formula:

$$R_{can-1} = R_{total-1} - R_{buy-1} = R_{total-0} - 10\% \times R_{use-0} \quad (4)$$

e) If there are enough container boats available for use this week ( $R_{can-1} \geq R_{use-1}$ ), then there will be free container boats  $R_{free-1}$  that need warranty.

Variable  $R_{keep-1}$  is established for the container boats that need warranty, and the relationship is as follows:

$$R_{free-1} = R_{can-1} - R_{use-1} \quad (5)$$

$$R_{keep-1} = R_{free-1} \quad (6)$$

Similar to the amount of debugging this week equals the amount of purchase this week, the number of free container boats and the number of container boats that need warranty is always the same, which can be deleted in the actual programming to build the model without affecting the whole. Two variables were retained in the analysis process, not only to conform to the process and analysis integrity of the problem set, but also because in the analysis of the operator's model, the free amount and the warranty amount were not equal, so as to compare and use them in the in-depth analysis of the subsequent data. This warranty variable can be removed during programming modeling, as long as it is matched in subsequent analysis.

f) If there are not enough working container boats for the week ( $R_{can-1} < R_{use-1}$ ), you will need to purchase a supplement. Since the newly purchased Container Boats cannot be used directly, but needs to be debugger for a week, the demand gap of this week needs to be purchased last week, so that it can be used this week after training. The relationship is as follows:

$$R_{buy-0}(new) = R_{buy-0}(old) + R_{use-1} - R_{can-1} \quad (7)$$

Note the following:

If there was demand for purchases last week, then this week's demand should be added to last week's demand. But in fact, the change in last week's purchase quantity is only related to this week's demand gap, and next week's demand gap will only affect this week's purchase quantity but not last week's purchase quantity, so it will only change once at most. Therefore, direct assignment does not affect the result. But in both analysis models and programming modeling, it is important to keep in mind that the purchase amount itself should be an additive amount.

This week's demand gap caused the change in the purchase volume of last week, which indirectly affected the total number of container boats last week, the number of container boats debugging last week, the total number of container boats this week, the number of available services this week and other data. However, it did not affect the previous weeks, so relevant data can be updated. However, similar to the way that the purchase quantity is accumulated rather than assigned, it should be noted in both the analysis model and the actual programming modeling that for the case of reverse order affecting variables, each change should be sorted out from the beginning, so as to keep the data updated for the overall coordination and once the purchase quantity changes, the data should be updated from the first week.

g) After the completion of the above operation, the operation of the next week is the same as that of this week. Once the container boats that can be used cannot meet the needs, it is also necessary to increase this part of the gap on the basis of the previous week's purchase, and then update the relevant data.

h) End up buying policy on price adjustment strategy approximation as piecewise function, purchase quantity and purchase price can present certain functional relation, using  $f(x)$  said gunner purchase policy function:

$$f(x) = \begin{cases} 200x, & x \leq 5 \\ 100+180x, & 5 < x \leq 10 \\ 300+160x, & x > 10 \end{cases} \quad (8)$$

### C. The Variable Relationship of the Operator Hands

Compared with the updated data of Container Boats, the operator data needs more consideration of the roles of "trainee" and "coach". In addition to the difference between the operators they can provide and those who need to work, the number of "coaches" should also be considered. Take a week (not the first or last week) as an example, and its flow diagram is shown in Figure 2.

The specific affectation is as follows:

a) The new operator purchased this week is  $C_{buy-1}$ . Note that this purchase quantity is also updated by the following weekly updates. According to the assumption, the week of purchase is the training, so the number of trainees this week  $C_{study-1}$  is equal to the number of purchases this week:

$$C_{study-1} = C_{buy-1} \quad (9)$$

In practice, the number of debugging this week is always equal to the number of purchase this week, and this rule is always maintained in the subsequent data update. Therefore, the variable of the number of students  $C_{study-1}$  can be removed at the early stage of programming model building. However, in addition to conforming to the process of the problem set and the integrity of the analysis, we also found in the subsequent analysis that, the number of students is a variable that contributes to the optimization of the analysis and model, so it is retained during the analysis and modeling process.

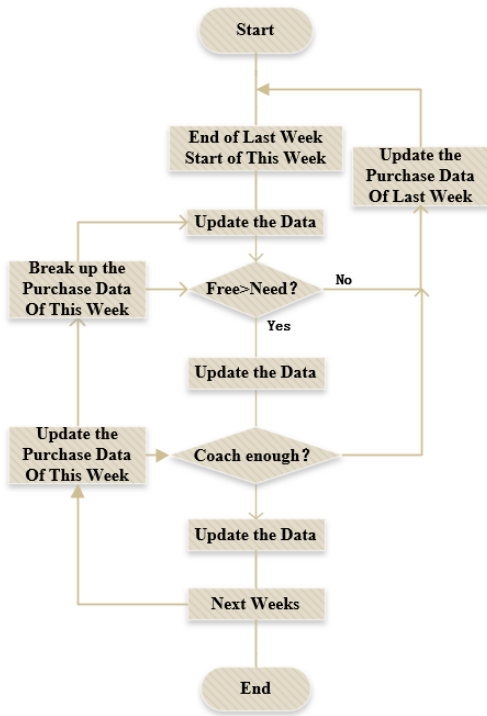


Figure. 2. Data update flow chart of the Operator Hands

It should also be noted that the number of trainers  $C_{coach-1}$  should be configured for the trainee training of the operator. When the number of trainees is 0, no coach is needed for that week; When the number of trainees is not 0, coaches are

needed in the week, and the number of coaches should be calculated according to the ratio. The matching assumption of the test environment is that one coach can guide 20 trainees at most. The conversion relationship is as follows:

b) The total number of operators this week is changed. Variable  $C_{total-1}$  is established, which includes the total number of operators last week  $C_{total-0}$  and the newly purchased operators this week  $C_{buy-1}$ . Meanwhile, the loss in use last week is subtracted, namely:

$$C_{total-1} = C_{total-0} + C_{buy-1} - 10\% \times C_{use-0} \quad (10)$$

c) The manipulators that need to be used this week  $C_{use-1}$  (to actually work in the patient's blood vessels) are obtained according to the demand of this week  $N$ . Each vascular robot carries 4 manipulators, namely:

$$C_{use-1} = 4N \quad (11)$$

d) The operators who participated in the actual work last week  $C_{use-0}$  need to maintain this week, so this part needs to set up a separate variable  $C_{upkeep-1}$  to record for analysis and modeling. The lost part does not need to be maintained:

$$C_{upkeep-1} = 90\% \times C_{use-0} \quad (12)$$

Since the newly purchased operator needs training, the actual standby operator that can be used this week  $C_{can-1}$  does not include the newly purchased part, but also excludes the maintenance part of this week. In essence, it is the total number of last week minus the number of last week's work:

$$C_{can-1} = C_{total-1} - C_{buy-1} - C_{upkeep-1} = C_{total-0} - C_{use-0} \quad (13)$$

There are three parts of the task this week: the actual working operator  $C_{use-1}$ , the trainer operator  $C_{coach-1}$  and the spare maintenance operator  $C_{free-1}$ .

e) First of all, consider whether the operator can meet the needs of the actual work. If the operator cannot provide the required hand, i.e.  $C_{use-1} > C_{can-1}$ , at this time, the demand gap of this part shall be fed back to last week, and the

purchase quantity of last week shall be accumulated:

$$C_{buy-0}(new) = C_{buy-0}(old) + C_{use-1} - C_{can-1} \quad (14)$$

At the same time, it should be noted that changes in the number of purchases may also lead to changes in the number of coaches and idle numbers of last week, or even new demand gaps (demand gaps will be generated due to the increase in the number of coaches and the shortage of idle numbers, which will be detailed in the subsequent analysis). Therefore, once the reverse demand update occurs, you need to update the data from scratch.

f) If the number of operators available this week is sufficient for regular work ( $C_{use-1} \leq C_{can-1}$ ), further consideration needs to be given to the number of trainers to train the trainees  $C_{coach-1}$ .

