Study of Decision-Making Problems in the Production Process

Tian Qin[1,a], Jiabao Lv[1,b], Mingyu Cao[1,c], Fang He[2,d*]

[1]College of integrated circuits and optoelectroinc chips, Shenzhen Technology University, Shenzhen, China
[2]College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China

[a]280626775@qq.com, [b]202100303031@stumail.sztu.edu.cn
[c]202100303103@stumail.sztu.edu.cn, [d*]hefang@sztu.edu.cn

Abstract. In this paper, the decision strategy under different scenarios is studied for the decision choice problem of manufacturing enterprises in the production process. The study aims to help improve the operational efficiency and agility of enterprises through the specific analysis of the decision-making problem in the production process to provide enterprises with feasible decision support tools, improve production efficiency and resource utilization, and help intelligent manufacturing upgrade. As in the production process, the spare parts used and the synthesized finished product may be defective, and defective products to the hands of the user will have to pay a certain amount of compensation or even reduce the credibility of the enterprise. At the same time, if you choose to test the spare parts and finished products also need to pay a certain amount of testing costs. Therefore, how to better find the most profitable decision has become a major problem. In order to solve these problems, this paper divides the production situation into two categories: single process and two spare parts, and multiple processes and multiple spare parts, and assumes specific values such as product failure rate and spare part cost. In the case of single process and two spare parts, this paper constructs a decision tree model and a cost effectiveness model, and solves them by Monte Carlo(MC) simulation. In the case of multi-process and multi-part production, the production process is too complex and high-dimensional when the MC simulation method requires a large number of samples, which usually shows a lower efficiency, so we use the Slime mold algorithm(SMA) to produce results more efficiently. After experiments with specific numerical settings, it is shown that the proposed model can produce good results in many cases. This not only helps to improve the operational efficiency and agility of enterprises, but also has strategic significance in promoting the digital transformation of the manufacturing industry and realizing sustainable development.

Keywords: decision-making problem, cost effectiveness model, Monte Carlo, Slime mold algorithm

P. S. Borah et al. (eds.), *Proceedings of the 2025 5th International Conference on Enterprise Management and Economic Development (ICEMED 2025)*, Advances in Economics, Business and Management Research 346,
https://doi.org/10.2991/978-94-6463-811-0_52

1 Introduction

With the intensification of global manufacturing competition, the increasing diversity of customer needs, and the acceleration of technological innovation, modern manufacturing processes are facing unprecedented complexity and uncertainty. Production decisions, as the core of business operations, have a direct impact on production efficiency, cost control, product quality, and on-time delivery. In today's increasingly competitive environment, companies need to strictly control each production process, and how to ensure production quality while reducing production costs has become one of the core issues. Research on decision-making in the production process not only helps to improve the operational efficiency [4 and agility of enterprises, but also has important strategic significance in promoting the digital transformation of the manufacturing industry and realizing sustainable development.[1]

Focusing on the key decision-making problems in the production process, this paper combines theoretical analysis and empirical research with the aim of constructing an intelligent decision-making framework adapted to dynamic environments. Taking an electronics processing factory as an example, its specific implementable strategies are thoroughly analyzed and the optimal strategies are found in different situations.

In the case of a single process and two spare parts, this paper establishes a cost-benefit model based on Decision Tree(DT) and Dynamic Programming(DP), analyzes the problem in detail to find out the key decision points and constructs a DT. A dynamic cost-benefit model is developed to calculate the total cost of different decisions by controlling the decision points such as whether to check the zero point or not, whether to disassemble or not through the given constraints and defining the binary decision variables, and then find the most appropriate decision by comparing the profit of each decision. Finally, MC simulation is performed to find the optimal decision path.

In the case of multiple processes and multiple spare parts, the use of MC simulation method requires a large number of samples, which usually shows low efficiency, and so on. In contrast, the SMA is able to quickly find an approximate optimal solution in complex environments by simulating the self-organizing behavior of slime molds, which is especially suitable for high-dimensional, nonlinear or multi-peak optimization problems. The SMA quickly converges to a feasible solution through a heuristic search mechanism that typically requires a small number of iterations. It often shows high efficiency in complex problems. Therefore, in this paper, we choose to use the SMA for solving in order to find the optimal decision path in the case of multiple processes and multiple spare parts.

2 Studies in Single Process and Two Spare Cases

2.1 Analysis of the Situation

The study of production decision-making in a single process or a single spare part scenario, although on the surface it seems to simplify the complexity of the production system, has a far-reaching background significance that involves the deepening of the theoretical research and also has an important guiding value for the actual production management. A single scenario provides a controlled experimental environment for studying the basic laws of the production system, which facilitates the construction of mathematical models, the verification of algorithms or theoretical frameworks, and lays the foundation for the study of complex systems. In addition, a large number of small and medium-sized manufacturing enterprises have relatively simple production processes, and the study of such scenarios can directly help them optimize resource utilization and reduce costs.