g) If the demand of the number of coaches cannot be met, i.e. ( $C_{can-1} - C_{use-1} < C_{coach-1}$ ), then the demand gap of the number of coaches is generated. Similar to the previous step, the demand gap of this part needs to be reversely updated into the purchase data of last week:

$$C_{buy-0}(new) = C_{buy-0}(old) + C_{coach-1} - (C_{can-1} - C_{use-1}) \quad (15)$$

Similarly, after the data is updated backwards, all the data is updated from scratch.

h) If the demand for the number of trainers can also be met, i.e., ( $C_{can-1} - C_{use-1} \geq C_{coach-1}$ ), then the number of free operators this week is:

$$C_{free-1} = C_{can-1} - C_{use-1} - C_{coach-1} \quad (16)$$

Note that the number of free operators in this part needs maintenance work, which should be distinguished from the operators in the maintenance part and also from the algorithm of the container boat.

i) After the above operations are completed, the operation of the next week is the same as that of this week. Once the available operators cannot meet the needs, it is also necessary to increase this

part of the gap on the basis of the previous week's purchase, and then update the relevant data.

j) Purchasing policy of price adjustment strategy approximation as piecewise function, purchase quantity and purchase price can present certain functional relation, use of  $g(x)$  operator to buy policy function:

$$g(x) = \begin{cases} 100x, & x \leq 20 \\ 200 + 90x, & 20 < x \leq 40 \\ 600 + 80x, & x > 40 \end{cases} \quad (17)$$

#### D. Programming

The python language and Jupyter platform are selected for the compilation. The logic is clear, which can be edited in blocks, and the data can be further analyzed on the basis of running.

##### 1) Initialization

In order to facilitate global analysis and derivation, data storage and extraction, and subsequent analysis, one-dimensional array is used to store weekly data changes of each variable in python language environment (matrix storage can also be used). In addition, the package guide and preset data import are also completed in this step.

##### 2) Write the model by parts

According to the separation and analysis of variables, the Container Boats and operator calculated separately and did not affect each other, so the two models were coded separately.

For the Container Boats part, under the preset premise and in accordance with the sequence of variable analysis, the value of this week's total amount, this week's available workload and this week's needed workload are updated respectively to judge the demand gap. If there is no demand gap, update the idle quantity and maintenance quantity and enter the next week calculation; If there is a shortfall, reverse add it to last week's purchases and update the data from the first week.

The operator part can be regarded as an extension of the Container Boat part. Modeling was carried out according to the sequence of variable analysis, and the total amount of this

week, the maintenance amount of this week, the available amount of this week, the workload of this week, the number of students of this week and the number of coaches of this week were updated respectively, and then judged step by step. If there is a gap between the available usage and the required workload, the reverse update purchase data is added to last week's purchase volume, and the data is updated from scratch; If the available amount can meet the workload, then judge whether it meets the demand of the number of trainers. If it does not meet the demand, reverse update the demand gap to the purchase amount of last week and update other data from scratch. If it meets the demand, update the free maintenance amount of this week and enter into the calculation of the next week.

After the monomer model of the two parts is established, the cycle structure corresponding to the number of weeks is loaded respectively and the function is packaged. The return value is each variable array updated respectively. See appendix for model code implementation.

3) *Control the operation part*

The individual parts of the operator and container boat are packaged as functions, and the running part simply calls the function, noting that the result of the function needs to be inherited with a preset number of arrays corresponding to the number of arrays each returns.

4) *Calculate operation cost module*

The calculation of operation cost adopts the summative method, setting the operator's purchase, operator's maintenance, operator's training, container boat's purchase, and container boat's maintenance. The cycle needs weekly, weekly sum, after the results are obtained, combined calculation is made according to the needs, and the required results are output.

5) *Other results output*

After the execution, you can call to read the finished data as needed.

E. *The Results of the Model*

The output results such as operator purchase array, container boat purchase array and operation and maintenance cost are summarized in Table 2.

TABLE II. THE OUTPUT OF THE SIMPLE GREEDY STRATEGY MODEL

The output of the simple greedy strategy model								
The purchase program of Container boat								
Week 1-8	0	0	0	5	0	0	0	1
Week 9-16	0	0	0	2	0	0	0	0
Week 17-24	0	0	2	19	0	0	0	7
Week 25-32	0	0	0	0	0	0	0	0
Week 33-40	0	0	0	0	0	0	0	4
Week 41-48	0	0	0	5	0	0	5	12
Week 49-56	0	2	15	18	0	4	18	13
Week 57-64	0	0	1	12	0	0	6	11
Week 65-72	0	0	2	11	0	0	0	10
Week 73-80	0	1	11	16	0	10	16	21
Week 81-88	0	9	18	24	0	6	11	19
Week 89-96	0	7	10	24	0	4	10	25
Week 97-104	1	8	14	20	1	7	16	0
Amount	484			Cost			91600	
The purchase program of operator hands								
Week 1-8	14	0	0	36	0	0	0	1
Week 9-16	0	0	0	8	0	0	0	0
Week 17-24	0	0	13	98	44	0	0	10
Week 25-32	0	0	0	0	0	0	0	0
Week 33-40	0	0	0	0	0	0	0	0
Week 41-48	0	0	0	0	0	0	18	68
Week 49-56	26	0	68	117	50	0	32	66
Week 57-64	0	0	0	36	10	0	9	57
Week 65-72	2	0	0	34	0	0	0	16
Week 73-80	10	0	47	89	37	19	89	126
Week 81-88	41	11	90	138	45	0	31	83
Week 89-96	24	0	35	99	36	0	23	104
Week 97-104	60	0	43	98	40	0	39	0
Amount	2290			Cost			311505	
Total Cost				403105				

IV. REVERSE CRRECTION MCHANISM

According to the greedy strategy thinking mode, according to the way of "need to buy", can perfectly solve the problem of "demand", but the "lowest operating cost" can be optimized, at this time need to reverse correction.

A. *Backward Distribution*

1) *Analysis of source*

As shown in Figure 3, it is the operator purchase demand matrix of the simple greedy algorithm of week 1-104. First, two consecutive



non-zero numbers in the result -- 18 and 68 are taken into consideration. It is noted that the free number of the next week corresponding to 68 is not 0, indicating that at least two more operators are purchased here, because the subsequent operator vacancies are directly added to the purchase volume of this week. This week created a gap in the number of coaches and continued to pass on purchases, but there is room for improvement. For the purposes of analysis, call purchase 18 last week and purchase 68 this week.

Simple - Operator Hands - Buy:																
0	14	0	0	36	0	0	0	0	0	0	8	0	0	0	0	0
0	13	98	44	0	0	10	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	18	68	26	0	68	117
0	32	66	0	0	0	36	10	0	9	57	2	0	0	34	0	0
16	10	0	47	89	37	19	89	126	41	11	90	138	45	0	31	83
0	35	99	36	0	23	104	60	0	43	98	40	0	39	0	1	0

Simple - Operator Hands - Free:																
0	5	0	24	12	0	1	39	32	0	1	28	21	0	5	32	26
6	17	0	2	5	41	36	0	20	81	88	56	68	106	100	74	79
93	63	57	72	57	22	18	36	28	0	0	13	0	2	4	2	0
6	4	0	4	8	53	30	3	0	18	0	2	3	34	16	0	8
30	0	1	5	0	2	3	0	0	4	1	0	0	4	2	12	0
5	11	0	3	5	21	0	3	6	9	0	3	5	11	0	1	0

Figure 3. The purchase and idle situation of the Operator Hands in the Simple Mode

Combined with the background analysis box selected this part of the data source. Due to the demand gap next week, the purchase volume of this week is 68, that is, there are 68 students and 4 coaches this week. However, there is also a demand gap this week, which makes the purchase volume of last week is 18, but there is no purchase volume last week, so there is no demand gap last week, and even there are spare operators.

It can be assumed that if the purchase volume of 68 students this week is transferred to last week, and there are enough spare numbers to provide coaches last week, then the demand of this week will be reduced by 4, and the planned purchase volume of last week will be reduced from 13 to 9. Although the purchase in advance increases the maintenance cost of 68 operators for a free week (5 yuan per operator per week) by 340 yuan in total, the purchase quantity of 4 operators reduced (100 yuan per operator) is saved by 400 yuan, so the comprehensive calculation saves some money.

On this basis, if 4 of the 68 newly purchased operators are purchased and trained last week in advance, and the remaining 64 are purchased this week, then the 4 operators purchased in advance can be used for training this week, which can save the maintenance cost of 64 operators for a week totaling 320 yuan.

Based on the above analysis, the optimization method based on this example can save 380 yuan locally. The essence is to advance the purchase of some of the operators originally needed this week to the last week. The amount of this demand gap is "self-training", and there is no need to supplement the number of trainers.

### 2) Optimization Mechanism

After analyzing the reasons for this problem, it is found that there are too many operators purchased and not enough coaches, which leads to the need to increase the purchase amount of coach operators.

Return to the algorithm process of analyzing the operator, using the method of comparing demand step by step. If there is a demand gap next week, directly increase the purchase volume this week, and then update the data from the first week. When the data is updated this week, there may be such a situation that although the work required by the available supply is sufficient, the number of coaches will increase due to the increase in the purchase amount, and the number of coaches will not be enough in the next step, so the demand gap of this week will be accumulated to the last week. The solution is to buy quantity forward in advance.

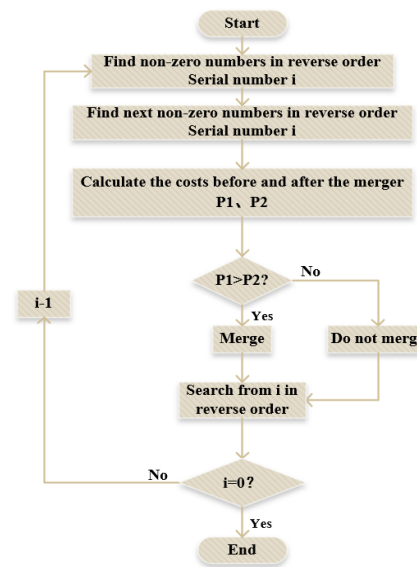


Figure 4. Update flow chart of the Operator Hands after the first optimization

Based on the hypothesis, it can be seen that the changes of the purchase quantity and the number

of trainers this week will not lead to the changes of the available operators and the required operators this week. In addition, as the purchase quantity is advanced, the available number this week will only increase, so there will be no additional demand gap. In the previous example, when the run to this week cannot provide enough coaches, if one purchase of this week is brought forward to last week, the number of coaches available for this week increases by 1, the number of students that can be accommodated increases by 20, and the actual number of students decreases by 1. Since each time of the forward accumulation of demand gap in this model algorithm, an operation of re-running the update array from the first week will be added, so there is no need to set the purchase quantity of forward specifically, nor to consider whether the number of coaches in the previous week is sufficient, because these problems can be found during the re-running and solved with this improved idea. The flow chart, shown in Figure 4, only needs to be optimized when there are not enough coaches available.

3) Effect of Optimization

After the optimized model is loaded with the demand sequence, the purchase and idle sequence of the operator is output again, as shown in Figure 5. It can be found that the local problems analyzed above have been solved. Throughout the series, the other parts of the "buy one week and then idle one week" scenario have also been resolved, and have been "minimized" from a purchase volume perspective.

The final result after optimization is shown in Table 3. To compare the purchase plan of the operator, the red dot in Figure 6 represents the situation before optimization, and the blue dot represents the situation after optimization. It can be found that the optimization module does have an adjustment effect on the purchase plan. From a global perspective, due to the requirement of the overall number of operators needed, the sequence of weekly demand quantity is contingent, and a certain proportion of losses in the operation process, the total number of operators needed to be purchased is the same in the end. However, the operation and maintenance cost of operators is reduced by 495 yuan after optimization, which is

because the "overbought" part in the early stage is transferred to the later stage. It reduces the maintenance cost. The optimizer does have an overall tuning effect along the way.

```

First Optimization - Operator Hands - Buy:
[ 0 14 0 0 0 36 0 0 0 1 0 0 0 8 0 0 0 0 0 0
 0 13 96 41 0 0 15 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 36 62 0 0 0 37 13 0 9 55 1 0 0 37 0 0 0 0
 16 9 0 48 87 37 21 89 122 44 12 90 134 47 0 33 80 22
 0 40 96 34 0 28 101 57 0 49 95 38 0 44 0]

First Optimization - Operator Hands - Free:
[ 0 5 0 24 12 0 1 39 32 0 1 28 21 0 5 32 26 4
 6 17 0 0 0 36 31 0 20 81 88 56 68 106 100 74 79 99
 93 63 57 72 57 22 18 36 28 0 0 13 0 0 0 0 0 0
 0 0 0 0 4 49 26 0 0 18 0 0 0 31 13 0 8 45
 30 0 0 4 0 0 0 0 0 0 0 0 0 0 0 10 1 0
 0 6 0 0 0 16 0 0 0 3 0 0 0 0 5 0]
    
```

Figure 5. The purchase and idle situation of the Operator Hands in the First Optimization Mode

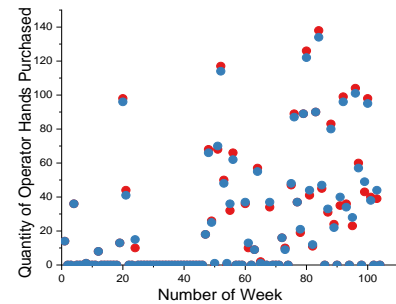


Figure 6. Comparison of purchasing plans of Operator Hands before and after the first optimization

TABLE III. THE OUTPUT OF THE FIRST OPTIMIZATION

The output of the first optimization							
The purchase program of Container boat							
Week 1-8	0	0	0	5	0	0	1
Week 9-16	0	0	0	2	0	0	0
Week 17-24	0	0	2	19	0	0	7
Week 25-32	0	0	0	0	0	0	0
Week 33-40	0	0	0	0	0	0	4
Week 41-48	0	0	0	5	0	0	12
Week 49-56	0	2	15	18	0	4	13
Week 57-64	0	0	1	12	0	0	11
Week 65-72	0	0	2	11	0	0	10
Week 73-80	0	1	11	16	0	10	21
Week 81-88	0	9	18	24	0	6	19
Week 89-96	0	7	10	24	0	4	25
Week 97-104	1	8	14	20	1	7	16
Amount	484			Cost			91600
The purchase program of operator hands							
Week 1-8	14	0	0	36	0	0	1
Week 9-16	0	0	0	8	0	0	0
Week 17-24	0	0	13	96	41	0	15
Week 25-32	0	0	0	0	0	0	0
Week 33-40	0	0	0	0	0	0	0
Week 41-48	0	0	0	0	0	0	18
Week 49-56	25	1	70	114	48	1	62
Week 57-64	0	0	0	37	13	0	55
Week 65-72	1	0	0	37	0	0	16
Week 73-80	9	0	48	87	37	21	122
Week 81-88	44	12	90	134	47	0	80
Week 89-96	22	0	40	96	34	0	101
Week 97-104	57	0	49	95	38	0	44
Amount	2290			Cost			311175
Total Cost				402775			

**B. Backward Combination**

In this model, since the purchase quantity must be an integer, the relationship between the purchase quantity and price should be discrete, but in order to facilitate analysis, it is treated as a function. Preferential policies can reduce the cost compared with the original purchase price.