Let's assume that a company manufactures a best-selling electronic product and needs to purchase two spare parts (Part 1 and Part 2), which are assembled into a finished product at the company. In the assembled finished product, as long as one of the spare parts is unqualified, the finished product must be unqualified; if both spare parts are qualified, the assembled finished product is not necessarily qualified. Enterprises can inspect the spare parts and the assembled finished product, but they must pay a certain inspection fee. For the unqualified finished product, the enterprise can choose to scrap, or disassemble it, the disassembly process will not cause damage to the spare parts, but need to spend disassembly costs, the disassembled parts repeat the direct decision. For the non-conforming products purchased by the user, the enterprise will unconditionally replace them and incur a certain replacement loss, and the returned non-conforming products repeat the direct decision.

In order to solve a series of decision-making problems such as whether to test spare parts and finished products in the process, we established a DT model and a cost-effectiveness model, and solved them by MC simulation method. Figure 1 illustrates the technology roadmap.[3]

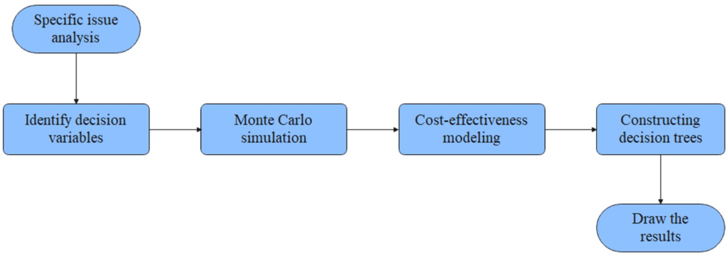

Fig. 1. Technology Roadmap.

The DT method, which first emerged in the 1960s to late 1970s, is a commonly used data mining method that is a model that represents decision rules and categorization results in a tree-like data structure[2] suitable for both classification and regression problems. The DT represents the decision process through a tree structure, where we divide each step of the decision problem into different nodes, and the leading branches represent the results of different decisions.

In this hypothesis, there are four decision nodes: whether or not to test spare part 1, whether or not to test spare part 2, whether or not to test each of the assembled finished products, and whether or not to disassemble the detected unqualified finished products. The production flowchart is shown below:

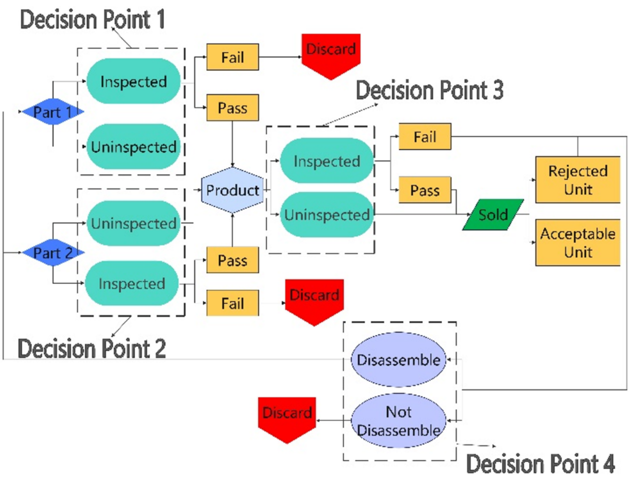

Fig. 2. Decision-making flowchart for a single process and two spare parts.

The four key decision points are indicated in the Figure 2, from which we can visualize the impact of each decision on the process to find the best decision.

2.2 Hypothetical Decision Variables

- $X_1 \in \{0,1\}$:Whether to test spare part 1 (0: No, 1:Yes)
- $X_2 \in \{0,1\}$:Whether to test part 2 (0: No, 1: Yes)
- $X_3 \in \{0,1\}$:Whether to test the finished product (0: No, 1: Yes)
- $X_4 \in \{0,1\}$:Whether to dismantle non-conforming products (0: No, 1: Yes)

2.3 Cost Effectiveness Modeling

Step1: Purchase cost of spare parts Y_1

Spare parts purchase cost is the product of unit price and quantity of spare parts, so:

$$Y_1 = \sum_{i=1}^{2} C_i A_i \tag{1}$$

Where C_i denotes the unit price of part i and A_i expresses the quantity of part i purchased.

Step2: Testing cost of spare parts Y_2

Spare parts testing cost is the product of spare parts testing cost and quantity, therefore:

$$Y_2 = \sum_{i=1}^{2} Z_i A_i X_i \tag{2}$$

Where Z_1 represents the detection cost of spare parts 1 and Z_2 represents the detection cost of spare parts 2.

Step3: The number of parts synthesized finished product C_1

The number of parts to synthesize the finished product depends on the number of parts, and the number of parts and the number of parts purchased and whether or not the parts tested. Therefore:

$$C_1 = \min\{(1 - P_1 X_1) * A_1, (1 - P_2 X_2) * A_2\} \tag{3}$$

Where p1 is the defective rate for part 1 and p2 is the defective rate for part 2.