The main variable related to the preferential purchase plan is the purchase quantity. Obviously, it is necessary to consider "the more you buy at one time, the better it will be". You can try to concentrate the purchase quantity of the neighboring weeks into the same week, whether it will be cheaper. In the context of the application of this study, it is not enough to consider the centralized purchase, but the following points should also be noted:

- a) Can only be the back of the weekly purchase volume forward, can not be merged in the future, otherwise it can not meet the demand for use;
- b) Due to the advance purchase of more warranty costs should be considered;
- c) For the operator, it is also necessary to consider whether the number of coaches is insufficient if the number of students is increased if the operator is purchased in advance.

*1) Analysis of source*

According to the preferential purchase method, it is obvious that the purchase plan should be selected forward and combined, and the situation of insufficient operator coaches should be considered at the same time. It is analyzed and optimized step by step here.

*a) Purchase volume consolidation*

Due to the existence of the purchase discount, it is ideal to get the discount through the combined purchase. However, it should be noted that the excess quantity purchased several weeks in advance will lead to the maintenance cost of this part in the waiting process. Therefore, the question of whether it is cost-effective to consolidate purchases forward needs to be considered.

Let the purchase quantity of one week (No. N) be a, the purchase quantity of the following week

(No. N+s) be b, the interval between two weeks be s, and the combined quantity be i. The value range of the combined quantity is 0 (no combined purchase) to b (all combined purchase). It is necessary to consider the increased warranty cost due to the part purchased in advance after the forward consolidation purchase. The calculation method of the cost is as follows:

$$R_{pay} = f(a+i) + f(b-i) + 10is \tag{18}$$

$$C_{pay} = g(a+i) + g(b-i) + 5is \tag{19}$$

Since the function is segmented and there are many independent variables, it is difficult to find out the law directly from the mathematical point of view. Therefore, it is considered to test by simulation. In the selected two weeks of respective purchase volume, simulate the results of different merger volumes under different intervals between two weeks. The original purchase parameters of the two weeks involved in the calculation are shown in Table 4.

TABLE IV. ORIGINAL PURCHASE PARAMETER SETTINGS

The number of container boats									
Week. N	2			7			15		
Week. N+s	2	7	15	2	7	15	2	7	15
The number of operator hands									
Week. N	10			30			50		
Week. N+s	10	30	50	10	30	50	10	30	50

The test results are presented in Figure 7, simulating the combined purchase of container boats. Each small icon is the comprehensive cost from no combined purchase to all combined purchase (with increasing abscissa) under a certain value combination of A and b, simulating different intervals (differentiated by different color lines in the same graph). Figure 8 shows the operator's combined purchase simulated in the same way.

From the trend analysis in the figure, it can be found that the Container Boats and the operator have the same change rules. With the change of the combined volume, when the purchase volume is small for two weeks, the required cost of the combined purchase will increase, as shown in FIG. 8 (a) and 9 (a). However, when the purchase

quantity is relatively large, the cost trend of combined purchase increases first and then decreases. In some cases, the cost of all purchase is even less than that of non-combined purchase, as shown in Figure. 8 (i) and Figure. 9 (i).

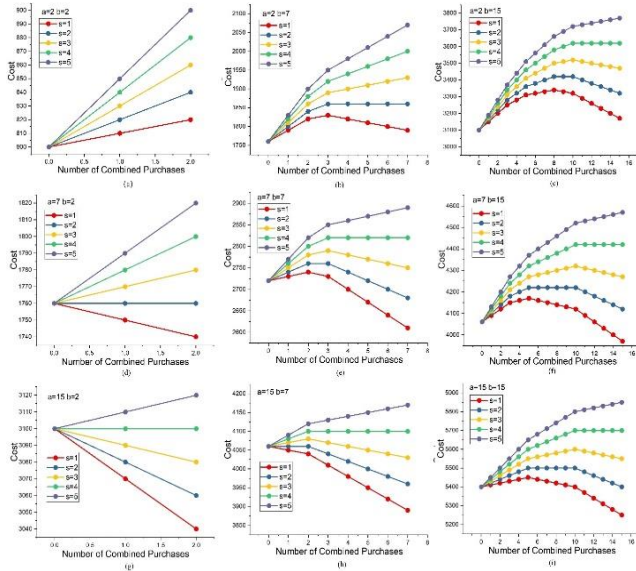


Figure. 7. Simulation of the combined purchase cost of Container Boats

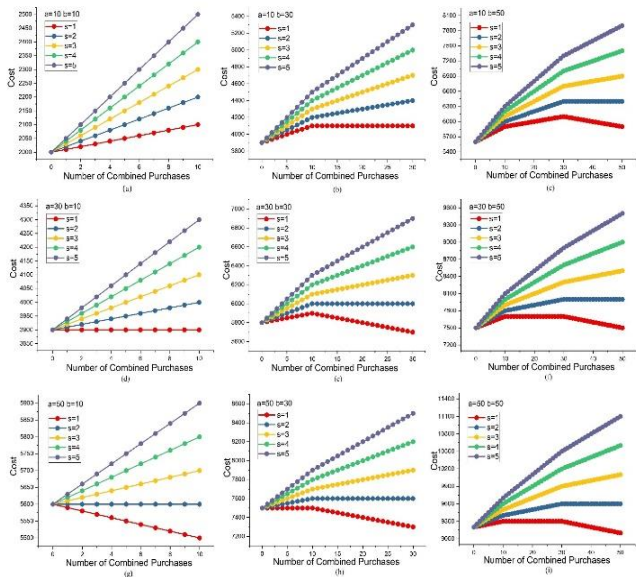


Figure. 8. Simulation of the combined purchase cost of Operator Hands

Then, when the minimum value needs to be selected, only one point between the starting point (no merge) and the end point (all merge) can be selected. Then only the two points need to be compared, without considering the middle case. A very important conclusion can be reached, that is,

without considering the specific amount of consolidation, you only need to compare the case of no merger and the case of all merger, and choose which one costs less.

The operator calculates the cost without merging and the cost algorithm of all merging as follows:

$$C_{pay1} = g(C_a) + g(C_b) \tag{20}$$

$$C_{pay2} = g(C_a + C_b) + 5C_a s \tag{21}$$

The cost algorithm for calculating the cost without merging and the cost algorithm for all merging is as follows:

$$R_{pay1} = f(R_a) + f(R_b) \tag{22}$$

$$R_{pay2} = f(R_a + R_b) + 10R_a s \tag{23}$$

Since the merge is from the back to the front, in the modeling, the reverse order is adopted to determine whether the merge is necessary, and the merging may occur after a certain week and then forward.

*b) Trainer number requirements and limitations of the operator*

For container boats, the change in purchase volume does not affect the actual regular operation except for the change in the regular array. It is only necessary to update the purchase sequence by adding the judgment function. However, whether the combined purchase of operators can provide enough operators needs to be considered. If there are not enough free operators, then the maximum number of trainee operators that can actually be merged is the maximum number of trainee operators that the free operators can support. Compare the cost of merging with the cost of not merging and choose the cheaper option.

Here, there are two options for the operator's operation: one operation is completed when the cost is combined and the number of coaches is counted at the same time; Performing the merge operation first, and then considering the fallback if there are not enough coaches, will increase the

amount of computation. The second algorithm is used in this study.

The operation is performed between three weeks, and the week numbers are named  $l$ ,  $m$ , and  $n$  in increasing order. According to algorithm 2, it is found that the merger is more cost-effective after performing the merger calculation from week  $n$  to week  $m$ , and then performing the merger calculation with week  $l$ , it is found that the merger can be performed again, and because there are enough free operators in week  $l$ , it is enough to meet the demand of purchase quantity. According to algorithm 1, when the number of coaches is moving from  $n$  to  $m$  weeks, there is a situation that the number of coaches is insufficient to backtrack, so the number of merges cannot reach the optimal level or even do not merge, but in fact, merging again to  $l$  weeks can solve this problem. So choose the second algorithm. Algorithm 1 is also logically feasible, but requires multiple iterations until the purchase sequence is no longer changing.

After performing the merge operation, perform the hand part rollback. Since not enough coaches can be provided, at least a certain number of coaches can be rolled back so that the number of coaches is just enough, or more can be rolled back. Combined with the analysis of the simulation results of the merging module, the forward calculation of the minimum rollback value (the more the rollback, the smaller the value of the abscissa in Figure. 10, 11, 12 and 13) shows that the value of the minimum cost in the rollback process must be a point between the starting state (minimum rollback) and the end state (all rollback). Therefore, You only need to consider which is less expensive between a minimum rollback (coach saturation) and a maximum rollback (total rollback).

When the number of free operators in a certain week is negative, the data of the current purchase number and the number of coaches in that week can be read at the same time to calculate the maximum number of coaches that can actually provide training:

$$coach\_total = coach + free \quad (24)$$

Each coach can undertake a maximum of 20 trainees, so the maximum number of purchases that can be undertaken is:

$$buy_{max} = 20coach\_total \quad (25)$$

The minimum number of purchases that need to be rolled back is:

$$buy_{back} = buy_{new} - buy_{max} \quad (26)$$

Read the original purchase number of the week in the original purchase data backup, and the original combined total is:

$$buy_2 = buy_{new} - buy_1 \quad (27)$$

Look for the weekly number of the nearest purchase, which is the weekly number of  $buy_2$ , and record the difference between the two weekly numbers for subsequent calculation. Note that you can't directly read the number of purchases from the source of the merge in the original array, because you can't rule out sequential merges near the source of the merge, but the rollback is gradual.

From this, we can calculate the cost of the total rollback  $pay_{back-all}$  (no consolidation in the original case) and the cost of the minimum rollback value  $pay_{back-min}$  (the original case, keep  $buy_{back}$ , only transfer  $buy_2 - buy_{back}$ ):

$$pay_{back-all} = g(buy_1) + g(buy_2) \quad (28)$$

$$pay_{back-min} = g(buy_{max}) + g(buy_{back}) + 5s(buy_2 - buy_{back}) \quad (29)$$

Compare the two data and choose a rollback method that costs less.

## 2) Optimization Mechanism

### a) Merge modules

To facilitate subsequent analysis and comparison, copy and reserve the original data before performing this operation.

According to the analysis, the module scans the purchase sequence in reverse order, and the specific process is shown in Figure 9.



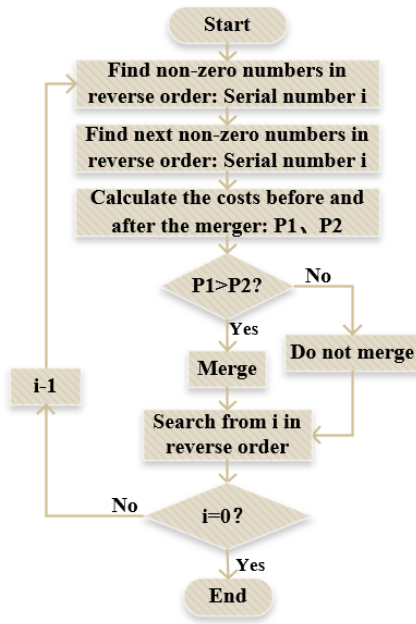


Figure. 9. Flow chart of merge purchase algorithm

Scan the purchase quantity sequence in reverse order scanning, find the first non-zero purchase quantity (let the ordinal number read is  $i$ ), then look forward to the most recent non-zero purchase quantity (let the ordinal number read is  $j$ ), and judge the cost before and after the merger. If the cost after the merger is higher and unchanged or higher, the merger will not be performed. If the cost is reduced after the merger, all the purchases in week  $i$  will be added up to week  $j$ , and the purchases in week  $i$  will be zero. Then continue scanning forward from the  $i-1$  ordinal and repeat the above procedure.

After the merge part is complete, the purchase sequence has been updated, and updates of the other arrays have been performed since the first week.

*b) Return module of the operator*

Since the combination module does not consider the limitation of the number of coaches, all or part of the data needs to be rolled back if the number of free operators is negative after the update. The operation process is shown in Figure 10.

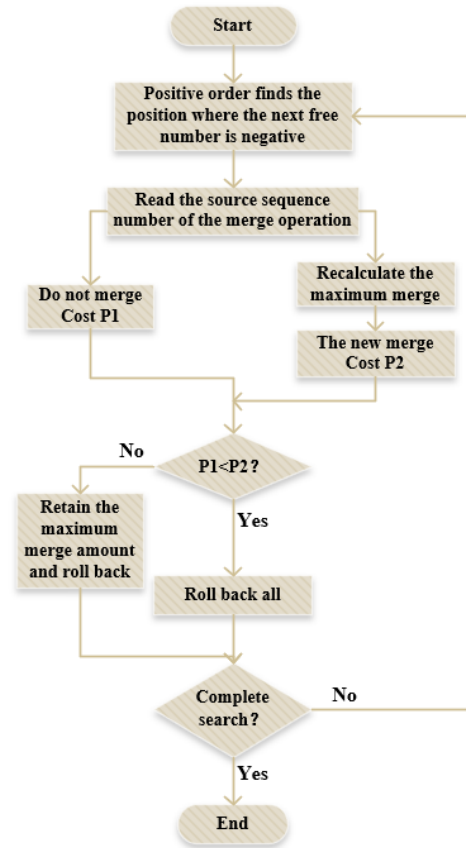


Figure. 10. Flowchart of the rollback algorithm

The positive sequence scans the array module that records the number of free operators every week, finds the first number less than zero (let the read ordinal number be  $i$ ), scans the original purchase data array before merging on this basis, and finds the nearest non-zero number (let the read ordinal number be  $j$ ), which is the ordinal number of merging the purchase number to  $i$ . The corresponding data is retrieved, and the cost is calculated in two ways: full rollback and only excess rollback, and the lower cost is selected to update the number of purchases. Since this operation must result in a change in the number of purchases, all the array data corresponding to the two ordinals must be updated with each rollback.

After the purchase sequence is updated, the other arrays are updated according to the rules from the first week.

*3) Effect of Optimization*

The running results after loading the reverse merge module are shown in Table 5.

TABLE V. THE OUTPUT OF THE SECOND OPTIMIZATION

The output of the second optimization								
The purchase program of Container boats								
Week 1-8	0	0	0	5	0	0	0	1
Week 9-16	0	0	0	2	0	0	0	0
Week 17-24	0	0	2	19	0	0	0	7
Week 25-32	0	0	0	0	0	0	0	0
Week 33-40	0	0	0	0	0	0	0	4
Week 41-48	0	0	0	5	0	0	5	12
Week 49-56	0	2	15	18	0	4	8	13
Week 57-64	0	0	1	12	0	0	6	11
Week 65-72	0	0	2	11	0	0	0	10
Week 73-80	0	1	11	16	0	10	16	21
Week 81-88	0	9	18	24	0	6	11	19
Week 89-96	0	7	10	24	0	4	10	25
Week 97-104	1	8	14	20	1	7	16	0
Amount	484			Cost			90140	
The purchase program of operator hands								
Week 1-8	14	0	0	36	0	0	0	1
Week 9-16	0	0	0	8	0	0	0	0
Week 17-24	0	0	13	96	41	0	0	15
Week 25-32	0	0	0	0	0	0	0	0
Week 33-40	0	0	0	0	0	0	0	0
Week 41-48	0	0	0	0	0	0	18	66
Week 49-56	25	1	70	114	48	1	36	62
Week 57-64	0	0	0	37	13	0	9	55
Week 65-72	1	0	0	37	0	0	0	16
Week 73-80	9	0	48	87	37	21	89	122
Week 81-88	44	12	90	134	47	0	33	80
Week 89-96	22	0	40	96	34	0	28	101
Week 97-104	57	0	49	95	38	0	44	0
Amount	2290			Cost			310525	
Total Cost							400665	