Step4: Assembly cost of the finished product Y_3

The assembly cost of the finished product is the product of the number of spare parts to synthesize the finished product and the assembly cost of the finished product, therefore:

$$Y_3 = C_1 * W_1 \tag{4}$$

Where W_1 expresses the assembly cost of the finished product.

Step5: Inspection cost of the finished product Y_4

The testing cost of the finished product is the product of the number of spare parts to synthesize the finished product and the testing cost of the finished product, therefore:

$$Y_4 = C_1 * Z_3 * X_3 \tag{5}$$

Where Z_3 is the testing cost of the finished product.

Step6: The defective rate of the finished product P_3

The defective rate of the finished product is related to whether the finished product contains defective parts and the inherent defective rate of the finished product, therefore:

$$P_3 = \{1 - [1 - (1 - X_1)p_1][1 - (1 - X_2)p_2]\} \quad + [1 - (1 - X_1)p_1][1 - (1 - X_2)p_2] * p_4 \tag{6}$$

Where p_4 is the inherent defective rate of the finished product. That is, two spare parts are qualified, the assembly of the finished product is also unqualified probability. $\{1 - [1 - (1 - X_1)p_1][1 - (1 - X_2)p_2]\}$for the assembly of the finished product after the probability of substandard spare parts, $[1 - (1 - X_2)p_1][1 - (1 - X_2)p_2] * p_4$for the assembly of the finished product are qualified spare parts and finished product for the probability of substandard.

Step7: Disassembly costs of non-conforming products Y_5

$$Y_5 = W_2 * C_1 * P_3 * X_3 * X_4 \tag{7}$$

Where W_2 is the disassembly cost of the finished product and P_3 is the defective rate of the finished product.

The decision to dismantle the defective parts is repeated, so we need to set up a recursive function so that the parts can continue to be added to the production process.

Step8: The replacement cost of substandard products Y_6

Enterprises need to replace the unqualified products in the hands of users and pay a certain amount of exchange loss, and the number of unqualified products in the hands of users and the rate of defective finished products and whether the finished product testing and production of finished products related to the number of units, therefore:

$$Y_6 = (1 - X_3) * P_3 * C_1 * (W_3 + \sum_{i=1}^{4} Y_i) \quad (8)$$

Where W_3 is the exchange loss of the finished product, $(1 - X_3) * P_3 * C_1$ is the number of nonconforming products to the user, and $(1 - X_3) * P_3 * C_1 * \sum_{i=1}^{4} Y_i$ is the price to re-compensate the user for a conforming product.

Step9: Revenue from the sale of finished goods by the enterprise M_1

The firm's revenue from the sale of finished goods is the product of the quantity of qualified goods that reach the user and the market price of the finished goods, therefore:

$$M_1 = C_1 * (1 - P_3) * M_2 \quad (9)$$

Where M_2 is the market selling price of the finished product.

Step10: Final Profit S_1

The final profit is the proceeds from the sale of the finished product by the firm minus all the costs spent in the production process and the cost of the admixture. Therefore:

$$S_1 = M_1 - \sum_{i=1}^{6} Y_i \quad (10)$$

2.4 CMC Simulation Method for Solving[4]

The MC method is a numerical computation method based on random sampling and statistical simulation. Its name is derived from the famous Monte Carlo casino in Monaco, a designation that metaphorically refers to the stochastic character that the method shares with gambling activities. As a computational technique systematically developed during the Manhattan Project in the early 1940s by scientists such as Stanislaw Ulam and John von Neumann, it transforms the solution of complex problems into statistical observations of random phenomena by constructing probabilistic models and resorting to large-scale stochastic experiments on computers. With its unique non-deterministic computational paradigm, the method has been successfully applied to interdisciplinary fields such as mathematical analysis, particle physics simulation, financial derivatives pricing, engineering reliability assessment, and artificial intelligence reinforcement learning.

The core steps of the MC method[5] can be broken down into the following four key stages, each with its own specific tasks and technicalities:

1. Modeling: Transforming the actual problem into a probabilistic model and determining the target quantity to be estimated. First we need to specify the random variable and its distribution (e.g., uniform distribution, normal distribution, etc.). Next, we need to transform the problem into a mathematical expectation form. In addition to this, we need to make sure that the probability distribution matches the problem scenario
2. Sampling (generating random samples): Generating a large number of independent random samples based on the probability distribution in the model. A pseudo-

random number generator is first used to generate uniformly distributed samples. Then the sample size is determined according to according to the accuracy requirements and the law of large numbers.
3. Calculation (Simulation and Statistics): Perform a simulation or calculation for each sample, collect the results and statistically estimate the target quantity. That is, perform an experiment for each sample and then save the output values for each sample.
4. Statistical analysis and error control: Estimating target values from sample statistics and assessing the precision of the results.