The simple greedy algorithm, the first optimization, the second optimization, and the three operation effects were compared and verified, mainly comparing the operator's purchase sequence and operation and maintenance cost, the container boat's purchase sequence and operation and maintenance cost, and the overall operation and maintenance total cost. The results are shown in Table 6. For the convenience of comparison, set the saving rate(SR):

$$\eta = \frac{Sum_{Before} - Sum_{After}}{Sum_{Before}} \times 100\% \quad (30)$$

TABLE VI. COMPARISON OF THE EFFECT BEFORE AND AFTER OPTIMIZATION

	OP-N	OP-C	CB-N	CB-C	TOTAL
Simple	2290	311505	484	91600	403105
Optimization 1	2290	311175	484	91600	402775
Optimization 2	2290	310525	484	90140	400665
SR total	0.6053%	SR 1	0.08187%	SR 2	0.5239%

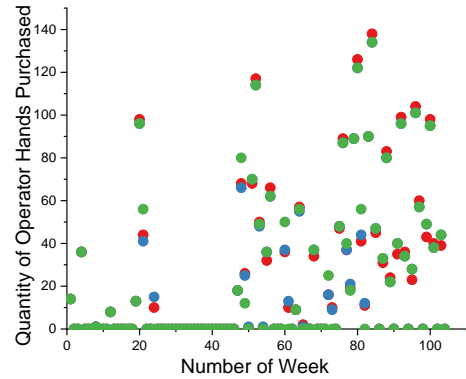


Figure 11. The purchase of Operator Hands before and after optimization

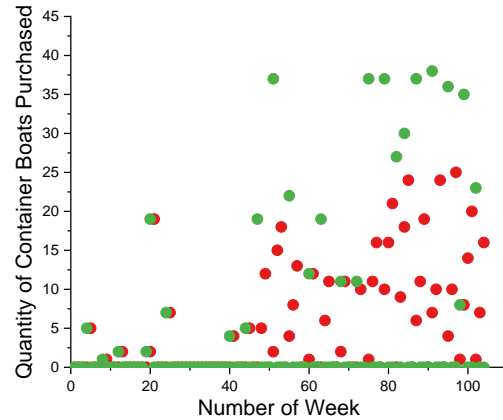


Figure 12. The purchase of Container Boats before and after optimization

Through data comparison, it is found that the two optimizations are effective in saving operation cost requirements. Figure 11 shows the situation of the operator buying the scheme in the two optimizations, where red represents before optimization, blue represents after the first optimization, and green represents after the second optimization. Figure 12 shows the purchase plan of container boats before optimization and after overall optimization. The container boats were not involved in the first optimization, so the purchase plan of container boats after the first optimization is not presented.

### C. Summary of optimization

Considering the minimum purchase amount, the reverse distribution mechanism is established to move the demand gap forward to reduce unnecessary purchase amount. Although the total purchase amount does not change under some special demand sequences, the maintenance waste

caused by "buying too early" is reduced. This module can achieve a saving rate of 0.08187%.

As for the preferential purchase scheme, the analysis found that "purchase in advance" could save more costs in some cases, so the reverse merger mechanism was designed to analyze and measure the costs of "merge" and "no merge" operation and choose a more favorable way. Among them, direct merging may lead to the operator "insufficient number of coaches", so a rollback module is designed. In the "coach oversaturation" section, the cost of the minimum rollback and the total rollback was calculated and the best choice was made. Compared with before loading this module, the saving rate of 0.5239% can be obtained. The two optimization modules work together to achieve a saving rate of 0.6053%.

Through the comparison of the chart, it can be found that the optimization module does have an adjustment effect on the purchase plan. Although the total number of purchases has not changed, the maintenance cost has been reduced due to the adjustment of the purchase plan. In other words, the purchase amount in the early stage has been adjusted to the purchase in the later stage, but the weekly work demand has not been affected. It should be noted that the total number of purchases does not change, which is also related to the demand sequence of the application scenario. In the subsequent simulation tests, the total number of purchases changes.

V. EXTENDED SIMULATION TEST

In order to more strongly verify the effect of the reverse correction mechanism, more array validation tests are performed in the original application context. Two test modes are planned: modifying application parameters and changing the requirements matrix.

A. Modifying Application Parameters

Under the application background of the study in this paper, although the addition of correction mechanism has a certain optimization effect, due to the limitations of environmental parameters, its effect is not obvious, so it is considered to modify different parameter ratios for further testing.

1) Change loss parameter

According to the application environment, there are many parameters besides the demand matrix, among which the main control parameters are more, and the appropriate change test should be done in combination with the actual application environment, so the loss related change test should be considered. In this application environment, the loss-related parameters were selected to change "damage rate(DR)" and "maximum number of instructions of operators (MN-O)", ten groups of test schemes with variable parameters were set up and the total operation and maintenance cost was tested, as shown in Table 7, Figure 13 and 14.

TABLE VII. TEST RESULTS OF LOSS PARAMETER CHANGES

DR	MN-O	Simple	Optimization	SR/%
0%	5	221290	217430	1.7443
	10	219205	216605	1.1861
	15	218565	216225	1.0706
	20	218260	216070	1.0034
	25	218000	215970	0.9312
	30	217895	216025	0.8582
	SR-average/%			1.1323
10%	5	408780	405120	0.8953
	10	404815	402320	0.6163
	15	403680	401365	0.5735
	20	403105	400665	0.6053
	25	402755	400365	0.5934
	30	402535	399625	0.7229
	SR-average/%			0.6678
20%	5	618710	611015	1.2437
	10	611970	606150	0.9510
	15	609710	604300	0.8873
	20	608975	603335	0.9261
	25	608230	602780	0.8960
	30	607465	602440	0.8272
	SR-average/%			0.9552
30%	5	826950	816590	1.2528
	10	817045	810275	0.8286
	15	813980	807800	0.7592
	20	812330	806920	0.6660
	25	811305	805955	0.6594
	30	810660	805260	0.6661
	SR-average/%			0.8054
40%	5	1030775	1019150	1.1278
	10	1018040	1011485	0.6439
	15	1014430	1008825	0.5525
	20	1012815	1007620	0.5129
	25	1011500	1006420	0.5022
	30	1011060	1005925	0.5079
	SR-average/%			0.6412
SR-average-total/%			0.8404	



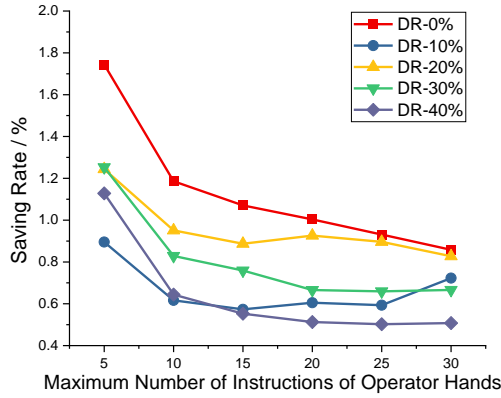


Figure. 13. The relationship between the MN-O and the SR with different DR

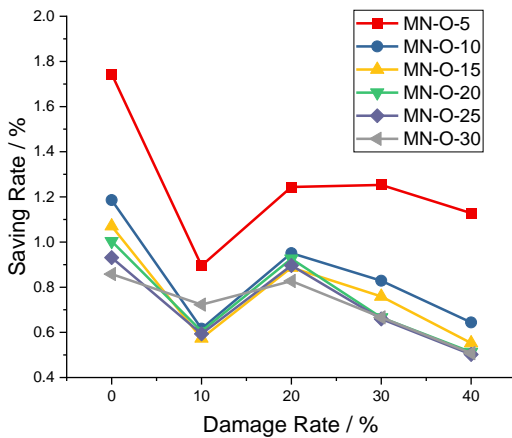


Figure. 14. The relationship between the DR and the SR with different MN-O

The test results show that under all parameter ratios, the total cost saving rate of the test results is greater than zero, indicating that the optimization algorithm is useful, the highest saving rate can be close to 1.8%, the lowest saving rate is more than 0.5%, and the average saving rate of all the test schemes is 0.8404%, and the related images show the same trend, the

optimization algorithm effect is stable and has an obvious effect.

2) Change the Discount Purchase Scheme

A large part of the model optimization in this study is based on preferential policies, and combined purchase is also the core decision basis in the model optimization, so it is considered to adjust the optimization policies and then conduct comparative tests. The experimental test Settings and results are shown in Table 8, Figure 15 and 16. Since the Container Boats and operator do not substantially affect each other, they can change simultaneously.

Scheme 1 Original purchase policy

Scheme 2 Add a layer of piecewise function each

Scheme 3 Add two layers of piecewise functions respectively

Scheme 4 Reduce one layer of piecewise function respectively

Scheme 5 Reduce two levels of piecewise functions each (no discount)

Scheme 6 The preferential power of the original piecewise function is reduced

Scheme 7 The preferential power of the original subsection function continues to decrease

Scheme 8 The original segmented function preferential efforts to increase

Scheme 9 We will continue to give more preferential treatment to original piecewise functions

TABLE VIII. TEST OF PARAMETER CHANGE OF PREFERENTIAL SCHEME

Scheme	Container Boats	Operator Hands	Simple	Optimization	SR/%
1	$f(x) = \begin{cases} 200x, & x \leq 5 \\ 100+180x, & 5 < x \leq 10 \\ 300+160x, & x > 10 \end{cases}$	$g(x) = \begin{cases} 100x, & x \leq 20 \\ 200+90x, & 20 < x \leq 40 \\ 600+80x, & x > 40 \end{cases}$	403105	400665	0.6053
2	$f(x) = \begin{cases} 200x, & x \leq 5 \\ 100+180x, & 5 < x \leq 10 \\ 300+160x, & 10 < x \leq 15 \\ 600+140x, & x > 15 \end{cases}$	$g(x) = \begin{cases} 100x, & x \leq 20 \\ 200+90x, & 20 < x \leq 40 \\ 600+80x, & 40 < x \leq 60 \\ 1200+70x, & x > 60 \end{cases}$	397055	389930	1.7945

3	$f(x) = \begin{cases} 200x, & x \leq 5 \\ 100+180x, & 5 < x \leq 10 \\ 300+160x, & 10 < x \leq 15 \\ 600+140x, & 15 < x \leq 20 \\ 1000+120x, & x > 20 \end{cases}$	$g(x) = \begin{cases} 100x, & x \leq 20 \\ 200+90x, & 20 < x \leq 40 \\ 600+80x, & 40 < x \leq 60 \\ 1200+70x, & 60 < x \leq 80 \\ 2000+60x, & x > 80 \end{cases}$	394265	379580	3.7247
4	$f(x) = \begin{cases} 200x, & x \leq 5 \\ 100+180x, & 5 < x \end{cases}$	$g(x) = \begin{cases} 100x, & x \leq 20 \\ 200+90x, & 20 < x \end{cases}$	414225	413355	0.2100
5	$f(x) = 200x$	$g(x) = 100x$	434295	433615	0.1566
6	$f(x) = \begin{cases} 200x, & x \leq 5 \\ 50+190x, & 5 < x \leq 10 \\ 150+180x, & x > 10 \end{cases}$	$g(x) = \begin{cases} 100x, & x \leq 20 \\ 100+95x, & 20 < x \leq 40 \\ 300+90x, & x > 40 \end{cases}$	418700	417880	0.1958
7	$f(x) = \begin{cases} 200x, & x \leq 5 \\ 25+195x, & 5 < x \leq 10 \\ 75+190x, & x > 10 \end{cases}$	$g(x) = \begin{cases} 100x, & x \leq 20 \\ 60+97x, & 20 < x \leq 40 \\ 140+95x, & x > 40 \end{cases}$	426193	425578	0.1443
8	$f(x) = \begin{cases} 200x, & x \leq 5 \\ 150+170x, & 5 < x \leq 10 \\ 450+140x, & x > 10 \end{cases}$	$g(x) = \begin{cases} 100x, & x \leq 20 \\ 400+80x, & 20 < x \leq 40 \\ 1200+60x, & x > 40 \end{cases}$	376065	365835	2.7203
9	$f(x) = \begin{cases} 200x, & x \leq 5 \\ 200+160x, & 5 < x \leq 10 \\ 600+120x, & x > 10 \end{cases}$	$g(x) = \begin{cases} 100x, & x \leq 20 \\ 600+70x, & 20 < x \leq 40 \\ 1800+40x, & x > 40 \end{cases}$	349025	329050	5.7231
SR-average %			1.6972		

According to the experimental scheme, the comparison images shown in Figure. 15 were drawn for groups 1-5, and the comparison images shown in FIG. 16 were drawn for groups 1 and 6-9.

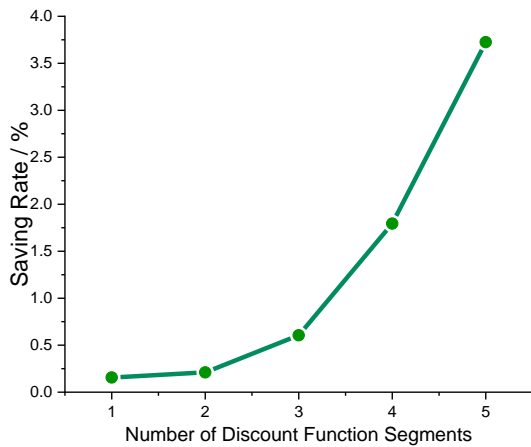


Figure. 15. Number of discount segments and SR

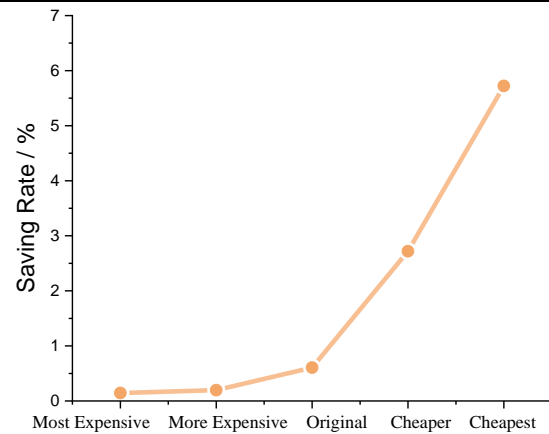


Figure. 16. Relationship between discount intensity and SR

The test results show that the saving rate is greater than zero in all cases, indicating that the optimization scheme is effective. In addition, with the increase of preferential intensity and the increase of preferential segments (also referred to as the increase of preferential intensity), the effect of the optimization scheme is getting better and

better, which is consistent with the optimization strategy "based on the lowest operating cost". At present, the best saving rate is close to 6%, the lowest is 0.14%, and the average saving rate is 1.69%.

### B. Changing the sequence of requirements

It is considered to generate three groups of arrays from different sources as the weekly demand function in the application scenario. Due to the limitations of application environment modeling, a new demand data is randomly generated near the weekly demand number according to the Gaussian distribution rule based on the array of the original application scenario, namely:

$$N_{new} = \text{random.gauss}(N_{old}, 10) \quad (31)$$

If the generated data is less than 0, it is generated again.

Use this rule to generate 100 test arrays. Then input each group of data into the simple greedy strategy model and the optimized model, and compare the saving rate of each group of input data. The experimental results are shown in Figure 17.