In the actual production situation of the enterprise, the defective rate of spare parts and the qualified rate of finished products have great randomness, and we can not be sure how many qualified products we can actually get from incoming goods and production. Therefore, in this problem, we use MC simulation to simulate the random fluctuations due to defective products in reality with a large number of random tests, never more realistically assess the expected profit and risk of a strategy.

Our specific steps are as follows:

1. The firm needs to decide whether to test spare parts and finished products, and whether to dismantle substandard finished products, so as to maximize profits under uncertain product quality. There are four decision nodes where decisions can be made, and two decisions per decision node, so there are a total of 16 decision options.
2. Perform a complete simulation (one trial) with random numbers: we simulate which of a certain batch of purchased spare parts are qualified by generating an array of random numbers, and then decide whether to eliminate substandard products based on whether to test them, calculate the number of finished products after assembly, and then simulate whether each finished product is substandard, and count the cost of disassembly, cost of replacement, and revenue from sales based on whether or not the simulated finished products are qualified. Finally return the profit value of this simulation.
3. Repeat the simulation in (2) 10,000 times to form a large sample where each simulation's profit is determined by a completely different "random sample". Finally, the average profit and profit fluctuation are calculated.
4. Finally, all strategy combinations (16 in total) are iterated, and for each combination 10,000 simulations are performed and their average profit and standard deviation are printed. In order to find the optimal strategy and can quantify its stability.

2.5 Specific Numerical Experiments

This paper assumes different situations that a firm may encounter in production. In different situations, the defective rate and inspection cost of parts, the defective rate and inspection cost of finished products and the replacement loss and dismantling cost of non-conforming finished products are different. We have assumed six different scenarios as Table 1 shown.

Table 1. Assume data for each program in production.

Scenarios1	Part1			Part2		
	Defective rate	Unit price	Testing costs	Defective rate	Unit price	Testing costs
	10%	4	2	10%	18	3

Product				substandard product	
Defective rate	assembly cost	Testing costs	Market price	Exchange losses	Disassembly costs
10%	6	3	56	6	5

Scenarios2	Part1			Part2		
	Defective rate	Unit price	Testing costs	Defective rate	Unit price	Testing costs
	20%	4	2	20%	18	3

Product				substandard product	
Defective rate	assembly cost	Testing costs	Market price	Exchange losses	Disassembly costs
20%	6	3	56	6	5

Scenarios3	Part1			Part2		
	Defective rate	Unit price	Testing costs	Defective rate	Unit price	Testing costs
	10%	4	2	10%	18	3

Product				substandard product	
Defective rate	assembly cost	Testing costs	Market price	Exchange losses	Disassembly costs
10%	6	3	56	30	5

Scenarios4	Part1			Part2		
	Defective rate	Unit price	Testing costs	Defective rate	Unit price	Testing costs
	20%	4	1	20%	18	1

Product				substandard product	
Defective rate	assembly cost	Testing costs	Market price	Exchange losses	Disassembly costs
20%	6	2	56	30	5

scenarios 5	Part1			Part2		
	Defective rate	Unit price	Testing costs	Defective rate	Unit price	Testing costs
	10%	4	8	20%	18	1

Product				substandard product	

Defective rate	assembly cost	Testing costs	Market price	Exchange losses	Disassembly costs
10%	6	2	56	10	5

scenarios 6	**Part1**			**Part2**		
	Defective rate	Unit price	Testing costs	Defective rate	Unit price	Testing costs
	5%	4	2	5%	18	3
Product				**substandard product**	
Defective rate	assembly cost	Testing costs	Market price	Exchange losses	Disassembly costs
5%	6	3	56	10	40

Assume that the enterprise initially purchases 10,000 units each of parts 1 and parts 2. Substitute the above six scenarios into the model constructed above, and analyze the optimal strategy in each case through MC simulation. In Table 2 we can get the results:

Table 2. Results of the Monte Carlo simulation method.

circumstances	x_1	x_2	x_3	x_4	Profit
1	0	0	1	1	198585
2	0	0	1	1	102570
3	0	0	1	1	198585
4	1	1	1	1	103000
5	0	1	1	1	159655
6	0	0	0	0	212907

2.6 Analysis of Results

From the results in the case of spare parts with a low rate of defective parts and a large cost of inspection, the spare parts should not be inspected as much as possible. Then you can choose to test the finished products to achieve high efficiency. Finally in the case of dismantling cost is not high, should try to dismantle the nonconforming products.

It can be seen that the study of a single process or spare parts scenario, which seems to be simple, is actually a bridge connecting theoretical innovation and practical application. It is not only the “smallest research unit” of the complex production system, but also the starting point of enterprise fine management. In the context of rapid technological iteration and highly uncertain supply chain, this kind of research provides a fundamental tool for improving production efficiency, reducing costs and enhancing resilience, which is of strategic significance for the transformation and upgrading of the manufacturing industry.