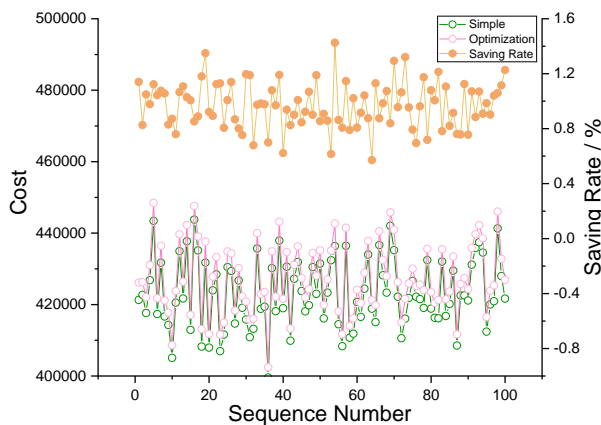


Figure 17. Test results after changing the requirement matrix

In Figure 17, the red ring represents the operation result before optimization, the green ring represents the operation result after optimization, and the orange origin represents the total cost saving rate of this test. It is found that all test results of the newly generated demand matrix are in line with expectations -- the saving rate is greater than zero, indicating that the optimization strategy is universal in this application scenario. In addition, it can be found that the optimization effect is also affected by the demand matrix. In the above 100 groups of tests, the maximum saving rate is close to 1.6%, the lowest saving rate is less than 0.6%, and the average saving rate is 0.9677%. The overall "optimization" effect has been achieved.

### C. Change the parameters and test again

In the experiment, it is found that due to the contingency of the original application scenario, constraint conditions and demand sequence, some optimization effects are not obvious. In the above simulation test, there is a better saving rate. Therefore, parameters with better optimization performance and reasonable performance were selected to modify the application scenario, and the simple model, the primary optimization model and the secondary optimization model were tested again.

The parameters that need to be changed are: the loss rate is 20%, the maximum number of instructions of the coach operator is 10, the preferential policies are selected as Article 9 in Table 5, and the demand sequence is selected as the group with the highest saving rate in the test in the previous section. The results of the run are shown in Table 9

TABLE IX. RETEST RESULTS OF CHANGING GLOBAL PARAMETERS

	OP-N	OP-C	CB-N	CB-C	TOTAL
Simple	4101	383245	939	146280	529525
Optimization 1	4080	381415	939	146280	527695
Optimization 2	4080	368045	939	137530	505575
SR total	4.5229%	SR 1	0.3456%	SR 2	4.1918%

In this group of tests, the "reverse purchase allocation" module expressed in optimization 1 showed obvious optimization effect on operator planning, which not only reduced the maintenance cost, but also clearly reduced the total number, indicating that this optimization module is effective. In addition, the optimization module 2 continues to maintain its original optimization effect and performs well.

Because of the contingency of parameter designation and requirement sequence, the effect of modules and models can not be expressed by a simple data, but should be analyzed concretely. Although the optimization rate (also known as saving rate in this paper) of the module in this application scenario is not large numerically, it can be seen from the variable parameter extension test and simulation test that the operation rate of the optimization module is 100 percent, that is, all the tests have optimization effect.

In the same application scenario, some researchers also choose linear programming model, overall programming model, ARIMA model, unitary linear regression model, etc. In contrast, the model proposed in this study does not focus on the expression of the overall function, but runs according to the time block, focuses on mining the influence relationship between variables, adopts the idea of greedy strategy to gradually select the optimal solution, and finally completes the model optimization module by gradually considering the "optimal condition" in

the reverse optimization. The final operation results show that the model proposed in this study can achieve approximate or even better operation effect, and the model analysis and parameter variable analysis are more clear and concise, the model segmentation is clear, the action mechanism is clear, and the stability and optimization effect of the optimization module are excellent.

## VI. CONCLUSIONS

In view of the application background tested in this paper, the superficial simple application of greedy strategy can indeed find very big drawbacks, especially under the influence of local optimal solution and after-effect, the so-called "optimal solution" produced in the early stage is often affected and changed. By analyzing the source, generation mechanism and solution direction of the problem and adding the "reverse correction" mechanism with "greedy strategy" as the core principle, it can meet the requirements of the application environment and optimize the results without destroying the original modeling logic. Some parameters of the application background were changed and tested regularly, and 100 groups of test sequences were randomly generated according to the original requirement sequence rules. Parameters and sequences with more obvious performance were selected to test the effect of the module again, and the optimization module had a better performance.

1) According to the requirements of application scenarios, the optimization strategy of this study proposes two optimization modules, reverse allocation and reverse merger. The saving rate of each module can be 0.08187% and 0.5239% respectively, and the comprehensive saving rate is 0.6053%. In the application scenario of this study,

the operating cost has reached hundreds of thousands or even millions, and the actual cost saved has reached tens of thousands.

2) The damage rate and the maximum number of instructions of the operator instructor were changed, and the highest saving rate was close to 1.8%, the lowest saving rate was more than 0.5%, and the average saving rate was 0.8404%.

3) When the parameters are changed regularly, the saving rate of the optimization module can also show a certain regular change, which shows that the optimization module has stability and scalability. It is not a simple numerical optimization, but an effective optimization from the mechanism of action.

4) Changing the purchase discount program for several times, the test results perfectly meet the optimization goal. The greater the preferential intensity of the purchase plan, the higher the optimized saving rate, the highest is close to 6%, the lowest is 0.14%, and the average saving rate is 1.69%.

5) Based on the original demand sequence, 100 groups of new demand sequence were randomly generated and tested using Gaussian distribution. Among them, the optimization efficiency is 100%, all the sequences can have the optimization effect, the best can be close to 1.6%, the average saving rate is 0.9677%, the optimization module has good effect and strong stability.

6) Select a set of parameters and demand sequences with good performance in the simulation test and in line with the actual situation, and test the optimization module effect again. The optimization module has obvious effect, the overall saving rate is more than 4.5%, and the cost is more than 20,000 yuan.

The application background of this research is a dynamic programming, multi-constraint and multi-objective topic. Its internal equation of state, state transition function and objective function are not a simple expression, with many variables and complex interconnections. The local optimal solution obtained by greedy strategy can directly influence each other, and the optimal solution cannot be obtained directly. However, this does not mean that greedy strategy cannot be used. In fact, in the solving idea of greedy strategy and dynamic programming, "decompose the problem and solve it step by step" is very conducive to analyzing the problem. Based on the preliminary greedy strategy model and through the analysis of the relationship between internal variables, the "reverse correction mechanism" proposed in this study can repair the shortcomings of the greedy algorithm and save tens of thousands of yuan of operating costs in the application environment of this study.

Compared with other modeling strategies in the same application scenario, the "compound optimization greedy strategy integrating reverse correction mechanism" proposed in this paper can not only obtain better operation results, but also dig deep into modeling analysis, module programming solutions and optimization strategies. It focuses more on the in-depth analysis of the mechanism of action and the analysis of the relationship between variables, so as to solve practical application problems by breaking the whole into pieces.

Greedy strategy is easy to be ignored because its local optimal solution does not mean the global optimal solution. However, the thinking of greedy strategy of "breaking the whole thing into pieces and breaking it one by one" is very conducive to analyzing the relationship between variables and

sorting out clear logical lines in a large number of variables and constraints. Greedy strategy should have more space to play, especially from the perspective of mathematical analysis, through the reverse correction of greedy strategy algorithm, repeated calls, comparative selection, etc., can get obvious and stable optimization effect.

In today's research environment where various kinds of high-order algorithm strategies and artificial intelligence machine learning strategies are blooming everywhere, sometimes simple problems are easy to be complicated. Traditional algorithms should not be abandoned directly, but can be modified or integrated with emerging

technology algorithms to get more possibilities and give play to greater potential.

#### REFERENCES

- [1] W.Shi, Application of data mining technology in biological sequence analysis and recognition algorithm [J]. Computer Programming Skills & Maintenance, 2020(07):77-79
- [2] J.Yu. Application analysis of Python language in big data analysis [J]. China New Telecommunications, 2021, 23(21):78-79.
- [3] S.Ou. A brief analysis of the application techniques of Python data analysis [J]. Youth of Shanxi, 2021(10):67-68.
- [4] Z.Liu, J.H.Guo. Explore the application of Jupyter Notebook programming tools in project-based learning [J]. Chinese modern educational equipment, 2020(14):3.
- [5] Y.Y.Wei, Y.F.Wu, Y.Y.Li. Research on the application of Python technology in data visualization [J]. Fujian Computer, 2022, 38(01):27-31.