3 Study in the Case of Multiple Processes and Multiple Spare Parts

3.1 Analysis of the Situation

In today's industry trend, the traditional assembly line is difficult to adapt to the multi-species mixed production, a single device may involve parts from 20 + countries which contains hundreds of processes. The current research focuses on a single link, and lacks a collaborative optimization framework of process-parts-resources. Therefore, multi-process and multi-part collaborative decision-making research is one of the core keys to the intelligent upgrading of the manufacturing industry.

In this paper, we assume that the assembly process of an electronics factory enterprise involves two processes and eight spare parts. The resulting decision nodes are consistent with a single process and two spare parts, that is, the spare parts and assembled each piece of finished product whether to test, on the detection of unqualified finished product whether to disassemble. Figure 3 illustrates the case of two processes and eight parts.

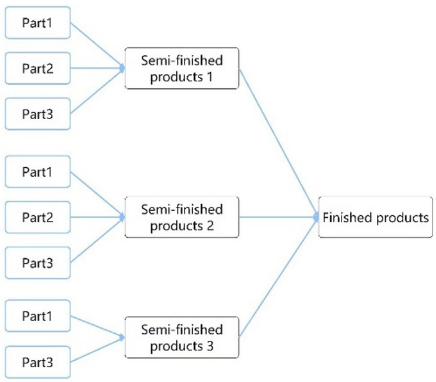

Fig. 3. low chart of 2 processes and 8 parts.

According to the specific situation of the problem we analyze the graph to know: there are 16 key decision points in the production system, there are eight decision nodes on whether to detect spare parts, four nodes on whether to detect semi-finished and finished products, and finally four nodes on whether to disassemble semi-finished and finished products. The great increase of decision nodes makes the number of decision schemes increase greatly, there are a total of kinds of schemes, and the paths to be traversed increase greatly, which leads to the workload of MC simulation is too large, so we need to further optimize the algorithm. First establish the mathematical model.

3.2 Hypothetical Decision Variables

Hypothetical decision variables:

- $X_i \in \{0,1\}$:Whether to test spare part i (0: No, 1: Yes)
- $B_i \in \{0,1\}$:Whether to dismantle semi-manufactured products i (0: No, 1: Yes)

- $b_i \in \{0,1\}$:Whether to test semi-manufactured products i (0: No, 1: Yes)
- $H_1 \in \{0,1\}$:Whether to test finished product (0: No, 1: Yes)
- $h_1 \in \{0,1\}$:Whether to dismantle finished product (0: No, 1: Yes)

3.3 Cost Effectiveness Modeling

Step1: Purchase cost of spare parts y_1

Spare parts purchase cost is the product of unit price and quantity of spare parts, so:

$$y_1 = \sum_{i=1}^{8} c_i a_i \tag{11}$$

Where c_i denotes the unit price of part i and a_i expresses the quantity of part i purchased.

Step2: Testing cost of spare parts y_2

Spare parts testing cost is the product of spare parts testing cost and quantity, therefore:

$$y_2 = \sum_{i=1}^{8} z_i a_i x_i \tag{12}$$

Where z_i represents the detection cost of spare parts i.

Step3: The number of parts synthesized finished product i c_i

The number of parts to synthesize the finished product depends on the number of parts, and the number of parts and the number of parts purchased and whether or not the parts tested. Therefore:

$$c_1 = \min\{(1 - p'_1 x_1) * a_1, (1 - p'_2 x_2) * a_2, (1 - p'_3 x_3) * a_3\} \tag{13}$$

$$c_2 = \min\{(1 - p'_4 x_4) * a_4, (1 - p'_5 x_5) * a_5, (1 - p'_6 x_6) * a_6\} \tag{14}$$

$$c_3 = \min\{(1 - p'_7 x_7) * a_7, (1 - p'_8 x_8) * a_8\} \tag{15}$$

Where p'_i is the defective rate for part i.

Step4: Assembly cost of semi-finished product y_3

The assembly cost of semi-finished products is the product of the quantity of spare parts to synthesize the finished product and the assembly cost of the finished product, therefore:

$$y_3 = \sum_{i=1}^{3} c_i * w_i \tag{16}$$

Where w_i expresses the assembly cost of semi-finished product i.

Step5: Testing cost of semi-finished products y_4

The testing cost of semi-finished products is the product of the number of spare parts to synthesize the finished product and the testing cost of the finished product, therefore:

$$y_4 = \sum_{i=1}^{3} c_i * z_i * B_i \tag{17}$$

Where z_i is the testing cost of semi-finished product i.

Step6: Defective rate of semi-finished products P'_i

The defective rate of semi-finished products is related to whether the semi-finished products contain defective parts and the inherent defective rate of semi-finished products, therefore:

$$P'_1 = \left\{1 - \left[1 - (1 - x_1)p'_1\right]\left[1 - (1 - x_2)p'_2\right][1 - (1 - x_3)p'_3]\right\} + \left[1 - (1 - X_1)p'_1\right]\left[1 - (1 - X_2)p'_2\right][1 - (1 - X_3)p'_3] * p''_1 \tag{18}$$

$$P'_2 = \left\{1 - [1 - (1 - x_4)p'_4]\left[1 - (1 - x_5)p'_5\right][1 - (1 - x_6)p'_6]\right\} + \left[1 - (1 - X_4)p'_4\right]\left[1 - (1 - X_5)p'_5\right][1 - (1 - X_6)p'_6] * p''_2 \tag{19}$$

$$P'_3 = \left\{1 - \left[1 - (1 - x_7)p'_7\right][1 - (1 - x_8)p'_8]\right\} + \left[1 - (1 - x_7)p'_7\right][1 - (1 - x_8)p'_8] * p''_3 \tag{20}$$

where p''_i is the rate of substandard products inherent in semi-finished product i. That is, the two spare parts are qualified, assembled semi-finished products also failed the probability.

Step7: Cost of dismantling non-conforming semi-finished products y_5

$$y_5 = \sum_{i=1}^{3} P'_i * B_i * b_i * c_i * w_i \tag{21}$$

where w_i is the dismantling cost of semi-finished product i.

The decision is repeated for the parts obtained after the disassembly of the defective part, for which we also need to set up a recursive function in order to allow the parts to be added to the production process.

Step8: Quantity of semi-finished products to be synthesized into finished products C'_1

The number of semi-finished products to be synthesized into finished products depends on the minimum number of semi-finished products, which in turn is related to the number of parts to be synthesized into semi-finished products. Therefore:

$$C'_1 = \min\left\{\left(1 - P'_1 \; B_1\right) * c_1, (1 - P'_2 B_2) * c_2, (1 - P'_3 B_3) * c_3\right\} \tag{22}$$

Step9: Assembly cost of the finished product y_6

The inspection cost of the finished product is the product of the number of semi-finished products to be synthesized into finished products and the assembly cost of the finished product, while the number of semi-finished products to be synthesized into finished products depends on the minimum number of semi-finished products therefore:

$$y_6 = C'_1 * W'_1 \tag{23}$$

Where $W_1{}'$ expresses the cost of assembly of the finished product.

Step10: Inspection cost of the finished product y_7

The testing cost of the finished product is the product of the number of spare parts to synthesize the finished product and the testing cost of the finished product, therefore:

$$y_7 = C'_1 * Z'_1 * X_3 \tag{24}$$

Where Z'_1 is the cost of testing the finished product.

Step11: The defective rate of finished products P'_4

The defective rate of the finished product is related to whether the finished product contains defective semi-finished products and the inherent defective rate of the finished product, therefore:

$$C'_1 = \min\left\{\left(1 - P'_1 \quad B_1\right) * c_1 \quad , (1 - P'_2 B_2) * c_2, (1 - P'_3 B_3) * c_3\right\} \tag{25}$$

Where p''_4 is the inherent defective rate of the finished product. That is, the two spare parts are qualified, the assembly of the finished product is also unqualified probability.

Step12: Dismantling costs for non-conforming finished products y_8

$$y_8 = W'_2 * C'_1 * P'_4 * \ H_1 * \ h_1 \tag{26}$$

Where W'_2 is the disassembly cost of the finished product and P'_4 is the defective rate of the finished product.

The decision is repeated for the semi-finished product after the disassembly of the non-conforming product, where a recursive function is also needed so that the new parts can continue to be added to the production.

Step13: The replacement cost of substandard products y_9

Enterprises need to exchange the unqualified finished products in the hands of the user and pay a certain amount of exchange loss, while the number of unqualified products in the hands of the user and the rate of defective finished products and whether the finished product testing and production of finished products number of related to, therefore:

$$y_9 = (1 - \ H_1) * P'_4 * C'_1 * (W'_3 + \textstyle\sum_{i=1}^{7} Y_i) \tag{27}$$

Where W'_3 is the loss on the exchange of finished goods, $(1 - \ H_1) * P'_4 * C'_1$ is the quantity of non-conforming goods in the hands of the user, and $(1 - \ H_1) * P'_4 * C'_1 * \sum_{i=1}^{7} Y_i$ is the price of re-compensating the user for a conforming product.

Step14: Revenue from the sale of finished goods by the enterprise M'_1

$$M'_1 = C'_1 * (1 - P'_4) * M'_2 \tag{28}$$

The firm's revenue from the sale of finished goods is the product of the quantity of qualified goods that reach the user and the market price of the finished goods, therefore:

Where M'_2 is the market selling price of the finished product.

Step15: Final Profit S'_1

The final profit is the proceeds from the sale of the finished product by the firm minus all the costs incurred in the production process and the costs paid out. Therefore:

$$S'_1 = M'_1 - \textstyle\sum_{i=1}^{9} y_i \tag{29}$$

3.4 SMA[6]

In the case of two processes and eight parts, the critical decision nodes have 16 points, and in this high-dimensional case MC simulation methods require a large amount of sampling and usually exhibit low efficiency, among other things. Therefore, in this case, we choose the SMA. The SMA adopts the dynamic network topology formed by the biomimetic slime mold population in foraging, uses a multithreaded parallel exploration strategy to cover the high-dimensional space synchronously, utilizes information entropy evaluation to achieve key dimension focusing in order to reduce the search complexity, and coordinates the pheromone interactions between the individuals through the bio-oscillatory model to form a dynamic balance between global exploration and local exploitation. It demonstrates high efficiency, robustness and scalability in high-dimensional complex optimization problems.

The SMA[7], proposed by Li S et al. in 2020, is a population intelligence optimization algorithm inspired by the foraging behaviors of natural slime molds (e.g., Pseudopodococcus multilocularis), and is mainly used to solve complex optimization problems. When searching for food, slime molds form an efficient network by adjusting their pseudopod morphology: expanding towards areas of high food concentration, contracting from areas of low concentration, and balancing exploration and exploitation capabilities through periodic oscillations. The algorithm abstracts this process into a mathematical model, and the core mechanisms include dynamic weight adjustment, stochastic perturbations, and adaptation-based positional updates. Weights are assigned so that high-quality solutions guide the search direction, stochastic perturbation avoids falling into local optimums, and contraction of the oscillation parameter coordinates global exploration and local exploitation. Its advantages lie in few parameters, fast convergence, and strong adaptability, especially in non-convex, multi-peak, and noise-containing complex optimization problems, which are widely used in engineering optimization, path planning, machine learning hyper-parameter tuning, etc., and it is an effective tool for solving high-dimensional nonlinear problems.

Our core steps can be broken down in detail into the following five key stages:

1. Initialize the population: Randomly generate 50 individuals (candidate solutions), each representing a location in the solution space $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. Where D is the problem dimension, in this problem D is 16. set the maximum number of iterations to 100, the contraction shock parameter and the perturbation strength 0.1.
2. Calculation of fitness: The fitness value $f(x_i)$ is calculated for each individual, reflecting the quality of its solution to determine the current optimal solution X_{best} and the worst solution X_{worst}, and the corresponding best fitness f_{best} and worst fitnessf_{worst} are recorded.
3. Dynamic weight update: Dynamically adjust the weights ψ according to the individual adaptation, and the quality solution weights are larger to guide the search direction:

$$\psi = 1 + \mathrm{r} * \log\left(\frac{f_{best}-f_i}{f_{best}-f_{worst}}\right) + 1 \tag{30}$$

Where $r \in [0,1]$ is a random number and the denominator is normalized by $f_{best} - f_{worst}$ to avoid unstable values.

4. Sticky Mushroom Position Update: Adjust the weights and update the position based on the fitness:

$$x_{new} = \begin{cases} X_{best} + \psi * (X_A - X_B) + \eta * rand(rand < z) \\ \text{Randomly perturbing new positions (else)} \end{cases} \tag{31}$$

X_A, X_B: two different individuals chosen at random, introducing random differences to enhance exploration.

rand: uniformly distributed random numbers, η controls the magnitude of the perturbation.

Contraction oscillation mechanism: the parameter z determines whether the individual exploits directly around the optimal solution with a certain probability (first branch) or explores completely randomly (second branch). Here we take z to be 0.03.

5. Iteration and termination: repeat the above steps until the maximum number of iterations or convergence is reached, and then output the global optimal solution and its fitness.

3.5 Specific Numerical Experiments

In this paper, it is assumed that a company may encounter a situation in production as Table 3 and Table 4 shown:

Table 3. Hypothetical parts data.

Parts	Defective rate	Unit price	Testing costs
1	10%	2	1
2	10%	8	1
3	10%	12	2
4	10%	2	1
5	10%	8	1
6	10%	12	2
7	10%	8	1
8	10%	12	2

Table 4. Hypothetical Semi-finished products and Finished products data.

<table>
<tr><td>Semi-finished products</td><td>Defective rate</td><td>assembly cost</td><td>Testing cost</td><td>Disassembly costs</td></tr>
<tr><td>1</td><td>10%</td><td>8</td><td>4</td><td>6</td></tr>
<tr><td>2</td><td>10%</td><td>8</td><td>4</td><td>6</td></tr>
<tr><td>3</td><td>10%</td><td>8</td><td>4</td><td>6</td></tr>
<tr><td>Finished products</td><td>10%</td><td>8</td><td>6</td><td>10</td></tr>
<tr><td rowspan="2">Finished products</td><td colspan="2">Market price</td><td colspan="2">Exchange losses</td></tr>
<tr><td colspan="2">200</td><td colspan="2">40</td></tr>
</table>

The defective rate of spare parts, semi-finished products and finished products are all 10%, but the purchasing unit price of spare parts and inspection cost are different.

The values in the icons are substituted into our formula and solved by python to get the optimal decision and the maximum profit calculated by the improved slime mold algorithm and the results are shown in the Table 5:

Table 5. Results of slime mold algorithm

Decision Points	Decision results
Parts 1 tested or not	Yes
Parts 2 tested or not	Yes
Parts 3 tested or not	No
Parts 4 tested or not	Yes
Parts 5 tested or not	No
Parts 6 tested or not	No
Parts 7 tested or not	Yes
Parts 8 tested or not	Yes
Semi-finished products1 tested or not	Yes
Semi-finished products2 tested or not	Yes
Semi-finished products3 tested or not	Yes
Semi-finished product 1 disassembled or not	Yes
Semi-finished product 2 disassembled or not	Yes
Semi-finished product 3 disassembled or not	Yes
Finished product tested or not	No
Finished product disassembled or not	No

4 Conclusion

Nowadays, the competition in the market has intensified, enterprises need to optimize the production process to reduce cost and improve efficiency in order to cope with the variability of the global supply chain and the price-sensitive market demand, whereas the optimization of production decision-making is one of the core challenges. This paper focuses on the decision optimization problem in the production process, and proposes a staged and strategic modeling and solving method for the different challenges in the single process and multi-process scenarios, aiming to provide intelligent decision support for enterprises in terms of both efficiency and cost. In the simplified scenarios of single process and two spare parts, the expected profit and risk of different decision paths can be accurately quantified by constructing a decision tree model and a dynamic cost-benefit model, combined with Monte Carlo simulation method. Experiments show that the framework can quickly converge to the optimal strategy in finite dimensions, which is especially suitable for the accurate solution of low-complexity problems. Monte Carlo simulation effectively handles the randomness of the defective rate and compensation cost through probabilistic sampling, providing a reliable decision basis for enterprises. Facing the high-dimensional complex problems with multiple processes and spare parts, traditional Monte Carlo simulation is limited in efficiency due to high computational cost. In this paper, the slime mold algorithm is introduced to achieve an efficient balance between global search and local exploitation by simulating the self-organized foraging behavior of the slime mold population. The slime mold algorithm rapidly approaches the near-optimal solution in the high-dimensional space with fewer iterations, which significantly reduces the consumption of computational resources, and avoids the defect of traditional optimization algorithms that are prone to fall into the local optimum. Numerical experiments prove that the method can effectively optimize the detection strategy and resource allocation in the complex production process, and improve the overall profit margin. Production decision optimization is one of the core challenges in the intelligent transformation of manufacturing industry, and this paper provides new ideas for solving this problem through sub-scenario modeling and innovative algorithm application. Future research can focus on algorithm generalization enhancement and actual industrial scenario validation, to further promote the transformation of theoretical results into industrial practice, and inject new kinetic energy into the sustainable development of the global manufacturing industry.

Acknowledgment

This work was supported by the Natural Science Foundation of Top Talent of SZTU (grant no. GDRC202333).

References

1. Han Chao, Xiao Xingzhi, Li Shu. How Did Industrial Policies Affect Business Performance: Does the Locus of Different Policies and Paths Matter[J]. Journal of Finance and Economics, 2017, 43(1): 122-133,144.
2. Malaslı Z M, Akkoyunlu C M, Pekel E , et al. Prediction of drying kinetics and energy consumption values of purple carrots dried in a temperature-controlled microwave dryer by decision tree, random forest and ada boost approaches[J]. Chemometricsand Intelligent Laboratory Systems,2025,260105352-105352.
3. Susanne Franke, Felix Franke, Ralph Riedel, Robustness evaluation of production plans using Monte Carlo simulation, Procedia Manufacturing, Volume 54, 2021, Pages 130-135, ISSN 2351-9789.
4. Da A J C, De L L N S, Marcos C S.An integrated production planning and inventory management problem for a perishable product: optimization and Monte Carlo simulation as a tool for planning in scenarios with uncertain demands[J].TOP,2024,32(2):263-303.
5. Wolde I, Starke A R, da Silva A K, et al. A detailed multi-component heat configuration assessment for complex industrial plants through Monte Carlo simulations: A case study for the cement industry[J]. Sustainable Energy Technologies and Assessments, 2025, 75: 104237.
6. Wang C W, Tao H W , Tian C W, et al. A Multi-strategy Slime Mould Algorithm for Solving Global Optimization and Engineering Optimization Problems[J].Evolutionary Intelligence,2024,17(5-6):3865-3889.
7. Li S, Chen H, Wang M, et al. Slime mould algorithm: A new method for stochastic optimization[J]. Future generation computer systems, 2020, 111: 300-323.